Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen
Gesamtfakultät
der Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Physiker Frederik Orlando Kaster
aus Kirchheimbolanden

Tag der mündlichen Prüfung: 11. Mai 2011

Bildanalyse für die Lebenswissenschaften – Rechnerunterstützte Tumordiagnostik und Digitale Embryomik

Gutachter: Prof. Dr. Fred A. Hamprecht

Prof. Dr. Wolfgang Schlegel

Dissertation

submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
Diplom-Physiker Frederik Orlando Kaster
Born in: Kirchheimbolanden

Oral examination: May 11, 2011

Image Analysis for the Life Sciences – Computer-assisted Tumor Diagnostics and Digital Embryomics

Referees: Prof. Dr. Fred A. Hamprecht

Prof. Dr. Wolfgang Schlegel

Zusammenfassung

Die moderne lebenswissenschaftliche Forschung erfordert die Analyse einer derart großen Menge von Bilddaten, dass sie nur noch automatisiert bewältigt werden kann. Diese Arbeit stellt einige Möglichkeiten vor, wie automatische Mustererkennungsverfahren zu verbesserter Tumordiagnostik und zur Entschlüsselung der Embryonalentwicklung von Wirbeltieren beitragen können.

Kapitel 1 untersucht einen Ansatz, wie räumliche Kontextinformation zur verbesserten Schätzung von Metabolitenkonzentrationen aus Magnetresonanzspektroskopiebildgebungs-(MRSI-)Daten zwecks robusterer Tumorerkennung verwendet werden kann, und vergleicht diesen mit einem neuen Alternativverfahren.

Kapitel 2 beschreibt eine Softwarebibliothek zum Training, Testen und Validieren von Klassifikationsalgorithmen zur Schätzung von Tumorwahrscheinlichkeiten an Hand von MRSI-Daten. Diese ermöglicht die Anpassung an geänderte experimentelle Bedingungen, den Vergleich verschiedener Klassifikatoren sowie Qualitätskontrolle: dafür ist kein Expertenwissen aus der Mustererkennung mehr erforderlich.

Kapitel 3 untersucht verschiedene Modelle zum Lernen von Tumorklassifikatoren unter Berücksichtigung der in der Praxis häufig auftretenden Unzuverlässigkeit menschlicher Segmentierungen. Zum ersten Mal werden Modelle für diese Klassifikationsaufgabe verwendet, welche zusätzlich die objektive Information aus den Bildmerkmalen nutzen.

Kapitel 4 enthalt zwei Beiträge zu einem Bildanalysesystem für die automatisierte Rekonstruktion der Entwicklung von Zebrabärbling-Embryonen an Hand von zeitaufgelösten Mikroskopiebildern: Zwei Verfahren zur Zellkernsegmentierung werden experimentell verglichen, und ein Verfahren zur Verfolgung von Zellkernen über die Zeit wird vorgestellt und ausgewertet.

Abstract

Current research in the life sciences involves the analysis of such a huge amount of image data that automatization is required. This thesis presents several ways how pattern recognition techniques may contribute to improved tumor diagnostics and to the elucidation of vertebrate embryonic development.

Chapter 1 studies an approach for exploiting spatial context for the improved estimation of metabolite concentrations from magnetic resonance spectroscopy imaging (MRSI) data with the aim of more robust tumor detection, and compares against a novel alternative.

Chapter 2 describes a software library for training, testing and validating classification algorithms that estimate tumor probability based on MRSI. It allows flexible adaptation towards changed experimental conditions, classifier comparison and quality control without need for expertise in pattern recognition.

Chapter 3 studies several models for learning tumor classifiers that allow for the common unreliability of human segmentations. For the first time, models are used for this task that additionally employ the objective image information.

Chapter 4 encompasses two contributions to an image analysis pipeline for automatically reconstructing zebrafish embryonic development based on time-resolved microscopy: Two approaches for nucleus segmentation are experimentally compared, and a procedure for tracking nuclei over time is presented and evaluated.

Acknowledgments

First of all, I would like to thank my supervisor Prof. Dr. Fred Hamprecht for the opportunity to conduct the research for this PhD thesis in his research group and for his constant advice during the last years. I thank Dr. Ullrich Köthe for his helpful advice concerning various areas of image processing, pattern recognition and software development. I thank my predecessors Dr. Björn Menze and Dr. Michael Kelm for their previous work on MRSI analysis, which paved the ground for parts of the research presented in this thesis, and for their helpful advice on the MRSI quantification and tumor segmentation projects. Dr. Björn Menze provided one of the expert label sets for the evaluation in chapter 1, and performed the registration of the real-world radiological data sets studied in chapter 3. Dr. Michael Kelm proposed the spatially regularized MRSI quantification approach that is validated in chapter 1, as well as implementing huge parts of the software foundation that was required for bringing the MRSI classification library presented in chapter 2 into clinical use. I thank Xinghua Lou, Martin Lindner and Bernhard Kausler for the productive collaboration on the zebra fish digital embryo project: Xinghua Lou developed one of the segmentation methods evaluated in chapter 4, Martin Lindner implemented the routines for the computation of the features required for the tracking procedure and Bernhard Kausler provided manual ground truth for the tracking evaluation. The other segmentation method studied in chapter 4 as well as the visualization functionality for segmentation validation was provided via the ILASTIK software developed by Dr. Christoph Sommer, Christoph Straehle and Dr. Ullrich Köthe: I thank them for their help with the usage and customization of this software. I thank Stephan Kassemeyer for helping with the implementation of the software described in chapter 2. Furthermore I thank all the other present and former members of the Multidimensional Image Processing group for the good group climate, for the lively discussions and for the help on various technical and scientific questions, namely Björn Andres, Sebastian Boppel, Joachim Börger, Luca Fiaschi, Jörg Greis, Matthias Griessinger, Dr. Michael Hanselmann, Nathan Hüsken, Dr. Marc Kirchner, Anna Kreshuk, Thorben Kröger, Rahul Nair, Dr. Bernhard Renard, Martin Riedl, Jens Röder, Patrick Sauer, Christian Scheelen, Björn Voss, Andreas Walstra and Matthias Wieler, as well as all the other researchers at the Heidelberg Collaboratory for Image Processing.

During my PhD research time, I was closely affiliated with the Software Development for Integrated Diagnostics and Therapy (SIDT) group of the German Cancer Research Center. I thank Prof. Dr. Wolfgang Schlegel for the all of the financial and academic support I received from the German Cancer Research Center. I thank the former and present heads of the SIDT group, Dr. Oliver Nix and Dr. Ralf Floca, for their advice particularly on questions of software development. Furthermore I thank all the group members for the good group climate, the lively discussions and the help on various technical questions, namely Markus Graf, Dr. Martina Hub, Andreas Jäger, Dr. Sarah Mang, Hermann Prüm, Dirk Simon, Dörte van Straaten and Lanlan Zhang.

Interdisciplinary projects as presented in this thesis would not have been possible without close interaction with the medical and biological collaborators. From the Radiological University Clinic of Heidelberg, I thank Dr. Marc-André Weber for providing the brain tumor images analyzed in chapter 3. From the Radiology group of the German Cancer Research Center, I thank Dr. Christian Zechmann, Dr. Patrik Zamecnik, Dr. Lars Gerigk, Dr. Bram Stieltjes and Dr. Christian Thieke for providing magnetic resonance spectroscopy imagery and expert annotations for the evaluation of the software presented in chapter 2 and for helpfully commenting upon the software interfaces from a clinical users' point of view. I also thank Bernd Merkel and Markus Harz from the Fraunhofer MeVis Institute for Medical Image Computing Bremen for developing the graphical user interface which makes for the screenshots in this chapter. For the acquisition and their helpful comments upon the MRSI spectra analyzed in chapter 1, I thank Prof. Dr. Peter Bachert, Sarah Snyder and Benjamin Schmitt from the Medical Physics in Radiology group of the German Cancer Research Center. From the Institute for Zoology at the University of Heidelberg, I thank Prof. Dr. Joachim Wittbrodt and Burkhard Höckendorf for acquiring the zebrafish microscopy images analyzed in chapter 4. From the Computer Graphics and Visualization group of the University of Heidelberg, I thank Prof. Dr. Heike Jänicke for providing software for the visualization of these data.

For their support in all administrative affairs, I would like to thank Barbara Werner, Stephanie Lindemann, Simone Casula, Sarina Faulhaber, Evelyn Verlinden and Karin Kruljac.

I gratefully acknowledge the financial support by the Helmholtz International Graduate School for Cancer Research, the Federal Ministry of Education and Research (BMBF) and the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp).

My final thanks go to Hans and Elfriede Botz for being the best landlord and landlady one could wish for, and to my family and friends for their constant love and emotional backing during these trying years.

Contents

Prologue

Computers are of ever-increasing importance for today's life sciences. Their influence is most established in genomics, where they were crucial for sequencing e.g. the human genome (Lander et al., 2001), and in proteomics, where they can be used in order to identify the proteins that are present in a biological sample (Colinge & Bennett, 2007). In general, their use is unavoidable whenever one encounters data sets that are too large for manual analysis. These data-intensive areas are typically designated with the suffix "-omics": besides genomics and proteomics, there are e.g. connectomics where the subject is the connections between all the neurons in a brain (Lichtman et al., 2008), embryomics which deals with the detailed study of embryonic development on a cellular level (Bourgine et al., 2010) or glycomics which studies the interactions between the polysaccharides covering the cellular membranes (Raman et al., 2005). Recently, the same computer-based high-throughput data analysis techniques have even transcended the boundaries of the life sciences, and have been fruitfully employed to study cultural trends by analyzing the usage frequencies of words and word sequences in digitized books from different time points, leading to the term "culturomics" (Michel et al., 2010).

While biological data can be structured in various ways (e.g. as sequences, trees, graphs or relational databases), this thesis concentrates on image data which show the spatial distribution of some interesting quantity. In the simplest case, each point in space is associated with a single scalar value, e.g. the intensity of emitted light. For multispectral or multimodal data, every point is associated with several scalar values: these may be e.g. the intensities of light emitted at different wavelengths. Most image data in the life sciences come from either of two sources:

- **Medical images** (Duncan & Ayache, 2000) are important for basic research, applied clinical research and routine diagnostics of diseases. Different physical mechanisms are exploited to gain information about the interior tissues of living humans or animals: e.g. X-ray attenuation (computed tomography), radiofrequency emission due to the relaxation of excited nuclei in a magnetic field (magnetic resonance imaging) or ultrasound scattering.
- **Microscopy images** (Rittscher, 2010) are mainly important for basic research, although they also have relevance for e.g. drug discovery (toxicity assays). Living (*in vivo*) or prepared (*in vitro*) tissues or organisms are illuminated

with either visible light or an electron beam, and magnified images are created using a lens system.

Chapters 1 – 3 deal with applications from medical image analysis, while a microscopy image analysis task is studied in chapter 4.

Computerized image analysis answers questions such as:

- **Classification**: Does a certain location in the image belong to a foreground class (e.g. a cell) or a background class?
- **Object detection**: Where is an interesting foreground object roughly located in the image?
- **Segmentation**: Exactly which pixels (2D) or voxels (3D) do belong to a particular contiguous foreground object?
- **Tracking**: If images are acquired at different time points, how do several objects in the image move over time?
- **Registration**: If several independent images are acquired which all show different aspects of an object, how can they be fused to a single multispectral image so that all points corresponding to the same location are matched to the same pixel or voxel?

In some cases, foreground and background can be discriminated by a simple criterion such as the absolute gray value of an image. More often, they differ in a more complicated way, and human experts are able to tell the both classes apart without being able to state explicit rules on which they base their decisions. Pattern recognition techniques allow to learn these rules automatically from example images together with annotations (or labels) provided by the human experts. This allows the use of generic techniques in order to solve a huge variety of specific image analysis tasks: often all task-specific information may be learned from a moderate set of annotated training data.

Chapter 1.

Experimental evaluation of MRSI quantification techniques using spatial context

1.1. Introduction and motivation

Tumor tissue can be distinguished from healthy tissue by its characteristic biochemical makeup, i.e. by the increase or depletion of characteristic metabolites due to the idiosyncrasies of tumor metabolism. Magnetic resonance spectroscopy imaging (MRSI) is a noninvasive technique by which the biochemical composition of tissues can be studied in the living body (*in vivo*) in a spatially resolved manner. Extracting the local metabolite concentrations from the MRSI signal is called *quantification*. This chapter deals with different approaches how quantification may be improved by exploiting the spatial smoothness of the MRSI data: rather than considering the spectrum in each voxel on its own, prior assumptions can be imposed that neighboring voxels should yield similar quantification results, and it is a plausible hypothesis that this will lead to a more robust estimation. As is shown in the following, it is experimentally preferable to impose the smoothness prior in a separate initialization stage in which the theoretically predicted spectra are roughly aligned to the data, rather than in the actual estimation step.[1]

[1] Parts of this chapter form part of (Kelm et al., 2011).

1.2. Background: Magnetic resonance spectroscopic imaging (MRSI)

Nuclear magnetism MRSI is a medical imaging[2] modality that makes use of the Zeeman splitting of nuclear energy states in an external magnetic field. The following exposition concerns common knowledge, see e.g. (de Graaf, 2008) for a good introductory text. Consider a nucleus ${}^{A}_{Z}\mathrm{X}$ (i.e. A nucleons, Z protons) with the nuclear spin $\vec{I}$: the associated magnetic moment is

$$\vec{\mu} = g_I \frac{e}{2Mc}\vec{I} = \gamma\vec{I}, \tag{1.1}$$

where g_I denotes the nuclear g-factor, M denotes the nuclear mass and γ the gyromagnetic ratio. For the nuclear state characterized by the quantum numbers I and m_I (with $m_I \in \{-I, -I+1, \ldots, I-1, I\}$), the expectation values of the squared magnitude of the magnetic moment and its z-component are given by

$$\langle \mu^2 \rangle = \gamma^2 \hbar^2 I(I+1), \langle \mu_z \rangle = \gamma\hbar m_I. \tag{1.2}$$

For most stable nuclei, both A and Z are even, and the nuclear spin I equals zero in the ground state. Very few stable nuclei (e.g. deuterium) have an even A and an odd Z, which leads to an integral value for I. The highest relevance for MRI have stable nuclei with an odd A, for which I takes a half-integral value (e.g. ^{1}H, ^{13}C, ^{19}F, ^{23}Na or ^{31}P).

Equilibrium magnetization In the absence of an external magnetic field, all nuclear states corresponding to the $2I+1$ different quantum numbers m_I are degenerate and hence equally populated in thermal equilibrium. However, once an external field B_0 is applied along the z-axis, Zeeman splitting occurs:

$$E = -\mu_z B_0 = -\gamma\hbar m_I B_0. \tag{1.3}$$

In the following we restrict ourselves to discussing the case of the protium (^{1}H), for which $I = 1/2$ and $\gamma = 2\pi \times 42.6\mathrm{MHz/T}$. Due to its high gyromagnetic ratio and its high natural abundance, this is the most sensitive nucleus for MR measurements. There are two Zeeman states ($m_I = 1/2$, i.e. parallel to the external field, and $m_I = -1/2$, i.e. antiparallel to the field). For a sample of matter (e.g. a human

[2] To be precise, while medical imaging is the most important application, other applications exist e.g. in food safety monitoring, non-destructive industrial testing or analyzing the composition of crude oil.

body), let $n_{\uparrow\uparrow}$ and $n_{\uparrow\downarrow}$ denote the numbers of nuclei in these two states. Then in thermal equilibrium,

$$\frac{n_{\uparrow\uparrow}}{n_{\uparrow\downarrow}} = \exp\left(\frac{\hbar\gamma B_0}{k_B T}\right) \approx 1 + \frac{\hbar\gamma B_0}{k_B T} \text{ for small } B_0. \tag{1.4}$$

It should be noted that the relative excess is small: e.g. for realistic values ($B_0 =$ 1.5T, $T =$ 300K), the ratio is $n_{\uparrow\uparrow}/n_{\uparrow\downarrow} = 1 + 3 \times 10^{-6}$. However, this minute excess is responsible for the macroscopic magnetization of the protons in the sample:

$$M_0 = (n_{\uparrow\uparrow} - n_{\uparrow\downarrow}) \cdot \mu_z \approx \frac{(\gamma\hbar)^2}{4k_B T} N B_0, \tag{1.5}$$

where $N = n_{\uparrow\uparrow} + n_{\uparrow\downarrow}$ is the total number of ^{1}H nuclei. At thermal equilibrium, the gross magnetization is completely aligned with the external field and no net transverse magnetization occurs (although the magnetic moments of the single spins precede around the external field, their precession is completely dephased, so that the transversal components of the magnetic moments cancel out).

Energy transitions by radio-frequency irradiation Transitions between the different energy levels can be driven by exciting the sample with electromagnetic radio-frequency (RFr) radiation near the resonance (or Larmor) frequency of $f_0 = \gamma B_0/2\pi$ (42.6 MHz/T for ^{1}H, corresponding to a wavelength of $7\text{m} \cdot \text{T}/B_0$), which can be generated by a transmitter coil. The irradiated RFr field must be orthogonal to the main external $\vec{B}_0$ field:

$$\vec{B}_1(t) = B_1 \cos(2\pi f t)\vec{e}_x + B_1 \sin(2\pi f t)\vec{e}_y. \tag{1.6}$$

The temporal evolution of the gross magnetization is then governed by the Bloch equations:

$$\frac{d\vec{M}}{dt} = \gamma\vec{M} \times \begin{pmatrix} B_1\cos(\omega t) \\ B_1\sin(\omega t) \\ B_0 \end{pmatrix} + \frac{\vec{M}_0 - \vec{M}_\parallel}{T_1} - \frac{\vec{M}_\perp}{T_2} \tag{1.7}$$

with $\vec{M}_\perp$ and $\vec{M}_\parallel$ denoting the magnetization components perpendicular and parallel to the $\vec{B}_0$ field. Eq. (1.7) consists of three terms: a precession term due to the excitation field, and two relaxation terms. The latter account for the fact that a gross magnetization perturbed away from the equilibrium magnetization $\vec{M}_0$ recovers to the equilibrium due to energy exchanges between the nuclear spins and the surrounding heat bath (T_1 relaxation, spin-lattice relaxation) and loss of coherence between the precessing spins (T_2 relaxation, spin-spin relaxation). Typical values for water-rich biological tissues are 1500–2000 ms for T_1 and 50–200 ms for T_2. Inhomogeneities in the external field $\vec{B}_0$ can further speed up the transversal spin dephasing and lead to effective values of $T_2^* < T_2$.

90° and 180° pulses The qualitative understanding of the magnetization dynamics is simplified if they are studied in a coordinate system $(\vec{e}_{x'}, \vec{e}_{y'}, \vec{e}_z)$ rotating in phase with the $\vec{B}_1$ vector. In such a system, Eq. (1.7) takes the following form:

$$\frac{d\vec{M}'}{dt} = \vec{M}' \times \begin{pmatrix} \gamma B_1 \\ 0 \\ 2\pi(f_0 - f) \end{pmatrix} + \frac{\vec{M}_0 - \vec{M}'_\parallel}{T_1} - \frac{\vec{M}'_\perp}{T_2} \tag{1.8}$$

Now it is obvious that in resonance $(f = f_0)$, $\vec{M}'$ rotates with angular frequency γB_1 around the $\vec{e}_{x'}$ axis. If such a resonant field is applied for a time of $\pi/(2\gamma B_1)$, the magnetization rotates into the xy-plane and is completely transversal (90° pulse): all spins precess with complete phase coherence, until they are again dephased due to the spin-spin relaxation. If the excitation field is applied for the double time (180° pulse), the spins first get into phase and then dephase again, so that the net magnetization points in the $-\vec{e}_z$ direction.

Signal acquisition: FID and spin echo sequence During relaxation, the precession of the non-equilibrium magnetization causes a transversal RF signal to be emitted, which can be detected in a receiver coil, typically both in x and in y direction (quadrature detection).[3] It is called the free induction decay (FID). The free induction decay of a single resonance can be described by a damped exponential in the time domain, and by a Lorentzian in the frequency domain:

$$g(t) \propto M_0 \exp\left(-\frac{t}{T_2^*} + 2\pi i f_0 t + i\phi_0\right) \tag{1.9}$$

$$\hat{g}(f) \propto \frac{M_0 T_2^* \exp(i\phi_0)}{1 + 2\pi i (f - f_0) T_2^*} \tag{1.10}$$

Often, Doppler broadening occurs due to the thermal motion of the protium nuclei in the sample: hence the Lorentzian is convolved with a Gaussian, resulting in a Voigt profile. As the FID is often perturbed by the previous RF pulse, a delayed signal acquisition is often preferable, which can be achieved by the spin-echo (SE) sequence: The idea is to reverse the rapid dephasing caused by the B_0 field inhomogeneities (T_2^*) with a 180° pulse in either x' or y' direction, which is applied after a time of TE/2. This causes all spin precessions to change their direction. Since the absolute precession speed stays the same, the spins come back into phase at the echo time TE: hence, a discernible echo signal occurs at that time, and then dephases again with time constant T_2^*. Compared to the original FID, the amplitude of the echo signal is reduced by a factor of $\exp(-\mathrm{TE}/T_2)$, which accounts for the stochastic dephasing effects that cannot be reverted by the 180° pulse.

[3] There may also be a single transceiver coil, which acts as both the transmitter and the receiver coil.

Chemical shift and MRS The previous discussion assumed that all ^{1}H nuclei in an external field have the same resonance frequency, irrespective of the molecules in which they occur. However, that is only approximatively correct: due to the magnetic properties of the surrounding electrons, all nuclei experience an effective external field that is slightly different from $\vec{B}_0$ (chemical shift):

$$\vec{B}_{\text{eff}} = \vec{B}_0(1 - \sigma) = \vec{B}_0 - \delta \tag{1.11}$$

Usually the induced magnetic field of the electrons opposes the $\vec{B}_0$ field (Lenz' rule) so that $\delta > 0$, but π electrons may also enhance $\vec{B}_0$ (e.g. for benzene, δ is negative). The typical order of magnitude for σ is 10^{-6}; hence the chemical shift is typically measured in parts per million (ppm). For ^{1}H spectroscopy, it is defined with respect to $Si(CH_3)_4$ (tetramethylsilane), which is assigned a chemical shift of 0. The total signal is a superposition of the FIDs of all metabolites contained in the sample: after a Fourier transformation, these FIDs appear as distinct peaks whose amplitude is proportional to the metabolite concentration (Fig. 1.1). Typically by far the most protium nuclei are part of water molecules, hence the metabolite signals may be undetectable against the water background signal, unless it is suppressed either by specific data acquisition protocols or by postprocessing steps. Experiments in which the spectral composition of the ^{1}H RFr signal is studied, are known as magnetic resonance spectroscopy (MRS).

Single-voxel localization In MRS, an entire sample is excited at once, and the emitted signal from the whole volume is received. This is usually sufficient for studies of homogeneous substances (e.g. in material characterization), and may also give valuable information in diagnostic medicine, e.g. about the presence and extent of a tumor in the brain (Cohen et al., 2005).[4] However, often one is interested not only in whether there is a tumor somewhere in the head, and how large it is, but also in its location: this information is particularly relevant for radiotherapy and surgery planning (see e.g. Chan et al. (2004)). Common to all spatial localization techniques (for a good recent overview over the different possibilities see Keevil (2006)) is the use of gradient fields, i.e. additional spatially varying magnetic fields which are parallel to the $\vec{B}_0$ field. Hence the resonance frequency becomes spatially dependent:

$$f_0(\vec{r}) = \frac{\gamma}{2\pi}(B_0 + \vec{G} \cdot \vec{r}). \tag{1.12}$$

These gradient fields are typically switched on only at specific phase during the measurement process. For slice-selective excitation, a z-gradient field is applied only

[4] Advantages of such whole-brain spectroscopy protocols are the good signal-to-noise ratio (SNR) and the robustness with respect to positioning errors.

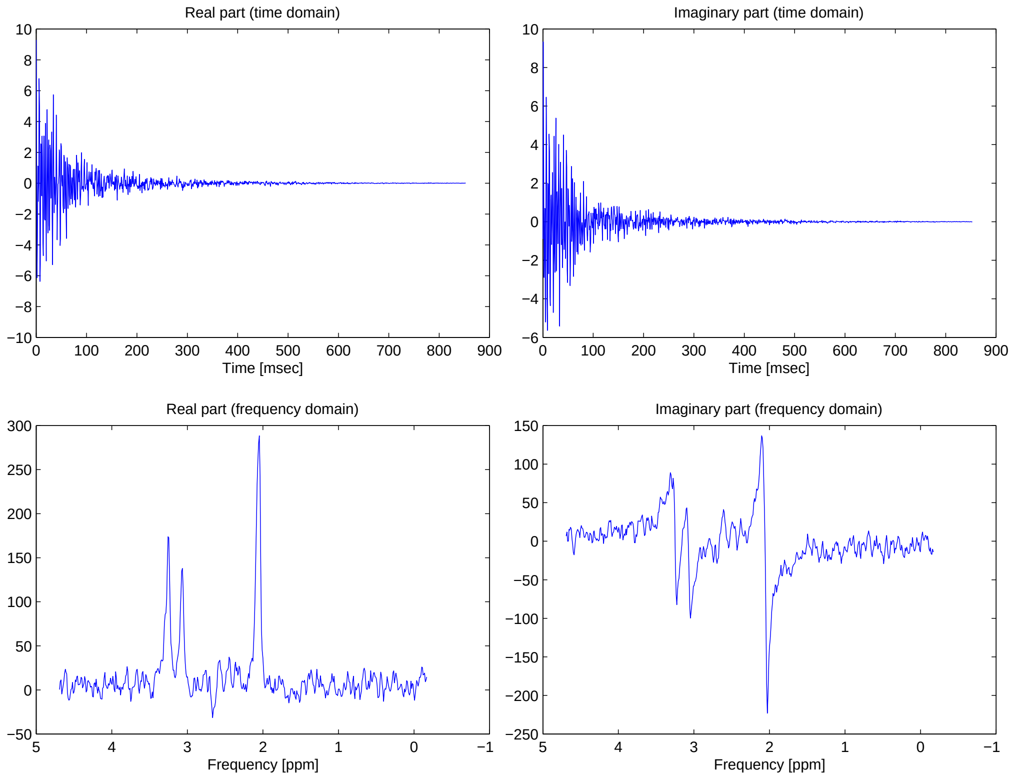


Figure 1.1. – Exemplary brain MRSI spectrum in the time and frequency domain. The three peaks correspond to the most important metabolites of the healthy brain, namely (from left to right) choline, creatine and N-acetylaspartate (NAA).

during the excitation with a bandwidth-limited RFr pulse: if the bandwidth is given by Δf, only the ^{1}H nuclei inside a axial slice of thickness

$$\Delta z = \frac{2\pi\Delta f}{\gamma G_z} \tag{1.13}$$

are excited.[5] The spectrum in a specific volume element (voxel) can be measured by single-voxel MRS techniques such as the PRESS (**P**oint-**RE**solved **S**pectro**S**copy)

[5] Strictly speaking, as the excitation pulse must be time-limited, it cannot be exactly frequency-limited at the same time, so that some signal bleeding from the other z slices always occurs. This is the reason why e.g. the 180° pulses in the PRESS sequence are commonly flanked by two symmetric spoiler gradient fields that dephase transversal magnetization that was caused by the imperfect selectivity.

sequence by Bottomley (1987), which consists of one 90° and two refocussing 180° pulses. Each pulse is accompanied by a slice selection gradient in a different direction (x, y and z), so that the second echo only occurs in the intersection of these three orthogonal planes. If the volume of interest lies near the surface of the sample, selective excitation can also be achieved by the use of a surface coil, as the B_1 field of a coil of radius a drops with the distance z from the coil as $(a^2 + z^2)^{-3/2}$ (B_1 gradient-based localization).

Magnetic resonance spectroscopy imaging (MRSI) If metabolite concentration maps are desired, the individual MR spectra of a whole grid of voxels inside a volume of interest must be measured at the same time: this is the application of MRSI. The easiest technique is based on the spin-echo sequence: it requires $N_x \cdot N_y \cdot N_z$ repetitions for measuring a grid of $N_x \times N_y \times N_z$ voxels. Each repetition is characterized by a different combination of gradients G_x, G_y and G_z. While the G_z gradient is applied during the 90° and the 180° pulse to achieve slice-selective excitation, the G_x and G_y gradients are simultaneously applied for a time of T between the 90° and the 180° pulse: they lead to a spatially dependent phase shift of

$$\Delta\phi = \gamma T(G_x x + G_y y) = k_x x + k_y y, \tag{1.14}$$

with $k_i := \gamma T G_i$. Measuring the signal for the different values of G_x and G_y (and hence k_x and k_y) can be interpreted as sampling the two-dimensional Fourier transformation of the spin density inside the excited slice:

$$\hat{\rho}(k_x, k_y) = \int dx \int dy \rho(x, y) e^{ik_x x + ik_y y}, \tag{1.15}$$

and the original spin density can be reconstructed via the inverse Fourier transform. A repetition TR $\gg$ TE must elapse between the different spin-echo cycles to avoid any remanent transverse magnetization from the previous cycle. This accounts for the long time required for MRSI measurements: with a typical repetition time of TR $= 2$ s, acquiring a coarse $16 \times 16 \times 8$ volume takes 4096 s, i.e. more than one hour.[6] For ^{1}H MRSI and standard clinical B_0 fields of 1.5 T, voxel sizes of 0.5–5 cm^3 can be achieved by these techniques. The limiting factor is the signal-to-noise ratio (SNR): too little signal can be captured from smaller voxels. As SNR improves roughly linearly with increasing B_0 field strength (Edelstein et al., 1986), improved spatial resolution can be achieved at higher field strengths that are currently under experimental investigation (Henning et al., 2009).

[6] Magnetic resonance imaging (MRI) uses similar encoding strategies and also samples the signal in the Fourier domain. However, it can be considerably sped up over MRSI by using the discussed phase modulation strategy only for one of the in-plane directions, and encoding the other direction in the frequency of the acquired signal (frequency modulation): i.e., the corresponding gradient is applied during signal acquisition. However, this is not an option for standard MRSI protocols, as the frequency of the acquired signal already encodes the chemical shift.

Clinically relevant metabolites In clinical applications of ^{1}H MRSI, detection is possible for metabolites having concentrations of down to 1 mmol/l: since the RFr sensitivity is typically not known, only relative quantification is possible (i.e. the ratios between the concentrations of different metabolites can be estimated, but not the absolute concentration values). Among the diagnostically most relevant metabolites that can be detected by ^{1}H spectroscopy are (Govindaraju et al., 2000):

1. ***N*-acetylaspartate** (NAA): This metabolite gives rise to the predominant resonance in healthy brain tissue. While its biochemical function is still only poorly understood, it is known to be a characteristic clinical marker for intact neurons: hence it is depleted in nearly every type of brain lesion (e.g. stroke, tumors or neurodegeneration).
2. **(Phospho-) Creatine** plays an important role as an energy buffer and storage medium, which is required for regeneration from adenosyldiphosphate (ADP) to adenosyltriphosphate (ADP), the most important free energy carrier in cell metabolism. Creatine is most useful as a normalization reference for other metabolite concentrations, but is not indicative for pathology by itself.
3. **Choline** is a precursor for the phospholipids making up the cellular membranes; hence it is enhanced in proliferating tissues with a high activity of membrane biogenesis (such as tumors).
4. **Lactate** is generated by anaerobic glycolysis; hence it is a marker for ischemia and hypoxia and it is commonly increased in tumors, particularly in the necrotic core.
5. **Lipid** resonances are broader than the signals of the metabolites mentioned above, and they typically cannot be captured by a simple parametric (Voigt or quantum mechanical) model. They arise mostly from free fatty acids, and are indicative for high-grade tumors or cell necrosis.
6. **Citrate** is one of the main ingredients of prostatic fluid: hence it is the predominant resonance in the healthy prostate, and it is characteristically depleted in prostatic cancer.

The sensitivity of MRSI and the metabolites visible in the spectrum can also be influenced by the choice of the echo time TE: As the MR signal decays with $\exp(-\mathrm{TE}/T_2)$, shorter echo times correspond to better SNR. However, many nuisance signals from proteins or liquids have very short T_2 and are decayed away in long-TE spectra, hence the signals from the interesting metabolites can be better discernible in these spectra.

1.3. Quantification with spatial context

Current state-of-the-art procedures for time-domain quantification of MRSI series, such as AMARES (Vanhamme et al., 1997) or QUEST (Ratiney et al., 2005), estimate the spatially resolved concentrations of relevant metabolites by solving a nonlinear least-squares (NLLS) problem:

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \left(g_{\theta}(t_n) - y_n\right)^2 \tag{1.16}$$

In the preceding formula, y_n denotes the complex MRSI signal for a specific voxel acquired at the time t_n and $g_{\theta}(t_n)$ is a parametric model for this time series, with the parameter vector θ comprising both the amplitudes of the relevant metabolites in this voxel (i.e. the final aim of quantification) and additional signal distortion parameters such as phase or frequency shifts or (Lorentzian or Gaussian) damping factors. In the following, this procedure will be called the Single Voxel (SV) method, since the estimation is performed for every voxel on its own and no information from neighboring voxels is used in this process. However, the non-convexity of this optimization problem may lead to convergence problems, or the procedure may converge to a wrong local optimum. The time course y_n also typically contains considerable noise (especially for high-resolution measurements) which may cause the parameter estimates to be biased and to have high variance (Cook et al., 1986).

Similar estimation problems arise also in the analysis of other medical imaging modalities, such as in the construction of kinetic parameter maps for the analysis of dynamic contrast-enhanced (DCE) MRI measurements. It could be shown that spatial regularization could improve both bias and variance of the parameter estimates and improve the robustness of the estimation with respect to noise (Kelm et al., 2009). "Spatial regularization" means that the parameters of different voxels are coupled via a regularization term penalizing large parameter differences between neighbor voxels, e.g. using a Generalized Gaussian Markov Random Field (GGMRF) model (Bouman & Sauer, 1993):

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta} \left[\sum_{s\in V}\sum_{n=1}^{N} \left(g_{\theta_s}(t_n) - y_n^s\right)^2 + \sigma^2 \sum_{s\sim t} \alpha_{st} \|W(\theta_s - \theta_t)\|_p^p\right] & (1.17)\\ &= \arg\max_{\theta} \log P\left((\theta_s)_s \mid (y_n^s)_{s,n}\right) & (1.18)\end{aligned}$$

In this formula, image voxels are indexed by s and t, with $s \sim t$ denoting a neighborhood relationship (usually only voxels in the same slice are considered as neighbors, and the standard 4-neighborhood or 8-neighborhood is used). y_n^s denotes the MRSI signal corresponding to the voxel s. The factor α_{st} allows one to e.g. weight diagonal and vertical or horizontal neighbors in an 8-neighborhood differently. W is a diagonal weighting matrix which controls how the different parameters (e.g. amplitudes, frequency shifts, phase shifts, ...) contribute to the penalty term: it is especially required for incommensurable parameters. σ^2 is the noise variance which can be estimated from the latest time points of the MRSI signal, and $\| \cdot \|_p$ with $1 < p \leq 2$ denotes the standard p-norm (using $p < 2$ can prevent an over-smoothing of edges, e.g. in the presence of lesions). In the language of Bayesian statistics, we can interpret the regularization terms as a prior distribution on the set of potential parameter maps.

The Hammersley-Clifford theorem (Clifford, 1990) states that for computing the optimal parameters on a subset of voxels A given the parameters at all other sites, only the parameter values in the Markov blanket of A must be known:

$$\arg\max_{\theta_A} \log P\left(\theta_A | \theta_{A^c}, (y_n^s)_{s,n}\right) = \arg\max_{\theta_A} \log P\left(\theta_A | \theta_{\partial A}, (y_n^s)_{s \in A, n}\right) \tag{1.19}$$

$$\text{with} \quad \partial A = \{s \in V | \exists t \in A : s \sim t\} \tag{1.20}$$

This property is used in the Iterated Conditional Modes (ICM) algorithm (Besag, 1986), which finds a local maximum of the joint log-probability by iteratively optimizing the parameters of each voxel given the current (fixed) values of its neighbors. Convergence may be sped up by the more general block-ICM scheme (Wu et al., 1994), which iterates over whole blocks of voxels and jointly optimizes the parameters over a whole block of voxels given the fixed parameter values from the Markov blanket of this block. This block-ICM scheme can be viewed as a compromise between ICM with single-voxel update and the (infeasible) global optimization problem in which the parameters from all voxels are jointly optimized: hence it may be plausibly expected that it also leads to a higher-energy solution than ICM with single-voxel updates (which is however not guaranteed).

Recently, Kelm (2007) proposed to impose a GGMRF prior on the MRSI parameter maps and to use the block-ICM algorithm in order to perform inference on this model: Preliminary studies on simulated MRSI measurements suggested that this spatial regularization improves the estimation robustness against noise, and decreases both bias and variance of the parameter estimates in comparison to the single voxel (SV) model, as had already been established for DCE MRI analysis. In this study, this claim was tested on real-world MRSI measurements. Preliminary evaluations on proband MRSI measurements (with a voxel size of $10 \times 10 \times 10\text{mm}^3$ as for standard clinical measurements) showed no improvement of using the GGMRF model over the SV model and the question arose how realistic the simulated data were and whether the GGMRF gives any practical advantages for MRSI analysis that justify the increased computation time: these findings necessitated a rigorous experimental analysis.

1.4. Related work

There exists a multitude of quantification techniques for MRSI data, so that only a cursory overview over the field can be given. For a more comprehensive recent survey, see (Poullet et al., 2008). They fall into two main categories: time-domain methods and frequency-domain methods, which may be overlapping.[7] Time-domain methods fit the measured signal to a parametric model by a non-linear least-squares (NLLS) estimation, which may be solved using local or global optimization techniques. The parametric model consists of the spectra of the constituting metabolites, which may be derived from simple parametric approximations (Lorentzian, Gaussian or Voigt model), quantum mechanical predictions or experimental *in vitro* measurements.[8] Other approaches do not make prior assumptions about the metabolites contributing to the spectrum, but e.g. use the expectation maximization (EM) algorithm or some modification of the singular value decomposition (SVD) to fit an optimal number of Lorentzians to the FID. Nuisance signals arising from macromolecules (proteins, lipids) can often neither be predicted theoretically nor measured *in vitro*, hence they are rather captured by a nonparametric model such as a spline decomposition, like in the AQSES procedure by Poullet et al. (2007). Many of the frequency-domain quantification methods also follow either the NLLS or the SVD approach; alternatives are peak integration (where no assumptions about the peak shape are made) or nonparametric regression techniques such as artificial neural networks.

Besides the work by Kelm (2007), upon which this chapter builds, there have been few comparable approaches on exploiting spatial regularity for improved quantification of magnetic resonance spectroscopy images. The approach by Croitor Sava et al. (2009) has the highest similarity to this line of research: Same as Kelm (2007), they formulate the spatially regularized fitting problem as a Gaussian Markov random field, and refine the solution over several iteration sweeps through the grid. They also solve the

[7] The procedures discussed later in this chapter only make use of scalar products between spectra: hence it does not matter for them whether the spectra are represented in the time domain or in the frequency domain, according to Parseval's theorem (i.e. the unitarity of the Fourier transform).

[8] For instance, the AMARES procedure by Vanhamme et al. (1997) uses Lorentzian spectra, while the QUEST procedure by Ratiney et al. (2005) can make use of experimental basis spectra.

intractable optimization problem approximately via an iterated conditional modes (ICM) approach, i.e. the nonlinear parameters of one voxel are optimized given the fixed values of its neighbors. Their work differs in two respects: firstly, they combine the spatial regularization with a semi-parametric baseline estimation as in the AQSES algorithm (Poullet et al., 2007) in order to account for the macromolecular nuisance signals that occur in the short-echo data they are studying. Secondly, they account for the parameters in the neighboring voxels not only in the energy functional, but also in the initialization and for determining the search bounds on the parameters. Sima et al. (2010) present a slight modification of this approach, which differs only in the implementation of the nonlinear optimization. Instead of solving the problem in Eqs. (1.16) and (1.21) by e.g. a Levenberg-Marquardt optimizer with respect to all parameters, the optimization with respect to the linear parameters is performed in closed form, so that gradients must only be computed with respect to the nonlinear parameters. This variable projection approach is known to speed up convergence (Golub & Pereyra, 2003).

Bao & Maudsley (2007) combine the two tasks of MRSI reconstruction (i.e. computing the spatial MRSI distribution from the signal that has been acquired in k-space) and metabolite quantification into a single probabilistic Bayesian model and add a spatial regularity prior: they then use an EM approach to find the maximum a posteriori (MAP) solution for this model. Hereby they differ from most other approaches (as well as the one presented in this chapter), where the MRSI reconstruction is performed before the quantification: this is typically done via a Fourier transform, which causes signal bleeding into adjacent voxels and Gibbs ringing due to the limited k-space sampling rate. Registered MRI data are used to identify the positions of tissue borders, so that the smoothness priors for the metabolite concentrations can be switched off across these borders.

Furthermore, the LCModel software by Provencher (2010) contains a "Bayesian learning" procedure which fits first the good-quality spectra in the center of the FOV, and propagates the phase and frequency corrections thus found towards the outer voxels, where they serve as soft constraints for the fit. This approach models the dependencies between the fit parameters in the different voxels as a directed graphical model in contrast to the undirected graphical models studied in this chapter. Furthermore the inference is solved in a greedy local manner instead of the global inference methods employed in this chapter: once an inner spectrum has been fitted, the information from the outer fits cannot be backpropagated to refine this fit. However, the technical details are kept as a trade secret, so that a thorough discussion of this method is not possible. Experimentally, it was shown to perform inferior to the approach by Croitor Sava et al. (2009).

1.5. Experimental setup

Spatially regularized models like GGMRF contain the underlying assumption that the parameters across neighboring voxels are positively correlated: this assumption holds especially for small voxel sizes. Since small voxels are also associated with a low signal-to-noise ratio, the advantages of GGMRF should then be particularly pronounced. In order to study this voxel size effect systematically, two MRSI measurement series of the brain of a healthy proband were run. The measurements were conducted on a Siemens MAGNETOM TrioTM® with the following parameters: spin-echo (SE) sequence, repetition time 1700 ms, echo time 135 ms, magnetic field 3 Tesla (corresponding to an imaging frequency of 123.23 MHz), dwell time $dt = 833$ μs, $N = 512$ recorded time points, matrix size $16 \times 16 \times 1$ voxels. Every series comprised six scans: in the first series, three scans each were performed with a constant slice thickness of 10 mm or 20 mm and the in-plane side length was reduced, leading to anisotropic voxels. In the second series, the voxel size was kept isotropic, and two scans were conducted for each of three different side lengths. These two setups allow to study the effects of increasing the lateral and axial resolution separately: However, the second setting (with isotropic voxel sizes) is more typical for clinical MRSI scans. Tables 1.1 and 1.2 show the voxel sizes and field of view (FOV) sizes for each measurement: Only voxels fully included in the FOV were used for the subsequent analysis (1433 voxels in total). The scan series also differ in the number of FIDs which were acquired and averaged in order to improve the signal-to-noise ratio (SNR). The mean SNR for all series is also reported in these tables: it is defined as absolute height of the highest peak in the frequency spectrum in the vicinity of the expected metabolite positions, divided by the root mean-square magnitude of the spectrum in a frequency band containing neither signal nor artifact peaks, as in (Kreis, 2004).[9]

All data were subjected to water suppression with a Hankel singular value decomposition (HSVD) scheme (Pijnappel et al., 1992) before further analysis (the 15 most prominent SVD components were computed, and all of these components with a chemical shift $>$ 3.6 ppm or $<$ 1.5 ppm were subtracted from the signal). Furthermore, exponential apodization with a time constant of $N \cdot dt/5$ was applied in order to improve the signal-to-noise ratio.

For evaluation, the SV estimation (i.e. a nonlinear least-squares fit for every single voxel) was compared with the results of a block-ICM optimization of the GGMRF model, using 3×3 voxel blocks with a "chessboard" sweep schedule as in (Kelm et al., 2009). Prototypical implementations written in MATLAB® were used: the nonlinear

[9] While this definition of the SNR is fairly common in the MR spectroscopy community, there are also other, subtly different conventions: this should be considered when comparing SNRs between different publications.

Voxel size [volume]	# Avg.	FOV size [grid size]	Mean SNR
$10 \times 10 \times 10\text{mm}^3$ [1000 μl]	3	$80 \times 80 \times 10\text{mm}^3$ [8×8]	8.56
$6.9 \times 6.9 \times 10\text{mm}^3$ [473 μl]	3	$80 \times 80 \times 10\text{mm}^3$ [11×11]	4.47
$5 \times 5 \times 10\text{mm}^3$ [250 μl]	3	$60 \times 60 \times 10\text{mm}^3$ [12×12]	3.18
$7 \times 7 \times 20\text{mm}^3$ [980 μl]	3	$90 \times 90 \times 20\text{mm}^3$ [13×13]	7.22
$10 \times 10 \times 20\text{mm}^3$ [2000 μl]	3	$100 \times 100 \times 20\text{mm}^3$ [10×10]	13.03
$3.4 \times 3.4 \times 20\text{mm}^3$ [236 μl]	3	$45 \times 45 \times 20\text{mm}^3$ [13×13]	2.86

Table 1.1. – Voxel sizes and field of view sizes of the first six MRSI series (constant slice thickness) used for the experimental evaluation of the GGMRF quantification procedure, together with the number of FID averages (# Avg.) and the mean signal-to-noise ratio over all spectra in the series.

Voxel size [volume]	# Avg.	FOV size [grid size]	Mean SNR
$10 \times 10 \times 10\text{mm}^3$ [1000 μl]	3	$80 \times 80 \times 10\text{mm}^3$ [8×8]	20.67
$10 \times 10 \times 10\text{mm}^3$ [1000 μl]	6	$80 \times 80 \times 10\text{mm}^3$ [8×8]	25.49
$8 \times 8 \times 8\text{mm}^3$ [512 μl]	6	$80 \times 80 \times 8\text{mm}^3$ [10×10]	15.50
$8 \times 8 \times 8\text{mm}^3$ [512 μl]	3	$80 \times 80 \times 8\text{mm}^3$ [10×10]	12.24
$6 \times 6 \times 6\text{mm}^3$ [216 μl]	6	$80 \times 80 \times 6\text{mm}^3$ [13×13]	7.12
$6 \times 6 \times 6\text{mm}^3$ [216 μl]	3	$80 \times 80 \times 6\text{mm}^3$ [13×13]	5.68

Table 1.2. – Voxel sizes and field of view sizes of the second six MRSI series (isotropic voxels) used for the experimental evaluation of the GGMRF quantification procedure, together with the number of FID averages and the mean signal-to-noise ratio.

optimization was performed with an interior trust-region method for constrained nonlinear least-squares estimation as implemented in the MATLAB® Optimization Toolbox (Coleman & Li, 1996). In order to compare the computational requirements of the two competing methods, the effective quantification time per voxel is reported (i.e. the quantification time for a whole slice divided by the number of voxels inside the field of view). The average values on a standard PC (Intel® Core Duo 2 CPU T9300 @ 2.50 GHz, 3 GB RAM) were 0.31 ± 0.03 sec for the SV method, and 1.24 ± 0.37 sec for the GGMRF method: hence spatial regularization leads to a fourfold increase in computation time.

The following data model (Lorentz model) for the MRSI signal was used:

$$g_\theta(t_n) = \sum_{m=1}^{M} a_m \exp\Big(\big(-(d_m^{(0)} + d_m) + 2\pi i(f_m^{(0)} + f_m)\big)t_n + i\phi_m\Big) \tag{1.21}$$

$M = 3$ metabolites were considered (choline / creatine / NAA) with expected frequency shifts $f_m^{(0)}$ of 196.55 Hz / 216.02 Hz / 341.6 Hz at 3 Tesla corresponding to chemical shifts of 3.161 ppm / 3.009 ppm / 2.026 ppm and expected damping constants $d_m^{(0)}$ of 8 s^{-1} for all three metabolites. a_m denotes the relative amplitudes (i.e. the parameters of interest to be estimated during quantification), ϕ_m denotes the phase shifts, and d_m and f_m are correction terms for the damping factors and frequency constants (corresponding to the width and the position of the Lorentz resonance lines). It is also possible to model the resonance lines as Voigt profiles (with an additional Gaussian damping term), which was neglected here. The resulting optimization problem therefore contains twelve free parameters per voxel (ten, if a common phase shift is shared across all metabolites, i.e. if the constraint $\phi_1 = \phi_2 = \phi_3$ is introduced). The spatial regularization term depends on five free parameters: the parameter p characterizing the p-norm and the entries w_a, w_f, w_d and w_ϕ of the diagonal weight matrix W, which control how much amplitude, frequency, damping and phase gradients are penalized (these values are shared across the different metabolites). As proposed by Kelm (2007), the parameter combination $p = 2$, $w_\mathrm{a} = 0$, $w_\mathrm{f} = 2$, $w_\mathrm{d} = 0.2$ and $w_\phi = 20/\pi$ was used for most experiments (which had there been determined from the variograms of the fitted parameter maps for another proband dataset). Note that the amplitudes are usually not explicitly regularized, since it suffices to regularize the other parameters, and since the eventual interest is on the amplitudes and any bias on them shall be avoided.

1.6. Preliminary evaluation by single rater (unblinded)

An objective evaluation of the GGMRF versus the classical SV quantification method is not possible, since the true metabolite concentrations inside a living brain are unknown and cannot be measured. MRSI phantoms (tubes containing metabolite of defined concentrations) are typically employed for the evaluation of SV quantification techniques, but they are inappropriate for comparing spatially resolved quantification methods, as the concentration is typically uniform inside the tube and there is no way to generate smooth concentration gradients. Hence a subjective evaluation approach was chosen: as long as the main metabolite peaks are identifiable (SNR > 1), a trained human can usually judge whether they are captured by the fitted model peaks. Fig. 1.2 shows an exemplary spectrum with its SV fit and several spatially regularized fits, which can be clearly distinguished into "good" and "poor" fits.

In a preliminary subjective comparison of the SV and GGMRF fits, the quality of each fit was labeled as "good" or "poor". With the standard settings of the algorithm as detailed above, 2 % of the "poor" SV fits could be improved to a "good" GGMRF

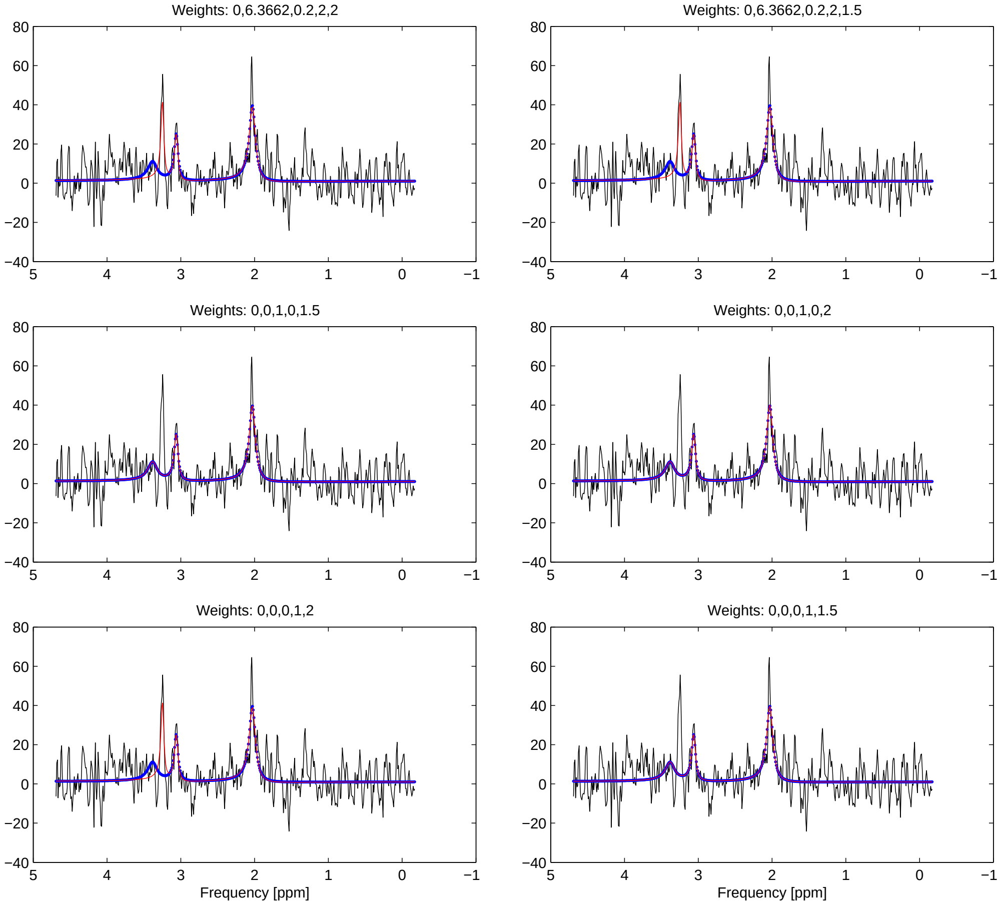


Figure 1.2. – Example spectrum [series 1, scan 3, voxel (7,5)] in the frequency domain (black) with SV fit (blue) and several GGMRF fits with different parameters (SP for "spatially regularized", red). Only the real parts of the complex spectra are shown. The black and the blue curve are identical for all six subplots, but the red (regularized) fits differ, as they correspond to different regularization parameters. These are listed in the subplot titles: e.g. "Weights: 0,6.3662,0.2,2,1.5" stands for $w_\mathrm{a} = 0$, $w_\phi = 20/\pi$ (6.3662), $w_\mathrm{d} = 0.2$, $w_\mathrm{f} = 2$, $p = 1.5$. The fit quality was rated "good" for the spatially regularized fits with parameters (0,6.3662,0.2,2,2), (0,6.3662,0.2,2,1.5) and (0,0,0,1,2) and "poor" for the single-voxel fit and the other spatially regularized fits, based on the criterion whether the choline peak was identified correctly or not. Note the particular relevance of frequency regularization for this example, which could be confirmed in the evaluation of the other spectra.

fit by the spatial regularization, while none of the "good" SV fits were degraded to "poor" GGMRF fits.

The following modifications of the algorithm were also tried:

1. Different value combinations for the weighting parameters w_{a}, w_{f}, w_{d} and w_{ϕ} and the norm parameter p.
2. Augmenting the SV and GGMRF models with a semiparametric baseline estimation to account for macromolecular background signals that cannot be modeled explicitly (as proposed by Sima & van Huffel (2006)).
3. Reparameterization of the frequency corrections f_m. The above data model assumes the central frequencies of the three metabolite peaks to jitter independently around their respective expected values. However, the mismatch between expected and true central frequencies may also be due to a miscalibrated frequency axis (e.g. if the local magnetic field deviates from exactly 3 Tesla). In this case, it is preferable to correct all metabolite frequencies by a common scale factor and offset, and then to add metabolite-specific frequency jitter within narrower bounds.
4. Constraining the phase shifts ϕ_m of the three metabolites to have equal values.

However, none of these modifications yielded better results than the 2 % improvement by the standard settings: the results were either comparable or worsened. Hence the standard settings were subjected to a decisive evaluation, thereby avoiding the multiple-comparison problem in statistical hypothesis testing (Shaffer, 1995).

1.7. Decisive evaluation by two raters (blinded) and results

The above preliminary analysis is insufficient for establishing the superiority of the GGMRF method over the SV method in a scientifically sound manner. The main reason is that it was performed unblinded (i.e. with the human rater knowing which curve corresponds to the SV fit and which curve corresponds to the GGMRF fit). Since the decision whether a fit is “good” or “poor” is necessarily subjective, the labels will be involuntarily biased by the prior expectations of the labelers, even if they try their best to label the fits carefully and fairly. Hence a subsequent decisive analysis was conducted, which was blinded: each spectrum was plotted twice with the two different fit curves (with no indication of the underlying model) and all plots were jumbled randomly. Two independent raters labeled the fit quality of each curve as either “good” or “poor” as above (for a “good” label, all three metabolite peaks had to be found with the correct peak position, width and amplitude).

Additionally the signal quality of each spectrum was labeled by the two raters as either “good”, “noisy” (SNR for the choline and creatine peaks < 1) or “containing artifacts” (presence of unidentifiable broad signal components in the spectrum, possibly caused by lipids). In borderline cases, the label “containing artifacts” took precedence over “noisy”. Fig. 1.3 shows examples of these three signal quality classes. Since every spectrum is plotted twice (once with the SV fit curve and once with the GGMRF fit curve) and the spectra are in random order, we get two independent signal quality labels from each rater. These labels were gathered in order to study the conditions more carefully under which spatial regularization leads to improved fits: For spectra degraded by considerable artifacts, no quantification method is expected to work well, and hence a beneficial effect of GGMRF may be diluted if these examples are included in the analysis. On the other hand, the spatial regularization is employed mainly to enhance the noise robustness of the fit and should hence prove advantageous especially on noisy spectra.

The main evaluation results are listed in Table 1.3: it shows the accuracies of the SV and GGMRF quantification for all of the twelve scans (i.e. the percentage of “good” fits among all spectra which have not been assigned a “containing artifacts” label by the respective rater). The alternative hypothesis that GGMRF quantification leads to an increase in this percentage, was tested against the null hypothesis that there is no effect. As the two raters clearly have differently strict criteria both for a good fit and for a good spectrum, a separate test was conducted for each rater. The values of the percentages also vary considerably between the different scans, which is understandable due to the differences in voxel size and hence in SNR. Hence an one-sided signed-rank test (Wilcoxon, 1945) was employed, which only assumes that the percentage differences between the two quantification methods are sampled independently from the same distribution, which is symmetric around its mean μ: the alternative hypothesis then corresponds to $\mu > 0$, while the null hypothesis corresponds to $\mu \leq 0$. The p-values were 0.0033 for rater A and 0.0294 for rater B, i.e. there is significant evidence that GGMRF indeed leads to an improved fit accuracy. However, the absolute value of the difference is small: the average improvement

$$\text{Accuracy of GGMRF} - \text{Accuracy of SV}$$

is 1.53 % for rater A and 1.25 % for rater B, while the average relative improvement

$$\frac{\text{Accuracy of GGMRF}}{\text{Accuracy of SV}} - 1$$

is 4.1 % for rater A, and 1.8 % for rater B. Fig. 1.4 shows the absolute and relative accuracy improvements as a function of in-plane resolution. As could be expected, the improvements by the spatial regularization are particularly pronounced for very small voxels: firstly, their smaller SNR causes the NLLS fit to be more prone to run into local maxima, and secondly, the spatial smoothness assumptions are obviously fulfilled better for smaller voxels.

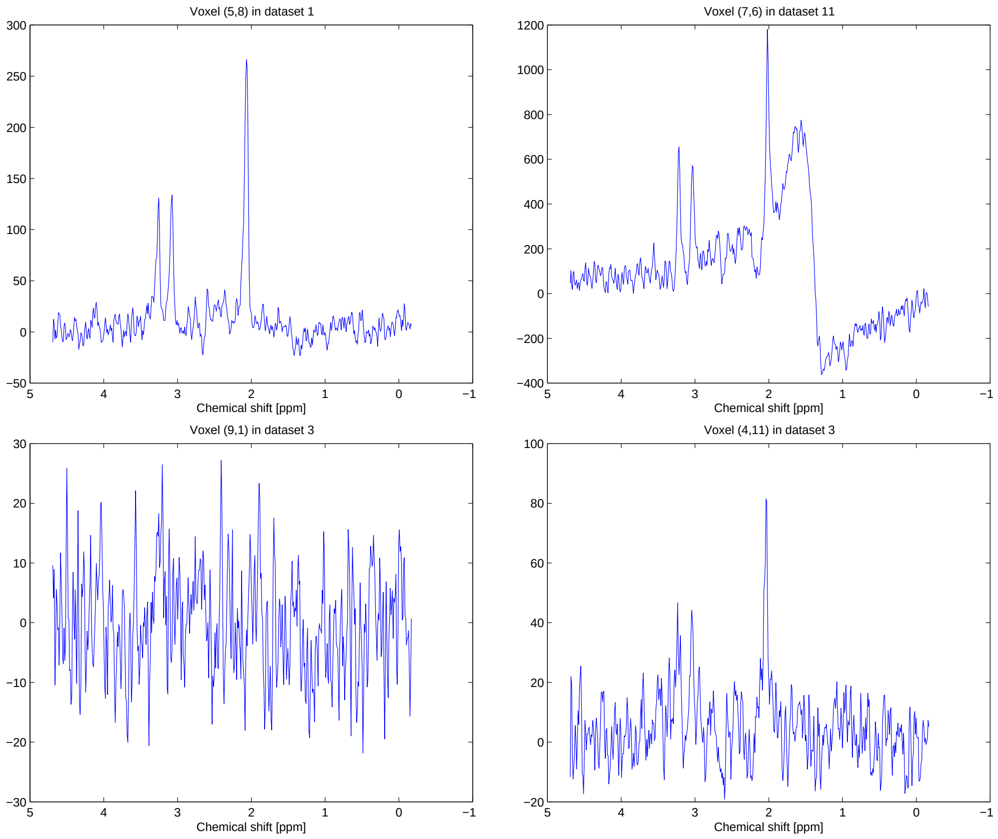


Figure 1.3. – Example spectra for the different signal quality labels. The first three spectra are exemplary for their respective quality classes and received unanimous votes: the top left spectrum was labeled “good” four out of four times, the top right spectrum was always labeled as “containing artifacts” and the bottom left spectrum was always labeled as “noisy”. The bottom right spectrum is a typical borderline example: each of the two raters labeled it once as “good” and once as “noisy”. The datasets in the two measurement series are labeled from 1 to 12, hence “Dataset 11” means the fifth scan in the second measurement series.

1.8. Alternative proposal: Regularized initialization by graph cuts

If the NLLS fit fails on a good-quality spectrum, this is typically due to one of the following three reasons: Either one peak in the spectrum is interpreted both as the choline and as the creatine peak (Fig. 1.5(a)), or the true choline peak is erroneously interpreted as the creatine peak and a small noise peak between the creatine and

	Rater A		Rater B	
Scan number	SV	GGMRF	SV	GGMRF
1	84.38 %	85.94 %	93.75 %	93.75 %
2	50.00 %	49.06 %	80.70 %	79.82 %
3	23.02 %	24.46 %	58.33 %	56.94 %
4	61.06 %	62.83 %	86.24 %	88.07 %
5	72.88 %	74.58 %	95.45 %	98.48 %
6	19.64 %	25.60 %	52.07 %	57.99 %
7	95.31 %	96.88 %	96.88 %	96.88 %
8	92.19 %	93.75 %	93.75 %	93.75 %
9	93.68 %	93.68 %	93.68 %	93.68 %
10	94.74 %	95.79 %	94.74 %	95.79 %
11	80.36 %	82.14 %	81.25 %	83.93 %
12	69.03 %	69.91 %	70.37 %	73.15 %

Table 1.3. – Percentage of SV and GGMRF fits that are labeled as "good" by the two raters, among all spectra in a scan that are assigned a "good" signal quality label by the respective rater. Scans 7–12 refer to the scans in the second acquisition series.

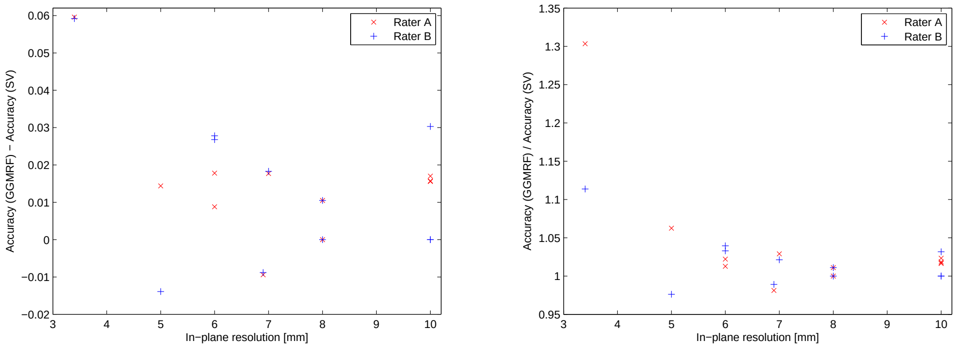
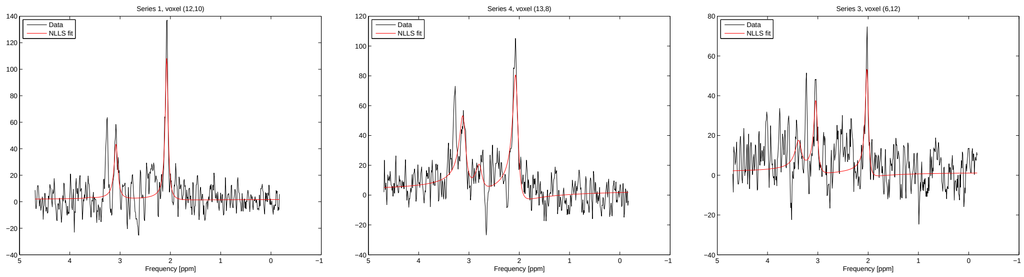


Figure 1.4. – Absolute and relative accuracy improvement of GGMRF quantification over SV quantification, as a function of in-plane voxel resolution, for the two raters.

---

the NAA peak is misinterpreted as the choline peak (Fig. 1.5(b)), or several small peaks are fitted as one by one overly wide peak instead of the correct (narrow) peak (Fig. 1.5(c)).

In order to analyze the reasons why the NLLS quantification fails, it is instructive to compare the actual peak positions in several spectra from one slice with their expected values, which can be computed from the B_0 field, the temporal sampling



(a) Merged choline and creatine peak

(b) Choline peak interpreted as creatine peak

(c) Several small peaks fitted as one

Figure 1.5. – Exemplary spectra showing the reasons for poor NLLS fits. The real part of spectra in the frequency domain is shown.

rate and the literature values of the chemical shift δ, e.g. as reported by Govindaraju et al. (2000). Fig. 1.6 shows a representative example: obviously the expected peak positions are systematically shifted with respect to their actual values. This phenomenon is probably caused by a small systematic deviation of either the B_0 field or the temporal sampling rate from their nominal values. Note that this is a plausible explanation for fitting results like in Figs. 1.5(a) or 1.5(b): if the initial position of the choline resonance in the model is closer to the real creatine peak than to the real choline peak, it gets fitted to this creatine peak, and the creatine resonance in the model gets either fitted to the same creatine peak (as in Fig. 1.5(a)) or to some other noise or nuisance peak (as in Fig. 1.5(b)).

Models like in Eqs. (1.16) or (1.18) that vary the parameters of each resonance in the model separately are ill-suited to correct such systematic errors. One possible solution would be to introduce couplings between the parameters of the different resonance, e.g. a repulsion term that prevents different resonances to be mapped to the same peak in the spectrum. However, a much simpler alternative is to initialize the model fitting by finding the optimal joint alignment between the model resonances and the spectrum: For this initialization, we simplify the nonlinear fitting problem in Eq. (1.16) by keeping the damping constants of the model (1.21) fixed ($d_1 = d_2 = d_3 = 0$) and constraining the frequency shifts to be equal for all metabolites ($f_1 = f_2 = f_3 = f$). Then f is the only remaining nonlinear parameter in Eq. (1.16). For a given value of f, the linear parameters (amplitudes and complex phases) and hence also the least-squares residuals can be computed in closed form: Let $y \in \mathbb{R}^N$ denote the complex signal time course as stacked into a column vector, $X(f) \in \mathbb{R}^{N \times M}$ be a matrix with entries

$$X_{nm}(f) = \exp\left[\left(-d_m^{(0)} + 2\pi i(f_m^{(0)} + f)\right)t_n\right] \tag{1.22}$$

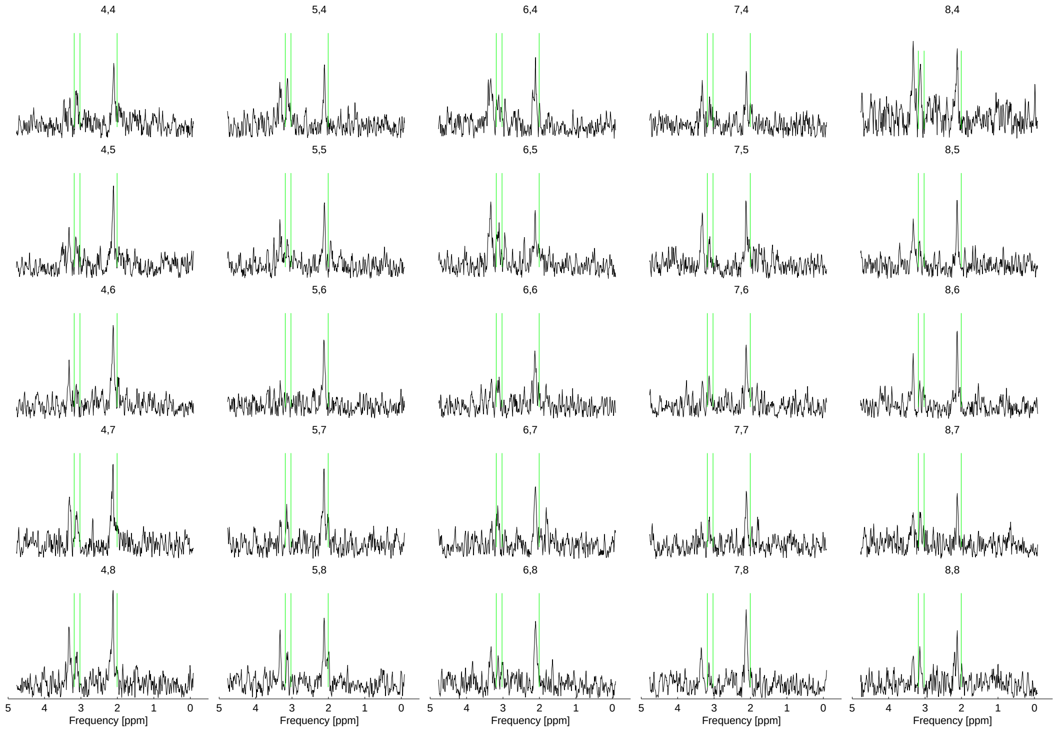


Figure 1.6. – Subgrid of magnitude spectra from dataset 3: the plot titles give the x and y index in the slice. The vertical green bars indicate the expected peak positions for the three main metabolite resonances, based on the nominal B_0 field strength: from left to right, they correspond to choline, creatine and NAA. One sees clearly that the actual peak positions are systematically shifted in all of the spectra.

---

and $b \in \mathbb{R}^M$ be a complex vector that comprises the metabolite amplitudes and their complex phases via $b_m = a_m e^{i\phi_m}$. Then the minimum residual sum of squares (RSS) for a given f is

$$\mathrm{RSS}(f) = \arg\min_b \|y - X(f)b\|^2 \tag{1.23}$$

$$= \|y - X(f)\big(X(f)^\dagger X(f)\big)^{-1} X(f)^\dagger y\|^2, \tag{1.24}$$

where $X(f)^\dagger$ denotes the Hermitian adjoint.

If we assume the metabolite signals to be non-overlapping in frequency space (i.e. the columns of $X(f)$ to be approximately orthogonal), Eq. (1.24) can be simplified considerably.[10] In this case, $X(f)^\dagger X(f) \approx N \cdot \mathbb{I}$ becomes nearly diagonal, and we can write

$$\begin{aligned} \mathrm{RSS}(f) &\approx \left\| y - \frac{1}{N} X(f) X(f)^\dagger y \right\|^2 && (1.25) \\ &\approx \|y\|^2 - \frac{1}{N} \|X(f)^\dagger y\|^2 && (1.26) \\ &= \|y\|^2 - \frac{1}{N} \left(|c_1(f)|^2 + |c_2(f)|^2 + \cdots + |c_M(f)|^2 \right), && (1.27) \end{aligned}$$

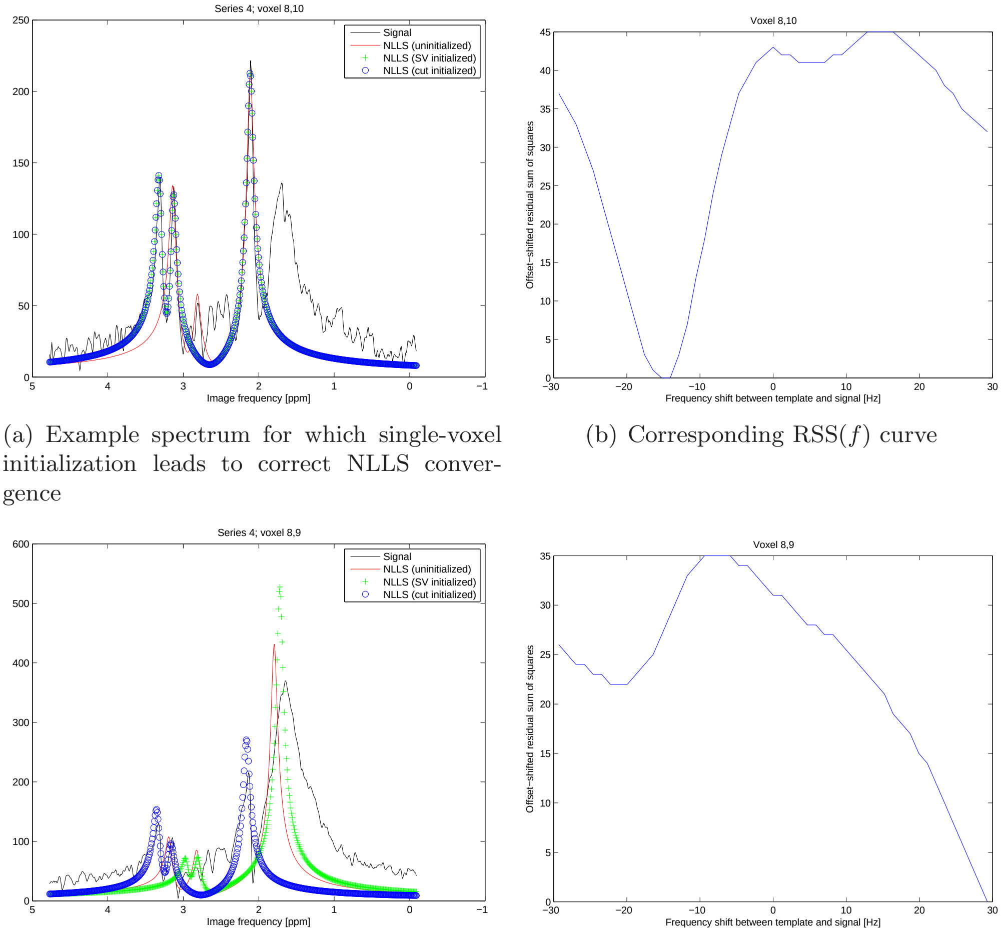
where $c_i(f) = X_i(f)^\dagger y$ and $X_i(f)$ is the i-th column of $X(f)$. Note that the cross-correlations $c_i(f)$ can be also computed from the Fourier transforms of y and $X_i(f)$ according to the unitarity of the Fourier transform (i.e. Parseval's theorem). Since the Fourier transform of the damped harmonic oscillation $X_i(f)$ is a Lorentzian, and varying f corresponds to shifting this Lorentzian along the frequency axis, the cross-correlations $c_i(f)$ can be efficiently computed for different values of f using a convolution. Then a line search can be performed to find the optimal frequency shift $f^* = \arg\min \mathrm{RSS}(f)$, and use this for initialization of the NLLS optimizer. Fig. 1.7(a) shows a spectrum for which the uninitialized NLLS fit fails. If the NLLS fit is run after initializing the frequency search values correctly (by a constant shift found from Eq. (1.27)), the correct minimum is found. Fig. 1.7(b) shows the corresponding $\mathrm{RSS}(f)$ curve: the correct initialization shift at -20 Hz is the global minimum of the curve.

In the presence of very strong spectral artifacts, the initialization according to Eq. (1.27) may cause the NAA peak to be mapped to the artifact signal instead of the true NAA signal peak (see Fig. 1.7(c)). Note that in the experiments this only happened for spectra which were labeled as "containing artifacts" by both raters, and which were therefore excluded from the evaluation in section 1.7. When examining the graph of the function $\mathrm{RSS}(f)$ for these pathological spectra, one notes that the true initialization appears as a local minimum, which is however overshadowed by the global minimum corresponding to the artifact signal (Fig. 1.7(d)). In this case, incorporating spatial context in the spirit of Eq. (1.18) is a plausible remedy: the initialization constants f_v of the different voxels are coupled by a GGMRF prior, and the joint optimum is found by solving

$$f^* = \arg\min_f E(f) = \arg\min_f \lambda \sum_v \mathrm{RSS}(f_v) + \sum_{v \sim w} |f_v - f_w|^p. \qquad (1.28)$$

While the pair potential is a convex function in the vector f, the single-site potentials $\mathrm{RSS}(f_v)$ are in general not convex: hence Eq. (1.28) cannot be tackled by convex

[10] This assumption holds very well between the NAA resonance and the two other resonances, but less well between the choline and the creatine resonance. However, since this step is only meant as a rough initialization of the fitting process, and the peak positions are refined afterwards, the increase in simplicity and computation speed warrants the slight inaccuracy.



(a) Example spectrum for which single-voxel initialization leads to correct NLLS convergence

(b) Corresponding RSS(f) curve

(c) Neighboring spectrum for which spatially regularized initialization is required

(d) Corresponding RSS(f) curve

Figure 1.7. – Exemplary spectra showing the benefits of single-voxel and regularized initialization. Note that the spectrum in Fig. 1.7(a) and other similar spectra are directly adjacent to the spectrum in Fig. 1.7(c). Hence the smoothness prior on the frequency initialization shift can be used to evade the global minimum caused by the artifact peak in Fig. 1.7(d). For illustration purposes, the RSS(f) curves were offset-shifted so that the minimum value of the curve is always zero: this does not influence the solution of the optimization problem.

---

optimization techniques. Using an ICM or block-ICM procedure as for the problem in Eq. (1.18) would be possible, but with possibly slow convergence and without any guarantee that the global optimum is attained eventually.

However, this problem differs from the one in Eq. (1.18) in that the state of each voxel can be described by a single frequency shift scalar f_v instead of several variables (frequency shifts, dampings and phases of several metabolites). Using an appropriate discretization for the f_v, the exact joint minimum can be computed efficiently by modelling it as a graph cut problem as in (Ishikawa, 2003). In general, for a set of linearly ordered labels l_i, the minimization problem

$$l^* = \arg\min_l \sum_i \psi_i(l_i) + \sum_{i \sim j} g(l_i - l_j) \tag{1.29}$$

for arbitrary single-site potentials ψ_i and an arbitrary convex function g can be transformed into an equivalent min-st-cut problem, which is then solved using e.g. the dual-tree max-flow algorithm (Boykov et al., 2001; Kolmogorov & Zabih, 2004; Boykov & Kolmogorov, 2004). Experimentally, it was shown that this max-flow implementation gives the best results for graph cut problems of this structure (Boykov & Kolmogorov, 2004). Note the conceptual difference from the GGMRF model and its block-ICM optimization heuristic described earlier: Instead of imposing a smoothness prior on the final model parameters, the regularization only affects their initialization value (i.e. their rough location), and they are then refined by a usual single-voxel NLLS optimization. Further differences are that only one nonlinear parameter is optimized over (the most important one, namely the global frequency calibration), and that therefore the global optimum for this single parameter can be found efficiently in contrast to the local optimality of block-ICM.

Tables 1.4 and 1.5 show the accuracy improvements of the NLLS quantification procedure by the single-voxel and the spatially regularized (graph-cut) initialization over the basic NLLS method where no special initialization is performed: for Table 1.4, all spectra are considered, while Table 1.5 only pertains to artifact-free spectra in analogy to Table 1.3. For the weighting factor from Eq. (1.28), $\lambda = 20$ was used, and the spatial prior was chosen to be linear ($p = 1$). It can be seen that already the single-voxel initialization leads to considerable improvements over the basic NLLS quantification, which are much more pronounced than the improvements by the GMRF prior on the fit parameters. The additional smoothness prior on the common frequency initialization shifts is mainly beneficial for artifact-containing spectra, but also gives small improvements over the single-voxel initialization for the artifact-free, but noisy spectra in e.g. series 6. The improvement of the single-voxel initialization over the uninitialized NLLS quantification is highly significant both when analyzing all spectra and when analyzing only the artifact-free spectra (in both cases $p = 1.26 \times 10^{-3}$ for a one-sided Wilcoxon test, if a fit with "wrong amplitudes" is counted as "poor"). In contrast, the improvement of the spatially regularized over the single-voxel initialization is significant only when considering all spectra ($p = 0.0113$), while $p = 0.0907$ when only the artifact-free spectra are considered. Figs. 1.8(a) and 1.8(b) show the accuracies as a function of in-plane resolution: as can be expected, the benefits of

	"Wrong amplitudes" as "poor"			"Wrong amplitudes" as "good"		
Series	NoInit	SVInit	GCInit	NoInit	SVInit	GCInit
1	84.4 %	100.0 %	100.0 %	84.4 %	100.0 %	100.0 %
2	42.1 %	97.5 %	97.5 %	43.0 %	97.5 %	97.5 %
3	21.5 %	92.4 %	93.1 %	22.2 %	93.8 %	95.1 %
4	42.6 %	83.4 %	88.8 %	42.6 %	83.4 %	98.8 %
5	50.0 %	85.0 %	89.0 %	50.0 %	86.0 %	96.0 %
6	23.1 %	76.9 %	82.8 %	24.9 %	82.8 %	87.6 %
7	96.9 %	100.0 %	100.0 %	96.9 %	100.0 %	100.0 %
8	93.8 %	100.0 %	100.0 %	93.8 %	100.0 %	100.0 %
9	89.0 %	100.0 %	100.0 %	89.0 %	100.0 %	100.0 %
10	90.0 %	98.0 %	100.0 %	90.0 %	98.0 %	100.0 %
11	55.0 %	79.9 %	90.5 %	55.0 %	80.5 %	93.5 %
12	49.1 %	76.3 %	87.6 %	49.1 %	77.5 %	89.9 %

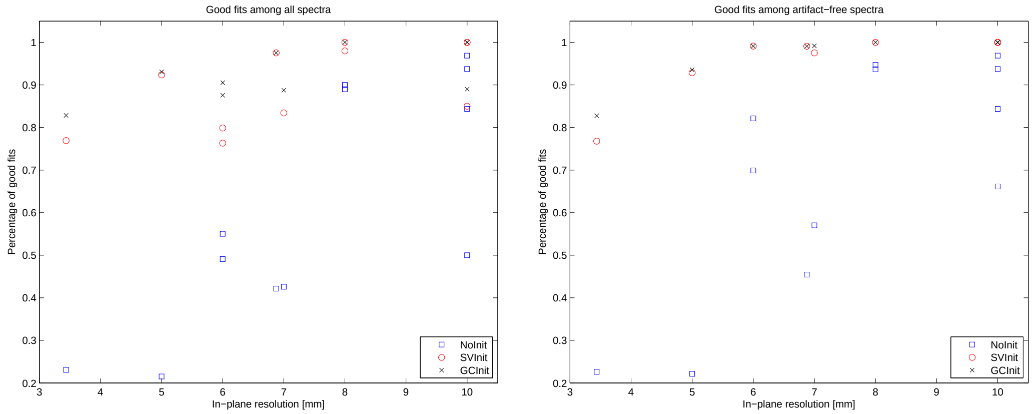
Table 1.4. – Ratio of good NLLS fits among all spectra, for three different initialization schemes of the frequency shifts: setting all to zero ("NoInit"), single-voxel initialization as by Eq. (1.27) ("SVInit") and spatially regularized graph cut initialization as by Eq. (1.28) ("GCInit"). Note that spectra with artifacts were not discarded before computing these numbers. The difference between columns 2–4 and columns 5–7 lies in how fits with a "wrong amplitudes" label were treated: in the former case, they were considered as "poor" fits, while in the latter case, they were considered to be "good" fits.

the initialization are the highest for highly resolved MRSI measurements with a poor SNR, for which NLLS is likely to run into local minima, as for the GGMRF model.

The computation times are shown in Fig. 1.9. Apparently, using a single-voxel initialization even saves time over the uninitialized NLLS fit (40 % on average): Computing the initialization is very fast, since all computations can be implemented via one-dimensional convolutions in the approximate formulation of Eq. (1.27), and the accelerated convergence of the subsequent NLLS fitting more than makes up for this initial investment. In contrast, using the spatially regularized initialization leads to an increase in computation time by 57 % on average, since solving the graph-cut optimization problem is costly. However, this is still well beneath the computation times required by the block-ICM algorithm.

	"Wrong amplitudes" as "poor"			"Wrong amplitudes" as "good"		
Series	NoInit	SVInit	GCInit	NoInit	SVInit	GCInit
1	84.4 %	100.0 %	100.0 %	84.4 %	100.0 %	100.0 %
2	45.5 %	99.1 %	99.1 %	46.4 %	99.1 %	99.1 %
3	22.1 %	92.9 %	93.6 %	22.9 %	94.3 %	95.7 %
4	57.0 %	97.5 %	99.2 %	57.0 %	97.5 %	99.2 %
5	66.2 %	100.0 %	100.0 %	66.2 %	100.0 %	100.0 %
6	22.6 %	76.8 %	82.7 %	24.4 %	82.7 %	87.5 %
7	96.9 %	100.0 %	100.0 %	96.9 %	100.0 %	100.0 %
8	93.8 %	100.0 %	100.0 %	93.8 %	100.0 %	100.0 %
9	93.7 %	100.0 %	100.0 %	93.7 %	100.0 %	100.0 %
10	94.7 %	100.0 %	100.0 %	94.7 %	100.0 %	100.0 %
11	82.1 %	99.1 %	99.1 %	82.1 %	100.0 %	100.0 %
12	69.9 %	99.1 %	99.1 %	69.9 %	99.1 %	99.1 %

Table 1.5. – Ratio of good NLLS fits among artifact-free spectra, for three different initialization schemes of the frequency shifts (as in Table 1.4). All spectra were discarded for which at least one of the two signal quality labels by rater A (see Table 1.3) was "containing artifacts". The differences between the numbers in the second column of this table, and the numbers in the second column of Table 1.3 are due to the limited intra-rater reliability.



(a) All spectra used (as in Table 1.4)

(b) Artifact spectra discarded (as in Table 1.5)

Figure 1.8. – Accuracy, i.e. percentage of "good" fits among all, for three different initialization schemes (see caption of Table 1.4), plotted against the in-plane voxel resolution.

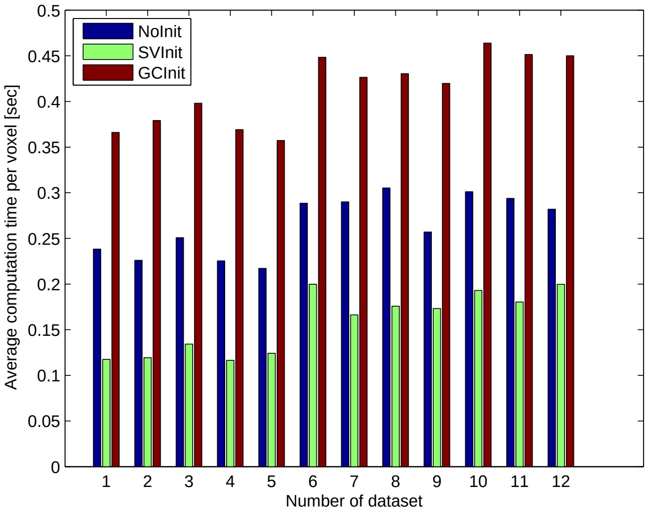


Figure 1.9. – Average computation time per voxel for quantifying the different datasets by the NLLS method, both without any initialization (`NoInit`), with a single-voxel initialization (`SVInit`) as given by Eq. (1.27) and with a spatially regularized initialization (`GCInit`) as given by the graph cut functional in Eq. (1.28).

Chapter 2.

Software for MRSI analysis

2.1. Introduction and motivation

Imaging methods for the *in vivo* diagnostics of tumors fall into three categories based on the different physical mechanisms they exploit: In computer tomography (CT), X-rays are transmitted through the body, which are attenuated differently in different tissue types. In nuclear medicine methods such as positron emission tomography (PET) or single photon emission computed tomography (SPECT), one detects the radiation of radioactive nuclides, which are selectively accumulated in the tumor region. Finally, magnetic resonance imaging (MRI) exploits the fact that various nuclei (namely protons) have a different energy when aligned in the direction of an external magnetic field than when they are aligned opposite to it. By injecting a radiofrequency wave into the imaged body, one can perturb some protons out of their equilibrium state into a higher-energy state: the radiofrequency signal which they emit upon relaxation is then measured, and its amplitude is proportional to the concentration of the protons in the imaged region. This measurement process can be performed in a spatially resolved fashion, so that a three-dimensional image is formed.

Standard MRI produces a scalar image based on the total signal of all protons, irrespective of the chemical compound to which they belong: typically, the protons in water molecules and in lipids make the highest contribution due to the large concentration of these molecules. However, the protons in different compounds can be distinguished by their resonance frequencies in the magnetic field (the so-called *chemical shift*), and it is possible to resolve the overall signal not only spatially, but also spectrally: this leads to magnetic resonance spectroscopy imaging (MRSI) or chemical shift imaging (CSI), for which a complex spectrum is obtained at each image voxel instead of a single scalar value as in MRI (de Graaf, 2008). Hence it is possible to measure the local abundance of various biochemical molecules non-invasively, and thereby gain information about the chemical make-up of the body at different locations: besides water and lipids, most major metabolites can be identified in the MRSI spectra, e.g. the most common amino acids (glutamate, alanine, ...), the reactants and products of glycolysis (glucose, ATP, pyruvate, lactate), precursors of membrane biosynthesis (choline, myo-inositol, ethanolamine), energy carriers (creatine) and tissue-specific marker metabolites (citrate for the prostate, N-acetylaspartate or NAA for the brain). As a downside, these metabolites occur in much lower concentrations than water, hence the spatial resolution must be far coarser than in MRI: only by collecting signal from a volume of typically 0.2–2 cm^3, a sufficient signal-to-noise ratio can be achieved.

MRSI provides valuable information for the noninvasive diagnosis of various human diseases, e.g. infantile brain damage (Xu & Vigneron, 2010), multiple sclerosis (Sajja et al., 2009), hepatitis (Cho et al., 2001) or several psychiatric disorders (Dager et al., 2008). The most important medical application field lies in tumor diagnostics, especially in the diagnosis and staging of brain, prostate and breast cancer as well as the monitoring of therapy response (Gillies & Morse, 2005). In tumors, healthy cells are destroyed and the signals of the biomarkers characteristic for healthy tissue (e.g. citrate for the prostate, NAA for the brain) are decreased. On the other hand, biomarkers for pathological metabolic processes often occur in increased concentrations: choline (excessive cell proliferation), lactate (anaerobic glycolysis), mobile lipids (impaired lipid metabolism). The top right and bottom right spectra in Fig. 2.1 are typical examples of spectra occurring in healthy brain tissue and in brain tumor, respectively.

While MRSI has proved its efficacy for radiological diagnostics, it is a fairly new technique that yet has to gain ground in routine radiology and in the training curricula of radiologists. Furthermore, the visual assessment is harder and more time-consuming than for MRI: while most medical imaging modalities provide two- or three-dimensional data, MRSI provides four-dimensional data due to the additional spectral dimension. Automated decision-support systems may assist the radiologists by visualizing the most relevant information in form of easily interpretable *nosologic images* (de Edelenyi et al., 2000): from each spectrum, a scalar classification score is extracted that discriminates well between healthy and tumorous tissue, and all scores are displayed as a color map. Ideally the scores can even be interpreted as the probability that the respective spectrum corresponds to a tumor. While such a decision support system may not completely obviate the need of manual inspection of the spectra, it can at least guide the radiologist towards suspicious regions that should be examined more closely, and facilitate the comparison with other imaging modalities.

Methods for computing the classification scores fall into two categories: quantification-based approaches (Poullet et al., 2008) and pattern recognition-based approaches (Hagberg, 1998). Quantification approaches exploit the fact that MRSI signals are

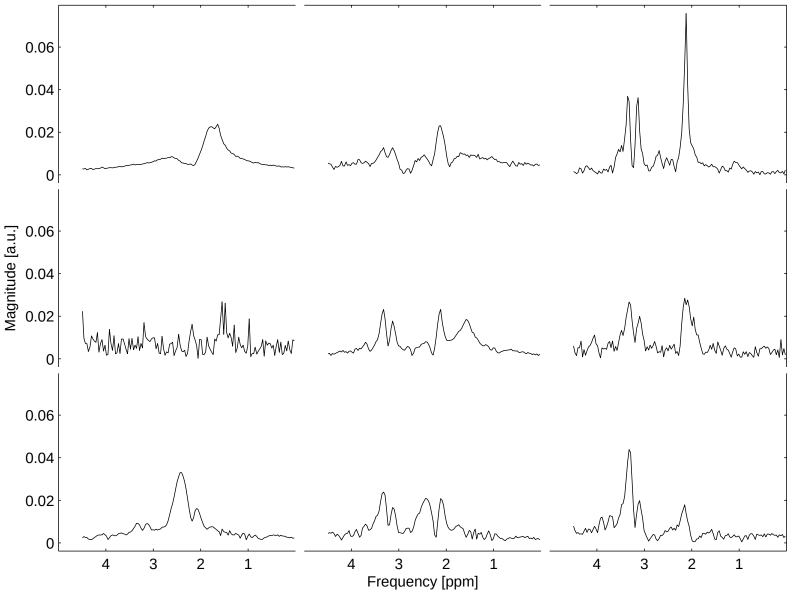


Figure 2.1. – Exemplary MRSI magnitude spectra of the brain, showing different voxel classes and signal qualities. All spectra have been water-suppressed and L_1 normalized (i.e. divided by the sum of all channel entries), and they are displayed on a common scale. Note the three distinct metabolite peaks, which are characteristic for brain MRSI: Choline (3.2 ppm), creatine (3.0 ppm) and N-acetylaspartate (NAA, 2.0 ppm). NAA is a marker for functional neurons, hence it has a high concentration in healthy tissue, and a low concentration in tumor tissue. On the other hand, choline is a marker for membrane biogenesis and has a higher concentration in tumor tissue than in healthy tissue. Left column: Spectra that are not evaluable owing to poor SNR or the presence of artifacts. Middle column: Spectra with poor signal quality, which however have sufficient quality so that the voxel class may be ascertained. Right column: Spectra with good signal quality. Top row: Spectra from healthy brain tissue. Middle row: Spectra of undecided voxel class. Bottom row: Spectra from tumor tissue. Note that the voxel class is only meaningful for the middle and the right column, and that the spectra in the left column were randomly assigned to the different rows.

physically interpretable as superpositions of metabolite spectra; they can hence be used to quantify the local relative concentrations of these metabolites by fitting measured or simulated basis spectra to the spectrum in every voxel. The fitting parameters (amplitudes, frequency shifts, ...) may be regarded as a low-dimensional representation of the signal. Classification scores are then usually computed from amplitude ratios of relevant metabolites: for instance, the choline/creatine and choline/-NAA ratios are frequently employed for the diagnosis of brain tumors (Martínez-Bisbal & Celda, 2009).

Pattern recognition approaches forego an explicit data model: instead, the MRSI signal is preprocessed to a (still high-dimensional) feature vector, and the mapping of feature vectors to classification scores is learned from manually annotated training vectors (the so-called *supervised learning* setting). Because of this need for manually annotated examples, pattern recognition techniques require higher effort from human experts than quantification-based techniques. Furthermore, they have to be retrained if the experimental measurement conditions change (e.g. different magnetic field strength, different imaged organ or different measurement protocol). However, comparative studies of quantification and pattern recognition methods for prostate tumor detection showed superior performance of the latter ones, as they are more robust against measurement artifacts and noise (Kelm et al., 2007). Given a sufficiently large and diverse training dataset, one can even use pattern recognition to distinguish between different tumor types, e.g. astrocytomas and glioblastomas (Tate et al., 2006).

MRSI data often have quality defects that render malignancy assessment difficult or even impossible: low signal-to-noise ratio, line widening because of shimming errors, head movement effects, lipid contamination, signal bleeding, ghosting etc. (Kreis, 2004). If these defects become sufficiently grave, even pattern recognition methods cannot tolerate them, and the resulting classification scores will be clinically meaningless and should not be used for diagnosis. Fig. 2.1 shows example spectra of good, poor, and very poor (not evaluable) quality for healthy, undecided and tumorous tissue. One can deal with this problem by augmenting the classification score for the malignancy (also called *voxel class*) with a second score for the signal quality: If this score is high, the users know that the spectrum has high quality and that the voxel class score is reliable, while for a low score they know that the voxel class score is unreliable and the spectrum should be ignored. This may also save the users' time, as poor-quality spectra need not be examined in detail. Pattern recognition approaches have been successfully employed for signal quality prediction, with similar performance to expert radiologists (Menze et al., 2008).

Most existing software products for MRSI classification incorporate quantification-based algorithms: for instance, they are typically included in the software packages supplied by MR scanner manufacturers. Furthermore, there are several stand-alone software products such as LCModel (Provencher, 2001), jMRUI (Stefan et al., 2009) or MIDAS (Maudsley et al., 2006).

In contrast, the application of pattern recognition-based methods still has to gain ground in clinical routine: This may be partially due to differences in the flexibility with which both categories of algorithms can be adjusted to different experimental conditions (e.g. changes in scanner hardware and in measurement protocols) or to a different imaged organ. For quantification-based methods one must only update the metabolite basis spectra to a given experimental setting, which can be achieved by quantum-mechanical simulation, e.g. with the GAMMA library (Smith et al., 1994). For pattern recognition-based methods on the other hand, one has to provide manual labels of spectra from many different patients with a histologically confirmed tumor, which is time-consuming and requires the effort of one or several medical experts. Since there exist many different techniques whose relative and absolute performance on a given task cannot be predicted beforehand, for every change in conditions a benchmarking experiment as in (Menze et al., 2006) or (García-Gomez et al., 2009) should also be conducted to select the best classifier and monitor the classification quality.

While the need for classifier retraining, benchmarking and quality assessment cannot be obviated, this chapter presents an object-oriented C++ library and a graphical user interface which assists this task better than existing software.[1] This work is an extension of the CLARET software (Kelm et al., 2006): While the original prototype of this software was written in MATLAB, an improved C++ reimplementation was created for the MeVisLab[2] environment. Most of the functionality described in this thesis does not exist in the original CLARET version and is hence novel: mainly the possibility to manually define labels and to train, test, evaluate and compare various classifiers and preprocessing schemes. The original software was only capable of analyzing MRSI data measured with a specific acquisition protocol (prostate measurements acquired with an endorectal coil at a 1.5 Tesla scanner with an echo time of 135 ms and a sampling interval of 0.8 ms). Retraining was only possible using both specialized tools and specialized knowledge about pattern recognition.

2.2. Background: Supervised classification

The following survey covers common knowledge; for a reference, see e.g. the book by Hastie et al. (2009).

[1] The contents of this chapter have been published as (Kaster et al., 2009, 2010a,b).

[2] http://www.mevislab.de

Aims and pitfalls of classification Supervised classification is a subarea of statistical learning. It deals with the following question: Assume we have a set of training examples with associated labels $\{(x_i, y_i)|i = 1, ..., n\} \subset \mathcal{X} \times \mathcal{Y}$, with a (discrete or continuous) feature space $\mathcal{X}$ and a finite label space $\mathcal{Y}$. In the following, we set $\mathcal{X} \subseteq \mathbb{R}^p$ and $\mathcal{Y} = \{0, \ldots, L-1\}$. A classifier is a rule that tells us which label $g(\hat{x})$ should be given to a new test example $\hat{x}$ for which the true label $\hat{y}$ is not known, based on the training input. Ideally one is also interested in estimates for the probabilities $\hat{p}_1, \ldots, \hat{p}_L$ that the label $\hat{y}$ belongs to the different possible classes, rather than a crisp assignment. The aim of classifier training is a low value for the expected classification error on a test example

$$\mathbb{E}_{(\hat{x},\hat{y})}\left[1 - \delta_{\hat{y},g(\hat{x})})\right] = p\big(\hat{y} \neq g(\hat{x})\big), \tag{2.1}$$

which is also known as the generalization error.[3] The theoretically optimal classifier (with the smallest generalization error) is the Bayes classifier:

$$g(x) = \arg\max_y p(y|x). \tag{2.2}$$

However, the conditional distribution $p(y|x)$ is not known in practice. For a suitably large training set, the training error

$$\frac{1}{n}\sum_{i=1}^{n}\big(1 - \delta_{g(x_i),y_i}\big) \tag{2.3}$$

is a lower bound on the generalization error, but it may be a severe underestimation: there are classifiers which are closely tuned to the training set so that their training error can go down to zero, but which may perform very poorly on test examples (this phenomenon is called “overfitting”). Better estimates for the generalization error can be achieved by cross-validation: the training data are partitioned into different folds, the classifier is repeatedly trained on all but one folds and tested on the remaining fold, and the average of all empirical test errors is reported. However, one should note that cross-validation estimates are in general biased (Bengio & Grandvalet, 2004). Finally, the bias-variance trade-off is important for understanding the dependence of many classifiers on their free parameters: For sake of illustration, consider a binary classification ($L = 2$). Then,

$$1 - \delta_{\hat{y},g(\hat{x})} = \big(\hat{y} - g(\hat{x})\big)^2 \tag{2.4}$$

[3] The generalization error is the simplest example of a loss function, namely one that treats all misclassifications as equally grave. More flexible loss functions may also be defined, e.g. for the automated tumor classification application considered in this chapter, false positives might be considered more permissible than false negatives: Then the goal of the classifier is to minimize the expected value of this loss.

and the generalization error can be decomposed as follows:

$$\mathbb{E}_{(\hat{x},\hat{y})}\left[\big(\hat{y}-g(\hat{x})\big)^2\right] = \mathbb{E}_{(\hat{x},\hat{y})}\left[\Big(\hat{y}-\mathbb{E}_{\hat{x}}\big[g(\hat{x})\big]+\mathbb{E}_{\hat{x}}\big[g(\hat{x})\big]-g(\hat{x})\Big)^2\right] \tag{2.5}$$

$$= \mathbb{E}_{\hat{y}}\left[\Big(\hat{y}-\mathbb{E}_{\hat{x}}\big[g(\hat{x})\big]\Big)^2\right] + \mathbb{E}_{\hat{x}}\left[\Big(g(\hat{x})-\mathbb{E}_{\hat{x}}\big[g(\hat{x})\big]\Big)^2\right]. \tag{2.6}$$

The second term in Eq. (2.6) measures how the classifier predictions varies around its expected prediction value (the *variance*), while the first term measures by which amount the expected prediction value deviates from the true label (the *bias*). Many classifier parameters increase or decrease the local smoothness (or regularity) of the classifier: by adjusting them, one can often trade higher bias for lesser variance and vice versa. Often, the optimum compromise between these two conflicting factors is achieved at a moderate parameter value, which may be found e.g. via cross-validation.

k nearest neighbors Arguably one of the simplest supervised learning techniques is the k nearest neighbors (kNN) classifier: for every test point, find the k closest examples among the training data (with respect to a suitable metric on the feature space $\mathcal{X}$) and assign their majority label. Despite its simplicity, this classifier has good theoretical guarantees: e.g. in the limit of infinite training data, its generalization error is at most twice as large as the generalization error of the Bayes classifier (Stone, 1977). However, for limited training examples the parameter k becomes important: large values of k enforce regularity of the classifier and decrease variance, while possible incurring a bias. In contrast, small values of k commonly lead to small bias and large variance.

Decision trees and random forests Decision tree classifiers (Hastie et al., 2009, chap. 9.2) iteratively partition the feature space into orthotopes: A binary tree data structure is initialized with the entire space $\mathcal{X}$ as the root node, then the tree is grown by selecting the best axis-parallel split of a leaf node into two daughter nodes by an axis-parallel split, i.e. a rule of the form "if feature i is larger than a threshold θ, then go to the right child, else go to the left child". The best split is commonly defined as the one causing a maximum decrease in some measure in node impurity among the training examples: I.e. if the mother node contains N training examples, of which a fraction $p_0 \in [0,1]$ belongs to class 0 and a fraction $p_1 = 1 - p_0$ belongs to class 1, and the left and right child contain N_L and N_R examples with fractions $p_{\mathrm{L}0}$, $p_{\mathrm{L}1}$, $p_{\mathrm{R}0}$ and $p_{\mathrm{R}1}$, common criteria are searching for the maximum entropy decrease

$$-p_0\log p_0 - p_1\log p_1 + \frac{N_\mathrm{L}}{N}\left(p_{\mathrm{L}0}\log p_{\mathrm{L}0} + p_{\mathrm{L}1}\log p_{\mathrm{L}1}\right) + \frac{N_\mathrm{R}}{N}\left(p_{\mathrm{R}0}\log p_{\mathrm{R}0} + p_{\mathrm{R}1}\log p_{\mathrm{R}1}\right)$$

$$\tag{2.7}$$

or the maximum Gini impurity decrease

$$2p_0p_1 - 2\frac{N_\mathrm{L}}{N}p_{\mathrm{L}0}p_{\mathrm{L}1} - 2\frac{N_\mathrm{R}}{N}p_{\mathrm{R}0}p_{\mathrm{R}1}. \tag{2.8}$$

This process is ended either once a maximum tree depth is reached, or once node purity is reached (i.e. all leaf orthotopes contain only training examples from a single class). An unlabeled test example is then classified according to the majority label inside the orthotope in which it is contained. Single decision trees are prone to overfitting, especially if the tree is grown up to purity. Random forests (Breiman, 2001) confer higher robustness: instead of growing a single tree, an ensemble of randomized trees is grown, and unlabeled test examples are assigned the majority label of the tree predictions. In the most common variant, randomization occurs at two stages: Firstly, each single tree is only trained using a random subset of the training examples, which is generated by bootstrapping (i.e. sampling with replacements). The remaining examples can be used to estimate the generalization error of this tree (this is called the out-of-bag estimate). Secondly, only a random subset of $m_\mathrm{try} \ll p$ features is considered for each fit. This number $m_\mathrm{try} \ll p$ is the main adjustable parameter[4] for random forests, as it determines the balance between two conflicting aims of random forest generation: the trees should be diverse to avoid overfitting (which encourages small m_try values), but also give accurate predictions (which encourages large m_try values). The rule of thumb $m_\mathrm{try} = \sqrt{p}$ often provides a good compromise.

Linear regression and regularized variants Linear regression (Hastie et al., 2009, chap. 3) is originally a regression problem, which aims to predict continuous labels $y_i \in \mathbb{R}$: however, binary classification may be reduced to this setting by using 0 and 1 as the training labels and binarizing the continuous test predictions via a threshold at e.g. 0.5. It searches for the optimal linear relationship (in a least-squares sense) between the features and the labels: if all training labels are stacked in an $n \times 1$ vector y, and all training features in an $n \times p + 1$ matrix X, it solves the problem

$$w^* = \arg\min_w (y - X'w)^2 \quad \text{with } w \in \mathbb{R}^{p+1}. \tag{2.9}$$

In order to allow for a constant offset, we assume that the last column of X is a vector of ones. Especially in high-dimensional feature spaces ($n < p$), linear regression

[4] The second parameter is the number of trees. However, this is mostly determined by the time available for training and prediction: In most cases, more trees give better prediction accuracies, but the effect saturates, and both training and prediction time grow linearly in the number of trees.

becomes an ill-posed problem and may suffer from severe overfitting, poor numerical conditioning and poor robustness towards noise. The solution lies in regularizing the regression, i.e. in restricting the effective number of parameters to a value smaller than n. One possible approach lies in imposing a Gaussian prior on the weight vector, which leads to ridge regression (RR):

$$w^* = \arg\min_w (y - Xw)^2 + \lambda w^2 \tag{2.10}$$

Large values of λ force the weights w_j to be small,[5] and will additionally make the problem well-conditioned. A different approach is principal components regression (PCR): if $V = (v_1, \ldots, v_{n_{\mathrm{PC}}})$ is a matrix built of the n_{PC} principal components of the feature matrix X (i.e. the eigenvectors of $X^\top X$ corresponding to the leading eigenvalues), then PCR solves the optimization problem

$$w^* = \arg\min_w (y - XVw)^2. \tag{2.11}$$

Hence the dimensionality of the features is reduced from p to n_{PC}. Often the leading principal components carry most of the discriminative information of the features, while the other components are mainly noise variables. Concerning the bias-variance tradeoff mentioned above, large values of λ and small values of n_{PC} will decrease the variance, but possibly incur a bias. Note that both linear regression and its variants are linear estimators, i.e. the predictions $\hat{y}$ of the trained regressor for the training examples linearly depend on the labels:

$$\hat{y} = Sy, \tag{2.12}$$

with S being a function of X: e.g., for linear regression,

$$S = X(X^\top X)^{-1} X^\top. \tag{2.13}$$

For this kind of estimators, the leave-one-out cross-validation estimate for the generalization error can be efficiently approximated by the generalized cross-validation (GCV):

$$\mathrm{GCV} = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{y_i - \hat{y}_i}{1 - \mathrm{trace}(S)/N} \right]^2 . \tag{2.14}$$

[5] Although typically no weights will be exactly zero. This behavior can be enforced by imposing a L_1 prior on w (LASSO), instead of the L_2 prior used in ridge regression. However, in contrast to ridge regression, a closed-form of the LASSO problem is no longer possible.

Margin-based methods: Support vector machines The support vector machine (Burges, 1998; Schölkopf & Smola, 2002, SVM) is a binary classification technique that aims to maximize the margin between the two classes (which are commonly denoted by -1 and 1 rather than 0 and 1). For the simplest case, assume that the training examples are linearly separable, i.e. there exists a vector w and a scalar b such that

$$y_i \left(w^\top x_i + b \right) > 0 \quad \text{for all } i. \tag{2.15}$$

Qualitatively, that means that the training examples with labels $+1$ and -1 lie on opposite sides of the separating hyperplane $\{x | w^\top x + b = 0\}$, which then acts as the decision boundary. In this case, w and b are not unique; and the support vector machine is defined as the separating hyperplane with the maximum margin, i.e. the separating hyperplane for which the distance to the closest training point is maximized:

$$(w^*, b^*) = \arg\min_{w,b} \frac{1}{2} w^2 \text{ s.t. } y_i \left(w^\top x_i + b \right) \geq 1 \text{ for all } i. \tag{2.16}$$

In practice, training data are rarely exactly linearly separable. If a linear classifier is appropriate, but there is always some overlap between the two classes due to noise, the separability constraints can be relaxed by the introduction of slack variables:

$$(w^*, b^*, \xi^*) = \frac{1}{2} w^2 + C \sum_{i=1}^{n} \xi_i \text{ s.t. } y_i \left(w^\top x_i + b \right) \geq 1 - \xi_i \text{ for all } i. \tag{2.17}$$

Note that all training examples with $\xi_i > 1$ will be misclassified by the trained SVM. Large values of C penalize such misclassifications severely, while for small values of C the criterion that the margin should be large becomes more important. If a nonlinear classifier is more appropriate, the features can be transformed into a higher-dimensional space via a transformation $x \to \phi(x)$: a linear classifier in this higher-dimensional space then becomes a nonlinear classifier in the original space. For example, a quadratic decision boundary can be achieved via the mapping $\phi(x) = (x, x^2)^\top$. It turns out that for solving the optimization problem in Eq. (2.17) only the scalar products $x_i^\top x_j$ are required: the solution in the higher-dimensional space follows directly by replacing these by $\phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$. This allows the use of infinite-dimensional mappings ϕ; an important example is the radial basis function (RBF) kernel

$$K(x_i, x_j) = \exp \left(-\frac{\|x_i - x_j\|^2}{2\gamma^2} \right). \tag{2.18}$$

Other methods For space reasons, the previous enumeration of supervised classification methods is incomplete: Important techniques that have not been covered are e.g. artificial neural networks in their shallow (Bishop, 1994) and deep variant (Bengio, 2009), boosting (Freund & Schapire, 1999) or Gaussian processes (Rasmussen & Williams, 2006). In general, it depends on the particular data which classifier has the best accuracy, and there are little theoretical results which could predict a superiority of a certain classifier under realistic conditions (limited amount of training data, unknown true distribution on $\mathcal{X} \times \mathcal{Y}$). However, comparative empirical evaluations have shown that randomized tree classifiers such as random forests or boosted decision trees typically have the highest overall accuracy over a range of real-world datasets of moderate (Caruana & Niculescu-Mizil, 2006) and high dimension (Caruana et al., 2008). Besides the classical supervised learning setting that has been discussed in this section, there has been recent research on how classifier accuracy may be improved by replacing some of the inherent assumptions in the supervised learning settings by more realistic alternatives. Three important examples for such assumptions are:

- That the training examples are sampled independently and identically distributed (i.i.d.) from $p(x, y)$. Accounting for statistical dependencies between different training examples leads to structured output learning (Bakir et al., 2007).
- That every training feature vector x_i comes with a label y_i. In practice, labeling is often costly, so that there may also be a huge pool of feature vectors x_i for which no label is available. Semi-supervised learning explores how to make use of the information contained in the unlabeled x_i (Chapelle et al., 2006).
- That the training procedure has no control over the selection of training data. In the active learning setting, a training procedure tries to identify feature vector candidates x_i whose labels would be particularly informative for the classification, and actively requests labels only for these examples (Settles, 2010).

2.3. Related work

There are two other alternative software products which employ pattern recognition methods for the analysis of MRSI spectra: HealthAgents by González-Vélez et al. (2009) and SpectraClassifier by Ortega-Martorell et al. (2010). What sets this software apart from these two systems, is the capability to statistically compare various different classifiers and to select the best one. SpectraClassifier provides statistical analysis functionalities for the trained classifiers, but linear discriminant analysis is the only available classification method. On the other hand, HealthAgent supports different classification algorithms but does not provide statistical evaluation functionality.

Extensibility was an important design criterion for the library: by providing abstract interfaces for classifiers, data preprocessing procedures and evaluation statistics, users may plug in their own classes with moderate effort. Hereby it follows similar ideas as general purpose classification frameworks such as Weka,[6] TunedIT[7] or RapidMiner[8]. However, it is much more focused in scope and tailored towards medical diagnostic applications. Furthermore, a similar plug-in concept for the analysis of MRSI data was used by Neuter et al. (2007), but with a focus on quantification techniques as opposed to pattern recognition techniques, and also lacking statistical evaluation functionalities.

2.4. Software architecture

2.4.1. Overview and design principles

The software is designed for the following use case: the users label several data volumes with respect to voxel class (tumor vs. healthy) and signal quality and save the results (Fig. 2.2). They specify several classifiers to be compared, the free classifier-specific parameters to be adjusted in parameter optimization (see Fig. 2.3) and preprocessing steps for the data. A training and test suite is then defined, which may contain the voxel class classification task, the signal quality classification task, or both. The users may partition all data volumes explicitly into a separate training and testing set, otherwise a cross-validation scheme is employed: the data is partitioned into several folds, and the classifiers are iteratively trained on all but one folds, and tested on the remaining fold. The latter option is advisable if only few data are available; it has the additional advantage that means and variances for the classifier results may be estimated.

Every classifier is assigned to a preprocessing pipeline, which transforms the observed spectra into training and test features. Some elements of this pipeline may be shared across several classifiers, while others are specific for one classifier. Input data (spectra and labels) are passed, preprocessed and partitioned into cross validation folds if no explicit test data are provided. The parameters of every classifier are optimized either on the designated training data or on the first fold by maximizing an estimate for the generalization error. The classifiers are then trained with the final parameter

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] http://tunedit.org/

[8] http://www.rapid-i.com

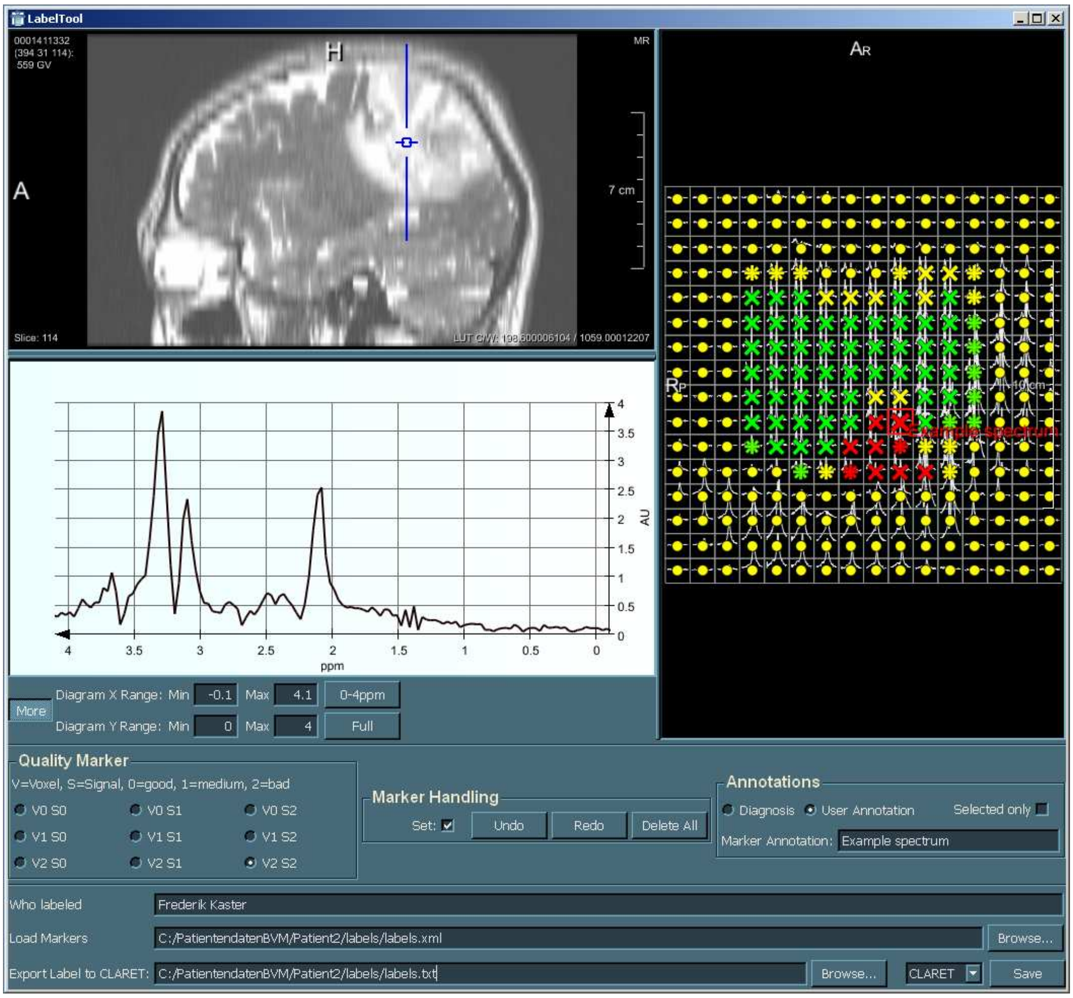


Diagram X Range: Min -0.1 Max 4.1

Diagram Y Range: Min 0 Max 4

○ V0 S0 ○ V0 S1 ○ V0 S2
○ V1 S0 ○ V1 S1 ○ V1 S2
○ V2 S0 ○ V2 S1 ● V2 S2

Set: ☑

○ Diagnosis ● User Annotation Selected only □

Marker Annotation: Example spectrum

Who labeled: Frederik Kaster

Load Markers: C:/PatientendatenBVM/Patient2/labels/labels.xml

Export Label to CLARET: C:/PatientendatenBVM/Patient2/labels/labels.txt CLARET

Figure 2.2. – User interface for the labeling functionality of the MRSI data, showing an exemplary dataset acquired at a 3 Tesla Siemens Trio scanner. This graphical interface was implemented by Bernd Merkel and Markus Harz, Fraunhofer MeVis Institute for Medical Image Computing. Top left: Corresponding morphological dataset in sagittal view (T_2-weighted turbo spin-echo sequence in this case). Users can place a marker (blue) to select a voxel of interest. Middle left: Magnitude spectrum of the selected voxel, which is typical for a cerebral tumor. Top right: Selected voxel (framed in red) together with the axial slice in which it is contained. The user-defined labels are overlayed over a synopsis of all spectra in the slice. The label shape encodes the signal quality (dot / asterisc / cross for “not evaluable” / “poor” / “good”), while the label color encodes the voxel class (green / yellow / red for “healthy” / “undecided” / “tumor”). The labels may also be annotated by free-text strings. Bottom panel: User interface with controls for label definition, text annotation and data import / export.

values, and performance statistics are computed by comparing the prediction results on the current test data with the actual test labels. Statistical tests are conducted to decide whether the classifiers differ significantly in performance. Typically not only two, but multiple classifiers are compared against each other, which must be considered when judging significance. Finally the classifiers are retrained on the total data for predicting the class of unlabeled examples. The user may perform quality control in order to assess if the performance statistics are sufficient for employment in the clinic (Fig. 2.4). The trained classifiers may then be loaded and applied to new datasets, for which no manual labels are available (Fig. 2.5).

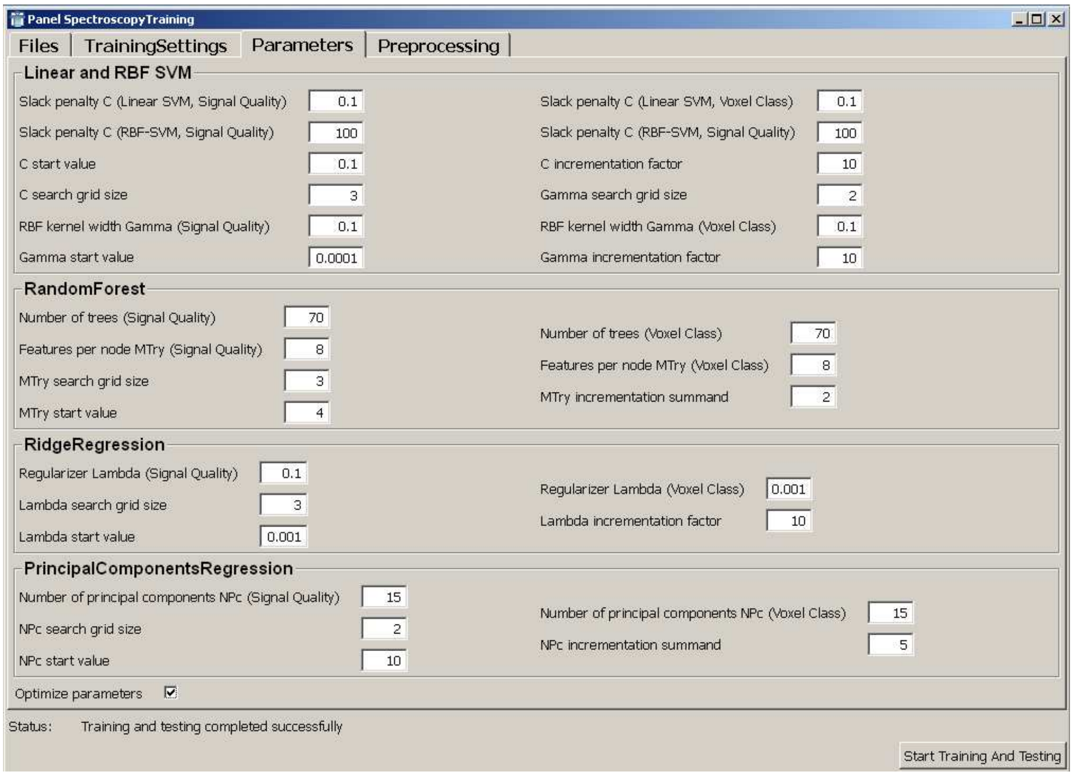
The main design criteria were extensibility, maintainability and exception safety. Extensibility was achieved by providing abstract base classes for classifiers, preprocessing procedures and evaluation statistics, so that it is easily possible to add e.g. new classification methods by deriving from the appropriate class. For maintainability, dedicated manager objects handle the data flow between the different modules of the software and maintain the mutual consistency of their internal states upon changes made by the user. Strong exception safety guarantees are necessitated by the quality requirements for medical software; it was achieved by creating additional resource management classes following the Resource Acquisition Is Initialization (RAII) idiom (Stroustrup, 2001).

2.4.2. The classification functionality

The design of the classification functionality of this library follows the main aim of separating between classifier-specific functionality (which must be provided by the user when introducing a new classifier) and common functionality that is used by all classifiers and does not need to be changed: the `ClassifierManager` class is responsible for the former, while the classes derived from the abstract `Classifier` basis class are responsible for the latter. Simple extensibility and avoiding code repetition were therefore the two main design principles.

A `ClassifierManager` object corresponds to each classification task, e.g. classification with respect to signal quality and with respect to voxel class (see Fig. 2.6). It controls all classifiers which are trained and benchmarked for this task, and ensures that operations such as training, testing, and the averaging of performance statistics over cross-validation folds as well as saving and loading are performed for each classifier. It also partitions the training features and labels into several cross-validation folds, if the users do not define a designated test dataset.

A `Classifier` object encapsulates an algorithm for mapping feature vectors to discrete labels after training. Alternatively, the output can also be a continuous score that gives information about the confidence that a spectrum corresponds to a tu-



Linear and RBF SVM

Slack penalty C (Linear SVM, Signal Quality) 0.1
Slack penalty C (RBF-SVM, Signal Quality) 100
C start value 0.1
C search grid size 3
RBF kernel width Gamma (Signal Quality) 0.1
Gamma start value 0.0001

Slack penalty C (Linear SVM, Voxel Class) 0.1
Slack penalty C (RBF-SVM, Signal Quality) 100
C incrementation factor 10
Gamma search grid size 2
RBF kernel width Gamma (Voxel Class) 0.1
Gamma incrementation factor 10

RandomForest

Number of trees (Signal Quality) 70
Features per node MTry (Signal Quality) 8
MTry search grid size 3
MTry start value 4

Number of trees (Voxel Class) 70
Features per node MTry (Voxel Class) 8
MTry incrementation summand 2

RidgeRegression

Regularizer Lambda (Signal Quality) 0.1
Lambda search grid size 3
Lambda start value 0.001

Regularizer Lambda (Voxel Class) 0.001
Lambda incrementation factor 10

PrincipalComponentsRegression

Number of principal components NPc (Signal Quality) 15
NPc search grid size 2
NPc start value 10

Number of principal components NPc (Voxel Class) 15
NPc incrementation summand 5

Optimize parameters ☑

Status: Training and testing completed successfully

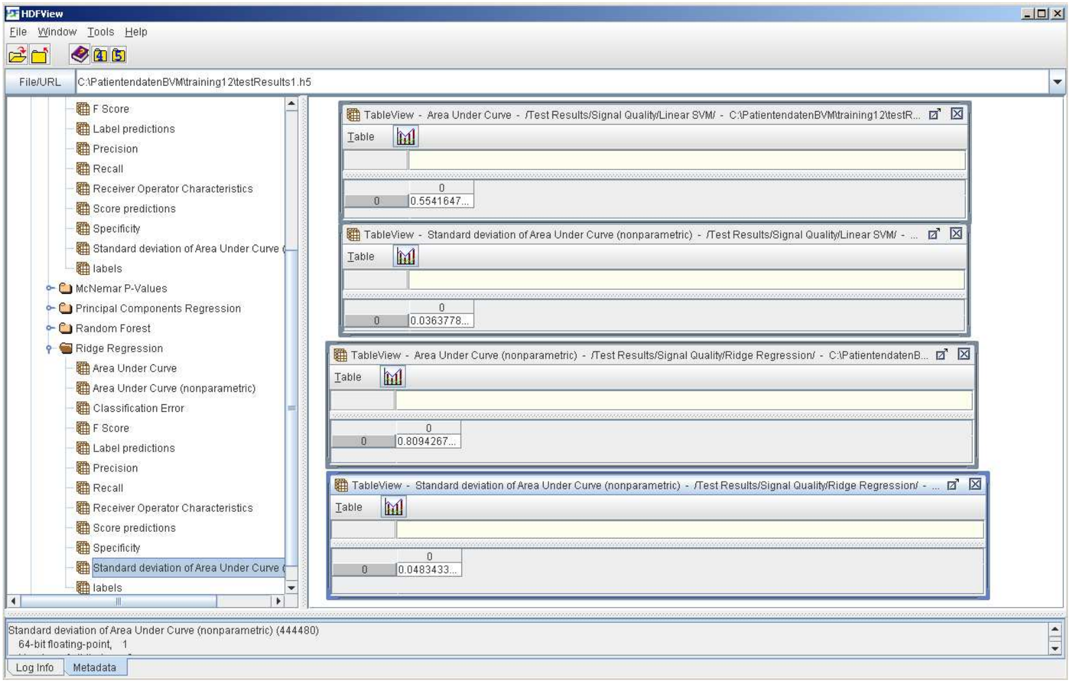
Figure 2.3. – Part of the user interface for classifier training and testing. In this panel, the search grids for automated parameter tuning of the different classifiers may be defined (default values, starting values, incrementation step sizes and numbers of steps).

---

mor. Bindings were implemented for several linear and nonlinear classifiers, which previously had been found to be well-suited for the classification of MRSI spectra (Menze et al., 2006): support vector machines (SVMs) with a linear and a radial basis function (RBF) kernel, random forests (RF), ridge regression (RR) and principal components regression (PCR); see (Hastie et al., 2009) for a description of these methods. The actual classification algorithms are provided by external libraries such as LIBSVM (Chang & Lin, 2001) and VIGRA (Köthe, 2000).

Both binary classification (with two labels) as well as multi-class classification (with more than two labels) are supported. Some classifiers (e.g. random forests) natively support multi-class classification, while for other classifiers (e.g. ridge regression and principal components regression),[9] it can be achieved via a *one-vs.-all* encoding

[9] To be precise, these two classifiers are actually regression methods and can be used for binary classification by assigning the label +1 and -1 to all positive and negative class examples and



C:\PatientendatenBVM\training12\testResults1.h5

- F Score
- Label predictions
- Precision
- Recall
- Receiver Operator Characteristics
- Score predictions
- Specificity
- Standard deviation of Area Under Curve (
- labels
- McNemar P-Values
- Principal Components Regression
- Random Forest
- Ridge Regression
  - Area Under Curve
  - Area Under Curve (nonparametric)
  - Classification Error
  - F Score
  - Label predictions
  - Precision
  - Recall
  - Receiver Operator Characteristics
  - Score predictions
  - Specificity
  - Standard deviation of Area Under Curve (
  - labels

TableView - Area Under Curve - /Test Results/Signal Quality/Linear SVM/ - C:\PatientendatenBVM\training12\testR...

	0
0	0.5541647...

TableView - Standard deviation of Area Under Curve (nonparametric) - /Test Results/Signal Quality/Linear SVM/ - ...

	0
0	0.0363778...

TableView - Area Under Curve (nonparametric) - /Test Results/Signal Quality/Ridge Regression/ - C:\PatientendatenB...

	0
0	0.8094267...

TableView - Standard deviation of Area Under Curve (nonparametric) - /Test Results/Signal Quality/Ridge Regression/ - ...

	0
0	0.0483433...

Standard deviation of Area Under Curve (nonparametric) (444480)
64-bit floating-point,   1

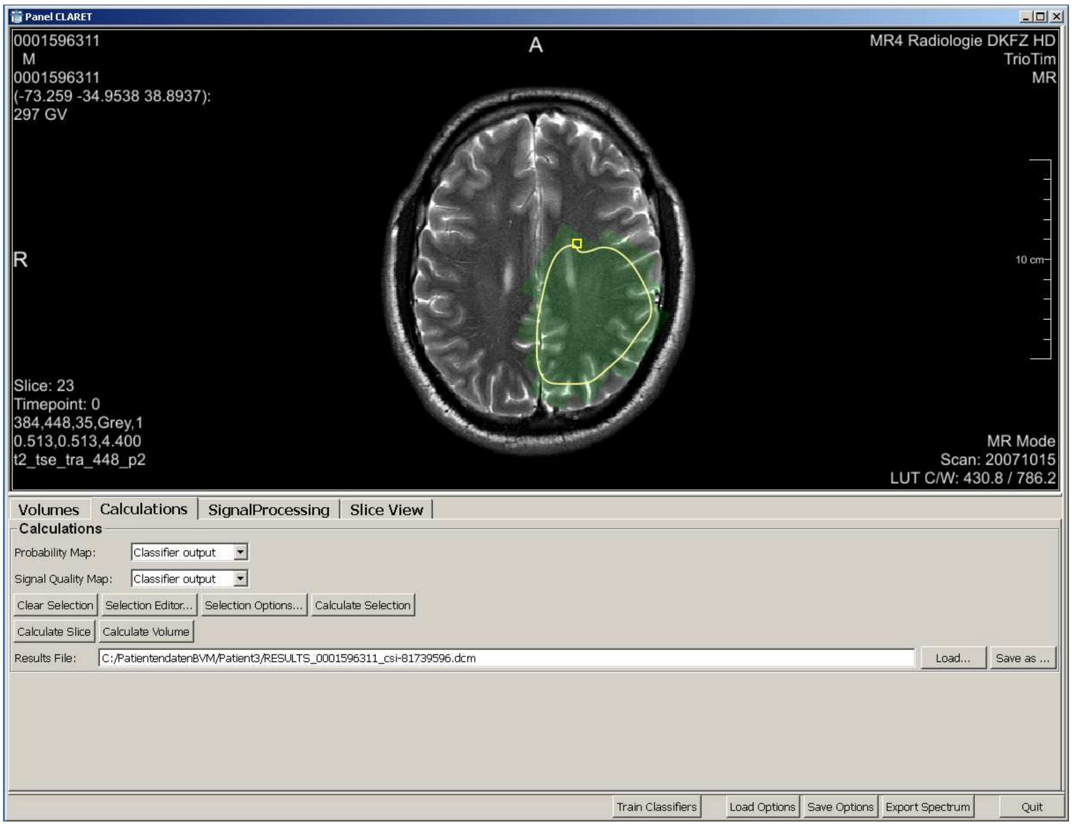
Figure 2.4. – Evaluation results for an exemplary training and testing suite. The upper two windows on the right-hand side show the estimated area under curve value for a linear support vector machine classifier and its estimated standard deviation (0.554±0.036), while the lower two windows show the same values for a ridge-regression classifier (0.809±0.048). This would allow a clinical user to draw the conclusion that only the latter one of these classifiers differs significantly from random guessing, and may sensibly be used for diagnostics. The poor quality of these classifiers is due to the fact that only a very small training set was used for the purpose of illustrating the user interface design (2 patients).

scheme,[10] in which each class is classified against all other classes in turn, and the class with the largest score is selected for the prediction (Rifkin & Klautau, 2004). This multi-class functionality allows the future extension of the library to the task of discriminating different tumor types against each other.

Furthermore, every classifier encapsulates an instance of the `ClassifierParameterManager` class controlling the parameter combinations that are tested during parameter optimization. Most classifiers have one or more internal parameters that ought to be optimized for each dataset in order to achieve optimal predictive performance

training a regressor. The `transformLabelsToBinary()` function maps the original labels to these two numbers.

[10] The virtual `isOnlyBinary()` function allows one to specify the affiliation of a classifier to these two categories.



0001596311
M
0001596311
(-73.259 -34.9538 38.8937):
297 GV

MR4 Radiologie DKFZ HD
TrioTim
MR

Slice: 23
Timepoint: 0
384,448,35,Grey,1
0.513,0.513,4.400
t2_tse_tra_448_p2

MR Mode
Scan: 20071015
LUT C/W: 430.8 / 786.2

Probability Map: Classifier output

Signal Quality Map: Classifier output

Results File: C:/PatientendatenBVM/Patient3/RESULTS_0001596311_csi-81739596.dcm

Figure 2.5. – Exemplary application of a trained classifier for the computer-assisted diagnosis of a new dataset. The classifier predictions for both voxel class and signal quality are depicted for a user-defined region of interest: the voxel class is encoded by the color (green for “healthy”, yellow for “undecided”, red for “tumor”), while the signal quality is encoded by the transparency (opaque for a good signal, invisible for a spectrum which is not evaluable). As an alternative to the classifier predictions, it is possible to display precomputed color maps as well as color maps based on the parametric quantification of relevant metabolites.

(see sec. 2.4.4). This is done by maximizing an estimate of the *generalization error* (i.e. the performance of the classifier on new test data that were not encountered during the training process) over a prescribed search grid, using the data from one of the cross-validation folds (or the whole training data, if no cross-validation is used). This generalization error could be estimated by dividing the training data into another training and test fold, training the classifier on the training part of the training data and testing it on the testing part of the training data.[11] However, this would be time-consuming. However, there exists considerable theoretical as well as empirical evidence (Golub et al., 1979; Breiman, 1996) that efficiently computable approximations for the generalization error may be sufficient for parameter adjustment: these are provided by the function `estimatePerformanceCvFold()`. For SVMs, this is an internal cross-validation estimate as described in (Lin et al., 2007), for random forests, the *out-of-bag error* and for regression-based classifiers the *generalized cross-validation* (Hastie et al., 2009). The optimal parameters are selected by the function `optimizeParametersCvFold()` based on the data from one specific cross-validation fold.

This part of the library may be easily extended by adding new classifiers, as long as they fit into the supervised classification settings (i.e. based on labeled training vectors, a function for mapping these vectors to the discrete labels is learnt). Artificial neural networks, boosted ensemble classifiers or Gaussian process classification are examples for alternative classification algorithms that could be added in this way. For this, one only needs to derive from the `Classifier` abstract base class and to provide implementations for its abstract methods (including the definition of the `Preprocessor` subclass with which this classifier type is associated). For parameter tuning, one also has to supply an estimate of the classifier accuracy: This may always be computed via cross-validation, but preferably this estimate should arise as a by-product of the training or be fast to compute (same as e.g. the out-of-bag error for the random forest or the generalized cross-validation). Furthermore one has to assume the existence of a continuous classification score, which ideally can be interpreted as a tumor probability. However, for classifiers without such a probabilistic interpretation it is sufficient to reuse the 0/1 label values as scores: as long as higher scores correspond to a higher likelihood for the positive (tumor) class, they can take any values. Only the single-voxel spectra are used for classification, hence the architecture does not allow classifiers that make explicit use of spatial context information (so-called *probabilistic graphical models*).

[11] Note that the actual test data must not be used during parameter tuning.

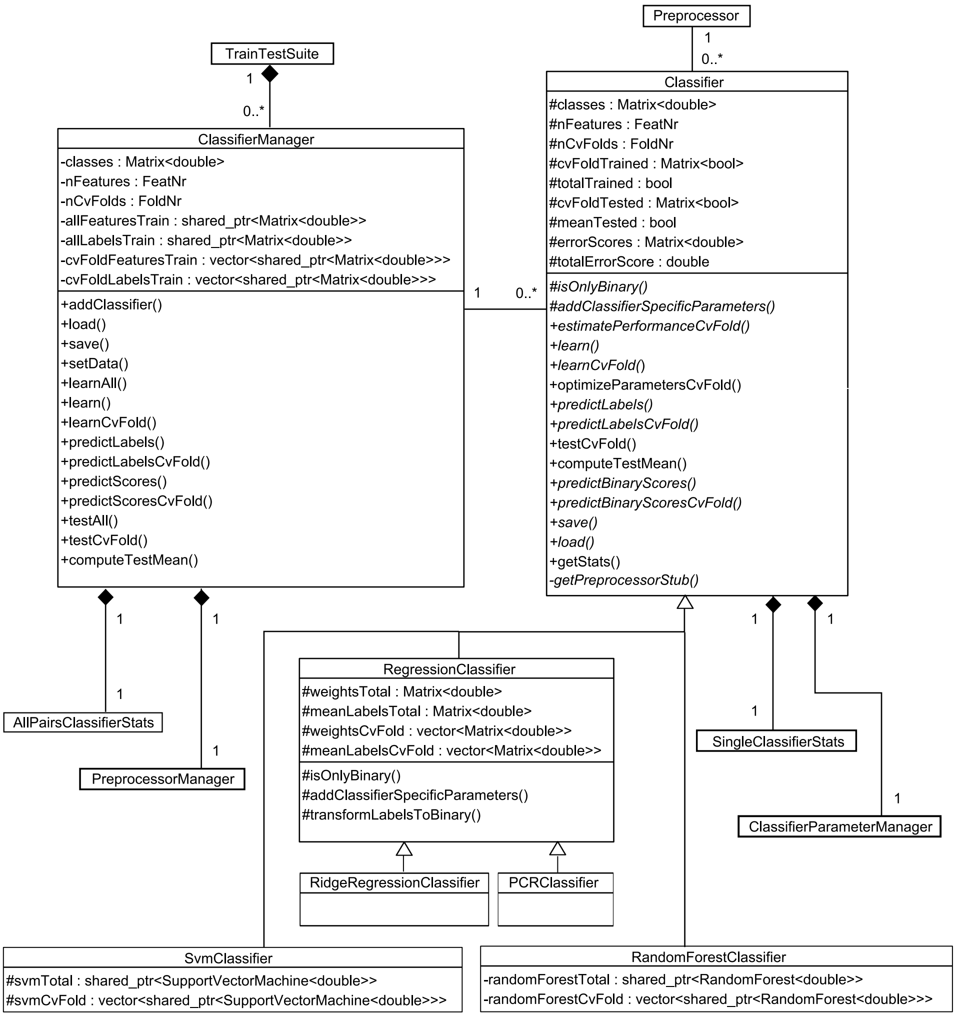

Preprocessor
1
0..*
TrainTestSuite
1
0..*
Classifier
#classes : Matrix<double>
#nFeatures : FeatNr
#nCvFolds : FoldNr
#cvFoldTrained : Matrix<bool>
#totalTrained : bool
#cvFoldTested : Matrix<bool>
#meanTested : bool
#errorScores : Matrix<double>
#totalErrorScore : double
#isOnlyBinary()
#addClassifierSpecificParameters()
+estimatePerformanceCvFold()
+learn()
+learnCvFold()
+optimizeParametersCvFold()
+predictLabels()
+predictLabelsCvFold()
+testCvFold()
+computeTestMean()
+predictBinaryScores()
+predictBinaryScoresCvFold()
+save()
+load()
+getStats()
-getPreprocessorStub()
ClassifierManager
-classes : Matrix<double>
-nFeatures : FeatNr
-nCvFolds : FoldNr
-allFeaturesTrain : shared_ptr<Matrix<double>>
-allLabelsTrain : shared_ptr<Matrix<double>>
-cvFoldFeaturesTrain : vector<shared_ptr<Matrix<double>>>
-cvFoldLabelsTrain : vector<shared_ptr<Matrix<double>>>
+addClassifier()
+load()
+save()
+setData()
+learnAll()
+learn()
+learnCvFold()
+predictLabels()
+predictLabelsCvFold()
+predictScores()
+predictScoresCvFold()
+testAll()
+testCvFold()
+computeTestMean()
1
0..*
1
1
1
1
AllPairsClassifierStats
1
PreprocessorManager
RegressionClassifier
#weightsTotal : Matrix<double>
#meanLabelsTotal : Matrix<double>
#weightsCvFold : vector<Matrix<double>>
#meanLabelsCvFold : vector<Matrix<double>>
#isOnlyBinary()
#addClassifierSpecificParameters()
#transformLabelsToBinary()
RidgeRegressionClassifier
PCRClassifier
1
1
1
SingleClassifierStats
1
ClassifierParameterManager
SvmClassifier
#svmTotal : shared_ptr<SupportVectorMachine<double>>
#svmCvFold : vector<shared_ptr<SupportVectorMachine<double>>>
RandomForestClassifier
-randomForestTotal : shared_ptr<RandomForest<double>>
-randomForestCvFold : vector<shared_ptr<RandomForest<double>>>

Figure 2.6. – Simplified UML diagram of the classification functionality of the software library: detailed explanations can be found in section 2.4.2. The connections to the classes `TrainTestSuite` (see Fig. 2.10), `Preprocessor` / `PreprocessorManager` (Fig. 2.7), `ClassifierParameterManager` (Fig. 2.8) and `SingleClassifierStats` / `AllPairClassifierStats` (Fig. 2.9) are shown. In this diagram, as in the following ones, abstract methods are printed in italics: to save space, the implementations of these abstract methods are not shown if they are provided in the leaves of the inheritance tree. The depiction here is simplified: actually the non-virtual interface principle is followed, so that protected visibility is given to all abstract methods, which are then encapsulated by non-virtual public methods.

2.4.3. The preprocessing functionality

Preprocessing (Fig. 2.7) is the extraction of a feature vector from the raw MRSI spectra with the aim of improved classification performance. While classification makes use of both the label and the feature information (supervised process), preprocessing only uses the feature information (unsupervised process). `Preprocessor` objects may act both on the total data (`transformTotal()`) and of the data of a single cross-validation fold (`transformCvFold()`): the distinction may be relevant since some preprocessing steps (e.g. singular value decomposition) depend on the actual training data used.

The main goal governing the design of the preprocessing functionality was training speed: data preprocessing steps which are common to multiple classifiers should only be performed once. Hence the different preprocessing steps are packaged into modules (deriving from the `Preprocessor` abstract base class) and arranged into cascades. A common `PreprocessorManager` ensures that every preprocessing step is only performed once. Hiding the preprocessing functionality from the library users was an additional criterion: Every subclass of `Classifier` is statically associated with a specific `Preprocessor` subclass and is responsible for registering this subclass with the `PreprocessorManager` and passing the data to be preprocessed.

First, since only the metabolite signals carry diagnostically relevant information, the nuisance signal caused by water molecules has to be suppressed, using e.g. a Hankel singular value decomposition filter (Pijnappel et al., 1992). Then the spectra are transformed from the time domain into the Fourier domain by means of the FFTW library (Frigo & Johnson, 2005), and the magnitude spectrum is computed. The subsequent steps may be adjusted by the user, and typically depend on the classifier:

Common MRSI preprocessing steps used by all classifiers are the rebinning of spectral vectors via a B-spline interpolation scheme, the extraction of diagnostically relevant parts of the spectrum and L_1 normalization (i.e. the spectral vector is normalized such that the sum of all component magnitudes in a prescribed interval equals one): these are performed by the class `MrsiPreprocessor`.[12] Other preprocessing steps are only relevant for some of the classifiers, e.g. the `RegressionPreprocessor` performs a singular value decomposition of the data which speeds up subsequent ridge regression or PCR. SVMs perform better when the features have zero mean and unit variance: this can be achieved by the `WhiteningPreprocessor`.

Two features of the software implementation support this modular structure: The `PreprocessorManager` incorporates a class factory, which ensures that only one instance of each preprocessor class is created. This allows to share preprocessors across

[12] More sophisticated steps such as the extraction of wavelet features might be added as well.

various classifiers and prevents duplicate preprocessing steps (such as e.g. performing the singular value decomposition twice on the same data). Furthermore, preprocessors are typically arranged in a tree structure (via the `predecessor` and `successors` references) and every classifier is assigned to one vertex of this tree, which ensures that all preprocessing steps on the path from the root to this vertex are applied in order (creating a pipeline of preprocessing steps). Once the data encapsulated inside one module changes, all successors are invalidated.

When new classifiers are added to the library, the preprocessing part may easily extended with new preprocessor modules as long as they fit into the unsupervised setting (i.e. they only make use of the features, but not of the labels). Besides implementing the abstract methods of the `Preprocessor` base class, the association between the classifier and the preprocessor must be included in the classifier definition by implementing its `getPreprocessorStub()` method: then the classifier object ensures that the new preprocessor is correctly registered with the preprocessor manager object. As a limitation, the new preprocessor has to be appended as a new leaf (or a new root node) to the preprocessor tree: the intermediate results from other preprocessing steps can only be reused if the order of these steps is not changed.

2.4.4. The parameter tuning functionality

All classifiers have adjustable parameters, which are encapsulated in the `ClassifierParameter` class (Fig. 2.8). The design of the parameter handling functionality was guided by the main rationale of handling parameters of different datatypes in a uniform way. Furthermore automated parameter adjustment over a search grid was enabled (which may have linear or logarithmic spacing depending on the range of reasonable parameter values), by hiding the details of the search mechanism from the class users.

Some parameters should be optimized for the specific classification task, as described in section 2.4.2: for the classifiers supplied by us, these are the slack penalty C for SVMs, the kernel width γ for SVMs with an RBF kernel, the random subspace dimension m_{try} for random forests, the number of principal components n_{PC} for PCR and the regularization parameter λ for ridge regression. They are represented as a `TypedOptimizableClassifierParameter`: besides the actual value, these objects also contain the search grid of the parameters, namely the starting and end value, the incrementation step and whether the value should be incremented additively or multiplicatively (encoded in the field `incrInLogSpace`). Multiplicative updates are appropriate for parameters that can span a large range of reasonable values.

There are also parameters which may not be optimized: these are encapsulated as a `TypedClassifierParameter`, which only contains the actual value. A good example

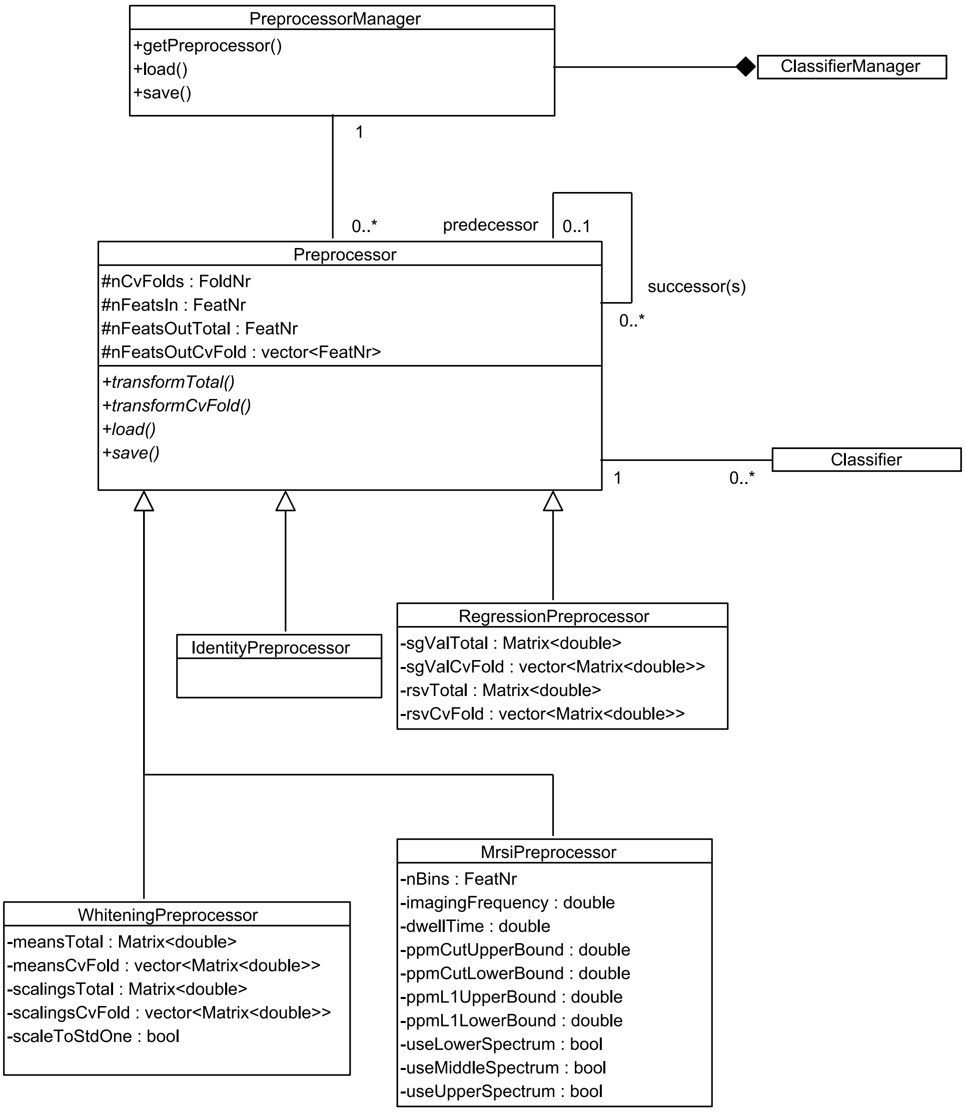

PreprocessorManager
+getPreprocessor()
+load()
+save()
ClassifierManager
1
0..*
predecessor
0..1
successor(s)
0..*
Preprocessor
#nCvFolds : FoldNr
#nFeatsIn : FeatNr
#nFeatsOutTotal : FeatNr
#nFeatsOutCvFold : vector<FeatNr>
+transformTotal()
+transformCvFold()
+load()
+save()
Classifier
1
0..*
IdentityPreprocessor
RegressionPreprocessor
-sgValTotal : Matrix<double>
-sgValCvFold : vector<Matrix<double>>
-rsvTotal : Matrix<double>
-rsvCvFold : vector<Matrix<double>>
MrsiPreprocessor
-nBins : FeatNr
-imagingFrequency : double
-dwellTime : double
-ppmCutUpperBound : double
-ppmCutLowerBound : double
-ppmL1UpperBound : double
-ppmL1LowerBound : double
-useLowerSpectrum : bool
-useMiddleSpectrum : bool
-useUpperSpectrum : bool
WhiteningPreprocessor
-meansTotal : Matrix<double>
-meansCvFold : vector<Matrix<double>>
-scalingsTotal : Matrix<double>
-scalingsCvFold : vector<Matrix<double>>
-scaleToStdOne : bool

Figure 2.7. – Simplified UML diagram of the preprocessing functionality; see section 2.4.3 for details. The connections to the classes `Classifier` and `ClassifierManager` (Fig. 2.6) are shown.

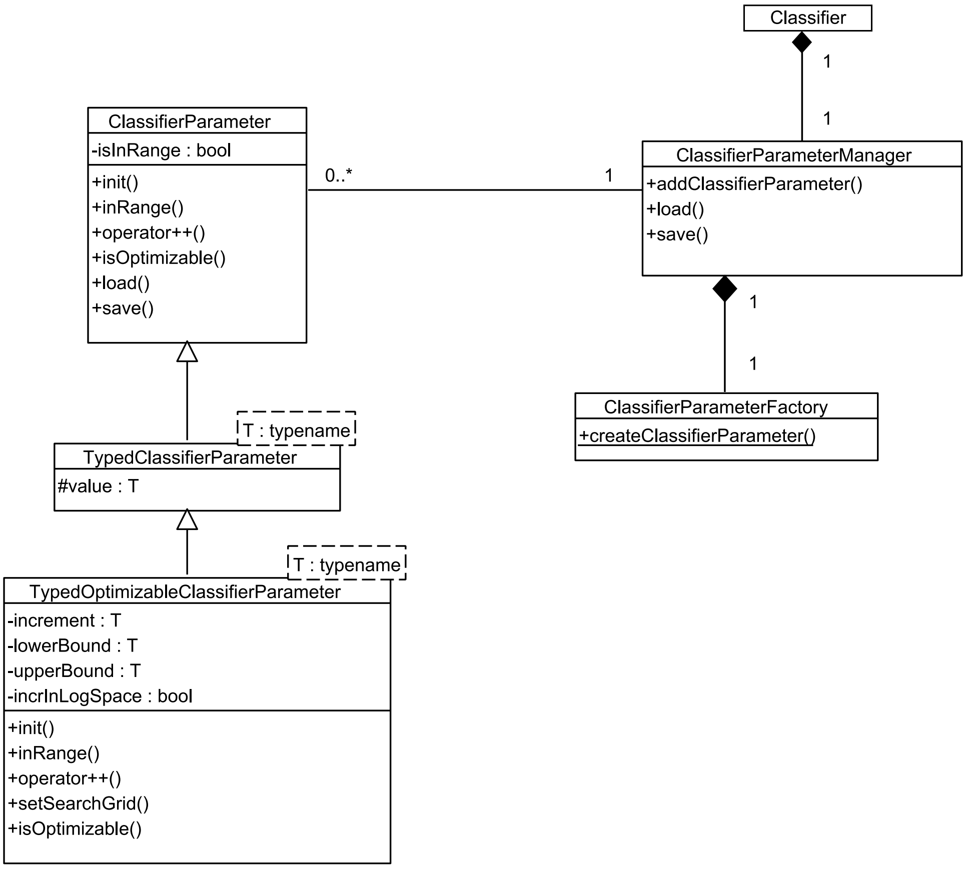


Figure 2.8. – Simplified UML diagram of the parameter tuning functionality; see section 2.4.4 for details. The connection to the class `Classifier` (Fig. 2.6) is shown.

would be the number of trees of a random forest classifier, since the generalization error typically saturates as more trees are added.

While all currently used parameters are either integers or floating-point numbers, one can define parameters of arbitrary type: however, one has to define how this data type can be written to or retrieved from a file or another I/O medium by implementing the corresponding I/O callbacks (see section 2.4.6 for detailed explanation). For optimizable parameters, it must also be defined what it means to increase the parameter by a fixed value (by overloading the `operator++()` member function). As a limitation, all parameters are assumed to vary completely independently and cannot encode constraints coupling the values of multiple parameters.

One should note that the parameter optimization process followed by this library is exactly the way a human expert would do it: in the absence of universal theoretical criteria about the choice of good parameters, they have to be tuned empirically so that a low generalization error is achieved.[13] However, this is the most time-consuming part of adapting a classifier to a new experiment, which is now completely automated by the software.

2.4.5. The statistics functionality

The computation of evaluation statistics is crucial for the automated quality control of trained classifiers (Fig. 2.9). This part of the library was designed with the following aims in mind: Needless recomputation of intermediate values should be avoided; thus the binary confusion matrix is computed only once and then cached within a `StatsDataManager` object, which can be queried for computing the different statistics derived from it (e.g. `Precision` and `Recall`). The library can be simply extended by new statistics characterizing a single classifier. Dedicated manager classes (such as `SingleFoldStats`, `SingleClassifierStats` as well as `PairClassifierStats` and `AllPairsClassifierStats`) are each responsible for a well-defined statistical evaluation task: namely, characterizing a classifier for a single cross-validation fold, characterizing a classifier over all folds, characterizing a single pair of classifiers and characterizing all existing pairs of classifiers. They ensure that this computation is performed in a consistent way for all classifiers, so that code redundancy is avoided.

The class `SingleClassifierStats` manages all statistics pertaining to one single classifier: it is composed of objects of type `SingleFoldStats`, which in turn manage all statistics either of a single cross-validation fold (`cvData`), or the mean and standard deviation values computed over all folds (`meanData`). A `StatsDataManager` is a helper class which caches several intermediate results required for the computation of the different `Statistics`.

There are different variants of how these statistics may be computed in a multi-class classification setting: some of them (e.g. the `MisclassificationRate`) can handle multiple classes natively; these statistics form the derived class `AllVsAllStat`. Other statistics (e.g. `Precision`, `Recall` or `FScore`) were originally designed for a binary classification setting. For the latter kind, one must report multiple values, namely one for each class when discriminated against all others (one-vs.-all encoding), and they inherit from the `OneVsAllStat` class. The `AreaUnderCurve` (AUC) value of the

[13] If sufficient data were available, it would be preferable to perform this parameter tuning on a separate tuning dataset that is not used in the training and testing of the classifier. Since typically clinics only have access to few validated MRSI data, this approach may not be practicable, and the cross-validation scheme used in this library is the best alternative to deal with scarce data.

receiver operating characteristic (ROC) curve (Fawcett, 2006) is a specialty: while it is also computed in a one-vs.-all fashion, the underlying ROC curves are stored as well. Standard deviation estimates are mostly available only for the `meanData` averaged over several cross-validation folds, with the exception of the AUC values for which nonparametric bootstrap estimates can be easily computed (Bandos et al., 2007).

Besides the statistical characterization of single classifiers, it is also relevant to compare pairs of classifiers in order to assess which one of them is best for the current task, and whether the differences are statistically significant. The `AllPairsClassifierStats` class manages the statistics characterizing the differences in misclassification rate between all pairs of classifiers, each of which is represented by a single `PairClassifierStats` instance. p-values are computed by statistical hypothesis tests with the null hypothesis that there is no difference between classifier performances. Implementations are provided for two tests: McNemar's test (Dietterich, 1998) is used when the data are provided as a separate training and test set, while a recently proposed conservative t-test variant (Grandvalet & Bengio, 2006) is used if the users provide only a training dataset, which is then internally partitioned into cross-validation folds. The latter test assumes that there is an upper border on the correlation of misclassification rates across different cross-validation folds, which is stored in the variable `maxCorrelationGrandvalet`.[14]

If there are more than two classifiers, the p-values must be adjusted for the effect of multiple comparisons: In the case of five classifiers with equal performance, there are ten pairwise comparisons and a significant difference ($p_{\text{raw}} < 0.001$) is expected to occur with a probability of $1 - 0.999^{10} \approx 0.01$. After computing all “raw” p-values, they are corrected using Holm's step-down or Hochberg's step-up method (Demšar, 2006), and all results are stored as `PValue` structures.

If there is need to extend the statistics functionality, it is simple to add any statistic characterizing a single classifier that can be computed from the true labels and the predicted labels and scores, as these values may be queried from the `StatsDataManager` object. This comprises all statistics which are commonly used for judging the quality of general classification algorithms. As a limitation, the evaluation statistics cannot use any information about the spatial distribution of the labels: hence it is impossible to compute e.g. the Hausdorff distance between the true and the predicted tumor segmentation. Among the statistical significance tests (like `McNemarPairClassifierStat`), one can add any technique that only requires the mean values of the statistic to be compared from each cross-validation fold. The cur-

[14] Note that a classical t-test may not be used, since the variance of misclassification rates is estimated from cross-validation and hence systematically underestimated. Bengio & Grandvalet (2004) showed that unbiased estimation of the variances is not possible; but the procedure used here provides an upper bound on the p-value if the assumptions are fulfilled.

rent design is not prepared for new methods of multi-comparison adjustment beyond Holm's or Hochberg's method: for every method acting only on p-values and computing an adjusted p-value, this would be possible, but requires moderate redesign of this part of the library. Also the assumption is hardwired that the mean and variance of these evaluation shall be estimated using a cross-validation scheme. The number of cross-validation folds can be specified at the `ClassifierManager` level: It is theoretically possible to run a leave-one-out validation scheme with this machinery, but that would lead to prohibitive computation times.

2.4.6. The input / output functionality

The input / output functionality was designed in order to keep it separated from the modules responsible for the internal computations: hence function objects are passed to the classifier, preprocessor etc. objects, which can then be invoked to serialize all types of the data that is encapsulated by these objects. Similar function objects are used for streaming relevant information outside and listening for user signals at check points.

For persistence, classifiers, preprocessors, statistics and all other classes with intrinsic state can be saved and reloaded in a hierarchical data format, and the data input/output can be customized by passing user-defined input and output function objects derived from the base classes `LoadFunctor` and `SaveFunctor` (see Fig. 2.10). For these function objects, the user must define how to enter and leave a new hierarchy level (`initGroup()` and `exitGroup()`) and how to serialize each supported data type (`save()` and `load()`): for the latter purpose, the function objects must implement all required instantiations of the `LoadFunctorInterface` or `SaveFunctorInterface` interface template. Exemplary support is provided for HDF5[15] as the main storage format (XML would be an obvious alternative). For integration into a user interface, other function objects may be passed that can either report progress information, e.g. for updating a progress bar (`StreamProgressFunctor`), or report status information (`StreamStatusFunctor`) or listen for abort requests (`AbortCheckFunctor`) at regular check points. A `ProgressStatusAbortFunctor` bundles these three different functions. The `TrainTestSuite` manages the actions of the library at the highest level: the library users mainly interact with this class by adding classifier manager objects, passing data and retrieving evaluation results.

The I/O functionality can simply be extended to other input and output streams, as long as the data can be stored in a key-value form with string keys, and as long as a hierarchical structure with group denoted by a name string can be imposed. Instead of only listening for abort signals, the `AbortCheckFunctor` could in principle

[15] http://www.hdfgroup.org/HDF5/

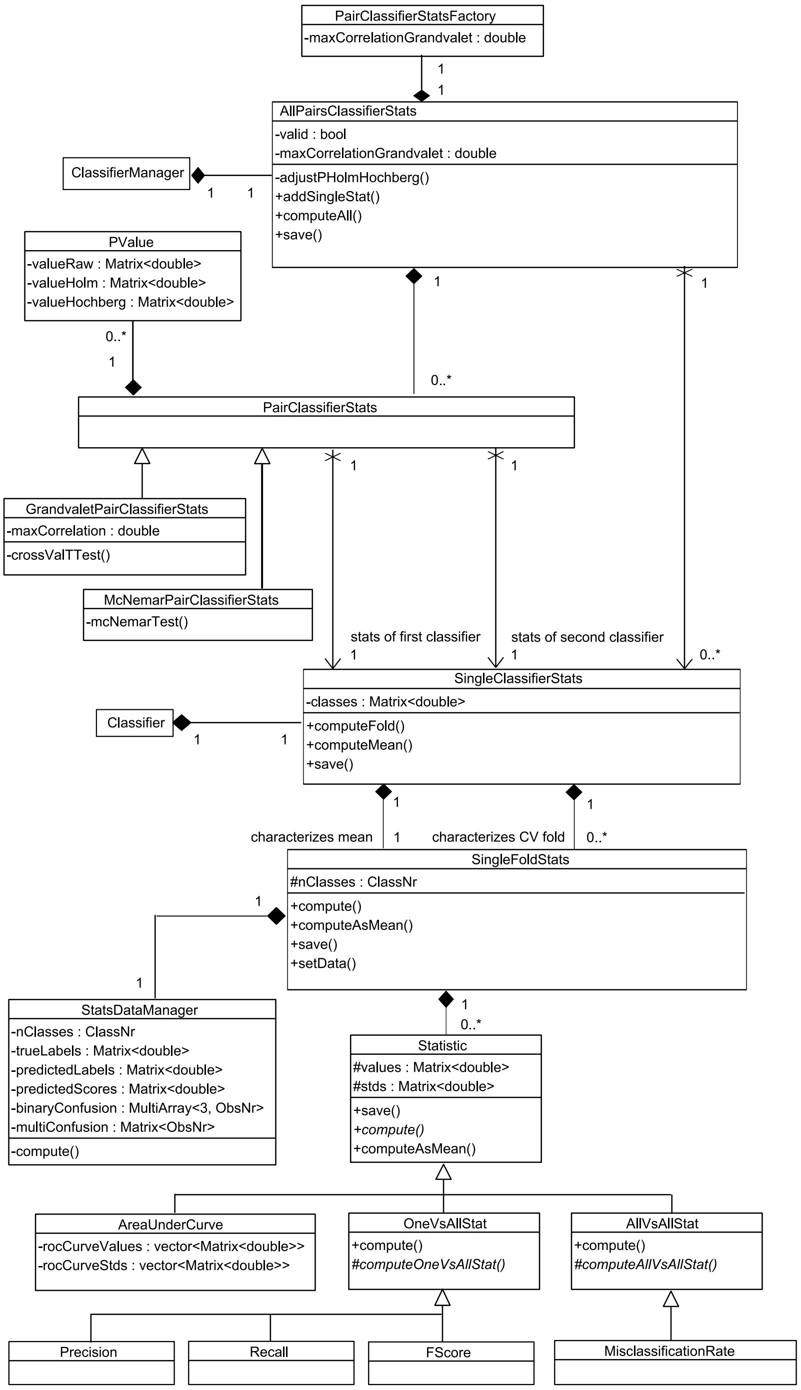


Figure 2.9. – Simplified UML diagram of the statistical evaluation functionality; see section 2.4.5 for details. The connections to the classes `Classifier` and `ClassifierManager` (Fig. 2.6) are shown.

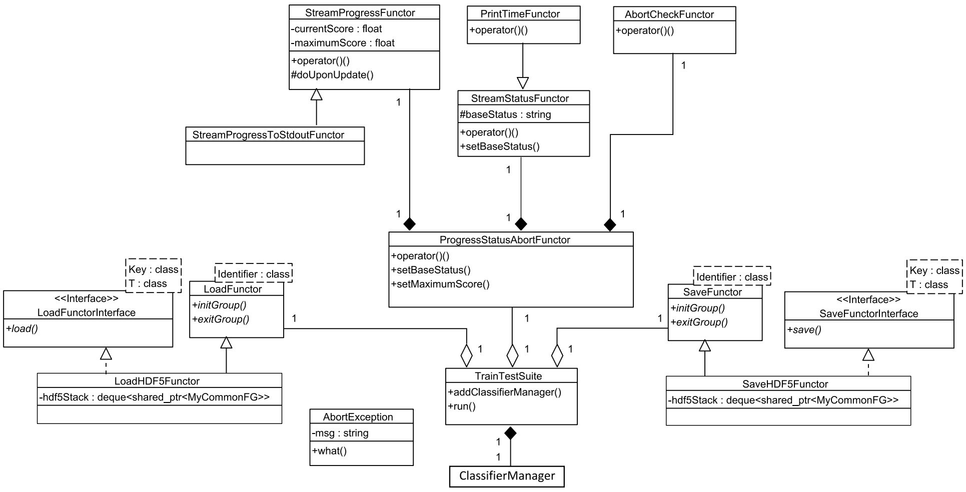


Figure 2.10. – Simplified UML diagram of the data input / output functionality; see section 2.4.6 for details. The connection to the class `ClassifierManager` (Fig. 2.6) is shown.

---

handle more general user requests: but aborting a time-consuming training process is presumably the main requirement for user interaction capabilities.

2.4.7. User interaction and graphical user interface

In order to further aid the clinical users in spectrum annotation, a graphical user interface was developed in MeVisLab that displays MRSI spectra from a selected slice in the context of its neighbor spectra, which can then be labeled on an ordinal scale by voxel class and signal quality and imported into the classification library (Fig. 2.2). Since clinical end users only interact with this user interface, they can start a training and testing experiment and evaluate the results without expert knowledge on pattern recognition techniques: they only have to provide their domain knowledge about the clinical interpretation of MRSI data. To this purpose, a graphical user interface displays the MRSI spectra of the different voxels both in their spatial context (upper right of Fig. 2.2) and as enlarged single spectra (middle left of this figure). It is known that the ability to view MRSI spectra in their surroundings and to incorporate the information from the neighboring voxels is one of the main reasons why human experts still perform better at classifying these spectra than automated methods (Zechmann et al., 2011). Simultaneously one can display a morphological MR image that is registered to the MRSI grid, which can give additional valuable information for the labeling process of the raters. Labels are provided on two axes (signal quality and voxel class / malignancy) that are encoded by marker shape and color; furthermore it is possible to add free-text annotations to interesting spectra.

After saving the label information in a human-readable text format, clinical users only have to provide the information which label files (and associated files with MRSI data) shall be used for training and testing. (As stated in section 2.4.6, it is not required to specify dedicated testing files; in this case, all data are used in turn for both training and testing via a hold-out scheme.) An expert mode provides the opportunity to select which classifiers to train and test and to set the classifier parameters manually (Fig. 2.3). Also default values are proposed for these parameters, which gave the best or close to the best accuracy on different prostate datasets acquired at 1.5 Tesla (table 2.1): these values can at least serve as plausible starting values for the parameter fine tuning on new classification tasks. Alternatively a search grid of parameter values may be specified, so that the best value is detected automatically: this allows to improve the classifier accuracy in some cases, while still requiring little understanding about the detailed effects of the different parameters on the side of the users.

Besides the weights of the trained classifiers, the training and testing procedures also generates test statistics that are estimated from the cross-validation schemes and saved in the HDF5 file format. By inspecting these files, one can get a detailed overview over the accuracy and reliability of the different classifiers and compare whether they yield significantly different results (Fig. 2.4).

Finally, the trained classifiers can be applied to predict the labels of new MRSI spectra for which no manual labels are available. For a user-selected region of interest, this information can be displayed in the CLARET software as an easily interpretable nosologic map overlayed over the morphological MR image (Fig. 2.5). The voxel class is encoded in the color (green for healthy tissue, red for tumor, yellow for undecided cases), while the signal quality is encoded in the alpha channel (for poor spectra the nosologic map is transparent, whereas for very good spectra it is nearly opaque).

2.5. Case studies

2.5.1. Exemplary application to 1.5 Tesla data of the prostate

The library was validated on 1.5 Tesla MRSI data of prostate carcinomas. Two different datasets were used for the training of signal quality and of voxel class classifiers: Dataset 1 (DS1) consisted of 36864 training spectra and 45312 test spectra, for which only signal quality labels were available; see (Menze et al., 2008) for further details. For joint signal quality and voxel class classification, 19456 training spectra from 24 patients with both signal quality and voxel class labels were provided; see (Kelm et al., 2007) for further details. During preprocessing, 101 magnitude channels were extracted as features for dataset 1, and 41 magnitude channels for dataset 2. No preprocessing steps besides rebinning and selection of the appropriate part of the spectrum were used. For training the voxel class classifier on dataset 2, only the 2746 spectra with "good" signal quality were used. Since relatively few spectra were available for dataset 2, an eight-fold cross-validation scheme was used on it rather than partitioning it into a separate training and test set.

Parameter (classifier)	Search grid values	Final values for DS1 (SQ) / DS2 (SQ) / DS2 (VC)
Slack penalty C (SVM)	$10^{-2}, 10^{-1}, \ldots, 10^{3}$	10^{1} / 10^{2} / 10^{2}
Number of features per node m_{try} (RF)	$4, 6, \ldots, 16$	16 / 14 / 16
L_2 norm penalty λ (RR)	$10^{-3}, 10^{-2}, \ldots, 10^{2}$	10^{-1} / 10^{-1} / 10^{-2}
Number of principal components n_{PC} (PCR)	$10, 15, \ldots, 40$	40 / 35 / 25

Table 2.1. – Search grid for automated classifier parameter selection and final values for signal quality (SQ) classification based on dataset 1 (DS1) and signal quality and voxel class (VC) classification based on dataset 2 (DS2).

As classifiers, support vector machines with linear kernel, random forests, principal component regression and ridge regression were trained, as the training of support vector machines with an RBF kernel was found to be too time-consuming. The optimal free hyperparameters were selected from the proposal values in table 2.1 by the automated parameter search capabilities of the library (using ten-fold cross-validation for the SVMs with linear kernel).

With these input data, one achieves state-of-the art classification performance: For signal quality prediction on dataset 1, the different classifiers achieved correct classification rates (CCR) of 96.5 % – 97.3 % and area under the ROC curve values of 98.9 % – 99.3 % (see table 2.2). On dataset 2, one obtains correct classification rates of 89.9 % – 92.2 % and area under curve values of 89.0 % – 94.6 % for the signal quality prediction task (table 2.3), and correct classification rates of 90.9 % – 93.7 % as well as area under curve values of 95 % – 98 % for the voxel class prediction task (table 2.4).

The automated parameter tuning functionality is especially relevant for the use of support vector machines, since wrong values of the parameter C may lead to a considerably degraded accuracy. If e.g. the starting value of 0.01 for C had been

	SVM	RF	RR	PCR
Precision	0.815	0.869	0.921	0.922
Recall	0.913	0.913	0.797	0.802
Specificity	0.972	0.982	0.991	0.991
F-score	0.861	0.891	0.855	0.857
CCR	0.965	0.973	0.968	0.968
AUC	0.989(14)	0.993(14)	0.990(14)	0.990(14)

Table 2.2. – Evaluation statistics for signal quality classifiers based on dataset 1. The standard deviation of the area under curve value (in parentheses) is estimated as proposed by Bandos et al. (2007). Note that the recall is also known as the "sensitivity".

	SVM	RF	RR	PCR
Precision	0.73(11)	0.832(57)	0.79(12)	0.79(12)
Recall	0.57(18)	0.58(17)	0.42(17)	0.43(17)
Specificity	0.964(23)	0.9820(62)	0.980(18)	0.979(19)
F-score	0.621(14)	0.67(13)	0.53(15)	0.54(16)
CCR	0.905(37)	0.922(32)	0.899(37)	0.899(38)
AUC	0.891(54)	0.946(57)	0.890(54)	0.890(54)

Table 2.3. – Average evaluation statistics for signal quality classifiers based on dataset 2 (with standard deviations in parentheses). While the standard deviation reported for the area under curve value is estimated as by Bandos et al. (2007) to facilitate the comparison with table 2.2, the other standard deviation estimates are computed from the cross-validation.

used for the signal quality classification of dataset 1, the correct classification rate would have dropped to 92.5 % (which means that the number of wrongly classified

	SVM	RF	RR	PCR
Precision	0.908(76)	0.864(27)	0.966(39)	0.900(14)
Recall	0.69(17)	0.753(16)	0.50(21)	0.50(21)
Specificity	0.983(23)	0.9771(87)	0.9966(39)	0.9928(78)
F-score	0.76(12)	0.79(11)	0.63(22)	0.63(21)
CCR	0.932(42)	0.937(42)	0.909(59)	0.909(62)
AUC	0.97(15)	0.98(15)	0.96(15)	0.95(15)

Table 2.4. – Average evaluation statistics for voxel class classifiers based on dataset 2 (see table 2.4 for further explanations).

spectra would have doubled). The other classifiers that are currently available in the library are more robust with respect to the values of their associated parameters.

While these absolute quality measures are highly relevant for the clinical practitioners, a research clinician may also be interested in the question which classifier to use for this particular task (and whether there is any difference between the different classifiers at all). This question could be answered with the statistical hypothesis testing capabilities of the library, since p-values from McNemar's test (for dataset 1) and the t-test variant (for dataset 2) characterizing the differences in the correct classification rates of various classifiers were automatically computed and corrected for multiple comparisons (both Holm's step-down and Hochberg's step-up method yielded qualitatively the same results). For the signal quality classifiers trained on dataset 1, random forests differed with high significance from all other classifiers ($p < 10^{-6}$). Support vector machines differed from principal components regression significantly ($p < 10^{-3}$), and ridge regression showed a barely significantly difference to both principal components regression and support vector machines ($p < 10^{-2}$), while all other differences were non-significant. For dataset 2, no (even barely) significant differences could be detected by Grandvalet's conservative t-test with an assumed upper bound of 0.7 for the between-fold correlation (even without Holm's or Hochberg's correction: this is presumably due to the small number of data points.

2.5.2. Extending the functionality with a k nearest neighbors classifier

As an exemplary case of how the functionality of the library may be extended, this subsection describes the addition of a new classifier method in detail, namely the k nearest neighbors (kNN) method as one of the simplest classifiers (Hastie et al., 2009). Every test spectrum is assigned the majority label of its k closest neighbors among the training spectra (with respect to the Euclidean distance).[16] This classifier is represented by a `NearestNeighborClassifier` class derived from the abstract `Classifier` base class:

```
class EXPORT_CLASSTRAIN
NearestNeighborClassifier : public Classifier {
private:
  // All training spectra
  vigra::Matrix<double> trainingSpectra;
  // All training labels
  vigra::Matrix<double> trainingLabels;
  // Training spectra for the different cross-validation folds
  std::vector<vigra::Matrix<double> > trainingSpectraCvFolds;
  // Training labels for the different cross-validation folds
  std::vector<vigra::Matrix<double> > trainingLabelsCvFolds;
  // Name strings associated with the kNN classifier
  static const std::string knn_name;
  static const std::string k_name;
  static const std::string cv_error_name;
```

[16] For binary classification, ties can easily be avoided by restricting k to odd values. However, if the user chooses an even k, the classifier errs on the safe side and classifies the spectrum as tumorous in case of a tie.

```
  static const std::string training_spectra_name;
  static const std::string training_labels_name;
protected:
  // Can be used for native multi-class classification
  virtual bool isOnlyBinary() const {
    return false;
  }
public:
  // Stub constructor
  NearestNeighborClassifier() : Classifier(),
    trainingSpectra(), trainingLabels(),
    trainingSpectraCvFolds(), trainingLabelsCvFolds(){
  }
  // Read-only access to classifier name string
  virtual std::string getClassifierName() const {
    return knn_name;
  }
  // Read-only access to error score name string
  virtual std::string getErrorScoreName() const {
    return cv_error_name;
  }
protected:
  /* The following virtual functions are discussed separately */
  ...
};
```

The only adjustable parameter is the number of nearest neighbors k. By default, the odd values $1, 3, \dots, 15$ shall be considered while optimizing over this parameter: they may also be adjusted afterwards by the library user. The last argument of the `addClassifierParameter` specifies that this parameter shall be incremented additively rather than multiplicatively.

```
void
NearestNeighborClassifier::
addClassifierSpecificParameters(){
  unsigned kValue=5;
  unsigned kLower=1;
  unsigned kUpper=15;
  unsigned kIncr=2;
  parameters->addClassifierParameter(k_name,kValue,kIncr,
                                     kLower,kUpper,false);
}
```

In this application case, the different spectral features correspond to MRSI channels and can assumed to be commensurable: hence no preprocessing except for the general MRSI preprocessing steps is required, and the associated preprocessor is an instance of the `IdentityPreprocessor` class, which leaves the features unchanged. In cases where one cannot assume the features to be commensurable, one should rather associate this classifier with a preprocessor of type `WhiteningPreprocessor` which brings all features to the same scale.

```
shared_ptr<Preprocessor>
NearestNeighborClassifier::getPreprocessorStubSpecific() const {
  shared_ptr<Preprocessor> output(new IdentityPreprocessor());
  return output;
}
```

For didactic reasons, a simple, but admittedly inefficient implementation is proposed. The training process consists simply of storing the training features and labels:

```
double
NearestNeighborClassifier::
estimatePerformanceCvFoldSpecific( FoldNr iF,
                                   const Matrix<double>& features,
                                  const Matrix<double>& labels ){
  double output = learnCvFoldSpecific(iF,features,labels);
  cvFoldTrained(iF,0)=true;
  return output;
}

double
NearestNeighborClassifier::
learnSpecific( const Matrix<double>& features,
               const Matrix<double>& labels ){
  trainingSpectra = features;
  trainingLabels = labels;
  return estimateByInternalVal( features, labels );
}

double
NearestNeighborClassifier::
learnCvFoldSpecific(FoldNr iFold,const Matrix<double>&
                    features, const Matrix<double>& labels){
  trainingSpectraCvFolds[iFold] = features;
  trainingLabelsCvFolds[iFold] = labels;
  return estimateByInternalVal( features, labels );
}
```

The automated parameter optimization requires an estimate for the generalization error, which must be obtained from one single cross-validation fold: if the data has for example been split into a training and a testing fold, only the training fold may be used for this estimation. Otherwise one would incur a bias for the test error that is computed on the separate testing dataset. Unlike many other classifiers (e.g. random forests), the kNN classifier does not automatically generate a generalization error estimate during training: hence one must resort to an internal validation step, in which the training data is split into an internal "training" and "testing" subset:

```
struct
NearestNeighborClassifier::
Comparison {
  operator()(const pair<double,double>& p1,
             const pair<double,double>& p2){
    return p1.first < p2.first;
  }
};

double
NearestNeighborClassifier::
estimateByInternalVal(const Matrix<double>& features,
                      const Matrix<double>& labels){
  unsigned k = parameters->getValue<unsigned>(k_name);
  // randomly group into two folds
  vector<int> folds( features.shape(0) );
  for( int i=0; i<features.shape(0); ++i ){
    folds[i] = rand() % 2;
  }
  unsigned correct = 0;
  unsigned wrong = 0;
  for(int i=0; i<features.shape(0); ++i ){
    if( folds[i]==0 ){ // 1 : test spectra, 0 : training spectra
      continue;
    }
    priority_queue<pair<double,double>, vector<pair<double,double> >,
      Comparison> currBest;
    unsigned nFound = 0;
    for(int j=0; j<features.shape(0); ++j){
      if( folds[j]==1 ){
        continue;
      }
      Matrix<double> tempVec = features.rowVector(i);
      tempVec -= features.rowVector(j);
      double newDist = tempVec.squaredNorm();
      if( nFound++ < k ){ // first k spectra automatically pushed
        currBest.push(pair<double,double>(newDist,labels(j,0)));
      } else {
        if( newDist < currBest.top().first ){
          currBest.pop();
          currBest.push( pair<double,double>(newDist,labels(j,0)));
        }
      }
    }
    double maxLabel = retrieveMajority(currBest);
    if( maxLabel==labels(i,0) ){
      correct++;
    } else {
      wrong++;
    }
  }
  return double(wrong)/(correct+wrong);
}
```

`retrieveMajority()` is a helper function to retrieve the most common label from the priority queue. Note that the implementation is deliberately simple for didactical reasons and has not been optimized for efficiency: in production code, one would store the training spectra in a balanced data structure like the box-decomposition trees (Arya et al., 1998) used in the ANN library[17] for faster retrieval. A similar implementation is used to predict the values of new test examples:

```
void
NearestNeighborClassifier::
predictLabelsAndScores(const Matrix<double>& featuresTrain,
                       const Matrix<double>& labelsTrain,
                       const Matrix<double>& featuresTest,
                       Matrix<double>& labelsTest,
                       Matrix<double>& scoresTest) const {
  unsigned k = parameters->getValue<unsigned>(k_name);
  labelsTest = Matrix<double>(featuresTest.shape(0),1);
  scoresTest = Matrix<double>(featuresTest.shape(0),classes.size(),0.);
  for(int i=0; i<featuresTest.shape(0); ++i){
    priority_queue<pair<double,double>, vector<pair<double,double> >,
      Comparison> currBest;
    unsigned nFound = 0;
    for(int j=0; j<featuresTrain.shape(0); ++j){
      Matrix<double> tempVec = featuresTest.rowVector(i);
      tempVec -= featuresTrain.rowVector(j);
      double newDist = tempVec.squaredNorm();
      if( nFound++ < k ){
        currBest.push(pair<double,double>(newDist,labelsTrain(j,0)));
      } else {
        if( newDist < currBest.top().first ){
          currBest.pop();
          currBest.push(pair<double,double>(newDist,labelsTrain(j,0)));
        }
      }
    }
    labelsTest(i,0) = retrieveMajority(currBest);
    while( !currBest.empty() ){
      scoresTest(i,classIndices.find(currBest.top().second)->second)+=1./k;
      currBest.pop();
    }
  }
}
```

This helper routine considerably simplifies the definition of the virtual prediction functions:

```
void
NearestNeighborClassifier::
predictBinaryScoresSpecific(const Matrix<double>& features,
                            Matrix<double>& scores) const {
  Matrix<double> labels;
  predictLabelsAndScores(trainingSpectra,trainingLabels,
```

[17] http://www.cs.umd.edu/˜mount/ANN/

```
                         features,labels,scores);
}

void
NearestNeighborClassifier::
predictBinaryScoresCvFoldSpecific(FoldNr iFold,
                                  const Matrix<double> &features,
                                  Matrix<double> &scores)const {
  Matrix<double> labels;
  predictLabelsAndScores(trainingSpectraCvFolds[iFold],
                         trainingLabelsCvFolds[iFold],
                         features,labels,scores);
}

void
NearestNeighborClassifier::
predictLabelsSpecific(const Matrix<double>& features,
                      Matrix<double>& labels) const {
  Matrix<double> scores;
  predictLabelsAndScores(trainingSpectra, trainingLabels,
                         features, labels, scores);
}

void
NearestNeighborClassifier::
predictLabelsCvFoldSpecific(FoldNr iFold, const Matrix<double>&
                            features, Matrix<double> &labels) const{
  Matrix<double> scores;
  predictLabelsAndScores(trainingSpectraCvFolds[iFold],
                         trainingLabelsCvFolds[iFold],
                         features,labels,scores);
}
```

Concerning serialization and deserialization, this classifier is only responsible for its internal data. In contrast, the serialization of the parameter k is handled by the associated `ParameterManager` object, while the evaluation statistics are serialized by the `ClassifierManager`.

```
void
NearestNeighborClassifier::
saveSpecific( shared_ptr<SaveFunctor<string> > saver) const {
  shared_ptr<SaveFunctorInterface<string, Matrix<double> > > matSaver =
    dynamic_pointer_cast<SaveFunctorInterface<string,Matrix<double> > >(
    saver);
  CSI_VERIFY( matSaver );
  matSaver->save(training_spectra_name, trainingSpectra);
  matSaver->save(training_labels_name, trainingLabels);
  for( FoldNr iF=0; iF<nCvFolds; ++iF ){
    ostringstream currMatName;
    currMatName << getFoldName() << iF << " " << training_spectra_name;
    matSaver->save(currMatName.str(), trainingSpectraCvFolds[iF]);
    currMatName.str() = "";
    currMatName << getFoldName() << iF << " " << training_labels_name;
    matSaver->save(currMatName.str(), trainingLabelsCvFolds[iF]);
  }
}

void
NearestNeighborClassifier::
loadSpecific( shared_ptr<LoadFunctor<string> >loader){
  shared_ptr<LoadFunctorInterface<string, Matrix<double> > > matLoader =
    dynamic_pointer_cast<LoadFunctorInterface<string,Matrix<double> > >(
    loader);
  CSI_VERIFY( matLoader );
  matLoader->load(training_spectra_name, trainingSpectra);
  matLoader->load(training_labels_name, trainingLabels);
  trainingSpectraCvFolds.resize(nCvFolds);
  for( FoldNr iF=0; iF<nCvFolds;++iF ){
    ostringstream currMatName;
    currMatName << getFoldName() << iF << " " << training_spectra_name;
    matLoader->load(currMatName.str(), trainingSpectraCvFolds[iF]);
    currMatName.str() = "";
    currMatName << getFoldName() << iF << " " << training_labels_name;
    matLoader->load(currMatName.str(), trainingLabelsCvFolds[iF]);
  }
}
```

On the signal quality task for dataset 1 (see section 2.5.1), this classifier achieves a correct classification rate of ca. 95 % across all tested values for the parameter k.

Chapter 3.

Brain tumor segmentation based on multiple unreliable annotations

3.1. Introduction and motivation

The use of machine learning methods for computer-assisted radiological diagnostics faces a common problem: In most situations, it is impossible to obtain reliable ground-truth information for e.g. the location of a tumor in the images. Instead one has to resort to manual segmentations by human labelers, which are necessarily imperfect due to two reasons. Firstly, humans make labeling mistakes due to insufficient knowledge or lack of time. Secondly, the medical images upon which they base their judgment may not have sufficient contrast to discriminate between tumor and non-tumor tissue. In general, this causes both a systematic bias (tumor outlines are consistently too large or small) and a stochastic fluctuation of the manual segmentations, both of which depend on the specific labeler and the specific imaging modality.

One can alleviate this problem by explicitly modelling the decision process of the human raters: in medical image analysis, this line of research started with the STAPLE algorithm (Warfield et al., 2004) and its extensions (Warfield et al., 2008), while in the field of general computer vision, it can already be traced back to the work of Smyth et al. (1995). Similar models were developed in other application areas of machine learning (Raykar et al., 2009; Whitehill et al., 2009; Rogers et al., 2010): some of them make also use of image information and produce a classifier, which may be applied to images for which no annotations are available. The effect of the different imaging modalities on the segmentation has not yet found as much attention.

In this chapter, all these competing methods as well as novel hybrid models are systematically evaluated for the task of computer-assisted tumor segmentation in radiological images: the same machinery is used on annotations provided by multiple human labelers with different quality and on annotations based on multiple imaging modalities. While traditionally these methods have been tackled by expectation maximization (EM; Dempster et al., 1977), here the underlying inference problems are formulated as probabilistic graphical models (Koller & Friedman, 2009) and thereby rendered amenable to generic inference methods. This facilitates the inference process and makes it easier to study the effect of modifications on the final inference results.[1]

3.2. Background

3.2.1. Imaging methods for brain tumor detection

T_1-, T_2- and PD-weightings in MRI For a general introduction to magnetic resonance imaging, such as principles of signal generation and spatial encoding, see section 1.2. In the following, some additional background about weightings and tissue contrast is provided, since these concepts are crucial for the detection of brain cancers from scalar MR images (in contrast to the spectral MRS images that were considered in the previous two chapters). For references, see e.g. (Yokoo et al., 2010; Kates et al., 1996). As can be derived from Eq. (1.8), the magnitude of the echo signal in a spin-echo sequence is approximately

$$A \propto \rho \left(1 - e^{-\mathrm{TR}/T_1}\right) e^{-\mathrm{TE}/T_2}, \tag{3.1}$$

with ρ being the density of MR-visible protium nuclei (PD), TR being the repetition time, i.e. the time between two subsequent 90° excitation pulses,[2] and TE being the echo time, i.e. the time between excitation and signal acquisition.[3] Image contrast between different tissues arises due to different values of the three relevant tissue parameters, ρ, T_1 and T_2. By appropriate choices for the sequence parameters TE and TR, one can weight the relative importance of these parameters: If very small values of TE are chosen (TE $\ll T_2$ for all relevant tissues), and TR is selected in the range of typical T_1 values,[4] the contrast mainly depends on T_1 and ρ (T_1 weighting). If very large values of TR are chosen (TR $\gg T_1$ for all relevant tissues)

[1] The contents of this chapter have been published as (Kaster et al., 2011).

[2] If a whole volume is imaged, multiple spin-echo sequences must be performed, which means that repeated excitation occurs before the longitudinal magnetization has completely relaxed to its equilibrium value. Eq. (3.1) describes the state after several previous excitations.

[3] Fast MR imaging techniques such as the FLASH sequence dispense with the refocussing 180° pulse and generate the echo signal purely by gradient pulses. For these techniques, the magnitude follows a similar formula, which however depends on the T_2^* instead of the T_2 time.

[4] These depend on the magnetic field strength. At 1.5 T, typical values are 250 ms for fat, 600 ms for white matter (WM), 750 ms for gray matter (GM) and 4000 ms for water and water-like liquids such as cerebrospinal fluid (CSF).

and TE is in the range of typical T_2 times,[5] the contrast mainly depends on T_2 and ρ (T_2 weighting). If both TE $\ll T_2$ and TR $\gg T_1$ is chosen, the contrast depends purely on ρ (PD-weighting).[6] The best characterization of tissues via MR is possible by combining the results from different series with different weightings (multimodal imaging).

MR contrast agents The presence of paramagnetic contrast agents in the vicinity of the precessing spins speeds up both T_1 relaxation and T_2 relaxation, by an amount which is approximately linear in the contrast agent concentration c_{CA}:

$$1/T_1^{(\mathrm{CA})} = 1/T_1 + r_1 \cdot c_{\mathrm{CA}}, \quad 1/T_2^{(\mathrm{CA})} = 1/T_2 + r_2 \cdot c_{\mathrm{CA}}, \tag{3.2}$$

where r_1 and r_2 are the relaxivities of the contrast agent. Most important for clinical applications are gadolinium(III) chelates, such as gadopentetate dimeglumine (Gd-DTPA), for which the predominant effect is on T_1 time. While the signal generation in MR imaging is due to the nuclear magnetic moments, the action of MR contrast agents is caused by the magnetic moment of their electron shell, for instance the half-filled f-shell of the Gd(III) atom. In the healthy brain, the blood-brain barrier prevents extravasation of contrast agents so that they stay in the blood pool: hence a contrast-enhancement in the brain tissue points to a disruption of blood-brain barrier integrity, which may be caused by immature blood vessels (that are often created by tumor angiogenesis), as well as inflammatory or degenerative diseases of the brain.

Inversion recovery and the FLAIR sequence The inversion recovery (IR) sequence is an alternative to the spin-echo sequence, in which the order of the 180° and the 90° pulse is interchanged: first the longitudinal magnetization is inverted by a 180° pulse, then after an inversion time TI, a transversal magnetization is created by a 90° pulse, and the FID signal is directly acquired after the 90° pulse. The signal magnitude is given by

$$A \propto \rho \left(1 - 2e^{-\mathrm{TI}/T_1}\right). \tag{3.3}$$

This sequence is frequently used for masking a certain compartment (e.g. fat or CSF) out of the MR image, by setting $\mathrm{TI}/\log(2)$ equal to the T_1 time of this compartment. An important modification is the fluid-attenuated inversion recovery (FLAIR) sequence, which combines inversion recovery with a spin echo (moderate TE, long

[5] Typical values are 60 ms for fat, 80 ms for WM, 90 ms for GM and 2000 ms for water or CSF. For T_2, the dependency on magnetic field strength is less pronounced than for T_1.

[6] Typical values for ρ are 0.7 g/ml for WM, 0.8 g/ml for GM and 1 g/ml for water or CSF. The difference between the chemical and the MR-visible proton concentration should be noted: lipids contain many immobilized protons that cannot contribute to the MR signal.

TR) in order to generate a T_2-weighted image where the CSF signal is masked out: the sequence schema is 180° – TI – 90° – TE/2 – 180° – TE/2 – ACQ.

Brain tumors The following description of medical imaging techniques for the detection of brain tumors contains common knowledge: for references see e.g. (DeAngelis et al., 2007; Debnam et al., 2007; Mikulis & Roberts, 2007). Brain tumors fall into two categories: primary brain tumors which originate from the brain (intra-axial tumor) or its direct surroundings (extra-axial tumor), and metastases of an extracranial cancer (e.g. lung cancer, breast cancer or malignant melanoma). Primary brain tumors seldom originate from neural cells, but more typically from the meninges (meningioma) or from a glia cell (e.g. astrocytoma, oligodendroglioma, glioblastoma multiforme, schwannoma). Prognostically relevant is the distinction between malignant brain tumors (which show uncontrolled proliferation, invade surrounding tissues and may metastasize) and benign tumors, which stay in a circumscribed area. However, even benign tumors may be fatal without treatment due to increased intracranial pressure. Due to their rapid proliferation, malignant brain tumors have a high demand for oxygen (and hence for blood perfusion): hence they build new blood vessels (tumor angiogenesis), which often have abnormal lining cells so that the blood-brain barrier may be disrupted inside the tumor. This is the reason why most tumors are surrounded by edema (i.e. blood plasma leaking in the intercellular space of the brain tissue). If the angiogenesis cannot keep step with the growth of the tumor, the core of the tumor becomes first hypoxic and later necrotic: this is indicative of highly aggressive malignancies. Radiological imaging diagnostics is typically indicated when neurological symptoms are observed, such as deficits in sensation, motion or language, seizures or impairments of alertness or cognition; also metastasis screening should be performed upon diagnosis of a primary tumor which is known often to metastasize to the brain.

Imaging of brain tumors The first choice for imaging diagnostics is magnetic resonance imaging (see section 1.2); computed tomography (CT) and positron emission tomography have typically lower sensitivity and specificity and are mainly useful either as a supplement or for patients which have a contraindication for high magnetic fields (e.g. metallic implants or cardiac pacemakers). Common tumor imaging protocols comprise two T_1-weighted scans (before and after injection of a contrast agent such as Gd-DTPA), a diffusion-weighted scan and either a T_2-weighted or a FLAIR scan. Gadolinium enhancement is the best indicator for aggressive (high-grade) malignancies. As necrotic tissue does not take up contrast agents, tumors with a necrotic core typically display a ring-shaped enhancement pattern, while tumors without a necrotic core are uniformly enhanced. However, low-grade and benign brain tumors show no enhancement after Gd-DTPA injection. They can be detected by the second radiological tumor sign, namely abnormal relaxation times: Most tumors are hypocellular (with increased T_1 and T_2 times) and appear as hypointensities in T_1-weighted and as hyperintensities in T_2-weighted (or FLAIR) images; while some tumors are hypercellular (with decreased relaxation times), where the effects are exactly reversed. In diffusion-weighted magnetic resonance imaging (DWI), the image intensity is attenuated by a factor of e^{-bD}, where b is a constant and D is the local diffusion coefficient. This is achieved by two gradient fields of equal strength that are applied symmetrically around the 180° pulse. For resting nuclei, they do not effect the signal, as the first gradient field causes a dephasing that is exactly rephased by the second gradient field. However, protium nuclei that have moved along the gradient direction experience a different field strength during rephasing than during dephasing, leading to the attenuation. Diffusion is increased in hypocellular regions; accordingly hypercellular tumors appear as hyperintensities and hypocellular tumors appear as hypointensities in diffusion-weighted imaging. Additional imaging techniques such as MRSI, functional MRI or perfusion-weighted imaging may further improve the differential diagnosis, but they are rarely used in clinical routine (mainly due to time constraints). The gold standard for tumor diagnosis and grading is the histopathological examination of an image-guided biopsy.

3.2.2. Variational inference for graphical models

Graphical models Probabilistic graphical models (Koller & Friedman, 2009; Wainwright & Jordan, 2008) are a tool for encoding the conditional independence relationships between random variables, and for inferring upon the values of unobserved (or hidden) variables $H = \{H_i | i = 1, \ldots, N_\mathrm{H}\}$ based on the values of observed variables $V = \{V_i | i = 1, \ldots, N_\mathrm{V}\}$. This chapter only considers directed graphical models (also known as Bayesian networks), which directly specify the factorization properties of the joint probability density over all variables: If $X = H \cup V$, a Bayesian network over the variables X is a directed graph with vertex set X, such that

$$p(X) = \prod_{i=1}^{N_\mathrm{H}+N_\mathrm{V}} p(X_i | \mathrm{pa}_i), \tag{3.4}$$

with pa_i denoting the parents of variable X_i in the graph (see Fig. 3.1 for an example). The factors $p(X_i|\mathrm{pa}_i)$ are called the conditional probability distributions (CPDs) of the Bayesian network.

Aims of inference Typical inference goals for such models are:

1. Computing the posterior marginals $p(H_i|V)$.

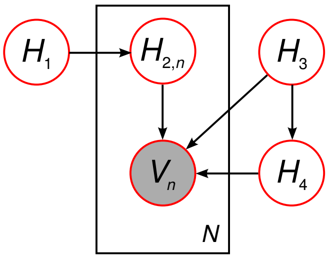


Figure 3.1. – Simple example for a Bayesian network. The graph nodes correspond to random variables; observed variables are denoted by a gray filling. All variables drawn inside the rectangle stand for an array of N variables $V_1, \ldots, V_N$ and $H_{2,1}, \ldots, H_{2,N}$ (plates notation, see Buntine (1994)). The edges denote the factorization properties of the joint probability distribution. For this example, $p(H, V) = p(H_1)p(H_3)p(H_4|H_3)\prod_{i=1}^{N}\left[p(H_{2,n}|H_1)p(V_n|H_{2,n}, H_3, H_4)\right]$.

2. Computing the evidence $p(V)$ of the observations given the current model. This may be useful for selecting a graphical model that captures the structure of the data well. A common problem in model selection is choosing the proper number of hidden variables: more hidden variables typically correspond to higher flexibility, so that the observations can be fitted more accurately, but at the same time the danger of overfitting arises. Bayesian model selection provides an elegant way to tackle this problem: consider two models $\mathcal{M}_1, \mathcal{M}_2$ with different numbers of variables. Then

$$p(V|\mathcal{M}_i) = \int dH\ p(V|H)p(H|\mathcal{M}_i) \tag{3.5}$$

results from a likelihood term $p(V|H)$ and an "Occam's razor" term $p(H|\mathcal{M}_i)$. For complex models with more parameters, the observations can usually be fitted better ($p(V|H)$ is higher for the best choice of H), but it becomes less likely that the hidden variables take this particular value out of the much larger space of possible values. Hence both overly simple and overly complex models are discouraged (Kass & Raftery, 1995).

3. Finding the maximum a posteriori (MAP) solution for the hidden variables $H^* = \arg\max_H p(H, V) = \arg\max_H p(H|V)$.

4. Computing the predictive distribution $p(\hat{v}|V)$ that specifies which observations $\hat{v}$ can be expected when sampling from the same graphical model with the same hidden variables.

Exact inference via junction trees Exact inference on Bayesian networks can be performed by the junction tree algorithm: First the directed graph is transformed into an undirected graph by moralization, i.e. by converting all directed edges into undirected edges and connecting all common parents of a node.[7] Afterwards the moralized graph is chordalized, i.e. edges are introduced in order to remove all chordless cycles of length greater than three. Then a junction tree is constructed on the chordalized graph, i.e. a tree graph whose nodes correspond to the maximum cliques C_i of the chordalized graph and whose edges link cliques sharing the same variables so that the running intersection property is respected (if a variable is present in two cliques, it must be present in all cliques on the unique path between those two cliques on the junction tree). Then every factor is assigned to some clique in this junction tree: $\psi_i(C_i)$ denotes the product of all CPDs assigned to the clique C_i. Finally, a message-passing algorithm is run, in which messages of the following kind are sent between neighboring cliques in a specific update order:[8]

$$\delta_{i\to j}(C_i \cup C_j) = \sum_{C_i \setminus C_j} \psi_i(C_i) \prod_{k\sim i, k\neq j} \delta_{k\to i}(C_k \cup C_i) \tag{3.6}$$

After messages have been passed along every edge in both directions, the clique marginals are given by

$$\beta_i(C_i) = \sum_{X\setminus C_i} p(X) = \psi_i(C_i) \prod_{k\sim i} \delta_{k\to i}. \tag{3.7}$$

Limitations of exact inference However, the complexity of this junction tree algorithm is exponential in the size of the largest clique in the junction tree for the optimum chordalization, which is called the treewidth of the original moralized graph.[9][10]

[7] This “marrying” of unconnected parents accounts for the “explaining away” property of Bayesian networks. This is best explained by the famous burglary-earthquake example by Pearl (1988). Both a burglary and an earthquake may set off an alarm bell in a house, and we can assume that both events occur independently from each other. However, once we know that the alarm bell rang, both the probability of a burglary and an earthquake become more likely; but if we know that a burglary occurred, the probability of an earthquake becomes less likely again and vice versa. This means that the common parent variables of a child variable are not conditionally independent given the child variable, even if they are independent when the child variable is marginalized over.

[8] Eqs. (3.6) and (3.7) describe the sum-product message-passing algorithm that is used to compute posterior marginals. For MAP estimation, all summations have to be replaced by maximizations (max-product algorithm).

[9] To be exact, the treewidth is defined as the minimum size of the largest clique of all chordal graphs containing the original graph minus one.

[10] There exist graphical models for which the junction tree algorithm has more favorable complexity: e.g. if all factors are Gaussians for which the marginalization can be performed analytically (Gaussian processes), the complexity is cubic in the treewidth.

Since there exist different possibilities for the chordalization, determining the optimum chordalization and hence the treewidth for a given Bayesian network is not straightforward: in fact, it is an NP-complete problem except for specialized classes of graphs (Bodlaender, 1992). As will be shown later, the graphical models that we analyze in this chapter have a treewidth linear in the number of raters and the number of image features used for the supervised classification; hence exact inference would only be practicable if there were very few raters and if the objective image information were disregarded.

Markov Chain Monte Carlo However, the computation time can be highly reduced if one dispenses with exact solutions and allows approximations. Most popular approximate inference techniques fall into one of two categories: Markov Chain Monte Carlo (MCMC) techniques (Andrieu et al., 2003) and variational approximations (Wainwright & Jordan, 2008). MCMC techniques approximate the (intractable) analytical marginal $p(H)$ by an empirical point mass density

$$p_N(H|V) = \frac{1}{T}\sum_{t=1}^{T}\delta(H - H^{(t)}), \tag{3.8}$$

where the T samples $H^{(t)}$ are drawn independently and identically distributed from the true $p(H|V)$. This sampling process is typically achieved by variants of the $\mathrm{MR_2T_2}$ algorithm (Metropolis et al., 1953) in which one or more particles perform random steps in the state space of all possible H, which may or may not be accepted based on the changes in $p(H, V)$: the states of the particle at the different points in their trajectory are then used as the random samples. An important special case is the Gibbs sampler (Geman & Geman, 1984), for which only one hidden variable H_i is updated in each step: namely, it is sampled from the conditional distribution $p(H_i|\{H_j^{(t)} : j \neq i\}, V)$ obtained by fixing all other hidden variables to their current values. MCMC techniques have been shown to be practically useful, though computationally expensive, and there are software products such as BUGS (Gilks et al., 1994; Lunn et al., 2000) or INFER.NET (Minka et al., 2009) that can perform generic MCMC inference on a variety of graphical models.

Variational inference and Rényi entropies Variational inference methods follow a different strategy: the true posterior $p(H|V)$, for which inference is intractable, is approximated by the closest $q(H)$ in a family $\mathcal{F}$ of distributions that allow tractable inference: "closest" is here defined with respect to a divergence measure between pairs of distributions $D(p\|q)$. Commonly $D(p\|q)$ is selected out of the family of

Rényi α-entropies (Rényi, 1961; Minka, 2005). If p and q are probability densities, then

$$D_\alpha(p\|q) = D_{1-\alpha}(q\|p) = \int dH \left[\frac{p(H)}{1-\alpha} + \frac{q(H)}{\alpha} - \frac{p(H)^\alpha q(H)^{1-\alpha}}{\alpha(1-\alpha)}\right]. \tag{3.9}$$

The two most important special cases are the inclusive ($\alpha = 1$) and exclusive ($\alpha = 0$) Kullback-Leibler (KL) divergence:

$$D_1(p\|q) = \mathrm{KL}(p\|q) = \int dH\ p(H) \log\left(\frac{p(H)}{q(H)}\right) + \int dH\big(q(H) - p(H)\big), \tag{3.10}$$

$$D_0(p\|q) = \mathrm{KL}(q\|p) = -\int dH\ q(H) \log\left(\frac{p(H)}{q(H)}\right) - \int dH\big(q(H) - p(H)\big). \tag{3.11}$$

For large values of α, the closest distribution q^* to a given distribution p tends towards majorization of p: for $\alpha \geq 1$, $p(H = h) > 0$ implies that also $q^*(H = h) > 0$ (zero-avoiding property), and in the limit $\alpha \to \infty$, $q^*(H) > p(H)$ holds everywhere.[11] The closest q^* hence tries to best fit the entire shape of the true p. In contrast, for small values of α the best approximation q^* tends towards minorization of the true p: for $\alpha \leq 0$, $p(H = h) = 0$ implies that also $q^*(H = h) = 0$, and in the limit $\alpha \to -\infty$, $q^*(H) < p(H)$ holds everywhere. The closest q^* hence tries to best fit the tails of the true distribution of the true p.

Inference by local updates Finding the closest q^* is achieved approximately via an iterative local update scheme (Minka, 2005), in which both the true p and the approximation q are partitioned into factors (the CPDs of the Bayesian network) and the factors of q are locally fit to the factors of p. Assume the following factorizations:

$$p(H) = \prod_i p_i(H), \quad q^*(H) = \prod_i q_i^*(H), \tag{3.12}$$

and define

$$p^{\setminus i}(H) = \prod_{j \neq i} p_j(H) = \frac{p(H)}{p_i(H)}. \tag{3.13}$$

We now want iteratively select q_i^* so that given the other factors, p is approximated best. The optimal local solution,

$$q_i^* \leftarrow \arg\min_{q_i} D\left(p_i p^{\setminus i} \| q_i q^{*\setminus i}\right), \tag{3.14}$$

[11] Note that we do not require q^* to be normalized: after normalization, this property does obviously no longer hold.

would be intractable, but if $q^{*\setminus i}$ approximates $p^{\setminus i}$ already adequately, Eq. (3.14) can be approximated by the tractable

$$q_i^* \leftarrow \arg\min_{q_i} D\left(p_i q^{*\setminus i} \| q_i q^{*\setminus i}\right). \tag{3.15}$$

Using the inclusive KL divergence ($\alpha = 1$) in this local update scheme, together with some additional assumptions leads to the expectation propagation algorithm by Minka (2001), while the use of the exclusive KL divergence ($\alpha = 0$) leads to variational message passing (Winn & Bishop, 2005). More general choices of α lead to the power expectation propagation algorithm (Minka, 2004). The advantage of choosing $\alpha = 0$ is that it provides an exact lower bound on the model evidence: note that

$$\log p(V) = \mathcal{L}(q) + \mathrm{KL}(q\|p) \geq \mathcal{L}(q) = \int dH q(H) \log\left(\frac{p(H,V)}{q(H)}\right), \tag{3.16}$$

which is tractable as it only involves a marginalization over $q(H)$.

Variational message passing After this generic view on variational inference techniques, we now discuss the variational message passing (VMP) algorithm by Winn & Bishop (2005) in detail. For the family $\mathcal{F}$, we choose all distributions q that factorize over all variables, and for which inference is hence trivially tractable:

$$q(H) = \prod_i q_i(H_i). \tag{3.17}$$

In this case, the solution of Eq. (3.15) is given by

$$\log q_j^*(H_j) = \mathbb{E}_{q_i^*(H_i), i\neq j}\left[\log p(H,V)\right]. \tag{3.18}$$

By the graphical model structure, $\log p(H,V)$ can be written as a sum of log-factors, most of which do not depend on H_j and which are hence irrelevant for the functional form of $q_j^*(H_j)$. For evaluating the expectation value in Eq. (3.18), we must only consider the local factors $q_i^*(H_i)$ for which i lies in the Markov blanket of j, i.e. is either a child, parent or coparent (i.e. another parent of a child) of j:

$$\log q_j^*(H_j) = \mathbb{E}_{i\in \mathrm{pa}_j}\left[\log p(H_j|\mathrm{pa}_j)\right] + \sum_{k\in \mathrm{ch}_j} \mathbb{E}_{i\in\left(\{k\}\cup \mathrm{cp}_k^{(j)}\right)\cap H}\left[\log p(X_k|\mathrm{pa}_k)\right] + \mathrm{const}, \tag{3.19}$$

with pa_j, ch_j being the sets of parents and children of j, and $\mathrm{cp}_k^{(j)} = \mathrm{pa}_k \setminus H_j$.

Conjugate-exponential models In order to evaluate Eq. (3.19) efficiently and to summarize the distribution q_j^* succinctly, we add the constraint that the factors of $p(H, V)$ must be conjugate-exponential models: Consider an arbitrary (observed or unobserved) variable of the graphical model, which shall be denoted by X_1 without loss of generality. Denote the parent nodes of X_1 by $Y_1, Y_2, \ldots$. Then two conditions must hold:

1. **Exponential family**: The CPD of X_1 given its parents has the following log-linear form:

$$\log p(X_1|Y_1, Y_2, \ldots) = \phi(Y_1, Y_2, \ldots)^\top u_{X_1}(X_1) - g_{X_1}\big(\phi(Y_1, Y_2, \ldots)\big). \quad (3.20)$$

   The vector u_{X_1} is called the natural statistics of X_1 and determines the family of distributions to which $p(X_1|Y_1, \ldots)$ belongs: e.g. for a Gaussian distribution, $u_{X_1}(X_1) = (X_1, X_1^2)^\top$, while for a Gamma distribution, $u_{X_1}(X_1) = (X_1, \log X_1)^\top$. The vector ϕ is called the natural parameters and parameterizes the specific distribution in the family, and the normalization summand g_X is known as the log-partition function.

2. **Conjugacy**: The prior distributions $\log p(Y_i|\mathrm{pa}_i)$ on the parents Y_i must have the same functional parameter dependence on Y_i as $p(X_1|Y_1, \ldots)$, i.e. if

$$\log p\left(Y_i|\mathrm{pa}_i\right) = \phi_{Y_i}\left(\mathrm{pa}_i\right)^\top u_{Y_i}(Y_i) - g_{Y_i}\left(\phi_{Y_i}(\mathrm{pa}_i)\right), \quad (3.21)$$

   then it must be possible for all i to write

$$\log p(X_1| \ldots, Y_i, \ldots) = \phi_{X_1 Y_i}\left(X_1, \mathrm{cp}_1^{(i)}\right)^\top u_{Y_i}(Y_i) + \lambda_i\left(X_1, \mathrm{cp}_1^{(i)}\right) \quad (3.22)$$

   with some functions λ_i and ϕ_{XY_i}. This is best explained with a simple example: consider a Gaussian variable X_1 with a mean Y_1 and a precision Y_2, which are themselves random variables:

$$\begin{aligned}
\log p(X_1|Y_1, Y_2) &= \log\left(\sqrt{\frac{Y_2}{2\pi}} \exp\left(-\frac{Y_2(X_1 - Y_1)^2}{2}\right)\right) && (3.23)\\
&= \begin{pmatrix} Y_1 Y_2 \\ -Y_2/2 \end{pmatrix}^\top \begin{pmatrix} X_1 \\ X_1^2 \end{pmatrix} + \frac{1}{2}\left(\log Y_2 - Y_2 Y_1^2 - \log(2\pi)\right) && (3.24)\\
&= \begin{pmatrix} X_1 Y_2 \\ -Y_2/2 \end{pmatrix}^\top \begin{pmatrix} Y_1 \\ Y_1^2 \end{pmatrix} + \frac{1}{2}\left(\log Y_2 - X_1^2 Y_2 - \log(2\pi)\right) && (3.25)\\
&= \begin{pmatrix} X_1 Y_1 - X_1^2/2 - Y_1^2/2 \\ 1/2 \end{pmatrix}^\top \begin{pmatrix} Y_2 \\ \log Y_2 \end{pmatrix} - \frac{1}{2}\log(2\pi) && (3.26)
\end{aligned}$$

If written as a function of Y_1, $P(X_1|Y_1,Y_2)$ has the form of a Gaussian, while written as a function of Y_2, it has the form of a Gamma distribution. Hence conjugacy is only fulfilled if the prior on the mean $p(Y_1|\mathrm{pa}_1)$ is also a Gaussian and the prior on the precision $p(Y_2|\mathrm{pa}_2)$ is also a Gamma distribution.

Mean parameterization and VMP updates If the natural statistics vector u_X of an exponential model is a minimal representation (meaning that its components are linearly independent), there are two equivalent parameterizations of this model: the natural parameter vector ϕ_X and the mean parameterization, also known as the gradient mapping

$$\mu_X = \mathbb{E}_{p(X)}[u_X(X)] = \nabla_{\phi_X} g_X(\phi_X). \tag{3.27}$$

For the simple case of a Gaussian with mean ψ and precision λ, the two parameterizations are given by

$$\phi_X = \begin{pmatrix} \lambda\psi \\ -\lambda/2 \end{pmatrix} = \begin{pmatrix} \mu_{X1}\left(\mu_{X2}-\mu_{X1}^2\right)^{-1/2} \\ -\left(\mu_{X2}-\mu_{X1}^2\right)^{-1/2}/2 \end{pmatrix}, \tag{3.28}$$

$$\mu_X = \begin{pmatrix} \psi \\ \psi^2+\lambda^{-2} \end{pmatrix} = \begin{pmatrix} -\phi_{X1}/\left(2\phi_{X2}\right) \\ \left(\phi_{X1}^2-1\right)/\left(4\phi_{X2}^2\right) \end{pmatrix}. \tag{3.29}$$

Let $\tilde{\phi}_X$ denote the inverse gradient mapping from μ_X to the corresponding ϕ_X. If all CPDs in the VMP problem are conjugate-exponential models, then the $q_j^*(H_j)$ solving Eq. (3.19) is in the same exponential family as $p(H_j|\mathrm{pa}_j)$, i.e. it is a multilinear function of the same statistics vector u_{H_j}. Its updated parameter vector is given by

$$\phi_{H_j}^* = \mathbb{E}\left[\phi_{H_j}(\mathrm{pa}_j)\right] + \sum_{k\in\mathrm{ch}_j} \mathbb{E}\left[\phi_{X_k H_j}\left(X_k, \mathrm{cp}_k^{(j)}\right)\right]. \tag{3.30}$$

Another key implication of the conjugacy is that the expectation values of the natural parameters in Eq. (3.30) can be uniquely determined from the expectation values of the natural parameters of the other variables in the Markov blanket via the inverse gradient mapping. As the latter are just the mean parameters of the distributions of these other parameters, these mean parameters capture all the information that H_j must know about its parents, children and coparents. Hence Eq. (3.30) may be written as

$$\begin{aligned} \phi_{H_j}^* &= \tilde{\phi}_{H_j}\left(\{\mu_{H_k}\}_{k\in\mathrm{pa}_j}\right) + \sum_{k\in\mathrm{ch}_j} \tilde{\phi}_{X_k H_j}\left(\mu_{X_k}, \{\mu_{H_i}\}_{H_i\in\mathrm{cp}_k^{(j)}}\right) && (3.31) \\ &= \tilde{\phi}_{H_j}\left(\{m_{X_i\to H_j}\}_{X_i\in\mathrm{pa}_j}\right) + \sum_{k\in\mathrm{ch}_j} m_{X_k\to H_j}, && (3.32) \end{aligned}$$

with the messages

$$m_{X_i \to H_j} = \mu_{X_i} \quad \text{for } X_i \in \mathrm{pa}_j, \tag{3.33}$$

$$m_{X_k \to H_j} = \tilde{\phi}_{X_k H_j}\left(\mu_{X_k}, \{\mu_{H_i}\}_{H_i \in \mathrm{cp}_k^{(j)}}\right) \quad \text{for } X_k \in \mathrm{ch}_j. \tag{3.34}$$

The variational message passing algorithm consists of iteratively updating the parameters of all nodes based on Eq. (3.32), and updating the lower bound on the evidence $\mathcal{L}$, until a local optimum is reached.

3.3. Related work

The work presented in this chapter lies in the intersection of two areas, which come together for the first time: latent variable and latent score models for learning with unreliable annotations (methodology), which are used for learning brain tumor segmentations from medical imagery (application area). First an overview over the different precious approaches for tackling the application task is given in subsection 3.3.1, while the methodologically related work is discussed in subsection 3.3.2.

3.3.1. Automated methods for brain tumor segmentation

Even for the constrained task of automated brain tumor segmentation in medical imagery, there exist so many previous approaches that a complete enumeration would go beyond the scope of this chapter. The following examples should hence be viewed only as a representative selection.

Methods based on generative models

Generative methods for tumor segmentation can often be formulated in the formalism of graphical models that is also used in this chapter for fusing the information from various different unreliable sources. However, instead of modelling the labeling process of the raters, these techniques usually propose probabilistic models for the generation of the visible image information given the hidden class labels.

For instance, Moon et al. (2002) and Prastawa et al. (2003b) propose an extension of the expectation maximization method by Leemput et al. (1999b) for brain segmentation with an atlas prior to joint brain, tumor and edema segmentation by adding class models for tumors and edema. The basic idea is to assume a Gaussian likelihood for each tissue class (with unknown parameters), to add a spatially varying prior for each class derived from a probabilistic brain atlas, and to jointly learn the likelihood parameters, the multiplicative bias field (which accounts for smooth intensity inhomogeneities in the image) and the class assignments of the voxels by an EM algorithm, with the class assignments and the bias field parameters being treated as hidden variables. Spatial priors for the tumor and the edema class are constructed as follows: The difference of two log-transformed T_1-weighted MR scans before and after gadolinium contrast enhancement is assumed as bias-free (since the multiplicative bias fields are assumed to have canceled out). The intensity histogram of this difference image is modeled by two Gaussians (corresponding to noise) and a gamma distribution (corresponding to tumor and other enhancing regions like blood vessels). The posterior probability of the gamma term is then interpreted as tumor prior. Since edema is mostly observed in WM regions, the edema prior is modeled experimentally as a fraction of the WM prior.

Nie et al. (2009) account for the different spatial resolutions of the different imaging modalities by proposing a spatial accuracy-weighted hidden Markov random field expectation maximization (SHE) procedure for fully automated segmentation of brain tumors from multi-channel MR images. Typically high-resolution (pre- and post-contrast) T_1-weighted images are combined with low-resolution T_2-weighted or FLAIR images by registration: since interpolation is required for resampling the low-resolution measurements, their accuracy is assumed to be lower. The geometric mean of distances to the voxels in the original image is used as the accuracy measure. As a generative model, a Gaussian hidden Markov Random Field (MRF) is used, for which the clique potentials are weighted by the product of accuracies of all neighbor pixels contributing to the interpolated signal. Parameter estimation is performed by the EM algorithm. The procedure is evaluated on the task of segmenting brain tumors from T_1-weighted, T_2-weighted and FLAIR MR images, after brain stripping and bias field correction as preprocessing steps. Compared to the results of two raters, no significant difference to the inter-rater results could be found (measured by Jaccard index and volume similarity).[12]

Particularly interesting is the approach by Corso et al. (2006, 2008), who propose a hybrid of two successful segmentation approaches: generative Bayesian models and normalized cut segmentation, the latter in the *segmentation by weighted aggregation* (SWA) approximation. As a generative model, a Gaussian mixture model is used for each of four classes (brain, tumor, edema, non-brain), whose parameters are estimated from the training data by the EM algorithm. The normal SWA algorithm generates a hierarchical segmentation by successively merging nodes based on their affinity (i.e. feature distance) and accumulating their statistics: this allows foreground objects of different scales to be detected (corresponding to different hierarchy

[12] The Jaccard index is the ratio of the intersection and the union of detected and true tumor volume, while the volume similarity is defined as $1 - |V_\mathrm{D} - V_\mathrm{T}|/(V_\mathrm{D} + V_\mathrm{T})$, where V_D and V_T are the detected and the true tumor volume.

levels). The newly proposed algorithm differs in two respects by incorporating the generative model: every node is assigned a model class, and the affinity is modulated such that nodes of the same class have an affinity near 1, and that nodes of different classes have an affinity near 0. The parameters are again learned from the training data by a stochastic search. Only the intensities in the different modalities are used as features. The algorithm has linear time complexity in the number of voxels v, but typically high memory requirements for storing the multi-level representation (scaling as $v \log(v)$); on a state-of-the-art PC, segmentation of an image volume takes 1-2 minutes (with ca. 5 minutes required for preprocessing). Evaluation against manual ground truth on multispectral datasets (pre- and post-contrast T_1-weighted MRI, T_2-weighted MRI, FLAIR, which are subsampled to the lowest resolution) yields average Jaccard scores for tumor and edema detection of 69 % and 62 %. For the majority of datasets, the median distance between automatic and ground-truth segmentation is 0 mm (meaning that most voxels of these two boundaries coincide).

Methods based on outlier detection

While generative models can capture well the intensity distributions of the different classes in healthy brain tissue, pathological lesions such as tumors or multiple sclerosis hyperintensities are often harder to model, and the common assumption of Gaussianity may be violated. This is the reasoning behind outlier-based segmentation methods, which fit a generative model to the normal tissues and detect all pathologies as outliers to this model.

Gering et al. (2002) propose a hierarchical classification procedure for learning models of healthy tissue classes and assigning the voxels to those classes, in which higher levels may correct wrong decisions made on lower levels. On the lowest level, an EM algorithm is used to learn the intensity distribution of GM, WM and CSF, treating the bias field and the class assignments of the single voxels as hidden variables, exactly as in (Leemput et al., 1999a). Spatial context is introduced on the second level by imposing a Potts model MRF prior on the class assignments, which is relaxed to a mean-field approximation for tractability as in (Leemput et al., 1999b). On the third level, the position of every voxel inside the structure of equally labeled voxels is considered, mainly its distance from the structure boundaries (e.g. if a WM voxel lies in the center of the white matter or borders neighboring structures). The prior probabilities for large distances from the boundary may then be increased, which favors large homogeneous regions and may remove spurious misclassifications. On the fourth level, global prior information such as digital atlas priors or priors on the distances between several structures (such as ventricles and skin) may be imposed. The fifth level is the interaction with the user, who initializes the iterative fitting of the models for the healthy classes by providing examples for each class with a quick brush stroke. Manual correction of misclassified voxels would also be possible on this level. Several iteration passes over these five levels are then performed until convergence; tumor voxels are identified as outliers with respect to the Mahalanobis distance to the center of the class they are assigned to.

Gering (2003) proposes a new metric called nearest neighbor pattern matching (NNPM) for judging the abnormality of an image window. For each window center position, a set of template windows corresponding to normal texture examples at this location is provided and the NNPM of the window is defined as the smallest root-mean-squared distance to any template in the set. In order to resolve texture similarity at different scales, a scale-space representation is used and a joint pathology probability is defined by treating the probabilities at each resolution as independent (i.e. the joint probability is the product of the pathology probabilities at the different scales, where a Gaussian assumption is used to extract a pathology probability from the distance).

Prastawa et al. (2003a, 2004) detect brain tumors as outliers in multispectral MR images, after robustly learning models for the healthy tissue classes: A probabilistic brain atlas is used to draw samples for all healthy classes (WM, GM, CSF) from locations characteristic for the respective class. A Gaussian model is assumed for each class, whose parameters are estimated with an outlier-robust estimator (Minimum Covariance Determinant); samples further than three standard deviations apart from the mean are discarded as outliers (tumor, edema) and assigned to an "abnormal" class. The distributions of all classes (GM, WM, CSF, abnormal, non-brain) are then re-estimated nonparametrically by a kernel density estimation, and the posterior probabilities are computed for all voxels. After estimating and correcting for a bias field, the whole process is iterated with the posterior probabilities in lieu of the prior atlas probabilities. After the abnormal class is finally segmented, it is partitioned into tumor and edema by k-means clustering with $k = 2$; if there exist two separate clusters (as measured by the Davies-Bouldin overlap index), the cluster with the lower mean T_2-weighted intensity is labeled as tumor. The tumor segmentation is then refined by performing a level set evolution initialized with the distance transform of the presegmented tumor; then false positives for the edema class are discarded by performing a connected component analysis and removing all components without contact to a tumor. This procedure is also iterated, disabling the level set in the final iteration step. Validation on bispectral datasets with T_1- and T_2-weighting yields overlap fractions of $77 \pm 5\%$ and Hausdorff distances of 12.7 ± 4.1 mm for tumor segmentation, while intra-rater comparison yields $77 \pm 15\%$ and 4.43 ± 0.68 mm.

Methods based on discriminative learning without explicit context information

The following methods are closest in spirit to the variants of logistic regression that will later be discussed in this chapter. Instead of directly modeling the joint distributions of features and labels $p(x, y)$, as generative models do, discriminative models restrict themselves to modeling the conditional distribution $p(y|x)$, which is also the relevant distribution for prediction purposes. This is an easier task as the feature distribution need not be detected, however it also poses the risk of overfitting if few training data are available. First we discuss only discriminative models that account for purely local image information, without taking spatial context into account:

Schmidt et al. (2005) explore support vector machine (SVM) classification with several combinations of alignment-based features for brain tumor segmentation in multispectral (pre- and post-contrast T_1-weighted and T_2-weighted) MR images in order to facilitate inter-patient training without need to provide patient-specific training examples. Preprocessing steps are noise reduction by nonlinear filtering, inter-slice intensity normalization, intra-volume bias field correction, mutual information-based multimodal registration, matching to an atlas template by a linear and a nonlinear step, resampling to the template coordinate system and inter-volume intensity standardization (in all steps methods were used that are mostly robust to the presence of tumors). Four types of alignment-based local features are then extracted: the distance transform of the brain area of the template (B feature), spatially dependent probabilities for the three main normal tissue classes (P features), spatially dependent average intensities for healthy brains in the different modalities (A features) and the intensity difference to the contralateral voxel to characterize local symmetry or asymmetry (S features). Also textural features are created by applying a multiscale Gaussian convolution filter bank. A linear kernel SVM is then trained, and the classification results of test images are postprocessed by repeated median filtering (in order to remove isolated labels) and selection of the largest connected component. For the best combination of alignment-based features (P, A and S) together with the texture features an average Jaccard score of 0.732 is obtained (which outperforms several other feature sets taken from previous literature).

Zhou et al. (2005) use a one-class learning procedure (one-class RBF kernel SVM) to learn the appearance of tumorous areas in pre- and post-contrast T_1-weighted images (only the gray values from both modalities are used as features). This yields a sensitivity of $83.3 \pm 5.1\%$ and a correspondence rate (true positives $-$ half of false positives, normalized by total number of tumor voxels) of 0.78 ± 0.06, while FCM (see section 3.3.1) only achieves values of $76.2 \pm 4.8\%$ and 0.73 ± 0.07.

Methods based on discriminative learning with incorporated context information

In cases where the local information is ambiguous, taking spatial context into account can often improve the segmentation: voxels that are surrounded by tumor voxels have an increased likelihood of being tumor voxels themselves, and likewise for healthy tissue. This increased model complexity comes at a price of increased computational complexity: finding the MAP solution of a spatially regularized model often leads to a discrete optimization problem that is intractable or only tractable in special cases, so that one has to resort to approximate solutions. The following approaches start from local discriminative classifiers as discussed in the previous section, and augment them with spatial context information:

Lee et al. (2005) compare three context-sensitive classification procedures (Markov random fields (MRF) as a generative model, discriminative random fields (DRF) and support vector random fields (SVRF) as discriminative models) with their context-free degenerated versions (naive Bayesian, logistic regression and support vector machines) for the task of segmenting brain tumors from multispectral MR images. The three context-sensitive models are all graphical models with single-site and pair potentials: for the MRF, the single-site potentials are Gaussians and the pair potential only depends on the local label assignments (e.g. a Potts potential); for the DRF, the single-site potentials are a generalized linear model (e.g. logistic regression terms) and the pair potential may be modulated by the (possibly non-local) features (here the penalty for different adjacent labels is attenuated if the features at the two voxels differ by a large amount). For the SVRF finally, the logit-transformed output of an SVM is chosen as single-site potential, and the same interaction term as for the DRF is chosen; it is assumed that the SVRF performs superior to the DRF in high-dimensional feature spaces with correlated features. The parameters of an SVRF can be trained by a solving a quadratic program. For inference, the label assignments of the context-sensitive classifiers are initialized with the locally optimal labels, and the final label assignment is computed using ICM (see section 1.3). Several preprocessing steps for noise reduction, bias-field correction, inter-slice intensity normalization and registration to an anatomical template are performed. Using alignment-based features as in (Schmidt et al., 2005) and evaluating the classifiers on three different tasks (segmenting the enhancing tumor region, the gross tumor region and the edema region), it turns out that SVRFs perform best for all three tasks (with average Jaccard indices of 0.825, 0.723 and 0.769).

Lee et al. (2006) propose semi-supervised discriminative random fields (SSDRF) as a semi-supervised generalization of classical discriminative random fields to be used for general computer vision problems, and use brain tumor segmentation as the main experimental application example of their article. The unlabeled data are used in order to decrease the risk of parameter overfitting, by adding the expected conditional entropy of the unlabeled dataset as a regularization term to the DRF posterior: the uncertainty for the labeling of the unlabeled training examples should be low. For parameter estimation, a gradient descent optimization is used (the marginalization over the unobserved labels may only be performed approximately by resorting to a pseudolikelihood approximation). Inference for the test examples is performed by ICM, as for a normal DRF. An evaluation on a dataset of multispectral 3D MRI scans (pre- and post-contrast T_1-weighted and T_2-weighted) against manual ground truth yields a significant increase in the average Jaccard index (0.66) compared to both logistic regression (0.54) and and DRF (0.55).

Corso et al. (2007) propose an algorithm called *extended graph-shifts* to minimize the energy function of a conditional random fields model for which the number of labels is unknown beforehand. The image label structure is represented by a hierarchical graph of progressively aggregated note such that each node takes the same label as its parent node: the root nodes correspond to the different clusters. The hierarchy may then be transformed by two types of graph shift operations (greedily selecting the operation at each iteration step that maximally decreases the global energy): changing the parent of a node (thus changing the label of all nodes in the sub-graph) and creating a new subgraph from a node. At the bottom layer (corresponding to the lattice voxels), every node is assigned a unary potential corresponding to the local evidence for the different possible labels, which is computed from the probabilistic output of a Viola-Jones-like boosting cascade trained on about 3000 features (e.g. Haar-like filters, gradients, local intensity curve-fitting). Also every pair of bottom layer nodes is assigned a Potts potential term; nodes and edges at the higher hierarchy layers aggregate the potentials of their children. The label assignment is initialized stochastically, and then the hierarchical structure allows the efficient decision which move decreases the total energy maximally. The procedure is evaluated on the tasks of brain tumor and edema segmentation from multispectral MR images (high-resolution pre- and post-contrast T_1-weighted MRI, and low-resolution T_2-weighted and FLAIR MRI), and of multiple sclerosis lesion segmentation from high-resolution unispectral MRI, training and testing on six datasets each. For tumor and edema segmentation, Jaccard scores, precision and recall of 86 % / 95 % / 90 % and 88 % / 89 % / 98 % respectively are achieved, while for multiple sclerosis lesion detection, the detection rate is 81 % on the test set.

Lee et al. (2008) propose a context-sensitive classifier called pseudo-conditional random fields that yields similar or better accuracy than DRF or SVRF, while being exactly solvable and computationally much more efficient than the traditional approaches. The local potentials are products of a generalized linear model (for the feature-conditional label distribution) and a Potts model term on the labels of adjacent voxels favoring smoothness, which is modulated by a multiplicative factor measuring the similarity of the features of both voxels. Only the generalized linear model term contains adjustable parameters, so that the spatial correlations can be neglected during training; and inference in the testing phase can be performed efficiently using graph cuts. An evaluation on the task of segmenting enhancing and necrotic glioblastomas from multispectral MR images (pre- and post-contrast T_1-weighted, and T_2-weighted) against manual ground truth leads to Jaccard scores in the range of 0.82–0.93, which are significantly superior to logistic regression and comparable to SVRF (see above), while the training time is over 30 times faster than for the SVRF (38 vs. 1276 seconds on average).

Wels et al. (2008a,b) propose two similar approaches for segmenting on the one hand pediatric brain tumors, and on the other hand multiple sclerosis lesions from multispectral MR images. The modalities used are T_1-weighted MRI with and without gadolinium enhancement and T_2-weighted MRI in the first case, and T_1-weighted, T_2-weighted and FLAIR MRI in the second case. For the tumor application, the images are preprocessed by brain stripping, anisotropic diffusion filtering, and intensity standardization by dynamic histogram warping. Segmentation is viewed as MAP estimation in a Markov random field, with the single-site potentials given by the probabilistic outputs of a probabilistic boosting trees (PBT) classifier trained on local features (multispectral intensities and gradient magnitudes, and Haar-like features efficiently computed for each of the modalities from an integral image representation). For the tumor application, a contrast- and distance-attenuated Ising pair potential is imposed and the MAP inference problem is solved exactly using graph cuts. For the MS application, a simple Ising pair potential is imposed and the MAP inference problem is solved approximately using ICM. In the latter case, the final segmentation is obtained by a Laplacian 2D level set evolution initialized from the MAP solution for every slice. Typical segmentation times are 5 minutes per dataset. For the tumor application, Jaccard scores of 0.78 ± 0.17 are obtained when comparing to manual segmentation. The evaluation of the MS application leads to total detection failure of one out of six datasets, and to similarity indices of 0.68 ± 0.15 for the other five examples.

Methods based on active contours / level set segmentation

Active contour methods model the segmentation contour as level set of a continuous function (the embedding function), and minimize a energy functional for the embedding function that accounts for data fidelity (the contour should coincide with local edge cures), regularity (e.g. the curvature of the contour) and prelearned shape assumptions. Mathematically, this energy minimization leads to the task of solving a partial differential equation (PDE). While this formalism can simply incorporate a large amount of prior knowledge about the final segmentation (such as shape information), it is prone to getting stuck in local minima.

Ho et al. (2002) use level set evolution to adapt an active contour to the tumor boundaries; the region competition formalism is employed in order to deal with the fuzzy tumor boundaries. First a tumor probability map is created from two T_1-weighted scans with and without gadolinium enhancement (by fitting a Gaussian mixture model with two components to the difference image), which tends to be noisy and show also blood vessels etc. The active contour is initialized with the 0.5 level set of this probability map, and then evolves by an PDE containing a region competition term (which causes shrinkage in low probability regions and expansion in high probability regions), a smoothness term penalizing high curvature and a uniform smoothing term for increased numerical stability. The procedure is validated on multispectral MR scans (T_1-weighted with and without gadolinium enhancement and T_2-weighted) of meningioma and glioblastoma patients, yielding Jaccard's scores in the range 0.85–0.93 and Hausdorff distances of 7–13 voxels as compared to manual segmentation.

Khotanlou et al. (2006) devise a method towards tumor segmentation on unispectral images (T_1 weighting only). After brain-stripping, the histogram-based fuzzy possibilistic c-means clustering method is used to create a rough tumor segmentation (which minimizes the sum of squared differences between the local gray level and the cluster center weighted by a sum of a fuzzy membership and a typicality value and thus ensures higher robustness than ordinary c-means). Misclassification errors are removed using morphological operations. The final tumor boundaries are obtained by evolving a deformable triangulated surface, containing an internal force (controlling surface tension and curvature) and an external force (a Generalized Gradient Vector Flow field, which is the equilibrium state of diffusing the gradient vector of a Canny edge map).

Cobzas et al. (2007) combine discriminative learning with problem-specific high-dimensional features, anatomical prior information and variational (level set) segmentation for the segmentation of brain tumors. The posterior probability as estimated by a logistic regression is used in the external force term of the level set evolution PDE leading to the final segmentation. After preprocessing the data by similar steps as in Schmidt et al. (2005) (see below), a logistic regression is trained based on alignment-based features as in Schmidt et al. (2005) and texture features (multi-scale Gabor features). The final segmentation is then obtained by running the level set evolution and removing small surface pieces as a post-processing step. Evaluation on T_1-weighted and T_2-weighted datasets yields average overlap fractions, Hausdorff distances and mean distances of 60±14%, 8.1±1.8 mm and 1.74±0.66 mm, which is considerable better than when using a Gaussian classifier.

Methods based on fuzzy clustering

Fuzzy clustering techniques work by grouping the set of features extracted from all voxels in the training images into several groups (or clusters), which are given different semantic interpretations: for example, some clusters may be identified with the different tissue classes (GM, WM, CSF) in the brain, while others may be identified with pathologies (tumor, edema) or extracerebral regions (bones, skin or air). For brain lesion detection applications, usually a fuzzy clustering approach is followed rather than a hard clustering: i.e. every voxel may be assigned to every cluster, with soft assignment weights that have to sum to 1. Most applications are based on the fuzzy c-means (FCM) technique that iteratively estimates the cluster centers and the soft assignments in an interleaved fashion.

Fletcher-Heath et al. (2001) combine FCM clustering with subsequent image processing and labeling operations based on explicit knowledge for segmenting non-enhancing brain tumors from multispectral MR images (T_1-, T_2- and PD-weighted). The input images are fuzzily oversegmented into ten clusters by FCM, and clusters corresponding to extracranial tissues, white matter and gray matter are identified and removed (but their locations are remembered in order to guide the subsequent steps). CSF, necrosis (if present) and tumors are then separated by several knowledge-guided image processing steps: If the T_1 histogram has a bimodal shape, the low-intensity peak corresponds to a necrosis which is then removed. The ventricles are identified by extracting a central shape bordered by GM and WM (left-right symmetry information is used if the tumor borders the ventricles). Isolated CSF pixels are then removed by morphological operations (this assumes a minimum spatial extent of the tumor). Finally, the most compact region(s) is/are selected as tumor(s), i.e. the number of tumors must also be known beforehand. The validation yields correct classification rates range from 53 % to 91 % per volume.

Segmentation in 4D images

While most of the other approaches described in this chapter only aim to segment an image volume acquired at a single time point, tumor progression monitoring studies require to track e.g. the volume of a tumor over time, so that the response to a therapy can be assessed. The following methods try to improve upon the single-volume segmentations by using the information from the different time points simultaneously:

Solomon et al. (2004) employ 4D segmentation to track the tumor volume over time and to assess changes in tumor size objectively; it is assumed that the additional temporal dimension may also lead to improved segmentation at the single time points. The basis for segmentation is a Gaussian mixture model fitted with an EM algorithm as in Leemput et al. (1999b), which is augmented with a temporal hidden Markov model (EM-HMM segmentation). Unispectral, nearly isotropic 3D MRI scans acquired at three different time points are registered and de-skulled. First a rough segmentation is obtained by k-means clustering, which is used as initialization to the EM estimation of the Gaussian model parameters (for this purpose the volumes at all different time points are used). Given the class-conditional observation models, the class assignment labels are estimated: It is assumed that every voxel at every time point is characterized by a status label (lesion vs. not lesion) which evolves by a Markov process (independently from all other voxels), and that the observed intensity only depends on the current status. Furthermore one assumes that the transition probability drops exponentially with the distance from the current tissue boundary, and the exponential coefficient is estimated from the results of the non-temporal EM segmentation at different time points. The posterior of the current status given all evidence acquired up to the current time point is then computed and used for fuzzy segmentation; it is also possible to reestimate the class assignments at earlier time points given the new information (smoothing). In a first experimental with three different time points, a correlation of 0.89 with the manual segmentation and a mean Dice similarity coefficient[13] of 0.71 are found. In an extension (Solomon et al., 2006), an MRF prior is added to the intensity distribution learned by the EM algorithm and the transition matrix is refined to accommodate more than two tissue classes (parenchyma, tumor, CSF and blood vessels), so that the Gaussian model assumptions become more accurate. Evaluation on simulated data shows that the MRF and the HMM priors and the smoothing step all lead to improvements as measured by sensitivity and the Jaccard index. Furthermore, evaluation on real data from three different time points yields segmentation results that are as good as comparable state-of-the-art segmentation techniques, and which have the same sensitivity as a manual segmentation, if a slightly smaller Jaccard similarity compared to the ground truth. The use of a multi-class tissue class leads to a slight decrease in sensitivity, but also to an increased similarity index (owing to less false positive detections).

Interactive segmentation methods

The segmentation is typically simplified if no full automation is required, and the clinical user has the opportunity to either initialize the segmentation by manual seed placement, or to refine the final segmentation.

The first approach is followed e.g. by Warfield et al. (2000) and Kaus et al. (1999, 2001), where the authors propose an adaptive, template-moderated spatially varying

[13] The Dice coefficient of two segmentations is the ratio between the overlap volume and the average volume of the single segmentations.

classification (ATM SVC) algorithm for multiple segmentation problems of both healthy and pathological structures, and apply it to brain tumor segmentation, amongst other tasks. The idea is to combine two segmentation strategies: classification based on local features (which does not account for anatomical information) and nonlinear registration to an anatomical template (which takes the local features only partially into account, and has only limited accuracy for pathological or highly variable organs). A unispectral three-dimensional MR image is initially registered to a template atlas by a nonlinear registration algorithm. The user has to provide three or four example labels for each class of interest (e.g. brain (WM & GM), CSF, tumor, skin, background), which typically requires 5 minutes of user interaction. The image is then segmented by a kNN classification (section 2.2), using as features both the voxel intensity and the distance to relevant brain structures, e.g. the ventricles. Then the registration is refined by matching the atlas with the segmented image, and the procedure is iterated. While the initial atlas only contains normal brain structures and no tumor, a tumor segment is added after the first iteration step from the initial segmentation. Compared to the majority vote of four experts, the tumor can be segmented with a voxelwise accuracy or 99.7%.

Level-set and active contour segmentations (see above) are also well-suited for a user-defined initialization, which may alleviate its problems with running into local energy minima. For instance, Jiang et al. (2004) provide a brain tumor segmentation method as part of a telemedicine CAD system. They use a level set segmentation starting from a coarse user-provided manual delineation, with standard terms for the external and internal force (local curvature and gradient of a simple edge map).

Droske et al. (2005) use level set evolution with an expanding force term to segment brain tumors on T_1-weighted gadolinium-enhanced images, starting from a user-provided initialization contour inside the tumor. The expansion speed is computed based on an edge map (expansion is slowed down if the edge intensity lies outside of a prescribed interval, which is estimated from the user-defined seed points). Since no automated convergence diagnostics are included, the user also has to specify the arrival time for the final segmentation. It is also possible to correct or add intermediate segmentations to ensure convergence to the correct final state.

Besides the dependence on the initial contour, level-set segmentation methods also typically depend on a number of free parameters whose optimal choice is not always clear beforehand, especially to clinical users. Lefohn et al. (2003) and Cates et al. (2004) employ fast level-set deformation solvers to interactively tune these free parameters of the level set partial differential equation (e.g. the trade-off between curvature term and data term, or the free parameters of the data term). A sparse approximation of the PDE is used in which only voxels near the isosurface are taken into account, and a further speed-up of 10–15 is achieved by implementing the solver on a GPU. Compared with the STAPLE-generated ground truth from four expert segmentations, even non-radiologist raters achieved a mean precision of 94 % (experts: 83 %) and an average correct volume fraction of 99.78% ± 0.13%, needing a total time of 6 ± 3 minutes per dataset, whereas the typical time for an unassisted three-dimensional manual segmentation is rather 3–5 hours.

The second approach, i.e. enabling the users to perform final corrections on the segmentation, is followed e.g. by Letteboer et al. (2004): A multiscale watershed segmentation of the tumor images is created as a preprocessing step (i.e. a scale-space representation is created by convolving with differentials of Gaussians at different scales, watershed segmentations are performed at the various scales and the catchment basins are linked across the different scales to ensure that each catchment basin at a fine scale is contained in exactly one catchment basin at every coarser scale). In a graphical user interface, the user may first create a rough segmentation by selecting segments at a coarse scale, and then interactively refine it by adding or deselecting subsegments at the finer scales: this leads to an increased intraobserver and interobserver similarity, and reduces the time needed for manually delineating the tumor is decreased from 22 minutes on average (10–40 minutes) to 7 minutes on average (1–15 minutes).

Cates et al. (2005) explore the opportunities of the ITK segmentation library for interactive segmentations of brain tumors and different anatomical structures (e.g. optic nerve, eyeball, lateral rectus muscle). Datasets are preprocessed by anisotropic diffusion, and a watershed over-segmentation is computed based on a lower-thresholded gradient map. A segment hierarchy is then constructed by successively merging watershed basins based on their watershed depth. The users then create the final segmentation by manual selection of regions in this hierarchy graph. Compared to the STAPLE consensus of several expert segmentations, this procedure yields a mean correct classification rate of 99.76 ± 0.14%. Giving the clinical user the opportunity for manual corrections at the final stage may also increase the acceptance of clinical radiologists for computer-assisted segmentation systems, and increase the safety of the patients during subsequent interventions that are planned on the basis of these segmentations.

Active learning approaches

Creating manual annotations for training a classifier is a time-consuming and tedious task, especially as it has to be performed by clinical radiologists, whose time is typically scarce. Active learning approaches can speed up this process by proposing the images (or image parts) for annotation which are expected to give the highest benefit to classifier accuracy. Farhangfar et al. (2009) propose such an active learning approach for the training of a DRF classifier, and apply it to the tasks of sky segmentation in natural images and brain tumor segmentation in MR images. Their approach is similar to the semi-supervised DRF model presented in Lee et al. (2006) (see above), but the regularization term consists of the expected conditional entropy of each queried new image to be labeled rather than the expected conditional entropy of all unlabeled images together. A pseudolikelihood approximation is employed to make the parameter estimation for this regularized likelihood tractable (for this approximation it is necessary to compute the MAP label estimate for the unlabeled image by ICM). There are two possible strategies how to request the next image to be labeled: firstly, select the image with the highest expected conditional entropy given the current estimate for the posterior distribution of the labels (which is approximated as sum over the pixel-wise entropies); this strategy is applied in all steps but the first where the posterior distribution is not yet initialized. Secondly, select the instance providing the maximum information about the labels of the other unlabeled instances (which can be computed from the solution of the regularized posterior); this strategy is only used in the initial step as it is computationally more expensive. Besides sky segmentation, this procedure is evaluated for the task of brain tumor segmentation from multispectral MR scans (pre- and post-contrast T_1-weighted and T_2-weighted). Four features are used for each pixel: the intensity in the T_2-weighted image, the difference between the post-contrast and pre-contrast T_1-weighted intensities, and the differences of these two gray values to the gray value of the contralateral voxel. Actively selecting two training images yielded (insignificantly) better F-measures than training on all 71 examples.

Methods exploiting left-right symmetry of the brain

Besides generic segmentation methods for medical imagery, there are also techniques that depend heavily on the specifical properties of brain imagery, namely the approximate left-right symmetry of the brain: this is e.g. exploited by the alignment-based features of Schmidt et al. (2005), cf. section 3.3.1. Another approach in this direction was proposed by Ray et al. (2008): The authors aim to quickly place a boundary box around a tumor in unispectral MR images, e.g. for retrieval purposes. For this they use asymmetry-based features specific for brain tumor segmentation, in order to profit from the knowledge that tumors tend to disturb the bilateral symmetry of the brain. A (healthy) template image is matched approximately to the input image, and for each coronal plane the Bhattacharyya distance[14] between the intensity histograms of the two images before this plane and after this plane is computed. The front and back face of the bounding box then delineate the region where this score

[14] The Bhattacharyya distance between two histograms is the sum of the geometric means of the entries in each histogram bin.

decreases from front to back, as the intensities of the two images tend to be uncorrelated in this area. Similarly, the left and right face are detected. Dice coefficients with bounding boxes drawn by expert radiologists lie in the range of 0.7–0.9.

Other approaches There are also multiple other approaches for brain lesion segmentation that cannot be discussed here due to space constraints. Amongst others, they comprise region growing (Broderick et al., 1996), rule-based techniques (Raya, 1990), semi-supervised classification (Song et al., 2006, 2009), template matching (Warfield et al., 1995; Hojjatoleslami et al., 1998), mathematical morphology (Gibbs et al., 1996; He & Narayana, 2002), fuzzy connectedness estimation (Udupa et al., 1997; Moonis et al., 2002), vector quantization (Karayiannis & Pai, 1999), pyramid segmentation (Pachai et al., 1998), eigenimages (Soltanian-Zadeh et al., 1998), texture-based classification (Kovalev et al., 2001; Iftekharuddin et al., 2005), Bayesian classification (Harmouche et al., 2006) and fuzzy logic (Zhu et al., 2005; Dou et al., 2007).

3.3.2. Learning from unreliable manual annotations

In the common formulation of supervised learning methods (see section 2.2), a mapping from training examples $x \in \mathcal{X}$ to targets $y \in \mathcal{Y}$ is learned from training examples (x_i, y_i). Typically, $\mathcal{X} \subseteq \mathbb{R}^p$ and $\mathcal{Y}$ is either continuous ($\mathcal{Y} \subseteq \mathbb{R}$, regression setting) or discrete ($\mathcal{Y} = \{1, \ldots, L\}$, classification setting). Often the targets y come from human judgment, and one assumes that this judgment is reliable, so that the training examples (x_i, y_i) can be viewed as samples from the true data distribution during the subsequent classifier training and testing. However, in many cases this assumption is overly optimistic, since the human labelers may be unreliable and assign some wrong labels. This is particularly the case for classification based on noisy or ambiguous image information, e.g. for the tasks of finding volcanoes in small aperture radar imagery of the Venus (Smyth et al., 1995) or distinguishing between genuine (Duchenne) and insincere (non-Duchenne) smiles (Whitehill et al., 2009). The most extreme phenomenon are adversarial labelers which deliberately cast wrong labels in order to degrade the classifier performance: they pose a severe challenge for e.g. collaborative e-mail spam filtering systems (Attenberg et al., 2009). Applications in medical image analysis include the segmentation of healthy brain images into the three main compartments of GM, WM and CSF (Warfield et al., 2004) or the classification of lung nodules detected in CT images into malignant or benign examples (Raykar et al., 2010). In the following, we will deal with the task of segmenting brain tumors from multimodal medical images. Fig. 3.2 gives an impression of the unreliability of human annotators for this task.

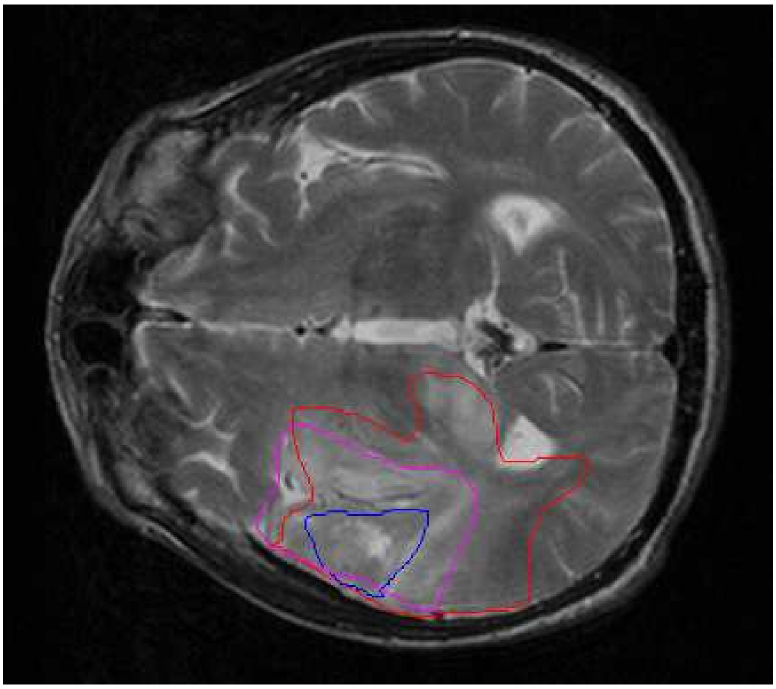


Figure 3.2. – Exemplary segmentations of a real-world brain tumor image by a single expert, based on different imaging modalities. In the background, an axial FLAIR section of an astrocytoma patient is displayed. The colored lines are the contours of manual tumor segmentations that were drawn by a senior radiologist on three different MR scans of the same slice: namely a T_2-weighted scan (magenta), a gadolinium-enhanced T_1-weighted scan (blue) and this FLAIR scan (red). The other two scans had been affinely registered to the FLAIR scan beforehand. Note the volume variability of ca. 400 % between the different modalities. This chapter deals with the question what single segmentation should be reported to summarize this information.

In cases in which only a single label and no additional information is provided about every training example, one can obviously not do better than treating this label as the truth. However, if several labels from multiple annotators are available, one can fuse these (possibly conflicting) votes to a consensus label, which should hopefully more reliable than every single vote, or even estimate the probabilities for the different possible values of the label. It can be expected that the multiple labelers may differ in their reliability: some may be experts for this tasks, some novices, some may be meticulous, some careless, and some may even be malicious as in the adversarial scenario. Ideally the fusion routine should identify the reliable labelers and assign their votes a higher weight for the final decision. Or, if objective feature information about the training example is available (that characterizes each example sufficiently well), one can check whether a rater consistently gives the same labels to examples having similar features, which may help one to decide whether he or she assigns the labels rather randomly or based on the visible image information. In the following, the previously proposed models for fusing unreliable manual annotations are reformulated in the language of probabilistic graphical models (more precisely Bayesian networks), which has not been done before (Fig. 3.3). This makes the similarities and differences between the different approaches clearer and allows the use of generic inference techniques.

In the STAPLE model proposed by Warfield et al. (2004, Fig. 3.3(a)), the discrete observations $s_{nr} \in \{0, 1\}$ are noisy views on the true scores $t_n \in \{0, 1\}$, with $n \in \{1, \ldots, N\}$ indexing the image pixels and $r \in \{1, \ldots, R\}$ indexing the raters. The r-th rater is characterized by the sensitivity γ_r and the specificity $1 - \delta_r$, and the observation model is $s_{nr} \sim t_n \mathrm{Ber}(\gamma_r) + (1 - t_n)\mathrm{Ber}(\delta_r)$, with "Ber" denoting a Bernoulli distribution. A Bernoulli prior is given for the true class: $t_n \sim \mathrm{Ber}(p)$. While the original formulation fixes $p = 0.5$ and uses uniform priors for γ_r and δ_r, the priors were modified in order to fulfil the conjugacy requirements for the chosen variational inference techniques: hence Beta priors are imposed on $\gamma_r \sim \mathrm{Beta}(a_{\mathrm{se}}, b_{\mathrm{se}})$, $\delta_r \sim \mathrm{Beta}(b_{\mathrm{sp}}, a_{\mathrm{sp}})$ and $p \sim \mathrm{Beta}(a_{\mathrm{p}}, b_{\mathrm{p}})$. A similar Beta prior was independently introduced by Commowick & Warfield (2010) in order to use prior knowledge about the relative quality of different raters: While in the following experiments the same values of a_{se}, b_{se}, a_{sp}, b_{sp} were used for all raters, it would also be possible to give higher a parameters and lower b parameters to raters who are supposed to be more reliable.[15] The prior on p is introduced in order to learn the share of tumor tissue among all voxels from the data.

The model by Raykar et al. (2009, Fig. 3.3(b)) is the same as (Warfield et al., 2004) except for the prior on t_n: here the authors assume that a feature vector φ_n is observed at the n-th pixel and that $t_n \sim \mathrm{Ber}\big(\{1 + \exp(-w^\top \varphi_n)\}^{-1}\big)$ follows a logistic regression model. A Gaussian prior is imposed on $w \sim \mathcal{N}(0, \lambda_{\mathrm{w}}^{-1} I)$. In contrast to (Warfield et al., 2004), they obtain a classifier that can be used to predict the tumor probability on unseen test images, for which one has access to the features φ_n but not to the annotations s_{nr}. One may hypothesize that the additional information of the features φ_n can help to resolve conflicts: in a two-rater scenario, one can decide that the rater has less noise who labels pixels with similar φ_n more consistently. In the modified graphical model formulation, a gamma prior for the weight precision is added: $\lambda_{\mathrm{w}} \sim \mathrm{Gam}(a_{\mathrm{w}}, b_{\mathrm{w}})$. Note that this model can be regarded as a direct multi-rater generalization of logistic regression (Hastie et al., 2009, Ch. 4).

[15] Note that the expected mean of a $\mathrm{Beta}(a, b)$ distribution is $a/(a + b)$.

Whitehill et al. (2009, Fig. 3.3(c)) propose a model in which the misclassification probability depends on both the pixel and the rater: $s_{nr} \sim \mathrm{Ber}\big(\{1+\exp(-t_n \alpha_r \epsilon_n)\}^{-1}\big)$ with the rater accuracy $\alpha_r \sim \mathcal{N}(\mu_\alpha, \lambda_\alpha^{-1})$ and the pixel difficulty ϵ_n with $\log(\epsilon_n) \sim \mathcal{N}(\mu_\epsilon, \lambda_\epsilon^{-1})$ (this parameterization is chosen to constrain ϵ_n to be positive).

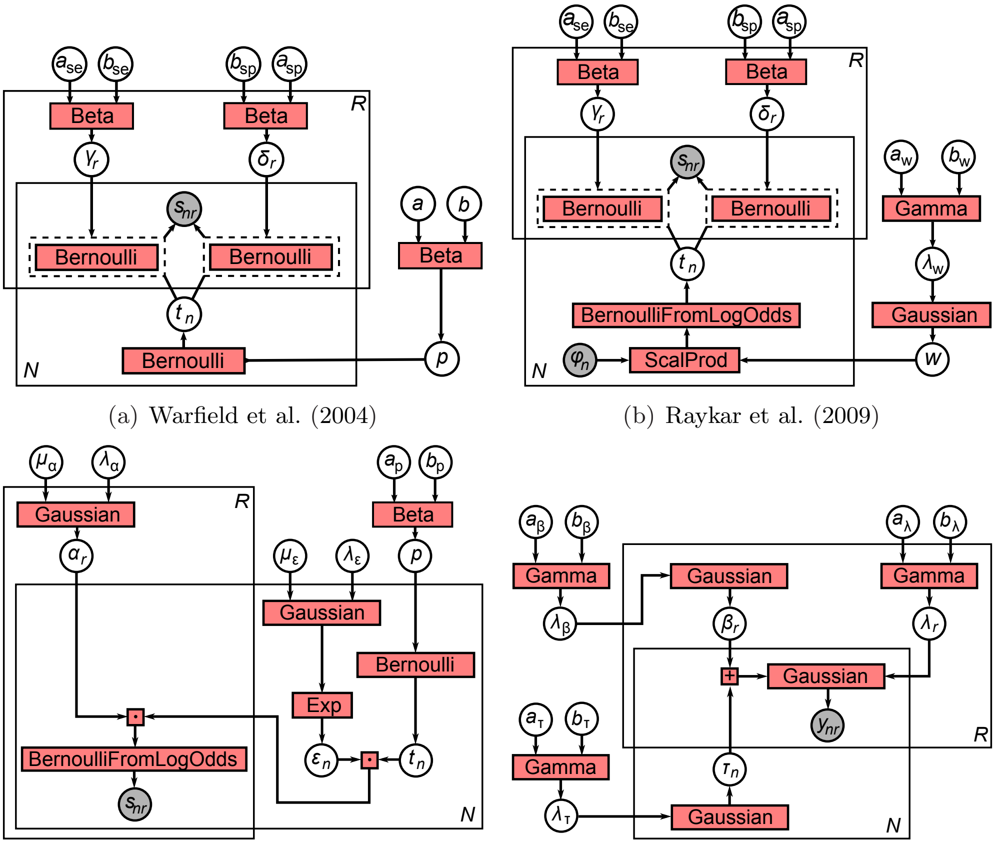
In the continuous variant of STAPLE by Warfield et al. (2008, Fig. 3.3(d)), the observations y_{nr} are continuous views on a continuous latent score τ_n. It is assumed that the noisy y_{nr} and the true τ_n give information not only whether a given voxel is tumor or not, but also how far it is away from the tumor boundary: Commonly y_{nr} is defined as the signed Euclidean distance function[16] of the r-th rater, and τ_n hence corresponds to the distance transform of the true tumor segmentation, so that the tumor contours are the zero-level set of τ. The r-th rater can be characterized by a bias β_r and a noise precision λ_r: $y_{nr} \sim \mathcal{N}(\tau_n + \beta_r, \lambda_r^{-1})$, with a Gaussian prior on the true scores: $\tau_n \sim \mathcal{N}(0, \lambda_\tau^{-1})$. In the modified graphical model formulation, Gaussian priors on the biases are added, i.e. $\beta_r \sim \mathcal{N}(0, \lambda_\beta^{-1})$. For the precisions of the Gaussians, gamma priors are used: $\lambda_\tau \sim \mathrm{Gam}(a_\tau, b_\tau)$, $\lambda_\beta \sim \mathrm{Gam}(a_\beta, b_\beta)$ and $\lambda_r \sim \mathrm{Gam}(a_\lambda, b_\lambda)$. Note that when thresholding the continuous scores, the tumor boundary may shift because of the noise, but misclassifications far away from the boundary are unlikely: this is an alternative to (Whitehill et al., 2009) for achieving a non-uniform noise model.

3.4. Modelling and implementation

3.4.1. Novel hybrid models

In addition to the previously proposed latent-class and latent-score models, four novel hybrid models are introduced, which incorporate all aspects of the previous proposals simultaneously: while they provide a classifier as in (Raykar et al., 2009), they do not assume misclassifications to occur everywhere equally likely. In the simplest variant (hybrid model 1, Fig. 3.4(a)), the model from (Warfield et al., 2008) is modified by a linear regression model for $\tau_n \sim \mathcal{N}(w^\top \varphi_n, \lambda_\tau^{-1})$ with $w \sim \mathcal{N}(0, \lambda_{\mathrm{w}}^{-1})$. Note that this model predicts a (noisy) linear relationship between the distance transform values y_{nr} and the features φ_n, while experimentally the local image appearance saturates in the interior of the tumor or the healthy tissue. To alleviate this concern (hybrid model 2, Fig. 3.4(b)), one can interpret y_{nr} as an unob-

[16] The unsigned Euclidean distance transform of a binary mask I is defined as 0 inside of I, and as the Euclidean distance to the closest point of I outside of I. The signed Euclidean distance transform is the difference of the unsigned distance transforms of I and its complement $\bar{I}$. Using a modification of Dijkstra's all-pairs shortest path algorithm, these measures can be computed for an entire binary image in a time linear in the number of pixels (Fabbri et al., 2008).



(a) Warfield et al. (2004)

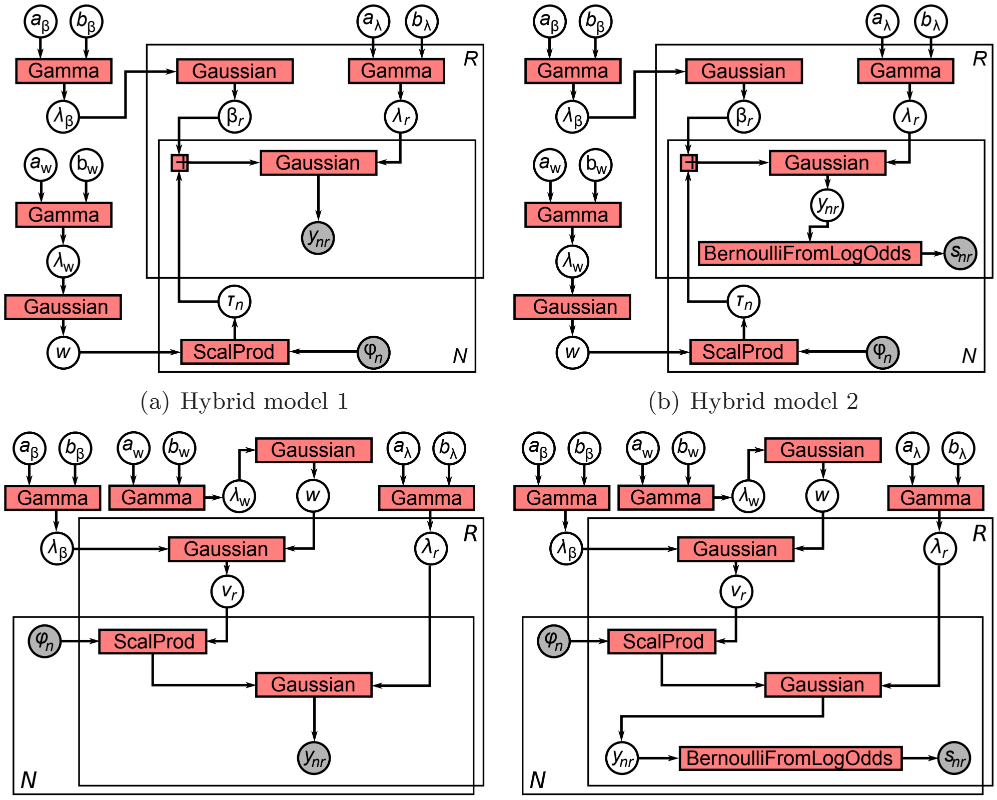
(b) Raykar et al. (2009)

(c) Whitehill et al. (2009)

(d) Warfield et al. (2008)

Figure 3.3. – Graphical model representations of the previously proposed fusion algorithms, partially with new priors added. Red boxes correspond to factors, circles correspond to observed (gray) and unobserved (white) variables. Some factors are deterministic: "Exp" refers to an exponential function, "ScalProd" to a scalar product, and the + and $\cdot$ factors to addition and multiplication. The "BernoulliFromLogOdds" factor means that the output y is a binary variable sampled from a Bernoulli distribution with parameter $(1+e^{-x})^{-1}$, where x is the input of the factor. Solid black rectangles are plates indicating an indexed array of variables (Buntine, 1994). The dashed rectangles are *"gates"* denoting a mixture model with a hidden selector variable (Minka & Winn, 2009).

served malignancy score, which influences the (observed) binary segmentations s_{nr} via $s_{nr} \sim \text{Ber}\big(\{1 + \exp(-y_{nr})\}^{-1}\big)$. This is a simplified version of the procedure presented in Rogers et al. (2010), with a linear regression model for the latent score instead of a Gaussian process regression. Alternatively one can model the raters as using a biased weight vector rather than having a biased view on an ideal score, i.e. $y_{rn} \sim \mathcal{N}(v_r^\top \varphi_n, \lambda_r^{-1})$ with $v_r \sim \mathcal{N}(w, \lambda_\beta^{-1} I)$. Again the score y_{nr} may be observed directly as a distance transform (hybrid model 3, Fig. 3.4(c)) or indirectly via s_{nr} (hybrid model 4, Fig. 3.4(d)).



(a) Hybrid model 1

(b) Hybrid model 2

(c) Hybrid model 3

(d) Hybrid model 4

Figure 3.4. – Newly proposed hybrid models: for the explanation of the symbols see the caption of Fig. 3.3.

3.4.2. Inference and implementation

For the graphical models considered here, exact inference by the junction tree algorithm is infeasible especially for the models that make use of the objective image information: If d is the number of features in the vector φ_n for the models that make use of the features φ, and $d = 1$ for the other models, the treewidth of the graphical models in Figs. 3.3 and 3.4 is given by $2R + d$. In absence of efficient exact algorithms for treewidth computation, this was found by computing experimental upper and lower bounds by the approximation techniques presented in (Bodlaender & Koster, 2010a) and (Bodlaender & Koster, 2010b). The tightest upper and lower bounds were found to coincide, giving the exact treewidth value.[17] However, one can perform approximate inference using e.g. variational message passing (Winn & Bishop, 2005): the true posterior for the latent variables is approximated by the closest factorizing distribution (as measured by the Kullback-Leibler distance), for which inference is tractable. As a prerequisite, all priors must be conjugate; this holds for all models discussed above except (Whitehill et al., 2009). Here one cannot apply the generic variational message passing scheme to this model, so that the results from the EM inference algorithm provided by the authors are reported instead.

The INFER.NET 2.3 Beta implementation for variational message passing (Minka et al., 2009) was employed to perform inference on the algorithms by Warfield et al. (2004), Warfield et al. (2008), Raykar et al. (2009) and the four hybrid models. The default value of 50 iteration steps was found to be sufficient for convergence, since doubling the number of steps led to virtually indistinguishable results. For the algorithm by Whitehill et al. (2009), the GLAD 1.0.2 reference implementation was used.[18] Alternative choices for the generic inference method would have been expectation propagation (Minka, 2001) and Gibbs sampling (Gelfand & Smith, 1990). We experimentally found out that expectation propagation had considerably higher memory requirements than variational message passing for our problems, which prevented its use for our problems on the available hardware. Gibbs sampling was not employed since some of the factors incorporated in our models (namely gates and factor arrays) are not supported by the current INFER.NET implementation. Note that these are purely practical reasons: in theory, it would have been possible to use also these two alternatives.

The results of the graphical models were also compared against three simple baseline procedures: majority voting, training a logistic regression classifier from the segmentations of every single rater and averaging the classifier predictions (ALR), and training a logistic regression classifier on soft labels (LRS): if S out of R raters voted for tumor in a certain pixel, it was assigned the soft label $S/R \in [0, 1]$.

3.5. Experiments

Two experiments were performed in order to study the influences of labeler quality and imaging modality separately. In the first experiment, multiple human annotations of varying quality based on one single imaging modality were collected and fused: for this task, simulated brain tumor measurements were used, for which ground truth information about the true tumor extent was available, so that the

[17] The LibTW library was used for these studies: `http://www.treewidth.com/docs/libtw.zip`

[18] `http://mplab.ucsd.edu/~jake/OptimalLabelingRelease1.0.2.tar.gz`

results could be evaluated quantitatively. In the second experiment, multiple human annotations based on real-world image data were collected and fused, which were all of high quality, but had been derived from different imaging modalities showing similar physical changes caused by glioma infiltration with different sensitivity.

3.5.1. Experiments on simulated brain tumor measurements

Tumor simulations Simulated brain tumor MR images were generated by means of the TumorSim 1.0 software (Prastawa et al., 2009).[19] The advantage of these simulations was the existence of ground truth about the true tumor extent (in form of probability maps for the distribution of white matter, gray matter, cerebrospinal fluid, tumor and edema). The final task of the classifiers was to discriminate between "pathological tissue" (tumor and edema) and "healthy tissue" (the rest). Nine image volumes were used: three for each tumor class that can be simulated by this software (ring-enhancing, uniformly enhancing and non-enhancing, see Fig. 3.5). Each volumetric images contained $256 \times 256 \times 181$ voxels, and the three different imaging modalities (T_1-weighted with and without gadolinium enhancement and T_2-weighted) were considered perfectly registered with respect to each other. The feature vectors φ_i consisted of four features for each modality: gray value, gradient magnitude and the responses of a minimum and maximum filter within a 3×3 neighborhood. A row with the constant value 1 was added to learn a constant offset for the linear or logistic models (since there was no reason to assume that features values at the tumor boundary are orthogonal to the final weight vector).

Justification of linear classification term Linear discrimination models like the model by Raykar et al. (2009) and the hybrid models are appropriate if the decision boundaries in the selected feature space can be regarded as linear, i.e. if a linear classifier can distinguish between pathological (tumor or edema) and healthy (GM / WM /CSF) features just as good as a state-of-the-art nonlinear classifier. In order to test this, a preparatory experiment was conducted, in which the ground-truth values for the tissue probabilities were assumed as known (i.e. no multirater setting). The generalization errors of both a linear classifier (logistic regression) and a nonlinear classifier (random forest, see section 2.2) were estimated for the task of distinguishing between characteristic pathological and characteristic healthy examples. "Characteristic" meant that the ground-truth probability for the respective class exceeded 0.98. For the estimation of variances, a twelve-fold cross-validation scheme was used, so that each of the twelve simulated volumes was selected as test dataset in some fold, and the remaining eleven simulated volumes were used for training.

[19] `http://www.sci.utah.edu/releases/tumorsim_v1.0/TumorSim_1.0_linux64.zip`

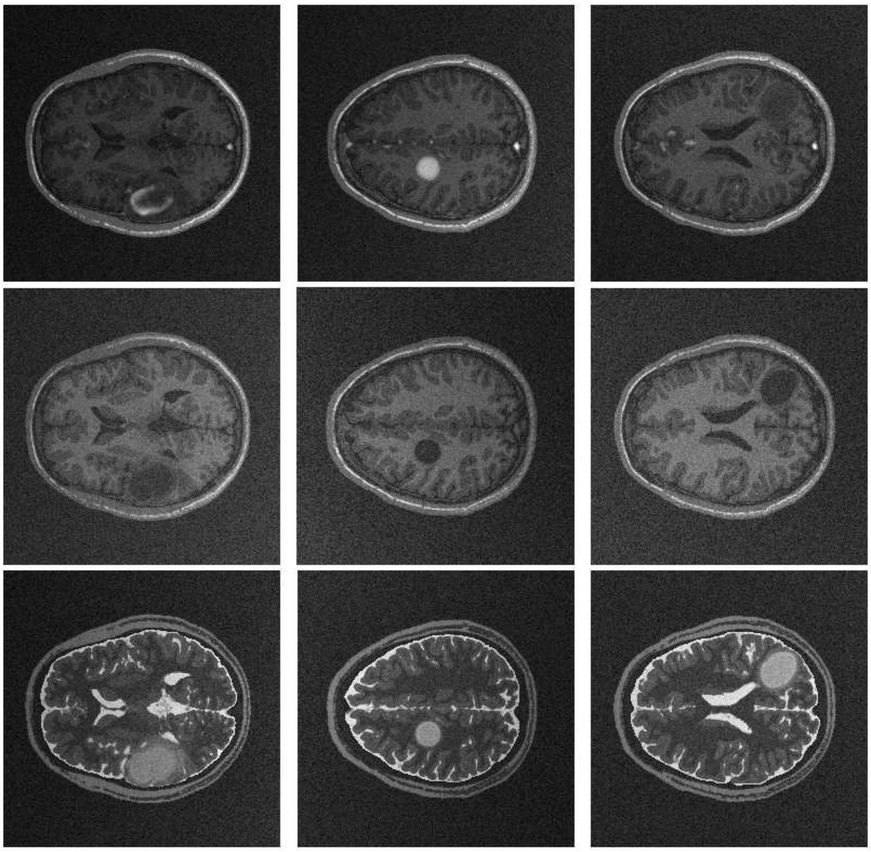


Figure 3.5. – Exemplary slices of the three simulated tumor classes: every column shows an exemplary simulated brain tumor image slice in the three weightings which can be produced by the TumorSim 1.0 software, namely T_1-weighting with gadolinium enhancement (top), T_1-weighting without gadolinium enhancement (middle) and T_2-weighting (bottom). The left column shows an example of a ring-enhancing tumor, the middle column of a uniformly enhancing tumor, and the right column of a non-enhancing tumor: this corresponds to decreasing tumor grade from left to right. Note that the appearance of the three classes only differs in the Gd-enhanced image; under T_1-weighting all appear as hypointensities, and under T_2-weighting as hyperintensities.

Logistic regression yielded a sensitivity of $97.8 \pm 4.8\%$ and a specificity of $97.2 \pm 1.0\%$ (average F-measure: 97.5%), while the random forest classifier yielded a sensitivity of $89 \pm 16\%$ and a specificity of $99.4 \pm 0.6\%$ (average F-measure: 93.9%). Since a high sensitivity is crucial for tumor detection, this means that linear classifiers (and especially variants of logistic regression) are superior to nonlinear methods for this classification task.[20]

Justification of feature set choice In an extension of the preliminary experiments described in the previous paragraph, several combinations of image features were tested in order to find a feature set that is sufficiently discriminative between healthy and pathological tissue in the ideal case that reliable labels are given. Table 3.1 shows the different features that were tried, while Fig. 3.6 shows the resulting sensitivities and specificities. The final choice fell on four features per image weighting (gray value, gradient, local minimum and local maximum): using fewer features would have impaired the classification specificity (Fig. 3.6(b)), while using more features would have given no additional improvements and would have increased the memory requirements.

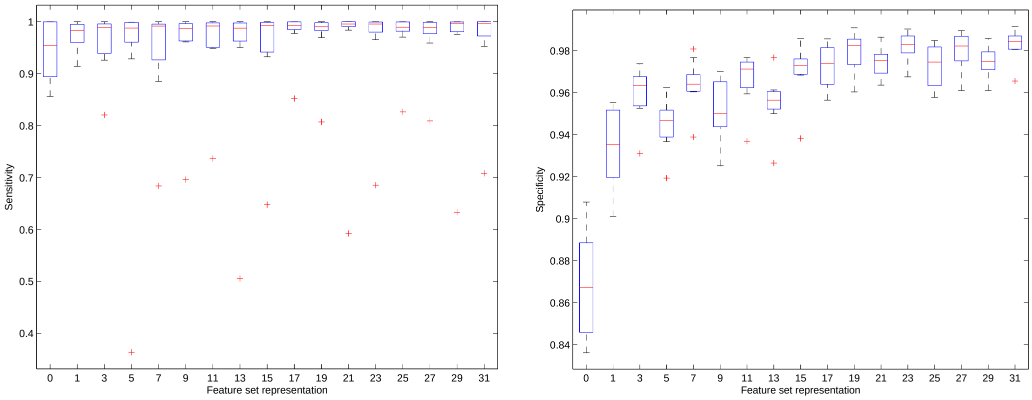
Feature	Length	Binary flag
Gradient magnitude	1	1
2D Hessian eigenvalues	2	2
2D structure tensor eigenvalues	2	4
Local entropy (3×3)	1	8
Local maximum & minimum (3×3)	2	16

Table 3.1. – Image features that were tested in order to find an optimal feature set for linear classification. Additionally the image gray values were part of each tentative feature set. While some features are scalars, others comprise several values: this is encoded in the column "Length". The final column gives the binary flag by which the features are encoded in Figs. 3.6(a) and 3.6(b). The mask size used for the computation of the local entropy, maximum and minimum is indicated in parentheses.

Label acquisition The image volumes were segmented manually based on hypointensities in the T_1-weighted images, using the manual segmentation functionality of the ITK-SNAP 2.0 software.[21] In order to control the rater precision, time limits of 60, 90, 120 and 180 seconds for labeling a 3D volume were imposed and five segmentations were created for each limit: one can expect the segmentations to be precise for generous time limits, and to be noisy when the rater had to label very

[20] Obviously linear decision boundaries can also be learned using a nonlinear classifier. However, for a limited amount of training data (i.e. for all practical purposes), linear classifiers will give superior classification accuracy if the decision boundary is (approximately) linear, as they are less prone to overfitting to noise in the data. As a rule, restrictive classifiers that make assumptions about the data are superior to more general classifiers if the assumptions actually hold in practice.

[21] http://www.itksnap.org/pmwiki/pmwiki.php?n=Main.Downloads



(a) Sensitivities

(b) Specificities

Figure 3.6. – Sensitivities and specificities for logistic regression on simulated brain tumor imagery using different feature subsets, when trained with randomly sampled characteristic examples for healthy and pathological tissues. Ground truth labels are provided to the classifier for this purpose. Each selected feature was computed for all three modalities, i.e. T_1-weighting with and without gadolinium-enhancement, and T_2 weighting. Furthermore, the image gray values were part of each feature set (and the only elements of the set with the label "0"). A cross-validation scheme is used to estimate the spread of the values that is visualized by the box plots (see the text for further details). The x label numbers encode the feature set composition (bit vector representation, see Table 3.1): e.g. $11 = 1 + 2 + 8$ corresponds to the set containing gradient, Hessian eigenvalues and entropy filter responses.

fast. The set of raters was the same for the different time constraints, and the other experimental conditions were also kept constant across the different time constraints. This was statistically validated: the area under curve value of the receiver operating characteristic of the ground-truth probability maps compared against the manual segmentations showed a significant positive trend with respect to the available time ($p = 1.8 \times 10^{-4}$, F test for a linear regression model). Since tight time constraints are typical for the clinical routine, this setting was considered as realistic, although it does not account for rater bias.

The slices with the highest amount of tumor lesion were extracted and partitioned into nine data subsets in order to estimate the variance of segmentation quality measures, with each subset containing one third of the slices extracted from three different tumor datasets (one for each enhancement type). Due to memory restriction, the pixels labeled as "background" by all raters were randomly subsampled to reduce the sample size. A cross-validation scheme was used to test the linear and log-linear classifiers (all except those by Warfield et al. (2004), Warfield et al. (2008) and Whitehill et al. (2009)) on features φ_n not seen during the training process: the training and testing process was repeated nine times, and each of the data subsets was chosen in turn as the training dataset (and two different subsets as the test data).

Choice of prior parameters The following default values for the prior parameters were used: $a_{\mathrm{Se}} = 10$, $b_{\mathrm{Se}} = 2$, $a_{\mathrm{Sp}} = 10$, $b_{\mathrm{Sp}} = 2$, $a_{\mathrm{w}} = 2$, $b_{\mathrm{w}} = 1$, $a_{\mathrm{p}} = 2$, $b_{\mathrm{p}} = 2$, $a_{\tau} = 2$, $b_{\tau} = 1$, $a_{\beta} = 2$, $b_{\beta} = 1$, $a_{\lambda} = 2$, $b_{\lambda} = 1$. Additional experiments verified that inference results changed only negligibly when these hyperparameters were varied over the range of a decade. In order to check the effect of the additional priors that were introduced into the models of Warfield et al. (2004), Warfield et al. (2008) and Raykar et al. (2009), additional experiments were run with exactly the same models as in the original papers (by fixing the corresponding variables or using uniform priors). However, this led to uniformly worse inference results than in the modified model formulations as described in section 3.3.2.

3.5.2. Experiments on real brain tumor measurements

For evaluation on real-world measurements, a set of twelve multimodal MR volumes acquired from glioma patients (T_1-, T_2-, FLAIR- and post-gadolinium T_1-weighting) was used. All images had previously been affinely registered to the FLAIR volume by an automated multi-resolution mutual information registration procedure as included in the MedINRIA[22] software. Manual segmentations of pathological tissue (tumor and edema) were provided separately for every modality on 60 slices extracted from these volumes (20 axial, sagittal and coronal slices each of which was intersecting with the tumor center). In these experiments, the described models are used to infer a single probability map summarizing all tumor-induced changes in the different imaging modalities. In particular, every modality is identified with a separate "rater" with a specific and consistent bias with respect to the joint probability map inferred.

3.6. Results

3.6.1. Simulated brain tumor measurements

Several scenarios (i.e. several compositions of the rating committee) were studied, which all gave qualitatively similar results for the accuracies of the different models, irrespective of whether "good" raters or "poor" raters were in the majority. Results are exemplarily reported for the 120/120/90 scenario (i.e. two raters with

[22] `https://gforge.inria.fr/projects/medinria`

	Specificity	Sensitivity	CCR	AUC	Dice
Majority vote	.987(007)	.882(051)	.910(032)	.972(008)	.827(020)
ALR	.953(018)	*.920(036)*	*.931(025)*	.981(005)	*.855(031)*
LRS	.953(019)	.919(037)	*.931(025)*	.981(005)	*.855(030)*
Warfield et al. (2004)	.987(007)	.882(051)	.910(032)	.972(008)	.827(020)
Warfield et al. (2008)	***1.000(001)***	.617(130)	.692(139)	**.989(003)**	.584(211)
Raykar et al. (2009)	.988(006)	.886(045)	.913(028)	***.993(003)***	.830(024)
Whitehill et al. (2009)	.988(004)	.913(016)	*.931(008)*	.980(003)	.845(063)
Hybrid model 1	.940(078)	.692(060)	.751(070)	.902(117)	.603(191)
Hybrid model 2	.972(019)	.716(048)	.770(057)	.953(015)	.628(163)

Table 3.2. – Evaluation statistics for the training data (i.e. the manual annotations of the raters were used for inference), under the 120/120/90 scenario. The first three rows show the outcome of the three baseline techniques. The best result in each column is marked *in italics*, while **bold numbers** indicate a significant improvement over the best baseline technique ($p < .05$, rank-sum test with multiple-comparison adjustment). Estimated standard deviations are given in parentheses. The outcome of the other scenarios was qualitatively similar (especially concerning the relative ranking between different inference methods). ALR = Averaged logistic regression. LRS = Logistic regression with soft labels. CCR = Correct classification rate (percentage of correctly classified pixels). AUC = Area Under Curve of the receiver operating characteristics curve obtained when thresholding the ground-truth probability map at 0.5. Dice = Dice coefficient of the segmentations obtained when thresholding both the inferred and the ground-truth probability map at 0.5.

a 120 sec constraint and one rater with a 90 sec constraint). Tables 3.2 and 3.3 show the results of various evaluation statistics both for training data (for which the human annotations were used) and test data. Sensitivity, specificity, correct classification rate (CCR) and Dice coefficient are computed from the binary images that are obtained by thresholding both the ground-truth probability map and the inferred posterior probability map at 0.5. If n_{fb} denotes the number of pixels that are thereby classified as foreground (tumor) in the ground truth and as background in the posterior probability map (and n_{bb}, n_{bf} and n_{ff} are defined likewise), these statistics are computed as follows:

$$\text{Sensitivity} = \frac{n_{\mathrm{ff}}}{n_{\mathrm{fb}} + n_{\mathrm{ff}}}, \qquad \text{Specificity} = \frac{n_{\mathrm{bb}}}{n_{\mathrm{fb}} + n_{\mathrm{bb}}},$$

$$\text{CCR} = \frac{n_{\mathrm{ff}} + n_{\mathrm{bb}}}{n_{\mathrm{ff}} + n_{\mathrm{bb}} + n_{\mathrm{bf}} + n_{\mathrm{fb}}}, \qquad \text{Dice} = \frac{2n_{\mathrm{ff}}}{2n_{\mathrm{ff}} + n_{\mathrm{bf}} + n_{\mathrm{fb}}}$$

	Sensitivity	Specificity	CCR	AUC	Dice
ALR	.937(017)	.924(038)	.928(029)	*.978(009)*	.837(065)
LRS	.936(017)	.925(038)	.928(029)	*.978(009)*	.837(066)
Raykar et al. (2009)	.927(019)	***.937(031)***	*.936(025)*	.977(013)	*.853(038)*
Hybrid model 1	.851(152)	.735(181)	.760(167)	.852(172)	.619(142)
Hybrid model 2	***.973(013)***	.727(174)	.786(116)	.952(026)	.667(084)

Table 3.3. – Evaluation statistics for the test data (i.e. the manual annotations of the raters were not used for inference), under the 120/120/90 scenario. Note that one can only employ the inference methods which make use of the image features φ_n and estimate a weight vector w: the unobserved test data labels are then treated as missing values and are marginalized over. All methods which only use the manual annotations (majority voting, and the methods by Warfield et al. (2004) and Warfield et al. (2008)) cannot be applied to these examples. The results for the other scenarios were qualitatively similar (especially concerning the relative ranking between different inference methods). Cf. the caption of table 3.2 for further details.

Additionally Area Under Curve (AUC) values are reported for the receiver operating curve obtained by binarizing the ground-truth probabilities with a fixed threshold of 0.5 and plotting sensitivity against $1 -$ specificity while the threshold for the posterior probability map is swept from 0 to 1. Most methods achieve Dice coefficients in the range of 0.8–0.85, except for the models operating on a continuous score (the hybrid models and the model by Warfield et al. (2008)). Since the chosen features are highly discriminative, even simple label fusion schemes such as majority voting give highly competitive results. Qualitatively, there is little difference between these two scenarios (and the other ones under study). While some graphical models perform better than the baseline methods on the training data (namely (Raykar et al., 2009) and (Warfield et al., 2008)), they bring no improvement on the test data.

Unexpectedly, the hybrid models perform worse and with lesser stability than the simple graphical models, and for hybrid models 3 and 4, the inference converges to a noninformative posterior probability of 0.5 everywhere. It should be noted that the posterior estimates of the rater properties do not differ considerably between corresponding algorithms such as (Warfield et al., 2008) and (Raykar et al., 2009), hence the usage of image features does not allow one to distinguish between better and poorer raters more robustly.

In order to account for partial volume effects and blurred boundaries between tumor and healthy tissue, it is preferable to visualize the tumors as soft probability maps rather than as crisp segmentations. In Fig. 3.7, the ground-truth tumor probabilities are compared with the posterior probabilities following from the different models. The models assuming a latent binary class label (i.e. those by Warfield et al. (2004); Raykar et al. (2009); Whitehill et al. (2009)) tend to sharpen the boundaries between

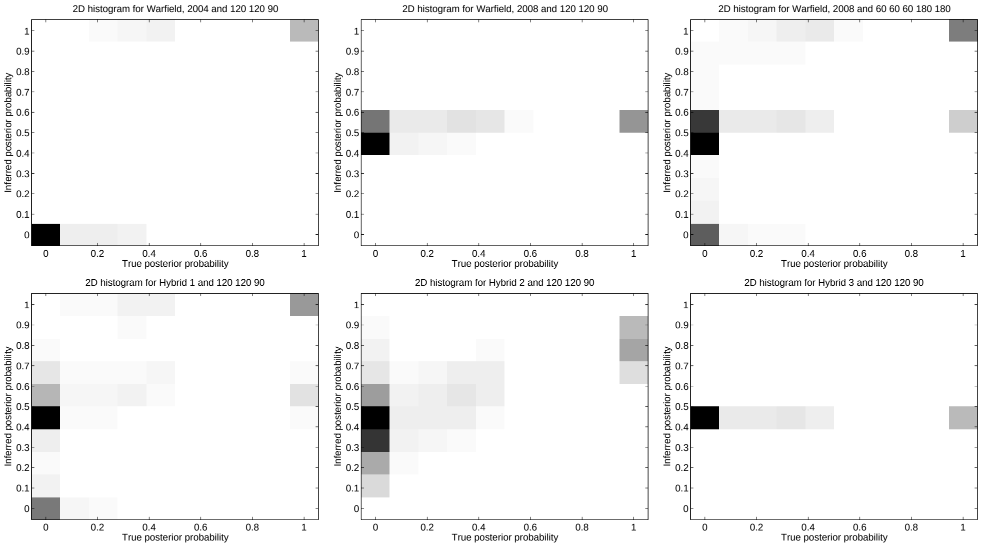


Figure 3.7. – Comparison of ground-truth (abscissa) and inferred posterior (ordinate) tumor probabilities for simulated brain tumor images, visualized as normalized 2D histograms. All histograms are normalized such that empty bins are white, and the most populated bin is drawn black. We show the inference results of (Warfield et al., 2004), (Warfield et al., 2008), and the hybrid models 1–3. The results of hybrid model 4 were similar to hybrid model 3, and the results of (Raykar et al., 2009) and (Whitehill et al., 2009) were similar to (Warfield et al., 2004). Most models gave similar results when the composition of the rater committee was altered, with the exception of (Warfield et al., 2008): Unexpectedly, this model gave slightly worse results for a scenario with a majority of better raters (e.g. 120/120/90, top middle) than for a scenario with a majority of poorer raters (e.g. 60/60/60/180/180, top right). For the ideal inference method, all bins outside the main diagonal would be white; Warfield et al. (2004) comes closest.

tumor and healthy tissue overly, while the latent score models (all others) smooth them. One can again note that the true and inferred probabilities are completely uncorrelated for hybrid model 3 (and 4).

3.6.2. Real brain tumor measurements

The optimal delineation of tumor borders in multi-modal image sequences and obtaining ground truth remains difficult. So, in the present study only a qualitative comparison of the different models is undertaken. Fig. 3.8 shows the posterior probability maps for a real-world brain image example. The results of the methods by (Warfield et al., 2004) and (Warfield et al., 2008) can be regarded as extreme cases: the former yields a crisp segmentation without accounting for uncertainty near the

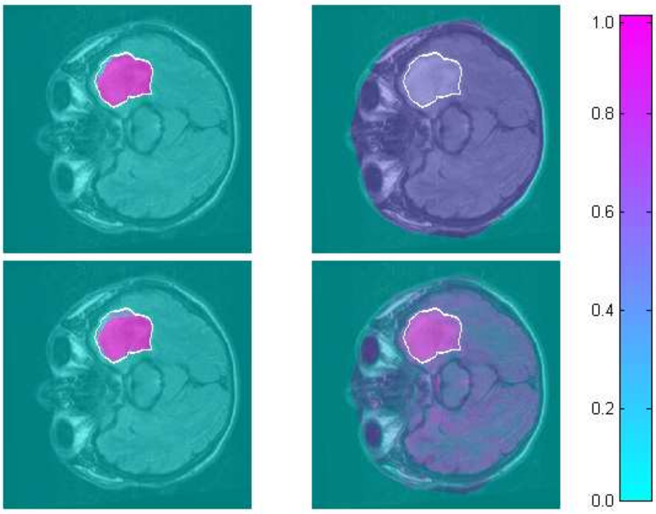


Figure 3.8. – Example of a FLAIR slice with manual segmentation of tumor drawn on the same FLAIR image (white contour), and inferred mean posterior tumor probability maps for (Warfield et al., 2004) (top left), Warfield et al. (2008) (top right), (Whitehill et al., 2009) (bottom left) and hybrid model 2 (bottom right). The results of hybrid model 3 and 4 were nearly identical to (Warfield et al., 2008), the results of hybrid model 1 to model 2, and the results of (Raykar et al., 2009) to (Whitehill et al., 2009). Tumor probabilities outside the skull were set to 0.

tumor borders, while the latter assigns a probability near 0.5 to all pixels and is hence inappropriate for this task. Hybrid model 1 (or 2) and the methods by (Whitehill et al., 2009) or (Raykar et al., 2009) are better suited for the visualization of uncertainties.

Chapter 4.

Live-cell microscopy image analysis for the study of zebrafish embryogenesis

4.1. Introduction and motivation

Digital Scanned Laser Light Sheet Fluorescence Microscopy (Keller & Stelzer, 2008, DSLM) is a recent live-cell imaging technique which provides unprecedented spatio-temporal resolution and signal-to-noise ratio at low energy load. This makes it an excellent tool for in-vivo studies of embryonic development at a cellular level: in particular, it allows one to determine the detailed fate of each single cell, its motion, divisions and in some cases eventual death, to construct a digital model of embryonic development (also called a "digital embryo") and to extract a cell lineage tree showing the ancestry and progeny of each cell. However, the huge number of images that are produced (due to the high spatio-temporal resolution) can no longer be analyzed manually: hence automated image processing methods are required in order to extract the biologically relevant information out of the raw image data.

This chapter describes two contributions to an image processing pipeline that shall eventually be used for high-throughput analysis of nucleus-labeled DSLM imagery.[1] The whole pipeline consists of the following parts:

Segmentation After interpolating the image stack in the z direction (so that all voxels are roughly isotropic), cell nuclei are segmented in a three-stage scheme developed by Lou et al. (2011b): Firstly, foreground seeds are generated by identifying local maxima (i.e. points where all eigenvalues of the Hessian are negative) that occur robustly across several levels in state-space and refining them via morphological closing and opening. These seeds serve as automatically generated foreground labels for a random forest classifier, while blurred watersheds between the basins flooded from the foreground seeds serve as background labels. The final segmentation is

[1] Parts of this chapter form part of (Lou et al., 2011a).

obtained by solving a discrete energy minimization problem via the graph cut algorithm (Boykov et al., 2001): the energy function incorporates single-site potentials (the classifier log-posterior probabilities) as well as higher-order terms corresponding to smoothness and shape priors, as well as flux priors guarding against the shrinking bias by which graph cut segmentation is commonly affected. For encoding the shape assumptions, a multi-object generalization of the gradient vector flow proposed by Kolmogorov & Boykov (2005) is used.

For several reasons, this is the hardest as well as the most crucial step of the pipeline. All later stages assume that the true nuclei form a subset of these segments: while some segments may later be discarded as misdetections, true nuclei that are missed cannot be recovered again. Hence the quality requirements for the segmentation are very high; in particular, the sensitivity should be close to 100 %, while a smaller specificity can be tolerated. For the same reasons, it should ideally never occur that two distinct nuclei are erroneously merged (undersegmentation), while the opposite case of oversegmenting one single nucleus into two segments can later be handled by discarding one of these segments. Further impeding factors are (see Fig. 4.1):

1. the high variability of nucleus brightness,
2. the inhomogeneous illumination of the images with characteristic striped artifacts, which are probably due to a combination of drifts in the illuminating laser intensity, and the linear scanning order (see section 4.2.2),
3. the presence of high-intensity speckles that can easily be mistaken for nuclei,
4. the varying texture and sometimes low contrast of the nuclei to be segmented,
5. the leakage of fluorescent dye into the cytoplasm as well as
6. the weak boundaries between neighboring nuclei, which bring a high risk of undersegmentation.

Besides finding the correct number and positions of nuclei, segmenting the correct size of the nuclei is an additional challenge, and many state-of-the-art segmentation methods are prone to shrinkage.

The first contribution of this chapter (section 4.4) is an experimental comparison of the segmentation scheme detailed above with the results obtained with a recently introduced interactive segmentation software (Sommer et al., 2010).

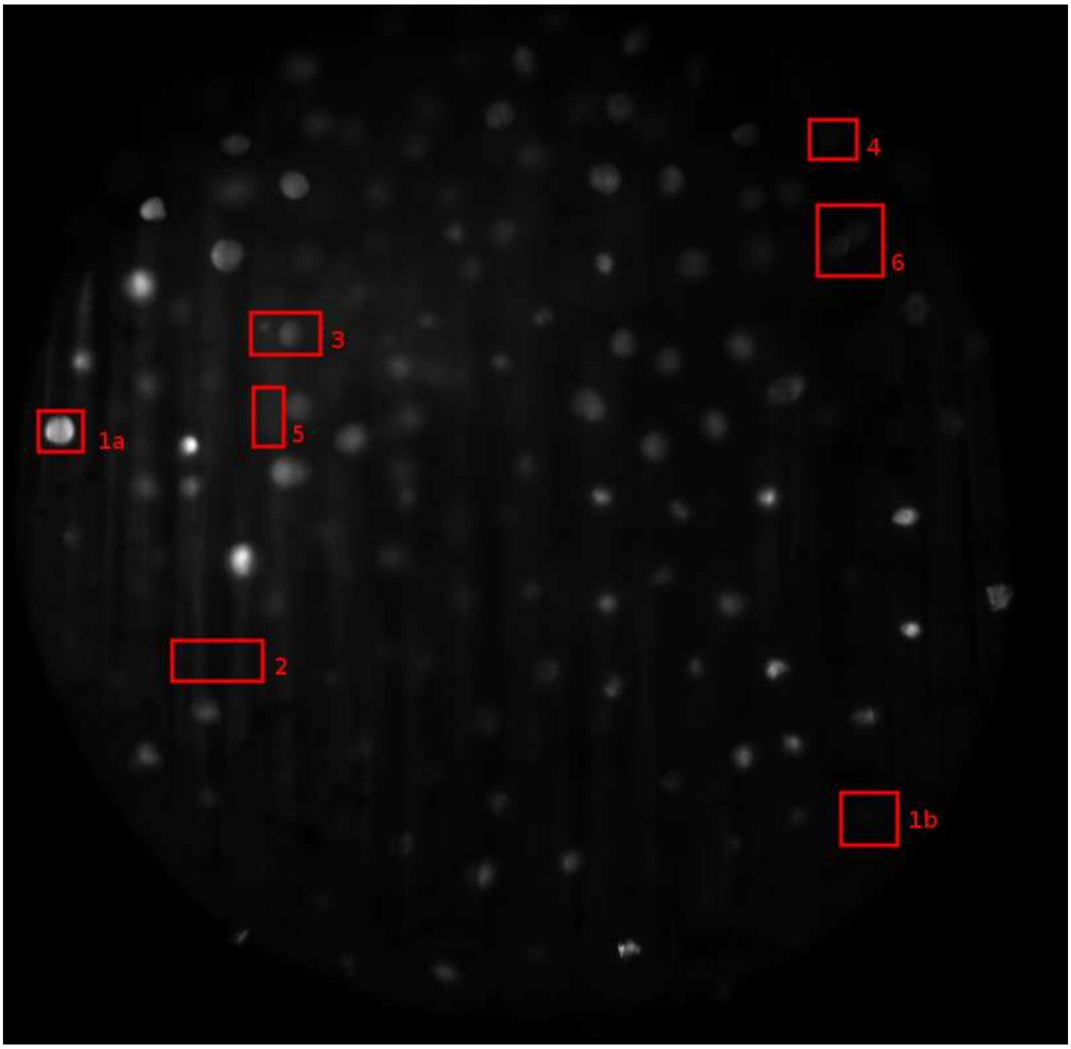


Figure 4.1. – Exemplary slice of a DSLM zebrafish image. The red rectangles mark areas where the different challenges of the data can best be illustrated: highly varying nucleus brightness (1a and 1b), striped illumination inhomogeneities (2), speckles which often occur close to real nuclei (3), low contrast (4), presence of fluorescent markers in the cytoplasm (5), weak boundaries between adjacent nuclei (6).

Feature extraction Connected component labeling is used to transform the binary image generated by the segmentation step into a list of individual nucleus objects. The individual objects are efficiently stored in a dictionary of keys-based[2] sparse matrix representation, and the segmented nucleus candidates are characterized by different features. These may be:

- **Geometrical features** such as the center of mass position (i.e. the intensity-weighted average position of the segment), the volume, the side lengths of the smallest bounding box around the segment or the principal components of the segment (i.e. the semiaxis lengths of an ellipsoid that is fitted to the intensity distribution).
- **Intensity distribution features**, i.e. both the leading central moments (mean, variance, skew, kurtosis), the maximum and minimum and the quartiles of the intensity distribution inside the segment.
- **Texture features**: for characterizing texture properties, the statistical geometric features (SGF) by Walker & Jackway (1996) are used. They are computed by binarizing the gray value images inside each segment at different thresholds, extracting intermediate features on each binary image (e.g. average squared distance of the connected component centers from the center of gravity) and aggregate statistics (such as mean or standard deviation) over all intermediate features, which are then used as the final features.

Cell tracking In order to efficiently track the large number of nuclei over time, the jointly optimal association of nuclei is found for every pair of subsequent time frames. The tracking algorithm is the second contribution of this chapter: hence it is described in detail in section 4.5.

Interactive visualization The results are interactively visualized by a software called Visbricks, which is based on the OpenSceneGraph 3D computer graphics library.[3] It offers the following capabilities:

- Visualization of all segmented nuclei in a given subvolume by their center-of-mass positions along with the principal components semiaxes or by volume rendering with smooth shading.

[2] The dictionary of keys representation describes as sparse matrix as a dictionary, with the keys being the row/column index tuples and the values being the nonzero entries of the matrix.

[3] http://www.openscenegraph.org

- Validation of individual nuclei by showing the cross-section of a selected nucleus across the plane defined by the leading principal components together with the segmentation isocontour.
- Visualization of the 3D trajectories of individual cells and their progeny over time.
- Synchronized display of the raw image data, nucleus segments and the cell lineage tree topology.

4.2. Background

4.2.1. The zebrafish *Danio rerio* as a model for vertebrate development

The zebrafish (*Danio rerio*) is a popular aquarium fish that has become one of the classical model organisms for vertebrate development, along with the Japanese ricefish (*Oryzias latipes*), the African clawed frog (*Xenopus laevis*), the chicken (*Gallus gallus domesticus*) and the mouse (*Mus musculus*). Due to the transparency of its embryos during their first 36 hours of development and its nearly constant size during the first 16 hours, it is particularly well-suited to in-vivo imaging studies.

In contrast to avertebrate model organisms such as the nematode *Caenorhabditis elegans*, the development of zebrafish embryos has no stereotypical course, and even genetically identical specimens may develop asynchronously. However, the usual development under optimal incubation conditions (28.5° C) can be roughly divided into the following eight periods (Kimmel et al., 1995):

Zygote During the first 45 minutes p.f.,[4] cytoplasm streams to the animal pole, where the nucleus is located: there it forms the so-called blastodisc. Meanwhile the yolk mass remains at the vegetal pole. At the animal pole, the fertilized egg undergoes its first mitotic division.

Cleavage From 45 to 145 minutes p.f., the second to seventh mitotic divisions occur rapidly (at 15 minute intervals), in which all cells in the embryo divide synchronously. However, the cell cleavage is not complete, and the cells are still connected by cytoplasmatic bridges. At the end, the 64 cell stadium is reached, and the cells are arranged in three regular layers.

[4] post fertilisationem, i.e. after fertilization.

Blastula During the next three hours (2.25 – 5.25 hours p.f.), the synchrony of cell cycles is gradually lost, and the average cell cycle duration increases. The cell arrangement also loses its regularity. The cell cycles 8 and 9 are still rapid and metasynchronous (i.e. the cells divide at nearly the same time), while the subsequent cell cycles are longer (up to 60 min) and asynchronous. From this stadium on, the cell cleavage is always completed and there are no cytoplasmatic bridges connecting adjacent cells. These cells in the lowest layer, which are neighboring the yolk, lose their integrity and release their cytoplasm and nuclei into the yolk: the yolk syncytial layer[5] arises, in which the nuclei still undergo mitosis, which is however not accompanied by a division of the cytoplasm. In the second half of the blastula period, epiboly sets in: both the blastodisc and the yolk syncytial layer thin and spread over the yolk sac, which is roughly halfway engulfed at the end of this stadium (50 % epiboly).

Gastrula This stadium lasts from 5.25 to 10 hours p.f., during which epiboly is completed (at 100 % epiboly, the yolk sac is fully enclosed by the embryo). In parallel, a thickened region (the germ ring) appears around the rim of the blastodisk, and cells accumulate at one particular position along this ring, the embryonic shield.[6] The germ ring consists of two germ layers, the epiblast and the hypoblast, with cells moving from the epiblast down into the interior of the embryo (towards the hypoblast). As the embryonic shields marks the later dorsal side of the embryo, this is the first time when the final embryonic axes can be discerned. Near the posterior end of the embryo, the tail bud starts to develop.

Segmentation From 10 up to 24 hours p.f., the tail extends futher extends from the tail bud. Along the anteroposterior axis, somites (i.e. primitive body segments) appear sequentially, which will later form the segments of the vertebral column as well as the associated muscles. Also, along this axis the notochord is formed, which induces neurulation: a ridge in the epiblast develops into the neural tube, which is segmented into neuromeres: these develop into the central nervous system, i.e. the brain and the spinal column. Motor axons grow out from the neuromeres towards the muscle precursors in the somites. This is also the period when the first body movements start. Rudiments of the kidneys and the eyes appear. In the head, the pharyngeal arches appear, which will later evolve into the gills and the jaws.

[5] The yolk syncytial layer is considered an extraembryonal tissue; it is unique to teleosts (bony fishes).

[6] This process is also called involution.

Pharyngula During the second day of development (24 – 48 hours p.f.), the body axis (which has hitherto be curved) starts to straighten. The circulatory system begins to develop, as well as the liver, the swim bladder and the gut tract, and around 36 hours all primary organ systems are present. During the end of the segmentation period and the beginning of the pharyngula period (16 – 32 hours p.f.), the embryo also experiences a rapid growth phase, in which it grows from its initial size of 1 mm to nearly 3 mm. Pigmentation sets in and the fins start to develop: Due to the pigmentation and the rapid growth, this is the time point where the organism can no longer easily be studied by live microscopy.

Hatching During the third day of development (48 – 72 hours p.f.), the morphogenesis of most organ systems except for the gut is completed. The gills and jaws are formed from the pharyngeal arches, and cartilage develops in the head and the fins. Sometime in this period, the larva hatches out of the chorion, in which it has been confined up to this point.

Early larva By the third day, the morphogenesis of the larva has been completed and the shape of the body and its organs stays mostly constant from then on. The swim bladder inflates, and the larva begins autonomous swimming and feeding movements. The larva eventually grows from its size of 3.5 mm (after hatching) to its final size of 4 cm, and reaches sexual maturity after 12 weeks.

4.2.2. Digital scanned laser light-sheet fluorescence microscopy (DSLM)

Fluorescence microscopy using GFP Fluorescence microscopy is a microscopy technique which detects the structures of interest by coupling them with fluorescent molecules and recording their light emission: since light emission occurs at a longer wavelength than the absorption of the illumination light by which the fluorophores are excited, wavelength-selective filters can be used to suppress the illumination background. Arguably the most important fluorophore in biology is the green fluorescent protein (GFP) of the jellyfish *Aequorea victoria*, which absorbs blue light at 395 nm and emits green light at 509 nm (Chalfie et al., 1994). By fusing the GFP gene with the gene for the histone protein H2B, and transferring this fusion gene to a living cell via mRNA injection, one can fluorescently label the chromatin within the nuclei of a cell and its daughter cells after mitosis (Kanda et al., 1998).

Light-sheet-based fluorescence microscopy Conventional fluorescence microscopy techniques use a single lens for illuminating the specimen and for gathering the emitted light: hence the whole specimen is flooded in light, even if only the focal plane is currently imaged. This poses problems due to phototoxicity and photobleaching: illuminated fluorophores may cause the death of the cells expressing them (possibly due to the formation of reactive oxygen radicals), and the fluorophore themselves may be destroyed after prolonged exposure due to chemical reactions and covalent bond forming while being in the excited state. This limits the applicability of fluorescence imaging, especially to time-lapse imaging series in which images are taken at regular intervals over a long time. Light-sheet-based fluorescence microscopy (LSFM) alleviates this problem by selectively illuminating only the focal plane that is currently imaged: for this purpose, two separate lenses are used for illumination and collecting the emitted light (such that the optical axes are perpendicular to each other), and a thin light sheet (formed by apertures) is used for illumination instead of flooding the entire specimen (Reynaud et al., 2008).

Digital scanned laser light-sheet fluorescence microscopy DSLM (Keller et al., 2008; Keller & Stelzer, 2008) is a variant of LSFM, in which a laser beam sequentially illuminates the entire specimen by a raster scan, thereby enabling 3D imaging. The advantages of using a laser instead of apertures for the light-sheet formation are an improved image quality (due to reduced optical aberrations), an increased intensity homogeneity and an increased illumination efficiency (due to the highly localized energy deposition). The SNR is typically 1000:1, and hence by a factor of 10–100 better than that of conventional techniques, at an energy deposition which is reduced by a factor of $10^3 - 10^6$ (leading to minimal photobleaching and phototoxicity and allowing time-lapse imaging over a long period). The specimen is typically fixated in a transparent gel such as agarose. CCD cameras are used for fast image capturing: typically, images of 4.2 megapixels (2048 × 2048) can be acquired with a frame rate of 15 frames per second, leading to a data rate of 1 Gbit/s for 16 bit images. Lateral and axial resolutions down to 300 nm and 1000 nm respectively can be achieved, and a multi-view image acquisition can be used to achieve isotropic image resolution (i.e. by taking several images from different angles and fusing them in silico).

4.2.3. Integer linear programming

The following mathematical background is common knowledge and covered in standard textbooks such as those by Papadimitriou & Steiglitz (1998) or Wolsey (1998). A linear program (LP) is a mathematical optimization problem for which both the optimization objective and the constraints are linear in the variables. In its canonical form, it is stated as

$$\min_{x} c^{\top} x \text{ subject to } Ax \geq b. \tag{4.1}$$

It can be viewed as the minimization of a linear function over the convex polytope defined by $Ax \geq b$. If we denote the number of variables with p and the number of constraints with m, then $x \in \mathbb{R}^p$ is the variable vector, $c \in \mathbb{R}^p$ the cost vector and $A \in \mathbb{R}^{m \times p}$ the constraint matrix.

State-of-the-art algorithms for globally solving the LP problem in Eq. (4.1) fall into two categories:

- **Simplex algorithm**: This algorithm by Dantzig (1949) makes use of the fact that the minimum must be attained on one of the vertices of the feasible polytope: starting from an initial vertex, one iteratively visits adjacent vertices such that the objective decreases. Different pivoting strategies exist which specify which neighbor to take if there are several possibilities, and may lead to vast differences in the practical performance of the algorithm (see (Terlaki & Zhang, 1993) for a somewhat dated overview). Most known pivoting schemes of the simplex algorithm give exponential worst-case complexity, and it is currently not known whether variants with polynomial complexity exist. However, these worst-case problem instances are mostly pathological, and the average-case complexity is typically cubic, both for random problem instances and for a variety of real-world instances.

- **Interior-point methods**: The first usable LP solver with proven polynomial complexity was proposed by Karmarkar (1984); the previously proposed ellipsoid algorithm (Aspvall & Stone, 1980) was unfit for practical use due to numerical stability problems and gave non-competitive performance on all real-world LP instances. In contrast to the simplex algorithm (where the current candidate solution is always a polytope vertex), it maintains an interior point of the polytope as the current solution and reaches the optimal solution on the border only asymptotically. The key idea is the replacement of the inequality constraints by adding a differentiable barrier function to the minimization objective, which becomes infinite at the border of the feasible region: the minimum of this adjusted objective is then found using Newton updates, and the contribution of the barrier term is iteratively decreased to zero (for a recent overview see (Nemirovski & Todd, 2008)).

Integer linear programming (ILP) is a mathematical optimization problem of the form as in Eq. (4.1), with the additional constraint that $x \in \mathbb{Z}^p$ must be a vector of integers. An important special case occurs when the x_i are further constrained to be either 0 or 1 (binary integer programming, BIP): this is one of the classical 21 NP-complete problems identified by Karp (1972) and is hence unlikely to be solvable in polynomial time. Nonetheless, powerful algorithms exist for finding the global optimum of ILP instances that can nowadays solve problem instances with up to a few hundred thousands of constraints and variables (Achterberg et al., 2006). They fall into three categories:

- **Branch and bound**: This strategy generates a search tree for all feasible solutions, and prunes unpromising subtrees to avoid complete enumeration (which would require exponential time). A subtree typically corresponds to the subproblem of fixing some variables of the original problem and finding the optimum over the remaining variables. It uses the fact that solving the LP relaxation of an ILP subproblem (with the integrality constraints dropped) gives a lower bound for the optimal solution of the ILP problem: if this lower bound already exceeds an upper bound for the global optimum (e.g. the best feasible solution that is currently known) then the subtree may be pruned.
- **Cutting plane**: This strategy tries to find the integer polytope, i.e. the convex hull of all integral feasible points, which is usually a strict subset of the convex polytope of the relaxed LP problem. Obviously, solving the LP relaxation over the integer polytope would give the solution for the original ILP problem, but finding this polytope is an NP-complete problem by itself. The cutting plane algorithm iteratively adds inequality constraints that are met by all feasible integral points, until the solution of the LP relaxation becomes integral.
- **Branch and cut**: This is a hybrid of the two other strategies, where the cutting plane algorithm is applied to the subproblems encountered while traversing the branch and bound search tree, leading to tighter lower bounds due to the additional constraints.

It should be noted that the performance of these methods depends on the individual ILP instance upon which they are applied: while they give fast solutions for many practically relevant instances, their worst-case complexity is still exponential. However, there is an important subclass of ILP instances, which are guaranteed to be solvable in polynomial time, namely those where the constraint matrix A is *totally unimodular* (TU): this means that the determinant of every quadratic minor must be either 0, +1 or -1. For these instances, the constraint polytope is at the same time the integral polytope, hence the ILP problem has the same solution as its LP relaxation. Several important network flow problems fall into this category, e.g. the *minimum-cost flow problem*, which asks how to route a given amount of flow f from a source s to a sink t, via a directed network of edges with a transport cost c_i and a maximum capacity b_i:

$$\min_x c^\top x \text{ s.t. } x \geq 0,\ x \leq b,\ \textstyle\sum_e x_e d_{ve} = f(\delta_{vs} - \delta_{vt}) \text{ for all } v. \tag{4.2}$$

In this equation, $(d_{ve})_{ve}$ denotes the directed incidence matrix, i.e. for every node v and edge e,

$$d_{ve} = \begin{cases} +1 & v \text{ start node of } e \\ -1 & v \text{ end node of } e \\ 0 & \text{else} \end{cases} \tag{4.3}$$

4.3. Related work

First, section 4.3.1 discusses previous research which has a scope similar to the entire pipeline to which this chapter contributes, namely reconstructing the cell lineage tree of an entire organism. In contrast, sections 4.3.2 and 4.3.3 present related work in the two most important subcomponents of this pipeline, namely nucleus segmentation and nucleus tracking. Due to the multitude of publications in those areas, only some selected articles can be discussed that have the highest relevance for putting the techniques discussed in this chapter into context.

4.3.1. Cell lineage tree reconstruction

Cell lineage reconstruction has been pioneered in the nematode *Caenorhabditis elegans*, which has a highly invariant cell lineage: 671 cells are generated, of which 113 (for hermaphrodites) undergo programmed cell death. Due to this moderate number of cells, the first lineage tree could be generated by manual tracing in interferometric microscopy images of living worms (Sulston et al., 1983).[7] However, this manual lineage reconstruction becomes impracticable when a large number of specimens shall be analyzed for their cell lineage, e.g. in order to elucidate the developmental effects of genetic variants.

Bao et al. (2006) present an automated lineage reconstruction system for confocal time-lapse microscopy imagery of H2B-GFP labeled *C. elegans* embryos up to the 350 cell stadium: nuclei are identified as local intensity maxima (with the constraint that there must be a minimum distance between all nucleus pairs) and approximated by the best spherical fit to the local intensity distribution. Nucleus tracking works by

[7] It should be noted that this work was awarded the 2006 Nobel Prize in Physiology or Medicine.

a greedy procedure. First each nucleus is tentatively assigned to its nearest neighbor in the previous time step, and then this assignment is refined by tackling cell divisions separately: if a nucleus at time t has several children at time $t+1$, all possible parents of these nuclei are computed (i.e. those whose distance is lower than a threshold), and a hand-crafted scoring function is used to select which of these possible parents are actually selected for each nucleus in the end. A graphical user interface is also provided for final manual lineage tree correction.

Recently, research has been undertaken with the aim of achieving automated cell lineage reconstruction also in a vertebrate, namely in the zebrafish *D. rerio*. It culminates in the publication of the zebrafish cell lineage tree up to the 1000 cell stadium (i.e. the first ten mitotic divisions, up to the mid-blastula stadium), based on label-free microscopic imagery (Olivier et al., 2010). The authors employ harmonic-generation microscopy in order to forego the need for fluorescent labeling: Mitotic nuclei and cell membrane positions are extracted from second-harmonics generation and third-harmonics generation images, as second harmonics are generated at anisotropic structures (such as microtubule spindles) and third harmonics are generated at interfaces between aqueous and lipidic media (such as cell membranes). A nearest neighbor scheme with interactive manual corrections is used for cell tracking. Additionally, a software for automated segmentation and tracking of cells in the zebrafish brain (based on laser scanning confocal microscopy) has be published (Langenberg et al., 2006), but no details about the technical workings are provided.

However, the only published evaluation pipeline for DSLM data is the one presented in (Keller et al., 2008): There the authors segment cell nuclei by local adaptive thresholding and perform local nucleus tracking by a nearest neighbor search. Nucleus detection efficiencies of 92 % and tracking accuracies of 99.5 % per frame-by-frame association can be achieved.

4.3.2. Cell or nucleus segmentation

Multi-scale initialization and graph-cut refinement The first of the segmentation methods studied in section 4.4, which was developed by Lou et al. (2011b), is most closely related to the nucleus segmentation presented in (Al-Kofahi et al., 2010): both approaches use blob filter responses that are coherent across multiple scales as initial seeds for the segmentation, and refine them via a discrete graph cut optimization. However, the method by Lou et al. (2011b) differs by the use of more flexible foreground cues based on discriminative random forest classifiers (see section 2.2) instead of the Poisson mixture model employed in the other article, by explicit shape regularization using a multi-object generalization of the graph cuts algorithm presented by Boykov et al. (2001), and by being a 3D segmentation in contrast to the 2D segmentation in (Al-Kofahi et al., 2010).

Classification based on local features The ILASTIK procedure, i.e. the second of the segmentation methods studied in section 4.4 is most closely related to the work of Yin et al. (2010). Both approaches extract local features from every pixel and classify them as either foreground or background; finally simple segmentation schemes are used to group spatially connected foreground pixels into segmented nuclei. However, the ILASTIK procedure uses the responses of convolutional filters computed at multiple scales as features, and a random forest as supervised classifier, while Yin et al. (2010) extract histograms from a patch window around each pixel, which are then clustered and classified using a Bayesian mixture of experts.

Level-set evolution A different approach is followed by Bourgine et al. (2010) and Zanella et al. (2010), who tackle the three steps of their segmentation pipeline (image denoising, center detection and pixel-accurate segmentation) with a common mathematical formalism, namely nonlinear partial differential equations. For denoising, a variant of anisotropic diffusion is used, objects are distinguished from background speckles by a level set evolution which causes small objects to vanish quickly, and voxelwise segmentation is achieved by a level set evolution that can account for missing boundaries by curvature regularization. This segmentation method is then applied for detecting and delineating cell nuclei in confocal time-lapse microscopy of zebrafish embryos. Compared to manual ground truth, they achieve mean Hausdorff distances (over all nuclei) in the range of 0.35 μm – 0.98 μm, with an average of 0.65 μm. Mosaliganti et al. (2009) propose a different level-set segmentation method for the same application: they fit a parametric model to the intensity distributions of training nuclei, and incorporate this as a prior into an active contour-based energy minimization problem, achieving Dice indices of 0.86 – 0.94 on different datasets. However, a disadvantage of continuous PDE-based segmentation methods (as opposed to discrete techniques like graph cuts) is that they are prone to get stuck in local optima of the energy functional.

Gradient flow tracking Li et al. (2007, 2008a) use a three-stage procedure for segmenting cell nuclei in 3D confocal fluorescence images of *C. elegans* and *D. rerio*. First, the image gradient field is denoised using gradient vector diffusion (i.e. solving a Navier-Stokes PDE). Secondly, the image is partitioned into the attraction basins of the gradient vector field, and it is assumed that each basin contains at most one nucleus. Finally, the local adaptive thresholding method by Otsu (1979) is used to compute the final nucleus segmentation, which achieves a volume overlap of 90 %

with the manual segmentation ground truth. These methods have been made publicly available in a software package called ZFIQ (Liu et al., 2008).

4.3.3. Cell or nucleus tracking

Previous approaches for the tracking of cells or nuclei fall into four categories:

1. Segmentation and frame-by-frame association,
2. level-set evolution and
3. stochastic filtering,
4. four-dimensional segmentation of spatiotemporal volumes.

Segmentation and frame-by-frame association In this formalism, independent nucleus segmentation is performed on each data volume (at each time step), and afterwards the optimal association of nuclei across different time points is found. In most cases, integer linear programming is used for matching nuclei between pairs of subsequent time steps (Al-Kofahi et al., 2006; Padfield et al., 2009a; Li et al., 2010): the objective is a suitably selected energy function, which makes sure that e.g. spatially close nuclei are more likely to be matched than distant nuclei, while the constraints make sure that no cell is used by more than one matching event. There are slight differences between the various papers (e.g. whether occlusions or entering and leaving the field of view are modelled), but the mathematical formulation of the models is nearly identical. The approach followed in this paper (section 4.5) follows the same formalism.

Level-set evolution Level-set evolution techniques model the cell boundaries as zero level sets of an implicit embedding function. Hence they are not restricted to a certain topology and can easily account for splits and merges of objects: Additional objects simply correspond to additional hills in the profile of the embedding function. Padfield et al. (2009b) segment and track GFP-labeled nuclei in time-lapse 2D fluorescence microscopy imagery by interpreting the images acquired at different time points as a 3D stack, and use level set evolution (initialized with automatically placed seed points, which are determined by a classifier) to segment the entire cell trajectories. For tracking fluorescent cells in time-lapsed 3D microscopy, Dufour et al. (2005) use a coupled active surfaces approach, which identifies every cell with the zero level set of a single embedding function and adds overlap penalties and volume preservation priors in order to prevent cells that overlap with other cells, or shrink or grow too rapidly. A level-set segmentation is then performed on the individual data volumes, in which the final segmentation of the previous time step always serves as initialization for the next segmentation task. This approach reaches a tracking accuracy of 99.2 %.

Stochastic filtering Li et al. (2008b) use a combination of stochastic motion filters with the techniques presented in the previous two paragraphs for cell tracking in 2D phase contrast microscopy imagery: first cells are detected using a combination of region-based and edge-based object detection techniques, then predictions for their central position in the next time step are cast using an interacting multiple models filter, which allows cells to have different motion patterns (such as e.g. Brownian motion or constant acceleration). This prediction is combined with the detection result from the next time step, and incorporated as one term into the energy functional of a level-set tracking scheme, which computes the definite tracking event across two subsequent time steps. Explicit rules are then used to compile tracking events into track segments (spanning multiple time steps), which are finally linked to a lineage tree using integer linear programming. Tracking accuracies between 87 % and 93 % can be achieved by this method on different datasets. A simplified version of this procedure (which uses only object detection and interacting multiple models filtering, but neither level-set evolution nor integer linear programming) is employed by Genovesio et al. (2006) for tracking quantum dot-labeled endocytosis vesicles in 3D fluorescence microscopy imagery, achieving a true positive rate of 85 % and a false positive rate of 6 % among all tracks. However, this is an easier task than cell or nucleus tracking, since vesicles do not divide.

Four-dimensional segmentation of spatio-temporal volumes If the cells or nuclei show sufficient overlap in subsequent time frames, one can unify segmentation (in space) and tracking (over time) by segmenting the nucleus tracks in a four-dimensional volume with three spatial and one temporal dimensions, as many three-dimensional segmentation techniques can be readily generalized to more dimensions. Luengo-Oroz et al. (2007) apply four-dimensional mathematical morphology to optical sectioning microscopy of zebrafish embryos with fluorescence-labeled nuclei and membranes. The nucleus lineage tree is finally used as the seeds in a seeded watershed segmentation of the cell outlines in the cellular membrane channel. 90 % of all mitosis events are identified correctly by this approach. A disadvantage of this approach is the high memory load, since a typical single three-dimensional microscopy image volume already has a size of several gigabytes and the four-dimensional segmentation has to access the data from all time points simultaneously.

4.4. Experimental comparison of two nucleus segmentation schemes

4.4.1. Introduction

Two nucleus segmentation procedures were experimentally compared: The first of these methods by Lou et al. (2011b) is fully automated; for a description see section 4.1. Since its final step is solving a shape-regularized graph-cut optimization problem, it is henceforward referred to as the "regularized graph cut" (RGC) segmentation. The second method uses the Interactive Learning and Segmentation Toolkit (ILASTIK) software by Sommer et al. (2010) for semiautomatic segmentation: the users interactively train a random forest classifier (see section 2.2) for the task of distinguishing between foreground (cell nuclei) and background (everything else) based on locally extracted image features. When a new label is placed on a training image volume, the current random forest predictions are automatically updated and displayed, so that the users can see where the classifier still performs poorly, and place their labels at these locations. The trained classifier can then be used to predict the foreground and background probabilities of all voxels either of the same volume that it was trained on, or of a new test image volume. These continuous probabilities can then be converted to a binary segmentation either by simple thresholding or by more sophisticated schemes that incorporate spatial regularity terms (e.g. graph cut segmentation). In this chapter, only the simple thresholding method will be used for this purpose. The main difference between the two methods is that RGC automatically generates foreground and background labels in order to train a classifier, and sophisticated spatial and shape regularization is used in order to transform the classifier predictions into a binary segmentation. For ILASTIK in contrast, the labels are placed manually by a user, and the segmentation is obtained from the classifier predictions by a trivial procedure.

4.4.2. Evaluation methodology

Dataset The experimental studies described in this chapter are based on a series of 100 DSLM image volumes, showing the animal pole of a H2B-eGFP labeled zebrafish embryo; see Fig. 4.1 for an exemplary slice. While the native voxel size was $0.3 \times 0.3 \times 1.0\ \mu\text{m}^3$, the data were resampled in the z-direction resulting in a nearly isotropic voxel size of $0.3 \times 0.3 \times 0.33\ \mu\text{m}^3$. The total number of voxels in a volume after resampling was $1161 \times 1111 \times 486 = 6.3 \times 10^8$. 60 seconds elapsed between the acquisition of two subsequent volumes. For comparison: A typical nucleus has a diameter of 7 μm, the typical mitosis duration in *D. rerio* is about 6–7 minutes and the typical migration speed of nuclei is less than 3 μm/min in the interphase of the cell cycle, and 8 μm/min in the metaphase (Kimmel et al., 1995).

Need for feature selection Classification-based segmentation methods such as the ones studied in this chapter require local image features, which are typically generated by convolving the image with Gaussian kernels of different scales and computing the responses of different image filters that capture properties like edge strength (e.g. gradient amplitudes), presence and orientation of blobs and ridges (e.g. the Hessian matrix and its eigenvalues) or local anisotropy (e.g. the structure tensor and its eigenvalues). The scale of the Gaussian kernel determines an interaction length between different locations in the image, i.e. how large a neighborhood should be chosen to take the context of each voxel into account for the classification. The large size of the image volumes acquired by DSLM necessitates feature selection in order to apply the ILASTIK segmentation method effectively: for a typical whole-embryo volume with a size of roughly 4×10^8 voxels, computing all routinely available local image features would require ca. 250 GB of main memory, which is by far out of reach for current desktop computers. However, not all features are equally useful for classification, and selecting a parsimonious feature set allows one to train a classifier if not on the whole data volume, then at least on a subvolume of the maximal possible extent.[8] This is important in order to obtain a classifier that is suitable for classifying an entire image volume, since the local appearance of the nuclei changes across the image due to illumination inhomogeneities.

Variable importance estimation Fortunately, the random forest classifier that is used as part of the ILASTIK segmentation provides two simple measures for the importance of the different features (Breiman, 2001): one generic measure that can be computed for every classifier, and one that is specific to the random forest. The generic method computes for every feature the decrease in classification accuracy that occurs when the values of this feature for the different training examples are randomly permuted. This means, if the rows of the $n \times p$ matrix X contain the features extracted for each of the n training examples, the classifier is retrained p times with a training matrix $X^{(p)}$ which differs from X by a random permutation of the p-th column. The decrease in classification accuracy can be computed separately for foreground and background examples, or averaged across all classes. A less costly alternative to this permutation-based variable importance measure uses the fact that the trees of the random forest classifier are grown by iteratively searching for the feature cut with the highest decrease in the Gini impurity (see Eq. (2.8)) among a

[8] This problem could be bypassed via the use of a lazy evaluation scheme, which computes the image features for the slice currently examined by the user on the fly. However, for the current processor speeds this procedure would be too slow for real-time responsiveness.

randomly selected feature subset. During classifier training, one can create a list for each feature containing the Gini decreases of cuts on this feature: The mean of this list is called the mean Gini decrease of this feature, and it has the advantage that it can be computed as a byproduct of the normal classifier training. For an overview over alternative possibilities for measuring variable importance, see (Guyon & Elisseeff, 2003). Note that there are natural groups of features that are sensibly computed simultaneously (e.g. the three different eigenvalues of a three-dimensional Hessian matrix): in this case it was decided to select or deselect these features as a whole, and to use the maximum mean Gini decrease in the group as a measure of the importance of the entire feature group.

Feature type / Feature scale	0.3	0.7	1	1.6	3.5	5	10
Gaussian smoothing (G)	3	2	3	4	4	3	3
Structure tensor (S)	2	2	4	4	4	4	3
Hessian of Gaussian (H)	2	2	3	4	5	5	5
Smoothed gradient magnitude (M)	1	1	1	2	3	2	2
Hessian of Gaussian eigenvalues (V)	1	2	2	3	4	5	5
Difference of Gaussians (D)	1	1	1	2	2	5	5
Structure tensor eigenvalues (E)	1	3	3	3	4	4	4
Laplacian of Gaussian (L)	1	1	1	1	3	5	4

Table 4.1. – Order in which the different image features were eliminated from the active feature set used by the ILASTIK software. For instance, a “1” means that the respective feature was eliminated already in the first iteration, and a “5” means that it was among the eight best features. The finally selected features are the best 20 ones that remained after the third iteration, i.e. all marked with either a “4” or a “5”.

Feature selection scheme Due to possible correlations between features, the variable importance of a particular feature depends on the other features that are used: specifically, it may gain importance once another feature is deselected. Hence the variable importance should be recomputed at times during the pruning of the feature set. As a compromise between evaluation time and accuracy, the following recursive feature elimination scheme was used, which is similar to the one employed by Menze et al. (2009):[9] The ILASTIK software was used for interactive labeling and classifier training of five image subvolumes of size 400 × 400 × 100 spaced at twenty minutes, and ten random forest classifiers (consisting of ten trees each) were trained on each volume using the same labels. Every random forest yielded a separate variable importance estimate for each feature, and the twelve feature groups with the smallest

[9] The main difference is that Menze et al. (2009) remove a certain fraction of the p % worst features in every iteration, while here the same absolute number of features is removed in each iteration.

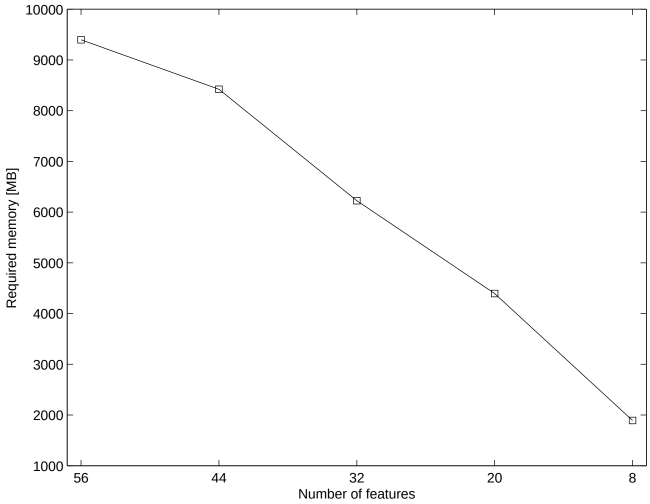
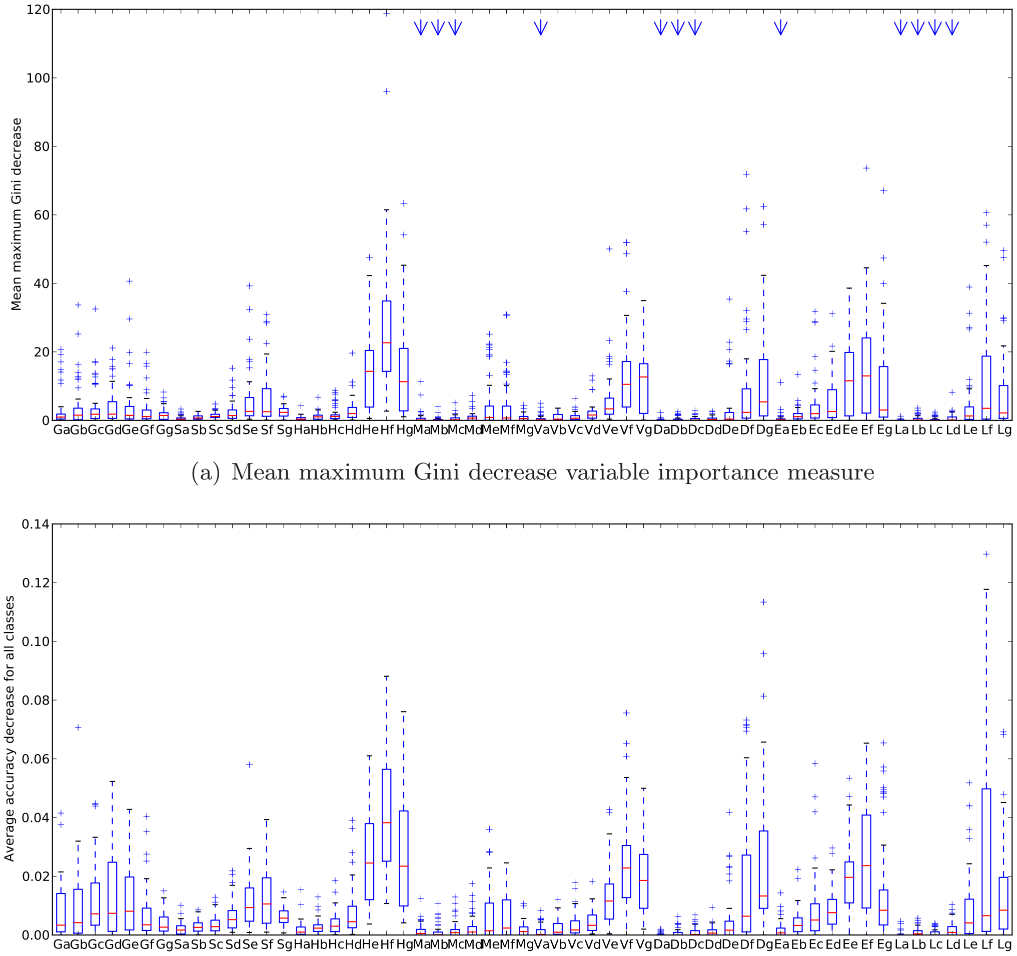


Figure 4.2. – ILASTIK memory requirements of the feature sets remaining at the end of the different iteration rounds for a $400 \times 400 \times 100$ data volume, assuming a 32-bit floating-point representation.

medians over all 50 estimates for the maximum mean Gini decrease were discarded, leaving 44 feature groups in the active feature set. The use of the variable-based variable importance instead of the Gini decrease would have been an obvious alternative, but both methods assign similar rankings to the features (compare Figs. 4.3(a) and 4.3(b)). This whole procedure was iterated three more times using the same labels for each image volume, with 32, 20 and 8 features remaining at the end of the different iteration rounds. Table 4.1 shows the order in which the features are pruned, and Fig. 4.2 shows the main memory requirements for the different feature sets. The final feature set was selected based on the quality of the segmentations obtained in the different iteration rounds, as computed from comparisons against manual ground truth. A threshold of 0.5 was used to generate a binary segmentation out of the continuous classifier outputs.

Segmentation evaluation After feature selection, the two competing segmentation methods (ILASTIK and RGC) are validated against the manual ground truth. Since it is impracticable to train the ILASTIK classifier on every single data volume separately, it was studied how well the trained classifiers generalize to close time points: Five image subvolumes were selected for interactive classifier training (at 1, 21, 41, 61 and 81 minutes after the start of the imaging series), and the trained



(a) Mean maximum Gini decrease variable importance measure

(b) Class-averaged permutation-based variable importance measure

Figure 4.3. – Comparison of two variable importance measures for the first feature selection iteration round. The boxplots show the distribution of 50 estimates computed for each feature (from ten random forest classifiers trained over five datasets). The arrows in Fig. 4.3(a) mark the features that were removed in this iteration due to having the lowest median value for the mean maximum Gini decrease. The upper-case letters in the x-axis labels encode the feature type (see Table 4.1), while the lower-case letters encode the feature scale: “a” stands for the smallest scale (0.3 voxel lengths), while “g” stands for the largest scale (10 voxel lengths).

---

classifiers were used for segmenting both these training subvolumes and separate testing subvolumes which had been acquired four minutes later (at 5, 25, 45, 65 and

85 minutes).[10] Again, the size of each subvolume was 400 × 400 × 100. While the current RGC implementation generates one single deterministic segmentation, the results of ILASTIK may vary depending on the number of labels and on the binarization threshold; furthermore there is an element of chance as the label placement is inherently subjective. In order to study the influence of these effects, 25 separate label sets were independently acquired for each training volume, and each was used for training a random forest classifier with 100 trees. Each set contained an equal number of foreground and background labels, and the total number of labels was varied systematically: five sets each had a total size of 40 (20 + 20), 80, 120, 160 and 200 labels.[11] This allowed to study both the effect of the label number (and hence of the user effort) on the segmentation quality on the segmentation quality, as well as the variability of the classifier for a fixed number of labels. Furthermore three different thresholds (0.25, 0.5 and 0.75) were used in order to transform the predicted probability maps into binary segmentations: One can expect that a low threshold leads to higher merge rates and recall while reducing split rates and precision (and that the opposite holds for a high threshold). The following segmentation quality measures were calculated:

- **Precision**: the percentage of segments in the computed segmentation that overlap with at least one nucleus in the ground truth.
- **Recall**: the percentage of nuclei in the ground truth that overlap with at least one segment in the computed segmentation.
- F_1 **measure**: The harmonic mean of precision and recall.[12]
- **Merge rate**: the percentage of segments in the computed segmentation that overlap with more than one nucleus in the ground truth.
- **Split rate**: the percentage of nuclei in the ground truth that overlap with more than one segment in the computed segmentation.
- **Dice index**: the ratio between the volume of the intersection between computed and ground-truth segmentation, and the average volume of these two segmentations.
- **Hausdorff distance**: the maximum distance of a point in the computed segmentation to the ground-truth segmentation.

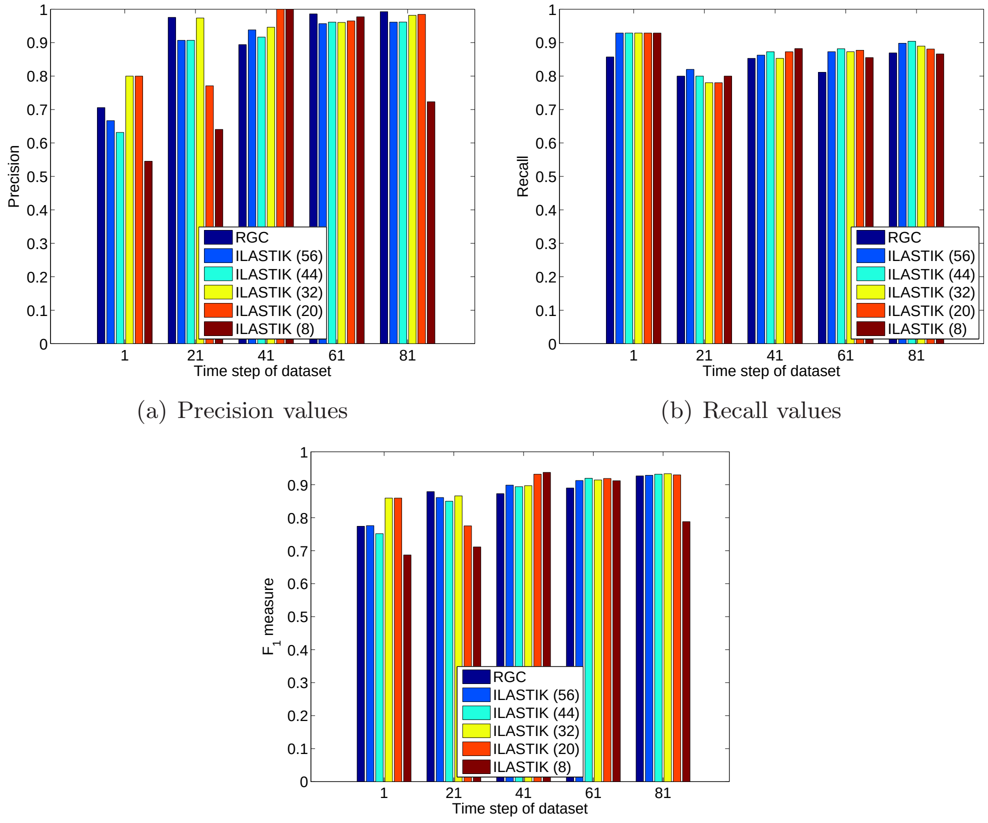
[10] Since generating the manual ground truth for these ten data volumes was already a time-consuming process, the same training volumes were used to select the number of features and to train the classifiers. It would have been methodologically preferable if a separate dataset would have been used for the feature selection.

[11] Typical human labeling speeds were 12–18 labels per minute.

[12] The subscript 1 indicates that there is a more general F_β measure, which allows to place higher weight on either precision or recall when forming the mean. For $\beta = 1$, both these measures are weighted equally.

The precision, recall and F_1 measure characterize the performance of the segmentation with respect to detecting precisely the cell nuclei as objects. Split and merge rate quantify the occurrence of oversegmentation and undersegmentation respectively, while Dice index and Hausdorff distance quantify the voxelwise accuracy of the segmentation.

4.4.3. Results for feature selection and evaluation



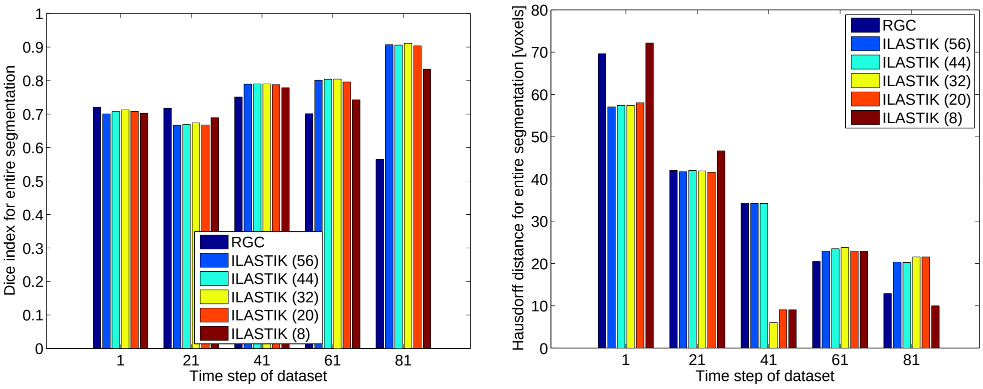
(a) Precision values

(b) Recall values

(c) F_1 measure values

Figure 4.4. – Object detection accuracy measures of the ILASTIK segmentations for different feature set sizes, with the RGC results shown for comparison purposes. Higher values correspond to better segmentations.

Feature selection results Figs. 4.4, 4.5 and 4.6 show the results of the different segmentation accuracy measures for the feature sets obtained after the different feature selection iterations. While there is little difference between the results obtained with 56, 44, 32 and 20 features, the segmentation quality degrades markedly when going down to eight features. Especially the precision declines considerably, which means that a classifier using only the final eight features erroneously detects a large number of false positive segments that do not correspond to actual cell nuclei. Fig. 4.5(a) shows that also the voxelwise accuracy (as measured by the Dice index) is impaired. On the other hand, using eight features favors oversegmentation over undersegmentation for the later time steps, which is advantageous for the later tracking procedure (compare Figs. 4.6(a) and 4.6(b)). Both the use of 20 and of eight features would hence be defensible: the decision fell on using the 20 features remaining after the fourth iteration round (i.e. those marked with “4” and “5” in Table 4.1), since this choice led to segmentation results that were nearly indistinguishable from using the entire feature set. It should be noted that 16 out of the 20 finally selected features were already among the 20 best features in the first iteration round (see Fig. 4.3(a)), so our procedure of gradually discarding only a small number of features per iteration step can be seen as somewhat overly cautious.

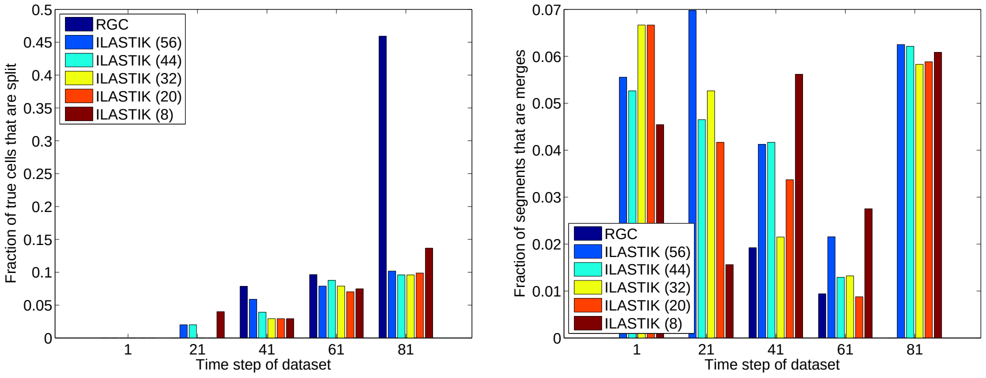


(a) Dice indices

(b) Hausdorff distances

Figure 4.5. – Voxelwise accuracy measures of the ILASTIK segmentations for different feature set sizes (in parentheses), with the RGC results shown for comparison purposes. For the left plot, higher values correspond to better segmentations, while it is the other way around for the right plot.

Selection of optimal binarization threshold Figs. 4.7, 4.8 and 4.9 show the effect of varying the ILASTIK binarization threshold on the F_1 measure and on the occurrence of oversegmentation and undersegmentation. Choosing an optimal threshold typically requires resolving a conflict between precision and recall, since raising the threshold typically increases the former and decreases the latter. The F_1 measure



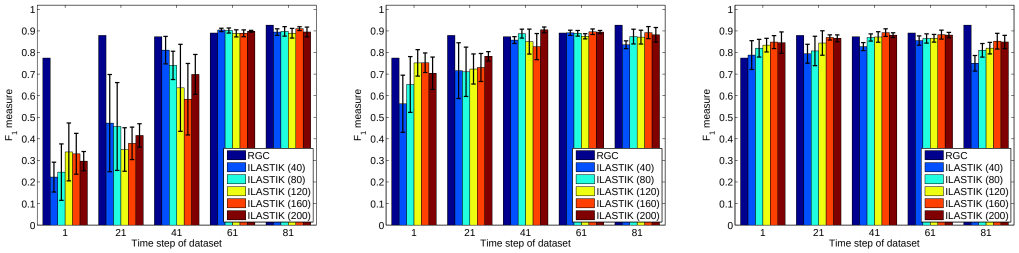
(a) Split rates

(b) Merge rates

Figure 4.6. – Oversegmentation and undersegmentation measures of the ILASTIK segmentations for different feature set sizes, with the RGC results shown for comparison purposes. Lower values correspond to better segmentations.

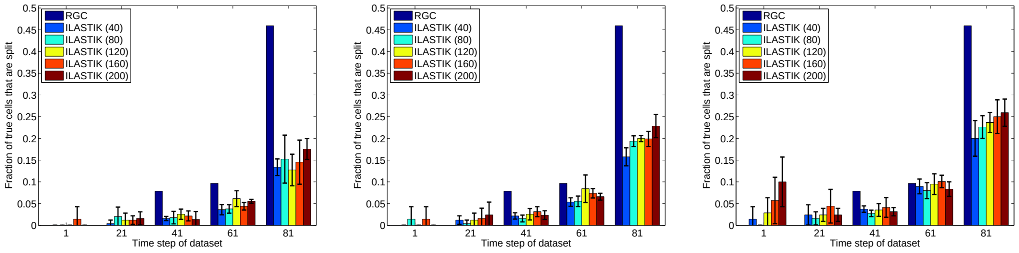
captures how well these two conflicting goals of good precision and recall can be met at the same time: as Fig. 4.8 shows, varying the threshold affects the recall more than the precision, so that higher F_1 measures are attained for a lower threshold. Having a high recall is also the more important goal than having a high precision, since extraneous nuclei may still be suppressed at a later stage during the tracking, while nuclei that are lost in the segmentation stage cannot be recovered later: this is a second argument for choosing a threshold of 0.25 rather than 0.5 or 0.75. On the other hand, lowering the threshold leads also to a decreased split rate (oversegmentation, Fig. 4.8) and an increased merge rate (undersegmentation, Fig. 4.9). Both these effects are nonnegligible, with the effect on merge rate being more pronounced. Since artificial splits can be tolerated better than artificial merges, this is an argument for selecting a high binarization threshold. Hence a compromise value of 0.5 was chosen, as in the preliminary studies on feature selection.

Comparison of training and test data For most performance measures, the differences between the training and the test datasets were negligible, with the largest differences occurring for the F_1 measures (Figs. 4.10(a) and 4.10(b)) and the recall (Figs. 4.10(e) and 4.10(f)): There the test values were slightly decreased compared to the training values, whereas no effect was noticeable for e.g. the precisions (Figs. 4.10(c) and 4.10(d)). Note that the absolute numbers should not be compared since the intrinsic difficulty of segmenting the training and the test datasets may be different: instead the relative performance of ILASTIK compared to the RGC segmentation should be used for the comparison, as the RGC method is not affected



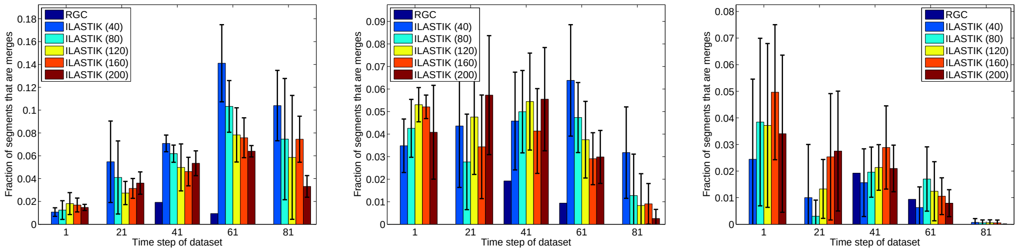
(a) Threshold = 0.25 (b) Threshold = 0.5 (c) Threshold = 0.75

Figure 4.7. – Effect of varying the segmentation threshold on the F_1 measure, for the training datasets. The bar lengths for the ILASTIK results indicate the mean values over the five separately trained classifiers with the same number of labels (in parentheses), and the error bars indicate the standard deviation.



(a) Threshold = 0.25 (b) Threshold = 0.5 (c) Threshold = 0.75

Figure 4.8. – Effect of varying the segmentation threshold on the occurrence of oversegmentation (split rate), for the training datasets.



(a) Threshold = 0.25 (b) Threshold = 0.5 (c) Threshold = 0.75

Figure 4.9. – Effect of varying the segmentation threshold on the occurrence of undersegmentation (merge rate), for the training datasets.

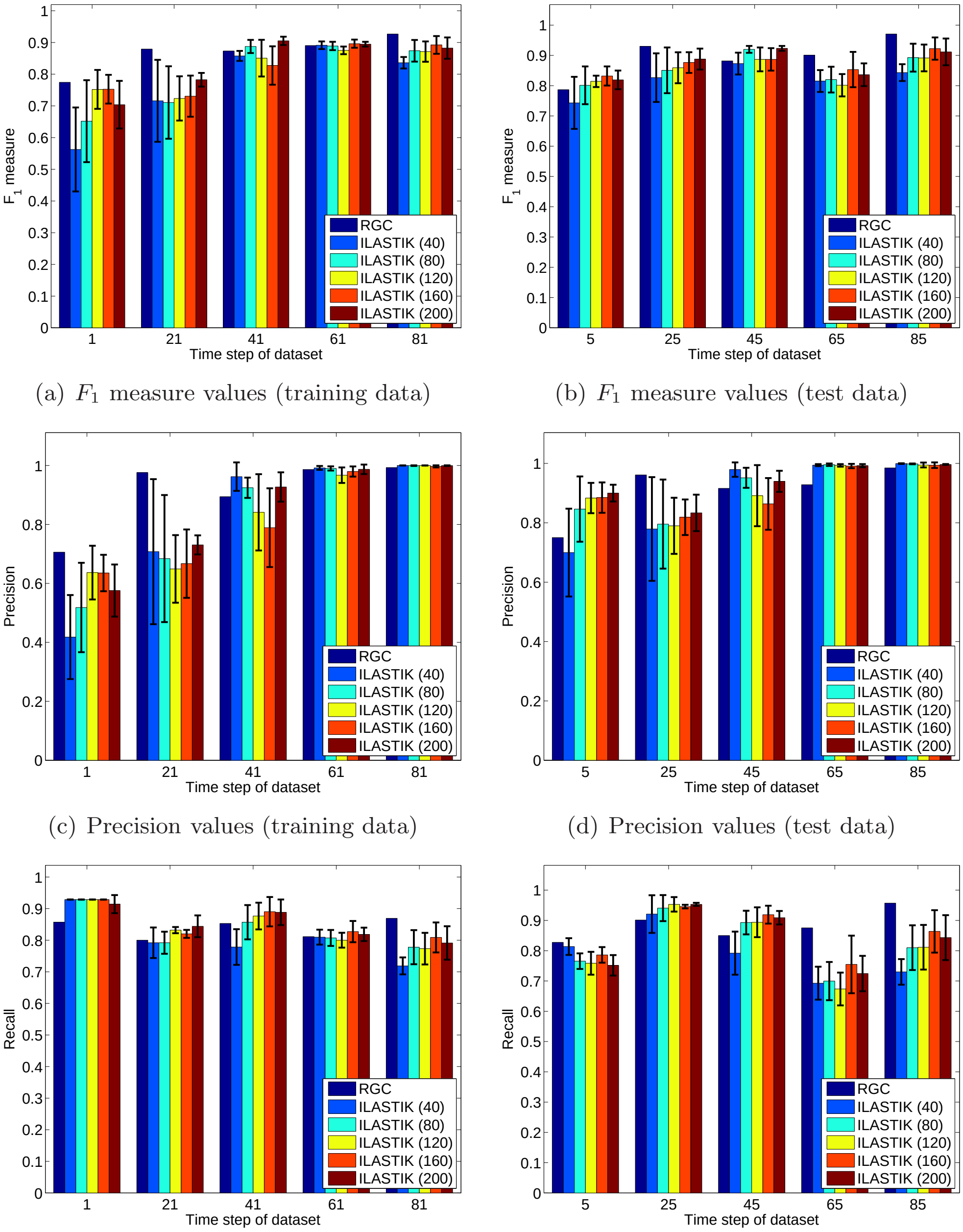
by the attribution to test or training data.[13] Note that the precision improves over time, while the recall is diminished: In the earlier time steps, the true nuclei have good contrast and are clearly detectable (high recall), while there are also numerous speckles that may be mistaken for nuclei by the segmentation method (poor precision). Most of these speckles disappear at later time steps, but at the same time the average nucleus contrast is reduced and several nuclei are not detected any longer.

Comparison between RGC and ILASTIK The differences between RGC and ILASTIK can be summarized as follows:

- If ILASTIK is trained with a sufficient number of training examples, both methods do not significantly differ in terms of precision, recall and F_1 measure. This holds both for segmenting the same dataset on which the classifiers are trained, and when applying the classifier to the data from a neighboring time step (Fig. 4.10). Typical values for the later time steps are 0.7–0.8 for the recall, > 0.99 for the precision and 0.8–0.9 for the F_1 measure.
- The voxelwise accuracy of both methods is also comparable, both when measured in terms of overlap volumes (Dice measure, Fig. 4.11(a)) and when measured in terms of surface distances (Hausdorff distance, Fig. 4.11(b)). Typical Dice indices for the later time steps lie between 0.55 and 0.65.
- RGC is more susceptible to oversegmentation (Fig. 4.8) and less susceptible to undersegmentation (Fig. 4.9), particularly for later time steps. In principle, the subsequent tracking is more robust towards oversegmentation than towards undersegmentation. However, the relative sizes of both effects should be taken into consideration: The merge rate of ILASTIK can be kept under 1 % using a sufficiently high number of training labels (Fig. 4.9(b)), while the split rate of RGC exceeds 35 % for the later time steps (Fig. 4.8). This places a heavy burden on the subsequent tracking and may cause tracking errors, by which true nuclei are matched with oversegmentation fragments.

In total, most differences are inconsiderable. If trained with a high number of labels (200 per data volume), ILASTIK has a slight advantage over RGC due to the markedly lower occurrence of oversegmentation: but this should be weighed against the increased human effort caused by the interactivity. Due to the suboptimal recall values, the subsequent tracking step needs to be robust towards false negatives, i.e. nuclei that are missed in some time step.

[13] This confounding effect could have been avoided by training the random forests on all ten datasets $(1, \ldots, 81, 5, \ldots, 85)$ and computing two segmentations for each of the datasets $1, \ldots, 81$, one with the classifier that was trained on the same dataset and one with the classifier that was trained on the dataset acquired four minutes later. However, this approach was not followed, as it would have been more time-consuming.



(a) F_1 measure values (training data)

(b) F_1 measure values (test data)

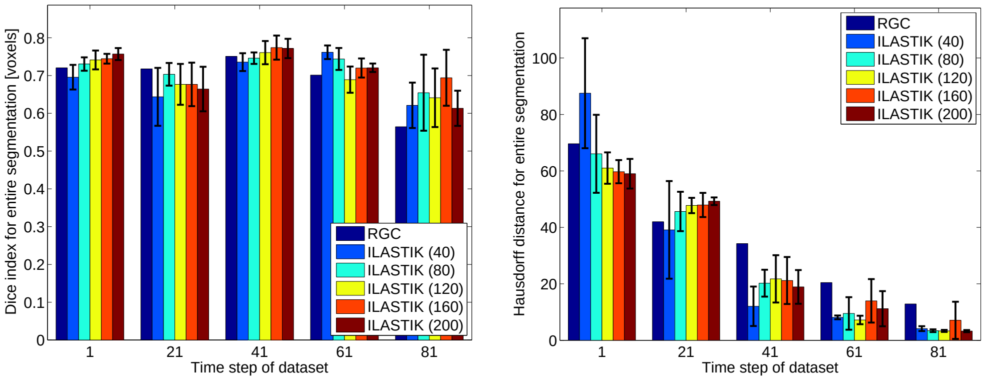
(c) Precision values (training data)

(d) Precision values (test data)

(e) Recall values (training data)

(f) Recall values (test data)

Figure 4.10. – Illustration of the difference between training and testing datasets for the counting accuracy measures, and comparison between the RGC and the ILASTIK segmentation. These graphics show the results for an ILASTIK segmentation threshold of 0.5.



(a) Dice indices

(b) Hausdorff distances

Figure 4.11. – Comparison of the RGC and the ILASTIK segmentation with respect to the voxelwise accuracy measures, for the training datasets. These graphics show the results for an ILASTIK segmentation threshold of 0.5. The decrease in Hausdorff distance is partially due to the higher cell density at the later time points. For the left-hand plot, higher values correspond to better segmentations, while the opposite holds for the right-hand plot.

4.5. Cell tracking by integer linear programming

4.5.1. Methodology

After generating segmented nucleus candidates by either of the methods discussed in the previous chapter, they have to be tracked over time in order to construct the cell lineage tree. This is achieved by finding the optimal joint association between nuclei for every pair of two subsequent time frames. The following events are possible:

1. Nucleus i *moves* to become nucleus j in the next time step ($i \to j$),
2. nucleus i *splits* into the nuclei j and k ($i \to j + k$),
3. nucleus i *disappears* in the next time step due to leaving the field of view, apoptosis or misdetection ($i \to \oslash$),
4. nucleus j from time step $t+1$ *appears* due to entering the field of view or being misdetected in the previous time step ($\oslash \to j$).

In order to rule out implausible events, children must be among the k nearest neighbors of their parent cell, and the parent-child distance must lie below a threshold $r_{\max}$. All these events have associated costs, which are chosen as follows (r_i denoting the center-of-mass position of nucleus i in voxel lengths):

$$\begin{aligned}
c_{i \to j} &= \|r_i - r_j\|^2 && (4.4)\\
c_{i \to j+k} &= \|r_i - r_j\|^2 + \|r_i - r_k\|^2 + c_{\text{Spl}} && (4.5)\\
c_{i \to \oslash} &= c_{\text{Dis}} && (4.6)\\
c_{\oslash \to j} &= c_{\text{App}} && (4.7)
\end{aligned}$$

Obviously, one could simply use also additional features for computing these costs. The constants c_{Spl}, c_{Dis} and c_{App} are chosen such that appearance and disappearance events are heavily penalized compared to splits and moves. Experimentally, the choice $k = 6$, $r_{\max} = 35$, $c_{\text{Spl}} = 100$, $c_{\text{Dis}} = c_{\text{App}} = 10000$ was found to yield acceptable results. Note that as long as c_{Dis} and c_{App} are above $2r_{\max}^2 + c_{\text{Spl}}$, their exact value does not matter since they preclude the disappearance or appearance of all cells which could be accounted for by some other event.

For each possible move (out of the set $\mathcal{M}$) and split (out of the set $\mathcal{S}$), define a binary variable x indicating whether this event takes place or not. Finding the optimum joint association is then a integer linear programming (ILP) problem:

$$\begin{aligned}
\min_x \; & \sum_{(i \to j) \in \mathcal{M}} x_{i \to j} (c_{i \to j} - c_{i \to \oslash} - c_{\oslash \to j}) \\
+ & \sum_{(i \to j+k) \in \mathcal{S}} x_{i \to j+k} (c_{i \to j+k} - c_{i \to \oslash} - c_{\oslash \to j} - c_{\oslash \to k}) \\
\text{s.t.} \; & \sum_{j:(i \to j) \in \mathcal{M}} x_{i \to j} + \sum_{j,k:(i \to j+k) \in \mathcal{S}} x_{i \to j+k} \leq 1 \quad \forall\, i, \\
& \sum_{i:(i \to j) \in \mathcal{M}} x_{i \to j} + \sum_{i,k:(i \to j+k) \in \mathcal{S}} x_{i \to j+k} \leq 1 \quad \forall\, j.
\end{aligned}$$

All cells not accounted for by either a split or a move are assumed to appear or disappear. Typically there are a few hundred thousands variables (one for each split or move) and a few ten thousands constraints (one for each nucleus in one of the two frames). Using a state-of-the-art ILP solver (ILOG CPLEX 12.2[14]), this problem can be solved to global optimality within less than a minute per frame pair on a standard desktop computer. Note that several frame pairs may trivially be processed in parallel. The ANN library[15] is used for efficiently extracting the k nearest potential child nuclei of each parent nucleus.

[14] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[15] http://www.cs.umd.edu/˜mount/ANN/

4.5.2. Experimental results

For a quantitative performance evaluation, the tracking was run on the first 25 data volumes of the same DSLM series that was used for the evaluation of the segmentation (see section 4.4.2), after the cell nuclei had been segmented by the RGC method. For these datasets, manual ground truth for the tracking was prepared based on their maximum intensity projection maps: this is a 2D image for which the gray value of the pixel with coordinates (x, y) is set to $\max_z I(x, y, z)$. This visualization technique is commonly used by biologists analyzing volumetric data, since the increased contrast simplifies the identification of nuclei, but the price is the loss of z information and the possible occurrence of occlusions. These shortcomings render the use of maximum intensity projections ineffectual for later time steps where the nucleus density becomes too high: hence the restriction to only the first 25 volumes. A cell lineage ground truth was constructed by manually tracking local intensity maxima in this 2D view over time.[16]

In order to use this tracking ground truth for the validation of the automated tracking results, the manually selected local intensity maxima had to be matched to the RGC segments. This was achieved by globally minimizing the sum of squared distances between the (x, y) positions of the placed marker and the intensity maximum of its assigned segment (with a distance cutoff of 20 voxel lengths). This optimization problem can be formulated as an ILP and solved as in section 4.5.1.[17] This matching is possibly error-prone due to possible occlusions and the disregard of the z dimension, but these imperfections are unavoidable given the origin of the ground truth.

	N_{gt}	N'_{gt}	N_{tr}	N'_{tr}	N_{ci}	Precision	Recall
Moves	3280	3006	3107	2941	2940	100.0 %	97.8 %
Splits	189	159	247	157	136	86.6 %	86.6 %
Appearances	2	2	107	67	1	1.5 %	50.0 %
Disappearances	4	3	181	72	2	2.8 %	33.3 %

Table 4.2. – Summary of statistics for the tracking evaluation.

We are interested in both the precision and the recall of the tracking, i.e. which percentage of detected events are actual, and which percentage of actual events are detected. Let N_{gt} denote the number of the different events (moves, splits, appearances, disappearances) in the ground truth, and N_{tr} denote the number of events

[16] The manual ground truth is courtesy by Bernhard X. Kausler.

[17] This particular problem can actually be solved more efficiently using e.g. the Kuhn-Munkres algorithm (Munkres, 1957), but the difference is irrelevant for the problem sizes encountered here.

found by the automated tracking. In order to disentangle the imperfections of the tracking from the imperfections of the segmentation, we discard all events for which the parent or one of the children could not be matched to an object in the other set: Hence N'_{gt} denotes the number of ground truth events for which all participating intensity maxima are matched to a segment, and N'_{tr} denotes the number of automated tracking events for which all participating segments are matched to an intensity maximum in the ground truth. If N_{ci} is the number of events that are correctly identified, then the precision is defined as $N_{\text{ci}}/N'_{\text{tr}}$ and the recall as $N_{\text{ci}}/N'_{\text{gt}}$. The results are summarized in Table 4.2. Note that precision and recall have similar values for the interesting events (moves and splits), while the precision exceeds the recall by far for the events that are caused by artifacts, i.e. appearances and disappearances.[18] This is unsurprising given the scarcity of these events in the ground-truth data, but indicates that incorrect appearances and disappearances are much too often introduced by the current tracking procedure.

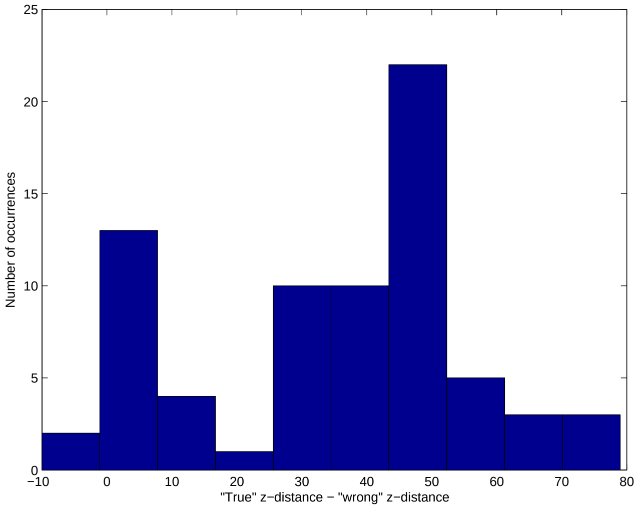


Figure 4.12. – Histogram showing the distribution of signed differences between the parent-child z distances for the ground-truth events and the events found by the automated tracking, aggregated over all events for which the ground truth and the automated tracking disagree. For this plot, the position of the maximum intensity voxel is used as the position of each segment.

It should be emphasized that the numbers in Table 4.2 are conservative estimates, i.e. lower bounds for the actual accuracy of the tracking: Firstly, since the ground truth is only derived from maximum intensity projections, it cannot handle occlusions properly. Fig. 4.12 shows that in most of the cases where the automated tracking

[18] Apoptosis normally does not occur at these early stage.

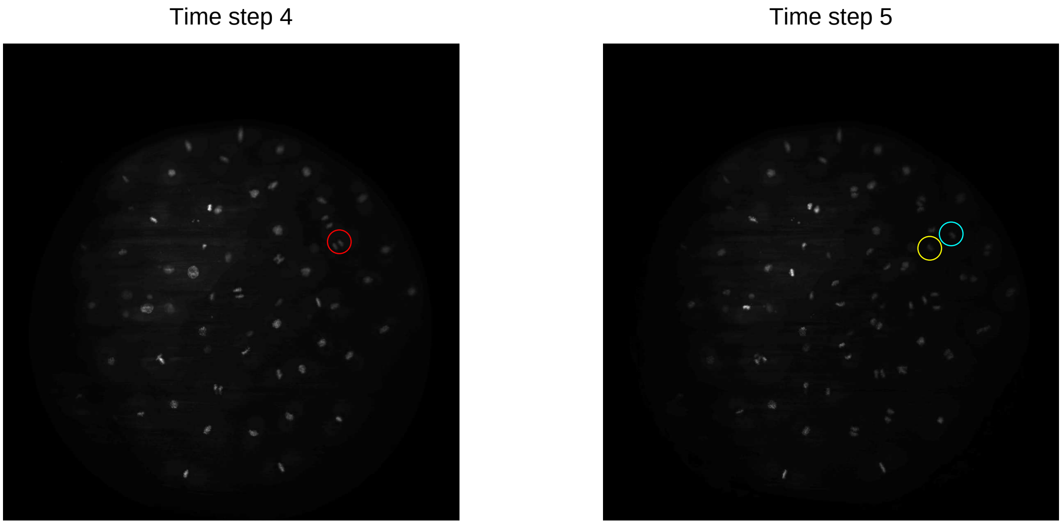


Figure 4.13. – Exemplary tracking error for which the daughter in the ground truth is well distinct from the daughter proposed by the automated tracking. The background image shows the maximum intensity projection, while the circles indicate the position of the parent nucleus (red), the daughter nucleus according to the ground truth (cyan) and the daughter nucleus proposed by the automated tracking (yellow). The circles are centered at the maximum intensity voxel of the respective segment.

---

and the ground truth disagree, the child segments according to the ground truth are more than 30 voxel lengths further away from the parent segment along the z direction than the child segments that are proposed by the automated tracking. This indicates that occlusion may be a problem, and that the ground truth may connect nuclei which have very different z positions. Secondly, mitoses typically span several time steps, and the exact time point of when a parent nucleus loses its identity and becomes two separate daughter nuclei is ill-defined. In Table 4.2, it is marked as a tracking error if the automated tracking places the split one minute earlier or later than in the ground truth, although such a variation has no biological relevance. Figs. 4.13 – 4.15 illustrate some typical tracking events that are marked as errors. Only rarely appears the daughter that is found by the automated tracking as clearly distinct from the ground-truth daughter in the maximum intensity projection, as in Fig. 4.13. More common is the case that these two segments lie in different z planes and occlude each other in the projection, as in Fig. 4.14. In some cases the daughter nucleus is tracked correctly, but an additional daughter is introduced by the tracking, changing a move into a split event (Fig. 4.15).

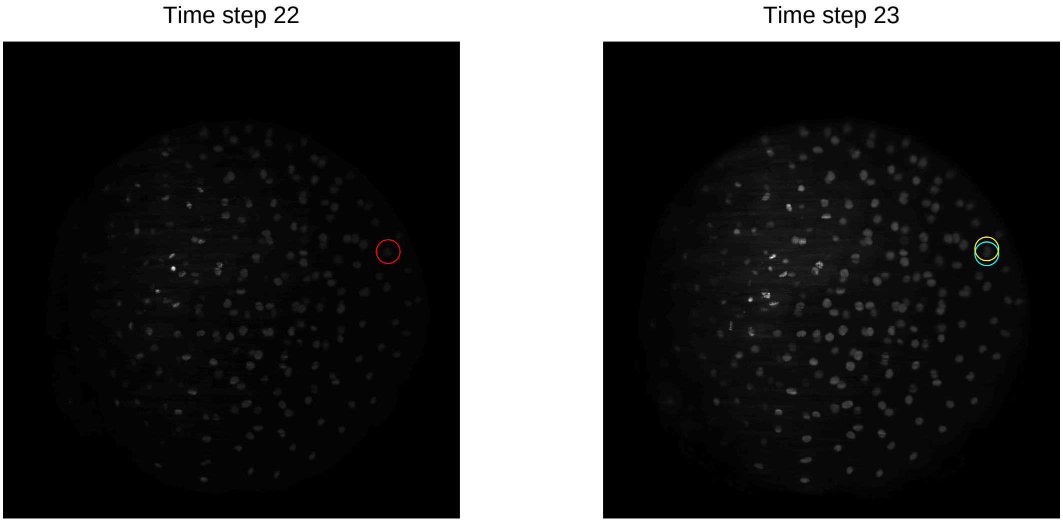


Figure 4.14. – Exemplary tracking error for which the two daughter candidate lie in different z-planes and occlude each other. Colors as in Fig. 4.13.

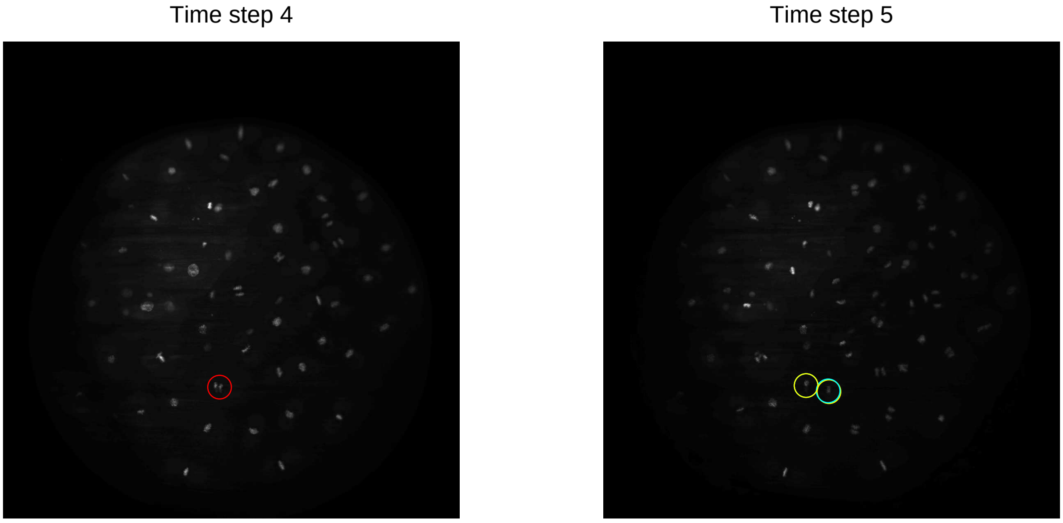


Figure 4.15. – Exemplary tracking error where a move event is mistaken for a split event, by introducing an additional parent-daughter track. Colors as in Fig. 4.13.

Chapter 5.

Final discussion and outlook

5.1. MRSI quantification with spatial context

In chapter 1, different methods for improving the accuracy of the simple single-voxel NLLS fit (AMARES) procedure were studied: it could be shown that imposing a Bayesian smoothness prior on the final fit parameters (GGMRF model) leads to small but significant improvements. However, improving the initialization step rather than the optimization step of NLLS fitting was found to give much higher gains in quantification accuracy, while requiring much less computation time. For most of the voxels, it was sufficient to optimize the initialization using only single-voxel information, but spatial smoothing of the initialization shifts was found to increase the robustness against pronounced spectral artifacts. However, the practical importance of the latter finding is dubious, as it only achieves significant improvements over the single-voxel initialization on artifact-ridden spectra that should not be used for diagnostic purposes anyway. Furthermore, the actual metabolite peak positions are typically stable across the entire volume: hence it may be sufficient to perform a global calibration of the fit model (for the whole scan) before fitting the single-voxel spectra.[1] As an additional caveat, the results in section 1.8 should be subjected to a double-blinded multi-rater evaluation before definite conclusions are drawn.

Further room for improvement is also in the MRSI datasets used for this study: a thorough experimental evaluation should comprise data from more probands and a higher variety of MR scanners, ideally from a multi-center study in the spirit of the INTERPRET project (Tate et al., 2006). It is particularly important to add pathological MRSI datasets coming from patients with e.g. tumor or multiple sclerosis, and to study whether the procedures can deal with the higher variability in these data. However, obtaining highly resolved spectral images (which are required for evaluations as performed in this chapter) from tumor patients may be difficult, since standard MR imaging protocols only comprise moderately resolved MRSI (if any),

[1] A plausible approach would be to use a robust estimator for the average minimum of RSS(f) over all voxels, such as the median.

which can be adequately quantified using existing quantification methods such as AMARES. Due to the long measurement time needed for MRSI scans and the stress that is thereby caused in the patients, acquiring such high-resolution measurements from highly diseased and mostly elderly people solely for the purpose of benchmarking quantification procedures may not be ethically defensible. Exploratory studies about the clinical applicability of high-field MR imaging may provide a way out and yield suitable high-resolution data as a by-product, since improved spatial resolution is one of the chief reasons for increasing magnetic field strength. It should be noted that pathologies mainly manifest themselves in the respective signal amplitudes, while the signal frequencies (on which the main smoothness assumptions are imposed both under the GGMRF and the GCInit model) mainly depend on magnetic field inhomogeneities and shimming problems which should be independent of biological phenomena such as tumors. Hence it is a plausible conjecture that the benefits of the GGMRF, SVInit and GCInit quantification schemes carry over to pathological data, but this needs to be checked experimentally.

A sensible extension of this study would be the comparison with a higher number of competing quantification schemes. Many concepts such as incorporating a semi-parametric baseline for nuisance signals (as in the AQSES approach by Poullet et al. (2007)) can be combined with both the GGMRF and the initialization procedures. However, the initialization optimization can also be combined with the QUEST approach of using experimental basis spectra for the fit (Ratiney et al., 2005), while GGMRF depends on an explicit parametric metabolite model. Particularly worthwhile would be the comparison with the "Bayesian learning" procedure provided in the LCModel software, as this software is commonly regarded as the current state of the art in MRSI quantification. Another interesting choice would be the proprietary quantification routines by the major MR scanner manufacturers such as Siemens or General Electric, which are typically used in clinical routine. As these are commercial products, they are expensive to obtain and their inner workings are opaque, which makes their use in methodological studies difficult. A comparison of only the final fit curves would not provide meaningful insights, as each software uses specific preprocessing steps, which are seldom reproducible by outsiders.

5.2. Software for MRSI analysis

Chapter 2 describes the first C++ library specifically designed for medical applications which allows principled comparison of classifier performance and significance testing. This will presumably help automated quality assessment and the conduction of clinical studies. While the absolute performance statistics of the single classifiers are most relevant for practical quality control in the clinic, the relative comparisons between different classifiers are interesting from a research-oriented point of view: for instance, they may answer the question which out-of-the-box classification techniques work best for the specific task of MRSI analysis, and can check whether newly proposed classification techniques give a significant advantage over established methods. Since quantification-based classifiers may easily be incorporated into the same framework, it will be possible to study the relative merits of quantification-based techniques as opposed to pattern recognition-based techniques on a large set of patient data.

The design of the library is deliberately restricted to single-voxel classifiers that predict the malignancy or signal quality of each voxel only based on the appearance of the spectrum inside this voxel, without considering the context of the surrounding spectra. The reason for this limitation is that automatic single-voxel classification is a mature technology whose efficacy has been proved in several independent studies, e.g. those by Tate et al. (2006), García-Gomez et al. (2009) or Menze et al. (2006). In contrast, classification with spatial context information has not yet been studied thoroughly: the two-dimensional conditional random field approach by Görlitz et al. (2007) is the only one in this direction to date. In that article, the authors achieve a promising, but moderate improvement in prediction accuracy over single-voxel classification on a simulated dataset (98.7 % compared to 98.2 %). However, it is yet far from clear which kinds of spatial context information may be beneficial for MRSI classification (2D neighborhoods, 3D neighborhoods, long-range context, comparison with registered MRI), and this question would have to be solved before a generic interface for such classifiers could be designed.

As next steps, the visualization and data reporting functionalities should be enhanced in order to improve usability: especially a more interpretable visualization of the statistical results may considerably benefit the medical users (for instance, plots of ROC curves could be provided, or the meaning of the AUC scores could be explained verbally). The clinical validation on 3 Tesla MRSI measurements of brain and prostate carcinomas is scheduled for the immediate future. Furthermore this software will eventually be integrated into the RONDO software platform for integrated tumor diagnostic and radiotherapy planning,[2] where it is planned to be a major workhorse for MRSI analysis. This will provide a good test for the usefulness of pattern recognition techniques in a clinical routine setting. Since the RONDO platform shall serve as a general-purpose tool for the radiological assessment of cancer, it must be tunable to different organ systems or measurement settings also by non-experts: hence the library is well-suited for this purpose.

[2] http://www.projekt-dot-mobi.de

5.3. Brain tumor segmentation based on multiple unreliable annotations

In chapter 3, graphical model formulations were introduced to the task of fusing noisy manual segmentations: e.g. the model by Raykar et al. (2009) had not been previously employed in this context, and it was found to improve upon simple logistic regression on the training data. However, these graphical models do not always have an advantage over simple baseline techniques: compare the results of the method by Warfield et al. (2004) to majority voting. Hybrid models combining the aspects of several models did not fare better than simple models. This ran contrary to the initial expectations, which were based on two assumptions: that different pixels have a different probability of being mislabeled, and that it is possible to detect these pixels based on the visual content (these pixels would be assigned high scores far away from the decision boundary). This may be an artifact of the time-constrained labeling experiment: if misclassifications can be attributed mostly to chance or carelessness rather than to ignorance or visual ambiguity, these assumptions obviously do not hold, and a uniform noise model as in the models by (Warfield et al., 2004) or (Raykar et al., 2009) should be used instead. It is furthermore not yet understood why the slight model change between hybrid models 1 / 2 and hybrid models 3 / 4 leads to the observed failure of inference. For the future, it should be checked if these effects arise from the use of an approximate inference engine or are inherent to these models: hence unbiased Gibbs sampling results should be obtained for comparison purposes, using e.g. the WinBUGS modelling environment (Lunn et al., 2000).

The use of simulated data for the evaluation is the main limitation of this approach, as simulations always present a simplification of reality and cannot account for all artifacts and other causes for image ambiguity that are encountered in real-world data. However, this limitation is practically unavoidable, since we are assessing the imperfections of the currently best clinical practice for the precise delineation of brain tumors, namely manual segmentation of MR images by human experts. This assessment requires a superior gold standard by which the human annotations may be judged, and this can only be obtained from an *in silico* ground truth. For animal studies, a possible alternative lies in sacrificing the animals and delineating the tumor on histological slices which can be examined with better spatial resolution. However, these kinds of studies are costly and raise ethical concerns. Additionally, even expert pathologists often differ considerably in their assessment of histological images (Giannini et al., 2001).

Better segmentations could presumably be achieved by two extensions: More informative features could be obtained by registration of the patient images to a brain atlas, e.g. in the spirit of (Schmidt et al., 2005). An explicit spatial regularization could be achieved by adding an MRF prior on the latent labels or scores, and employing a mean-field approximation (Zhang, 1992) to jointly estimate the optimum segmentation and the model parameters.

5.4. Live-cell microscopy image analysis

Chapter 4 compares two alternative approaches for segmenting cell nuclei in DSLM images of zebrafish embryos, a fully automated approach that uses prior knowledge about the nucleus shape, and an interactive approach that does not account for shape. It establishes that there is no clear advantage of one approach over the other: While the fully automated method is more susceptible to oversegmentation (erroneous fragmentation of nuclei), the semiautomated method rather encounters problems with undersegmentation (erroneous merging of distinct nuclei). This results hold even when the classifier that forms the core of the interactive classifier is applied to another image volume than the one it was trained on. Furthermore, the chapter presents a new method for tracking nuclei over time, which uses integer linear programming for finding a jointly optimal association between segments at different time points, and shows that it correctly assigns around 90 % of all matches, as compared against manual ground truth.

At the current stage, neither the segmentation nor the tracking is of sufficient quality to reconstruct an entire cell lineage of *D. rerio* over several hours. Since the accuracy of the tracking is limited by the accuracy of the segmentation, and since tracking errors accumulate over time, an accuracy of over 99.9 % would be required to keep the accuracy of the entire lineage tree over 90 %, when it is constructed from 100 time steps. However, the recall values of both segmentation and tracking lie below 90 %: hence their error rates still need to be reduced by a factor of 100. The two segmentation methods studied in this chapter (ILASTIK and RGC) do not significantly differ with respect to quality. However, the current accuracy may be sufficient to answer biological questions that are concerned with average values over ensembles of cells and do not need to account for the precise fate of every single cell: e.g. how the average cell motion speed changes over time, or how it is affected by different genetic mutations.

The gravest problem of the tracking procedure is the relatively high number of erroneous appearance and disappearance events. These cause discontinuities in the cell lineage which preclude the long-term analysis of cell fate. The reason lies in the limitations of the greedy frame-by-frame processing approach that is currently employed for the tracking. While this is well-suited for quickly reducing the problem size and finding all obvious associations between nuclei, it cannot handle artifacts or ambiguous cases where information from more time points needs to be used to find the correct matching. For instance, if one nucleus is missed by the segmentation in a particular time step, this leads to an appearance of its daughter nucleus. The optimal adoption of appearing nuclei by grandparent nuclei from more than one time step earlier can be found by solving a similar ILP problem as is used for the frame-by-frame tracking.

Other promising approaches for improvement include:

- **Additional features for cell matching**: Only the nucleus position is currently used for the frame-to-frame association, as this is an easily interpretable criterion which is also used by human annotators. To resolve ambiguous cases, it may be useful to include e.g. the segment volume or the average intensity, as these values can be expected to vary little over time within one nucleus.
- **Automated cell cycle phase classification**: Particularly apoptosis and cell division events can be identified with high confidence by biological experts, based on the characteristic appearance of the cells. For 2D cell microscopy images, Wang et al. (2008) were already able to predict the cell cycle phase based on shape descriptors with 99 % accuracy using statistical learning techniques. If similar accuracy rates could be achieved for the 3D DSLM images, reduced confusion between split and move events can be expected.
- **Use of motion information**: The current optimization objective is to achieve a low squared distance between the position of each parent nucleus at time t and the position(s) of its daughter(s) at time $t + 1$. Since the cells are moving, it is more plausible to extrapolate the trajectory of the parent nucleus to time $t + 1$, and to minimize the distance between the daughter position(s) and the extrapolated parent position instead. This could be achieved using stochastic motion modelling as in (Li et al., 2008b), but as a simpler alternative one could also fit a low-parametric model (e.g. a straight line) to the nucleus trajectory in the previous few time steps.
- **Interleaved segmentation and tracking**: The current segmentation uses no temporal information. However, for determining whether an ambiguous image patch belongs to the foreground or to the background, it may help to know whether or not a nucleus exists at the same position in the previous time step. This information could be incorporated e.g. by propagating the positions of the nuclei found at time t to the following time step (according to some motion model) and adding a potential to the regularized graph cut objective that encourages the new segments to lie close to the previous segments.

Further concern is warranted about the reliability of the ground truth that was used for assessing the accuracies of both segmentation and tracking. For the imagery that is analyzed here, both segmentation and tracking are ill-defined tasks. After time step 50, the foreground / background contrast becomes so low that the decision whether to label a particular patch as nucleus or background becomes highly subjective. Nor is it then a clear-cut decision whether two bright blobs belong to one single segment, or two separate segments. Possible shortcomings of the tracking ground truth in the presence of occlusions were already mentioned. Furthermore, if there are several potential daughter candidates with similar distances from a parent nucleus, there exists no reliable criterion even for human raters by which the correct association could be determined. A remedy may be the random expression of fluorescent markers in the spirit of the Brainbow project in neurobiology (Livet et al., 2007): if only a few nuclei emit fluorescence light at a particular wavelength, they are easier to identify at subsequent time points. Additionally, one could accept the fact that there are unavoidable uncertainties in the reconstruction of the cell lineage, and convey to the biologist user the information which parts of the cell lineage are certain and which are ambiguous.

List of Symbols and Expressions

Acronyms

AMM	Adaptive Mixture Model
AUC	Area Under Curve
Cho	Choline
CPD	Conditional Probability Distribution
Cre	Creatine
CSF	Cerebro-Spinal Fluid
CSI	Chemical Shift Imaging
CT	Computer Tomography
DCE	Dynamic Contrast Enhancement
DRF	Discriminative Random Field
DS	Data Set
DSLM	Digital Scanned Light Sheet Microscopy
DWI	Diffusion-Weighted Imaging
EM	Expectation Maximization
FCM	Fuzzy c-Means
FID	Free Induction Decay
FLAIR	Fluid-Attenuated Inverse Recovery
FOV	Field of View
GC	Graph Cut
Gd	Gadolinium
GFP	Green Fluorescent Protein

GGMRF	Generalized Gaussian Markov Random Field
GM	Gray Matter
GMM	Gaussian Mixture Model
GMRF	Gaussian Markov Random Field
GPU	Graphical Processing Unit
HSVD	Hankel Singular Value Decomposition
ICM	Iterated Conditional Modes
ILASTIK	Interactive Learning and Segmentation Toolkit
IR	Inversion Recovery
KL	Kullback-Leibler
LSFM	Light sheet-based fluorescence microscopy
MAP	Maximum a posteriori
MCMC	Markov Chain Monte Carlo
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MR	Magnetic Resonance
MRSI	Magnetic Resonance Spectroscopic Imaging
MRS	Magnetic Resonance Spectroscopy
NAA	*N*-acetylaspartate
NNPM	Nearest Neighbor Pattern Matching
PCR	Principal Components Regression
PDE	Partial Differential Equation
PD	Protium Density
PET	Positron Emission Tomography
p.f.	post fertilisationem
ppm	parts per million
PRESS	Point-Resolved Spectroscopy

RBF	Radial Basis Function
RF	Random Forest
RFr	Radio-Frequency
RGC	Regularized Graph Cut
ROC	Receiver Operating Characteristic
RR	Ridge Regression
RSS	Residual Sum of Squares
SE	Spin Echo
SGF	Statistical Geometric Features
SPECT	Single-Photon Emission Computer Tomography
SP	Spatial regularization
SQ	Signal Quality
STAPLE	Simultaneous Truth and Performance Level Estimation
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVRF	Support Vector Random Field
SV	Single Voxel
SWA	Segmentation by Weighted Aggregation
TE	Echo Time
TR	Repetition Time
TU	totally unimodular
VC	Voxel Class
VMP	Variational Message Passing
WM	White Matter

Greek Symbols

γ	Gyromagnetic ratio
μ	Magnetic moment

Latin Symbols

B_0 / B_1	Static longitudinal / oscillating transverse magnetic field
f	Frequency
I	Nuclear spin quantum number
M_0	Equilibrium magnetization
$M_\perp$ / $M_\|$	Transverse / longitudinal magnetization
T_1 / T_2	Spin-lattice / spin-spin relaxation time

List of Figures

List of Tables

Bibliography

T. Achterberg, T. Koch, A. Martin (2006). "MIPLIB 2003." Operations Research Letters, 34(4), 361–372. The current state of which problems are solved can be found at `http://miplib.zib.de/miplib2003.php`.

O. Al-Kofahi, R. Radke, S. Goderie, et al. (2006). "Automated Cell Lineage Construction." Cell Cycle, 5(3), 327–335.

Y. Al-Kofahi, W. Lassoued, W. Lee, et al. (2010). "Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images." IEEE Transactions on Biomedical Engineering, 57(4), 841–852.

C. Andrieu, N. De Freitas, A. Doucet, et al. (2003). "An introduction to MCMC for machine learning." Machine Learning, 50(1), 5–43.

S. Arya, D. Mount, N. Netanyahu, et al. (1998). "An Optimal Algorithm for Approximate Nearest Neighbor Searching." Journal of the ACM, 45, 891–923.

B. Aspvall, R. Stone (1980). "Khachiyan's Linear Programming Algorithm." Journal of Algorithms, 1, 1–13.

J. Attenberg, K. Weinberger, A. Dasgupta, et al. (2009). "Collaborative Email-Spam Filtering with Consistently Bad Labels using Feature Hashing." In: Conference on Email and Anti-Spam (CEAS).

G. Bakir, R. Hofmann, B. Schölkopf, et al. (eds.) (2007). Predicting Structured Data. MIT Press.

A. Bandos, H. Rockette, D. Gur (2007). "Exact Bootstrap Variances of the Area Under ROC curve." Communications in Statistics: Theory and Methods, 36, 2443–2461.

Y. Bao, A. Maudsley (2007). "Improved Resolution for MR Spectroscopic Imaging." IEEE Transactions on Medical Imaging, 26(5), 686–695.

Z. Bao, J. Murray, T. Boyle, et al. (2006). "Automated cell lineage tracing in *Caenorhabditis elegans*." Proceedings of the National Academy of Sciences, 103(8), 2707–2712.

Y. Bengio (2009). "Learning Deep Architectures for AI." Foundations and Trends® in Machine Learning, 2(1), 1–127.

Y. Bengio, Y. Grandvalet (2004). "No Unbiased Estimator of the Variance of K-Fold Cross-Validation." Journal of Machine Learning Research, 5, 1089–1105.

J. Besag (1986). "On the statistical analysis of dirty pictures." Journal of the Royal Statistical Society B (Methodological), 48(3), 259–302.

C. Bishop (1994). "Neural networks and their applications." Reviews of Scientific Instruments, 65(6), 1803–1832.

H. Bodlaender (1992). "A Tourist Guide through Treewidth." Tech. Rep. RUU-CS-92-12, Utrecht University.

H. Bodlaender, A. Koster (2010a). "Treewidth computations I: Upper bounds." Information and Computation, 208(3), 259–275.

H. Bodlaender, A. Koster (2010b). "Treewidth Computations II: Lower Bounds." Tech. Rep. UU-CS-2010-022, Utrecht University.

P. Bottomley (1987). "Spatial Localization in NMR Spectroscopy *in Vivo*." Annals of the New York Academy of Sciences, 508, 333–348.

S. Bouman, K. Sauer (1993). "A generalized Gaussian image model for edge-preserving MAP estimation." IEEE Transactions on Image Processing, 2(3), 296–310.

P. Bourgine, R. Čunderlık, O. Drblıková-Stašová, et al. (2010). "4D embryogenesis image analysis using PDE methods of image processing." Kybernetika, 46(2), 226–259.

Y. Boykov, V. Kolmogorov (2004). "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision." IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9), 1124–1137.

Y. Boykov, O. Veksler, R. Zabih (2001). "Fast Approximate Energy Minimization via Graph Cuts." IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), 1222–1239.

L. Breiman (1996). "Out-of-Bag Estimation." Tech. Rep., UC Berkeley.

L. Breiman (2001). "Random Forests." Machine Learning, 45(1), 5–32.

J. Broderick, S. Narayan, M. Gaskill, et al. (1996). "Volumetric measurement of multifocal brain lesions." Journal of Neuroimaging, 6, 36–43.

W. Buntine (1994). “Operations for Learning with Graphical Models.” Journal of Artificial Intelligence Research, 2, 159–225.

C. Burges (1998). “A Tutorial on Support Vector Machines for Pattern Recognition.” Data Mining and Knowledge Discovery, 2(2), 121–167.

R. Caruana, N. Karampatziakis, A. Yessenalina (2008). “An Empirical Evaluation of Supervised Learning in High Dimensions.” In: International Conference on Machine Learning (ICML), 96 – 103.

R. Caruana, A. Niculescu-Mizil (2006). “An Empirical Comparison of Supervised Learning Algorithms.” In: International Conference on Machine Learning (ICML), 161–168.

J. Cates, A. Lefohn, R. Whitaker (2004). “GIST: an interactive, GPU-based level set segmentation tool for 3D medical images.” Medical Image Analysis, 8(3), 217–231.

J. Cates, R. Whitaker, G. Jones (2005). “Case study: an evaluation of user-assisted hierarchical watershed segmentation.” Medical Image Analysis, 9(6), 566–578.

M. Chalfie, Y. Tu, G. Euskirchen, et al. (1994). “Green fluorescent protein as a marker for gene expression.” Science, 263(5148), 802–805.

A. Chan, A. Lau, A. Pirzkall, et al. (2004). “Proton magnetic resonance spectroscopy imaging in the evaluation of patients undergoing gamma knife surgery for Grade IV glioma.” Journal of Neurosurgery, 101, 467–475.

C. Chang, C. Lin (2001). “LIBSVM: a library for support vector machines.” Software available at http://www.csie.ntu.tw/ cjlin/libsvm.

O. Chapelle, B. Schölkopf, A. Zien (eds.) (2006). Semi-Supervised Learning. MIT Press.

S. Cho, M. Kim, H. Kim, et al. (2001). “Chronic hepatitis: in vivo proton MR spectroscopic evaluation of the liver and correlation with histopathologic findings.” Radiology, 221(3), 740–746.

P. Clifford (1990). “Markov random fields in statistics.” In: G. Grimmett, D. Welsh (eds.), Disorder in Physical Systems. A Volume in Honour of John M. Hammersley. Oxford University Press, Oxford.

D. Cobzas, N. Birkbeck, M. Schmidt, et al. (2007). “3D variational brain tumor segmentation using a high dimensional feature set.” In: International Conference on Computer Vision (ICCV 2007).

B. Cohen, E. Knopp, H. Rusinek, et al. (2005). “Assessing Global Invasion of Newly Diagnosed Glial Tumors with Whole-Brain Proton MR Spectroscopy.” American Journal of Neuroradiology, 26, 2170–2177.

T. Coleman, Y. Li (1996). “An interior trust-region approach for nonlinear minimization subject to bounds.” SIAM Journal on Optimization, 6, 418–445.

J. Colinge, K. Bennett (2007). PLoS Computational Biology, 3(7), e114.

O. Commowick, S. Warfield (2010). “Incorporating Priors on Expert Performance Parameters for Segmentation Validation and Label Fusion: A Maximum a Posteriori STAPLE.” In: T. Jiang, et al. (eds.), Proceedings of the 13th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2010), Part III, *Lecture Notes on Computer Science*, vol. 6363/2010, 25–32. Springer, Berlin.

R. D. Cook, C.-L. Tsai, B. C. Wei (1986). “Bias in nonlinear regression.” Biometrika, 73(3), 615–623.

J. Corso, E. Sharon, S. Dube, et al. (2008). “Efficient multilevel brain tumor segmentation with integrated bayesian model classification.” IEEE Transactions on Medical Imaging, 27(5), 629–640.

J. Corso, E. Sharon, A. Yuille (2006). “Multilevel segmentation and integrated Bayesian model classification with an application to brain tumor segmentation.” In: Medical Image Computing and Computer-Assisted Interventions (MICCAI), *Lecture Notes in Computer Science*, vol. 4191, 790–798.

J. Corso, A. Yuille, N. Sicotte, et al. (2007). “Detection and Segmentation of Pathological Structures by the Extended Graph-Shifts Algorithm.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 4791/2007, 985–993. Springer.

A. Croitor Sava, D. Sima, J. Poullet, et al. (2009). “Exploiting spatial information to estimate metabolite levels in 2D MRSI of heterogeneous brain lesions.” Tech. Rep. ESAT-SISTA 09-182, Katholieke Universiteit Leuven.

S. Dager, N. Oskin, T. Richards, et al. (2008). “Research Applications of Magnetic Resonance Spectroscopy (MRS) to Investigate Psychiatric Disorders.” Topics in Magnetic Resonance Imaging, 19(2), 81–96.

G. Dantzig (1949). “Programming of Interdependent Activities II: Mathematical Model.” Econometrica, 17(3/4), 200–211.

F. S. de Edelenyi, C. Rubin, F. Estève, et al. (2000). “A new approach for analyzing proton magnetic resonance spectroscopic images of brain tumors: nosologic images.” Nature Medicine, 6, 1287–1289.

R. de Graaf (2008). In Vivo NMR Spectroscopy: Principles and Techniques. Wiley, New York.

L. DeAngelis, J. Loeffler, A. Mamelak (2007). “Primary and metastatic brain tumors.” In: R. Pazdur, L. Wagman, K.A.Camphausen, et al. (eds.), Cancer Management: A Multidisciplinary Approach. CMP Healthcare Media, San Francisco CA.

J. Debnam, L. Ketonen, L. Hamberg, et al. (2007). “Current Techniques Used for the Radiological Assessment of Intracranial Neoplasms.” Archives of Pathology and Laboratory Medicine, 131, 252–260.

A. Dempster, N. Laird, D. Rubin, et al. (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” Journal of the Royal Statistical Society. Series B (Methodological), 39(1), 1–38.

J. Demšar (2006). “Statistical comparisons of classifiers over multiple data sets.” Journal of Machine Learning Research, 7, 1–30.

T. Dietterich (1998). “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.” Neural Computation, 10, 1895–1923.

W. Dou, S. Ruan, Y. Chen, et al. (2007). “A framework of fuzzy information fusion for the segmentation of brain tumor tissues on MR images.” Image and Vision Computing, 25(2), 164–171.

M. Droske, B. Meyer, M. Rumpf, et al. (2005). “An adaptive level set method for interactive segmentation of intracranial tumors.” Neurological Research, 27(4), 363–370.

A. Dufour, V. Shinin, S. Tajbakhsh, et al. (2005). “Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces.” IEEE Transactions on Image Processing, 14(9), 1396–410.

J. Duncan, N. Ayache (2000). “Medical Image Analysis: Progress over Two Decades and the Challenges Ahead.” IEEE Transactions on Pattern Recognition and Machine Intelligence, 22(1), 85–106.

W. Edelstein, G. Glover, C. Hardy, et al. (1986). “The Intrinsic Signal-to-Noise Ratio in NMR Imaging.” Magnetic Resonance in Medicine, 3, 604–618.

R. Fabbri, L. D. F. Costa, J. Torelli, et al. (2008). “2D Euclidean Distance Transform Algorithms: A Comparative Survey.” ACM Computing Surveys, 40(1), 2:1 – 2:44.

A. Farhangfar, R. Greiner, C. Szepesvári (2009). “Learning to Segment from a Few Well-Selected Training Images.” In: International Conference on Machine Learning (ICML), 305–312.

T. Fawcett (2006). “An introduction to ROC analysis.” Pattern Recognition Letters, 27(8), 861–874.

L. M. Fletcher-Heath, L. O. Hall, D. B. Goldgof, et al. (2001). “Automatic segmentation of non-enhancing brain tumors in magnetic resonance images.” Artificial Intelligence in Medicine, 21(1-3), 43–63.

Y. Freund, R. Schapire (1999). “A Short Introduction to Boosting.” Journal of the Japanese Society for Artificial Intelligence, 14(5), 771–780.

M. Frigo, S. Johnson (2005). “The Design and Implementation of FFTW3.” Proceedings of the IEEE, 93(2), 216–231.

J. García-Gomez, J. Luts, M. Julià-Sapé, et al. (2009). “Multiproject-multicenter evaluation of automatic brain tumor classification by magnetic resonance spectroscopy.” Magnetic Resonance Materials in Physics, Biology and Medicine, 22, 5–18.

A. E. Gelfand, A. F. Smith (1990). “Sampling-Based Approaches to Calculating Marginal Densities.” Journal of the American Statistical Association, 85(410), 398–409.

S. Geman, D. Geman (1984). “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 721–741.

A. Genovesio, T. Liedl, V. Emiliani, et al. (2006). “Multiple Particle Tracking in 3-D+t Microscopy: Method and Application to the Tracking of Endocytosed Quantum Dots.” IEEE Transactions on Image Processing, 15(5), 1062–1070.

D. Gering (2003). “Diagonalized Nearest Neighbor Pattern Matching for Brain Tumor Segmentation.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 2879/2003, 670–677. Springer.

D. Gering, W. Grimson, R. Kikinis (2002). “Recognizing Deviations from Normalcy for Brain Tumor Segmentation.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 2488/2002, 388–395. Springer.

C. Giannini, B. Scheithauer, A. Weaver, et al. (2001). “Oligodendrogliomas: reproducibility and prognostic value of histologic diagnosis and grading.” Journal of Neuropathology & Experimental Neurology, 60(3), 248.

P. Gibbs, D. Buckley, S. Blackband, et al. (1996). “Tumour volume determination from MR images by morphological segmentation.” Physics in Medicine and Biology, 13, 2437–2446.

W. Gilks, A. Thomas, D. Spiegelhalter (1994). “A language and program for complex Bayesian modelling.” The Statistician, 43, 169–178.

R. Gillies, D. Morse (2005). “In Vivo Magnetic Resonance Spectroscopy in Cancer.” Annual Review of Biomedical Engineering, 7, 287–326.

G. Golub, M. Heath, G. Wahba (1979). “Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter.” Technometrics, 21(2), 215–223.

G. Golub, V. Pereyra (2003). “Separable nonlinear least squares: the variable projection method and its applications.” Inverse Problems, 19, R1–R26.

H. González-Vélez, M. Mier, M. Julià-Sapé, et al. (2009). “HealthAgents: distributed multi-agent brain tumor diagnosis and prognosis.” Applied Intelligence, 30, 191–202.

L. Görlitz, B. H. Menze, M.-A. Weber, et al. (2007). “Semi-Supervised Tumor Detection in Magnetic Resonance Spectroscopic Images Using Discriminative Random Fields.” In: Proceedings of the DAGM 2007, *Lecture Notes in Computer Science*, vol. 4713/2007, 224–233.

V. Govindaraju, K. Young, A. Maudsley (2000). “Proton NMR chemical shifts and coupling constants for brain metabolites.” NMR in Biomedicine, 13, 129–153.

Y. Grandvalet, Y. Bengio (2006). “Hypothesis Testing for Cross-Validation.” Tech. Rep. TR 1285, Département d’Informatique et Recherche Opérationelle, University of Montréal.

I. Guyon, A. Elisseeff (2003). “An Introduction to Variable and Feature Selection.” Journal of Machine Learning Research, 3, 1157 – 1182.

G. Hagberg (1998). “From magnetic resonance spectroscopy to classification of tumors: A review of pattern recognition methods.” NMR in Biomedicine, 11(4–5), 148–156.

R. Harmouche, L. Collins, D. Arnold, et al. (2006). “Bayesian MS Lesion Classification Modeling Regional and Local Spatial Information.” In: 18th International Conference on Pattern Recognition (ICPR).

T. Hastie, R. Tibshirani, J. Friedman (2009). The Elements of Statistical Learning. Springer, New York.

R. He, P. Narayana (2002). "Automatic delineation of Gd enhancements on magnetic resonance images in multiple sclerosis." Medical Physics, 29, 1536–1546.

A. Henning, A. Fuchs, J. Murdoch, et al. (2009). "Slice-selective FID acquisition, localized by outer volume suppression (FIDLOVS) for ^{1}H-MRSI of the human brain at 7 T with minimal signal loss." NMR in Biomedicine, 22(7), 683–696.

S. Ho, E. Bullitt, G. Gerig (2002). "Level-set evolution with region competition: automatic 3-D segmentation of brain tumors." In: 16th International Conference on Pattern Recognition (ICPR).

S. Hojjatoleslami, F. Kruggel, D. Von Cramon (1998). "Segmentation of white matter lesions from volumetric MR images." In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 1496/1998, 52–61. Springer.

K. Iftekharuddin, M. Islam, J. Shaik, et al. (2005). "Automatic brain tumor detection in MRI: methodology and statistical validation." In: Medical Imaging 2005: Image Processing, *Proceedings of SPIE*, vol. 5747, 2012–2022.

H. Ishikawa (2003). "Exact optimization for Markov random fields with convex priors." IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10), 1333–1336.

C. Jiang, X. Zhang, W. Huang, et al. (2004). "Segmentation and Quantification of Brain Tumor." In: IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS).

T. Kanda, K. Sullivan, G. Wahl (1998). "Histone-GFP fusion protein enables sensitive analysis of chromosome dynamics in living mammalian cells." Current Biology, 8(7), 377.

N. Karayiannis, P. Pai (1999). "Segmentation of magnetic resonance images using fuzzy algorithms for learning vector quantization." IEEE transactions on medical imaging, 18(2), 172–180.

N. Karmarkar (1984). "A New Polynomial-Time Algorithm for Linear Programming." Combinatorica, 4(4), 373–395.

R. Karp (1972). "Reducibility Among Combinatorial Problems." In: J. T. R.E. Miller (ed.), Complexity of Computer Computations, 85–103. Plenum, New York.

R. Kass, A. Raftery (1995). "Bayes Factors." Journal of the American Statistical Association, 90(430), 773–795.

F. Kaster, S. Kassemeyer, B. Merkel, et al. (2010a). "An object-oriented library for systematic training and comparison of classifiers for computer-assisted tumor diagnosis from MRSI measurements." In: Bildverarbeitung für die Medizin 2010 – Algorithmen, Systeme, Anwendungen, 97–101.

F. Kaster, B. Kelm, C. Zechmann, et al. (2009). "Classification of Spectroscopic Images in the DIROlab Environment." In: World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany, *IFMBE Proceedings*, vol. 25/V, 252–255.

F. Kaster, B. Menze, M.-A. Weber, et al. (2011). "Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations." In: B. Menze, et al. (eds.), MICCAI 2010 Workshop on Medical Computer Vision (MCV), *Lecture Notes in Computer Science*, vol. 6533, 74–85. Springer, Heidelberg.

F. Kaster, B. Merkel, O. Nix, et al. (2010b). "An object-oriented library for systematic training and comparison of classifiers for computer-assisted tumor diagnosis from MRSI measurements." Computer Science – Research and Development, in press.

R. Kates, D. Atkinson, M. Brant-Zawadzki (1996). "Fluid-attenuated Inversion Recovery (FLAIR): Clinical Prospectus of Current and Future Applications." Topics in Magnetic Resonance Imaging, 8(6), 389–396.

M. Kaus, S. Warfield, A. Nabavi, et al. (1999). "Segmentation of meningiomas and low grade gliomas in MRI." In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 1679/1999, 1–10. Springer.

M. Kaus, S. Warfield, A. Nabavi, et al. (2001). "Automated Segmentation of MR Images of Brain Tumors." Radiology, 218(2), 586–591.

S. Keevil (2006). "Spatial localization in nuclear magnetic resonance spectroscopy." Physics in Medicine and Biology, 51, R579 – R636.

P. Keller, A. Schmidt, J. Wittbrodt, et al. (2008). "Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy." Science, 322(5904), 1065–1069.

P. Keller, E. Stelzer (2008). "Quantitative in vivo imaging of entire embryos with Digital Scanned Laser Light Sheet Fluorescence Microscopy." Current Opinion in Neurobiology, 18(6), 624–632.

B. Kelm (2007). Evaluation of Vector-Valued Clinical Image Data Using Probabilistic Graphical Models: Quantification and Pattern Recognition. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg.

B. Kelm, F. Kaster, A. Henning, et al. (2011). "Using Spatial Prior Knowledge in the Spectral Fitting of Magnetic Resonance Spectroscopic Images." NMR in Biomedicine, accepted.

B. Kelm, B. Menze, T. Neff, et al. (2006). "CLARET: a tool for fully automated evaluation of MRSI with pattern recognition methods." In: H. Handels, J. Ehrhardt, A. Horsch, et al. (eds.), Bildverarbeitung für die Medizin 2006 – Algorithmen, Systeme, Anwendungen, 51–55.

B. Kelm, B. Menze, O. Nix, et al. (2009). "Estimating Kinetic Parameter Maps from Dynamic Contrast-Enhanced MRI using Spatial Prior Knowledge." IEEE Transactions on Medical Imaging, 28(10), 1534 – 1547.

B. Kelm, B. Menze, C. Zechmann, et al. (2007). "Automated Estimation of Tumor Probability in Prostate Magnetic Resonance Spectroscopic Imaging: Pattern Recognition vs. Quantification." Magnetic Resonance in Medicine, 57, 150–159.

H. Khotanlou, J. Atif, O. Colliot, et al. (2006). "3D brain tumor segmentation using fuzzy classification and deformable models." In: Fuzzy Logic and Applications, *Lecture Notes in Computer Science*, vol. 3849/2006, 312–318. Springer.

C. Kimmel, W. Ballard, S. Kimmel, et al. (1995). "Stages of Embryonic Development of the Zebrafish." Developmental Dynamics, 203, 253–310.

D. Koller, N. Friedman (2009). Probabilistic Graphical Models – Principles and Techniques. MIT Press.

V. Kolmogorov, Y. Boykov (2005). "What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/Area and Flux." In: International Conference on Computer Vision (ICCV 2005).

V. Kolmogorov, R. Zabih (2004). "What Energy Functions can be Minimized via Graph Cuts?" IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2), 147–159.

U. Köthe (2000). Generische Programmierung für die Bildverarbeitung. Ph.D. thesis, Universität Hamburg. Software available at http://hci.iwr.uni-heidelberg.de/vigra/.

V. Kovalev, F. Kruggel, H. Gertz, et al. (2001). "Three-Dimensional Texture Analysis of MRI Brain Datasets." IEEE Transactions on Medical Imaging, 20, 424–433.

R. Kreis (2004). "Issues of spectral quality in clinical ^{1}H magnetic resonance spectroscopy and a gallery of artifacts." NMR in Biomedicine, 17(6), 361–381.

E. Lander, L. Linton, B. Birren, et al. (2001). "Initial sequencing and analysis of the human genome." Nature, 409, 860–921.

T. Langenberg, T. Dracz, A. Oates, et al. (2006). "Analysis and Visualization of Cell Movement in the Developing Zebrafish Brain." Developmental Dynamics, 235, 928–933.

C. Lee, M. Schmidt, A. Murtha, et al. (2005). "Segmenting brain tumors with conditional random fields and support vector machines." In: First International Workshop for Computer Vision for Biomedical Image Applications (CVBIA), *Lecture Notes in Computer Science*, vol. 3765/2005, 469–478. Springer.

C. Lee, S. Wang, F. Jiao, et al. (2006). "Learning to model spatial dependency: Semi-supervised discriminative random fields." In: Advances in Neural Information Processing Systems (NIPS), vol. 19, 793–800.

C. Lee, S. Wang, A. Murtha, et al. (2008). "Segmenting Brain Tumors using Pseudo–Conditional Random Fields." In: Medical Image Computing and Computer Assisted Intervention (MICCAI), vol. 5241/2008, 359–366. Springer.

K. V. Leemput, F. Maes, D. Vandermeulen, et al. (1999a). "Automated Model-based Bias Field Correction of MR Images of the Brain." IEEE Transactions on Medical Imaging, 18(10), 885–896.

K. V. Leemput, F. Maes, D. Vandermeulen, et al. (1999b). "Automated Model-based Tissue Classification of MR Images of the Brain." IEEE Transactions on Medical Imaging, 18(10), 897–908.

A. Lefohn, J. Cates, R. Whitaker (2003). "Interactive, GPU-Based Level Sets for 3D Brain Tumor Segmentation." In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 2878/2003, 564–572.

M. Letteboer, O. Olsen, E. Dam, et al. (2004). "Segmentation of Tumors in Magnetic Resonance Brain Images Using an Interactive Multiscale Watershed Algorithm." Academic Radiology, 11, 1125–1138.

F. Li, X. Zhou, J. Ma, et al. (2010). "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis." IEEE Transactions on Medical Imaging, 29(1), 96–105.

G. Li, T. Liu, J. Nie, et al. (2008a). "Segmentation of touching cell nuclei using gradient flow tracking." Journal of Microscopy, 231(1), 47–58.

G. Li, T. Liu, A. Tarokh, et al. (2007). "3D cell nuclei segmentation based on gradient flow tracking." BMC Cell Biology, 8, 40.

K. Li, E. Miller, M. Chen, et al. (2008b). "Cell population tracking and lineage construction with spatiotemporal context." Medical Image Analysis, 12(5), 546–566.

J. Lichtman, J. Livet, J. Sanes (2008). "A technicolour approach to the connectome." Nature Reviews Neuroscience, 9, 417–422.

H. Lin, C. Lin, R. Weng (2007). "A note on Platt's probabilistic outputs for support vector machines." Machine Learning, 68, 267–276.

T. Liu, J. Nie, G. Li, et al. (2008). "ZFIQ: a software package for zebrafish biology." Bioinformatics, 24(3), 438–439.

J. Livet, T. Weissman, H. Kang, et al. (2007). "Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system." Nature, 450, 56–62.

X. Lou, F. Kaster, M. Lindner, et al. (2011a). "DELTR: Digital Embryo Lineage Tree Reconstructor." In: International Symposium on Biomedical Imaging (ISBI), submitted.

X. Lou, U. Köthe, P. Keller, et al. (2011b). "Accurate Reconstruction of Digital Embryo Volume with Multi-Object Shape Regularization." Medical Image Analysis, to be submitted.

M. A. Luengo-Oroz, B. Lombardot, E. Faure, et al. (2007). "A Mathematical Morphology Framework for the 4D Reconstruction of the Early Zebrafish Embryogenesis." In: International Symposium on Mathematical Morphology.

D. Lunn, A. Thomas, N. Best, et al. (2000). "WinBUGS – A Bayesian modelling framework: Concepts, structure and extensibility." Statistics and Computing, 10(4), 325–337.

M. Martínez-Bisbal, B. Celda (2009). "Proton magnetic resonance spectroscopy imaging in the study of human brain cancer." Quarterly Journal of Nuclear Medicine and Molecular Imaging, 53(6), 618–630.

A. Maudsley, A. Darkazanli, J. Alger, et al. (2006). "Comprehensive processing, display and analysis for *in vivo* MR spectroscopic imaging." NMR in Biomedicine, 19(4), 492–503.

B. Menze, B. Kelm, R. Masuch, et al. (2009). “A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data.” BMC Bioinformatics, 10, 213.

B. H. Menze, B. M. Kelm, M.-A. Weber, et al. (2008). “Mimicking the human expert: Pattern recognition for an automated assessment of data quality in MRSI.” Magnetic Resonance in Medicine, 59(6), 1457–1466.

B. H. Menze, M. P. Lichy, P. Bachert, et al. (2006). “Optimal classification of long echo time in vivo magnetic resonance spectra in the detection of recurrent brain tumors.” NMR in Biomedicine, 19(5), 599–609.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, et al. (1953). “Equation of state calculations by fast computing machines.” Journal of Chemical Physics, 21(6), 1087–1092.

J.-B. Michel, Y. Shen, A. Aiden, et al. (2010). “Quantitative Analysis of Culture Using Millions of Digitized Books.” Science, 331(6014), 176–182.

D. Mikulis, T. Roberts (2007). “Neuro MR: protocols.” Journal of Magnetic Resonance Imaging, 26(4), 838–847.

T. Minka (2001). “Expectation Propagation for approximate Bayesian inference.” In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI), 362 – 369.

T. Minka (2004). “Power EP.” Tech. Rep. MSR-TR-2004-149, Microsoft Research.

T. Minka (2005). “Divergence measures and message passing.” Tech. Rep. MSR-TR-2005-173, Microsoft Research Cambridge.

T. Minka, J. Winn (2009). “Gates.” In: D. Koller, D. Schuurmans, Y. Bengio, et al. (eds.), Advances in Neural Information Processing Systems (NIPS), vol. 21, 1073–1080. MIT Press, Cambridge MA.

T. Minka, J. Winn, J. Guiver, et al. (2009). “Infer.NET 2.2.” Microsoft Research Cambridge. `http://research.microsoft.com/infernet`.

N. Moon, E. Bullitt, K. Van Leemput, et al. (2002). “Automatic Brain and Tumor Segmentation.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 2488/2002, 372–379. Springer.

G. Moonis, J. Liu, J. Udupa, et al. (2002). “Estimation of tumor volume with fuzzy-connectedness segmentation of MR images.” American Journal of Neuroradiology, 23(3), 356–363.

K. Mosaliganti, A. Gelas, A. Gouaillard, et al. (2009). "Detection of Spatially Correlated Objects in $3D$ Images Using Appearance Models and Coupled Active Contours." In: G.-Z. Yang, et al. (eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI 2009), Part II, *Lecture Notes in Computer Science*, vol. 5762, 641–648. Springer, Berlin.

J. Munkres (1957). "Algorithms for the Assignment and Transportation Problems." Journal of the Society for Industrial and Applied Mathematics, 5(1), 32–38.

A. Nemirovski, M. Todd (2008). "Interior-point methods for optimization." Acta Numerica, 17, 191–234.

B. D. Neuter, J. Luts, L. Vanhamme, et al. (2007). "Java-based framework for processing and displaying short-echo-time magnetic resonance spectroscopy signals." Computational Methods and Programs in Biomedicine, 85, 129–137.

J. Nie, Z. Xue, T. Liu, et al. (2009). "Automated brain tumor segmentation using spatial accuracy-weighted hidden Markov Random Field." Computerized Medical Imaging and Graphics, 33, 431–441.

N. Olivier, M. Luengo-Oroz, L. Duloquin, et al. (2010). "Cell Lineage Reconstruction of Early Zebrafish Embryos Using Label-Free Nonlinear Microscopy." Science, 329(5994), 967–971.

S. Ortega-Martorell, I. Olier, M. Julià-Sapé, et al. (2010). "SpectraClassifier 1.0: a user friendly, automated MRS-based classifier-development system." BMC Bioinformatics, 11, 106.

N. Otsu (1979). "A threshold selection method from gray-level histograms." IEEE Transactions on Systems, Man, and Cybernetics, 9, 62–66.

C. Pachai, Y. Zhu, J. Grimaud, et al. (1998). "Pyramidal approach for automatic segmentation of multiple sclerosis lesions in brain MRI." Computerized Medical Imaging and Graphics, 22(5), 399–408.

D. Padfield, J. Rittscher, B. Roysam (2009a). "Coupled Minimum-Cost Flow Cell Tracking." In: J. Prince, D. Pham, K. Myers (eds.), Information Processing in Medical Imaging (IPMI 2009), *Lecture Notes in Computer Science*, vol. 5636, 374–385. Springer, Berlin.

D. Padfield, J. Rittscher, N. Thomas, et al. (2009b). "Spatio-temporal cell cycle phase analysis using level sets and fast marching methods." Medical Image Analysis, 13(1), 143–155.

C. Papadimitriou, K. Steiglitz (1998). Combinatorial Optimization: Algorithms and Complexity. Dover Publications.

J. Pearl (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann.

W. Pijnappel, A. van den Boogaart, R. de Beer, et al. (1992). "SVD-Based Quantification of Magnetic Resonance Signals." Journal of Magnetic Resonance, 97, 122–134.

J. Poullet, D. Sima, A. Simonetti, et al. (2007). "An automated quantitation of short echo time MRS spectra in an open source software environment: AQSES." NMR in Biomedicine, 20(5), 493–504.

J. Poullet, D. Sima, S. Van Huffel (2008). "MRS signal quantitation: A review of time- and frequency-domain methods." Journal of Magnetic Resonance, 195(2), 134–144.

M. Prastawa, E. Bullitt, G. Gerig (2009). "Simulation of Brain Tumors in MR Images for Evaluation of Segmentation Efficacy." Medical Image Analysis, 13(2), 297–311.

M. Prastawa, E. Bullitt, S. Ho, et al. (2003a). "Robust estimation for brain tumor segmentation." In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 2879/2003, 530–537. Springer.

M. Prastawa, E. Bullitt, S. Ho, et al. (2004). "A brain tumor segmentation framework based on outlier detection." Medical Image Analysis, 8(3), 275–283.

M. Prastawa, E. Bullitt, N. Moon, et al. (2003b). "Automatic Brain Tumor Segmentation by Subject Specific Modification of Atlas Priors." Academic Radiology, 10(12), 1341–1348.

S. Provencher (2001). "Automatic quantitation of localized in vivo ^{1}H spectra with LCModel." NMR in Biomedicine, 14(4), 260–264.

S. Provencher (2010). LCModel and LCMgui user's manual, version 6.2-2. Http://s-provencher.com/pub/LCModel/manual/manual.pdf.

R. Raman, S. Raguram, G. Venkataraman, et al. (2005). "Glycomics: an integrated systems approach to structure-function relationships of glycans." Nature Methods, 2, 817–824.

C. Rasmussen, C. Williams (2006). Gaussian Processes for Machine Learning. MIT Press.

H. Ratiney, M. Sdika, Y. Coenradie, et al. (2005). "Time-domain semi-parametric estimation based on a metabolite basis set." NMR in Biomedicine, 18, 1–13.

N. Ray, R. Greiner, A. Murtha (2008). "Using Symmetry to Detect Abnormalities in Brain MRI." Computer Society of India Communications, 31(19), 7–10.

S. Raya (1990). "Low-level segmentation of 3D Magnetic Resonance brain images: A rule-based system." IEEE Transactions on Medical Imaging, 9, 327–337.

V. Raykar, S. Yu, L. Zhao, et al. (2009). "Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit." In: International Conference on Machine Learning (ICML), 889 – 896.

V. Raykar, S. Yu, L. Zhao, et al. (2010). "Learning From Crowds." Journal of Machine Learning Research, 11, 1297–1322.

A. Rényi (1961). "On measures of entropy and information." In: Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 547–561.

E. Reynaud, U. Kržič, K. Greger, et al. (2008). "Light sheet-based fluorescence microscopy: more dimensions, more photons, and less photodamage." HFSP Journal, 2(5), 266–275.

R. Rifkin, A. Klautau (2004). "In Defense of One-Vs-All Classification." Journal of Machine Learning Research, 5, 101–141.

J. Rittscher (2010). "Characterization of Biological Processes through Automated Image Analysis." Annual Review of Biomedical Engineering, 12, 315–344.

S. Rogers, M. Girolami, T. Polajnar (2010). "Semi-parametric analysis of multi-rater data." Statistics and Computing, 20(3), 317 – 334.

B. Sajja, J. Wolinsky, P. Narayana (2009). "Proton Magnetic Resonance Spectroscopy in Multiple Sclerosis." Neuroimaging Clinics of North America, 19(1), 45–58.

M. Schmidt, I. Levner, R. Greiner, et al. (2005). "Segmenting Brain Tumors using Alignment-Based Features." In: International Conference on Machine Learning and Applications (ICMLA), 215–220.

B. Schölkopf, A. Smola (2002). Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge MA.

B. Settles (2010). "Active Learning Literature Survey." Tech. Rep. 1648, University of Wisconsin-Madison.

J. Shaffer (1995). "Multiple Hypothesis Testing." Annual Review of Psychology, 46, 561–584.

D. Sima, A. Croitor Sava, S. V. Huffel (2010). “Adaptive Alternating Minimization for Fitting Magnetic Resonance Spectroscopic Imaging Signals.” In: M. Diehl, et al. (eds.), Recent Advances in Optimization and its Applications in Engineering, vol. 7, 511–520. Springer, Berlin.

D. Sima, S. van Huffel (2006). “Regularized semiparametric model identification with application to NMR signal quantification with unknown macromolecular baseline.” Journal of the Royal Statistical Society B (Methodological), 68(3), 383–409.

S. Smith, T. Levante, B. Meier, et al. (1994). “Computer Simulations in Magnetic Resonance. An Object-Oriented Programming Approach.” Journal of Magnetic Resonance, A 106(1), 75–105.

P. Smyth, U. Fayyad, M. Burl, et al. (1995). “Inferring Ground Truth From Subjective Labelling of Venus Images.” In: G. Tesauro, D. Toretzy, T. Leen (eds.), Advances in Neural Information Processing Systems (NIPS), vol. 7, 1085–1092. MIT Press.

J. Solomon, J. Butman, A. Sood (2004). “Data Driven Brain Tumor Segmentation in MRI Using Probabilistic Reasoning over Space and Time.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 3216/2004, 301–309. Springer.

J. Solomon, J. Butman, A. Sood (2006). “Segmentation of brain tumors in 4D MR images using the hidden Markov model.” Computer Methods and Programs in Biomedicine, 84(2–3), 76–85.

H. Soltanian-Zadeh, D. Peck, J. Windham, et al. (1998). “Brain tumor segmentation and characterization by pattern analysis of multispectral NMR images.” NMR in Biomedicine, 11(4–5), 201–208.

C. Sommer, C. Straehle, U. Köthe, et al. (2010). “Interactive Learning and Segmentation Tool Kit.” `http://gitorious.org/ilastik/ilastik.git`. “master” branch, commit 087fd66d4db165ff6c14c8573b6543b3e62d5b7e with personal customizations.

Y. Song, C. Zhang, J. Lee, et al. (2006). “A Discriminative Method for Semi-Automated Tumorous Tissues Segmentation of MR Brain Images.” In: Computer Vision and Pattern Recognition Workshop (CVPRW).

Y. Song, C. Zhang, J. Lee, et al. (2009). “Semi-supervised discriminative classification with application to tumorous tissues segmentation of MR brain images.” Pattern Analysis & Applications, 12(2), 99–115.

D. Stefan, F. D. Cesare, A. Andrasescu, et al. (2009). “Quantitation of magnetic resonance spectroscopy signals: the jMRUI software package.” Measurement Science and Technology, 20, 104035.

C. Stone (1977). “Consistent Nonparametric Regression.” Annals of Statistics, 5(4), 595–620.

B. Stroustrup (2001). “Exception Safety: Concepts and Techniques.” In: C. Dony, J. Knudsen, A. Romanovsky, et al. (eds.), Advances in Exception Handling Techniques, 60–76. Springer, New York.

J. Sulston, E. Schierenberg, J. White, et al. (1983). “The embryonic cell lineage of the nematode *Caenorhabditis elegans*.” Developmental Biology, 100(1), 64–119.

A. Tate, J. Underwood, D. Acosta, et al. (2006). “Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra.” NMR in Biomedicine, 19, 411–434.

T. Terlaki, S. Zhang (1993). “Pivot rules for linear programming: a survey on recent theoretical developments.” Annals of Operation Research, 46, 202–233.

J. Udupa, L. Wei, S. Samarasekera, et al. (1997). “Multiple sclerosis lesion quantification using fuzzy-connectedness principles.” IEEE Transactions on Medical Imaging, 16(5), 598–609.

L. Vanhamme, A. van den Boogaart, S. van Huffel (1997). “Improved method for accurate and efficient quantification of MRS data with use of prior knowledge.” Journal of Magnetic Resonance, 129(1), 35–43.

M. Wainwright, M. Jordan (2008). “Graphical models, exponential families, and variational inference.” Foundations and Trends® in Machine Learning, 1(1-2), 1–305.

R. Walker, P. Jackway (1996). “Statistical Geometric Features – Extensions for Cytological Texture Analysis.” In: Proceedings of the 13th International Conference on Pattern Recognition (ICPR).

M. Wang, X. Zhou, F. Li, et al. (2008). “Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy.” Bioinformatics, 24(1), 94–101.

S. Warfield, J. Dengler, J. Zaers, et al. (1995). “Automatic identification of gray matter structures from MRI to improve the segmentation of white matter lesions.” Journal of Image-Guided Surgery, 1(6), 326–338.

S. Warfield, M. Kaus, F. A. Jolesz, et al. (2000). “Adaptive, template moderated, spatially varying statistical classification.” Medical Image Analysis, 4(1), 43–55.

S. Warfield, K. Zou, W. Wells (2004). “Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation.” IEEE Transactions on Medical Imaging, 23(7), 903–921.

S. Warfield, K. Zou, W. Wells (2008). “Validation of image segmentation by estimating rater bias and variance.” Philosophical Transactions of the Royal Society A, 366(1874), 2361–2375.

M. Wels, G. Carneiro, A. Aplas, et al. (2008a). “A Discriminative Model-Constrained Graph Cuts Approach to Fully Automated Pediatric Brain Tumor Segmentation in 3-D MRI.” In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), *Lecture Notes in Computer Science*, vol. 5241/2008, 67–75. Springer.

M. Wels, M. Huber, J. Hornegger (2008b). “Fully Automated Segmentation of Multiple Sclerosis Lesions in Multispectral MRI.” In: Pattern Recognition and Image Analysis, vol. 18, 347–350. Pleiades.

J. Whitehill, P. Ruvolo, T. Wu, et al. (2009). “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise.” In: Y. Bengio, D. Schuurmans, J. Lafferty, et al. (eds.), Advances in Neural Information Processing Systems 22, 2035–2043. MIT Press.

F. Wilcoxon (1945). “Individual Comparisons by Ranking Methods.” Biometrics Bulletin, 1(6), 80–83.

J. Winn, C. Bishop (2005). “Variational Message Passing.” Journal of Machine Learning Research, 6, 661–694.

L. Wolsey (1998). Integer programming. Wiley-Interscience.

Z. Wu, H.-W. Chung, F. Wehrli (1994). “A Bayesian approach to subvoxel tissue classification in NMR microscopic images of trabecular bone.” Magnetic Resonance in Medicine, 31(3), 302–308.

D. Xu, D. Vigneron (2010). “Magnetic Resonance Spectroscopy Imaging of the Newborn Brain – A Technical Review.” Seminars in Perinatology, 34(1), 20–27.

Z. Yin, R. Bise, M. Chen, et al. (2010). “Cell Segmentation in Microscopy Imagery Using a Bag of Local Bayesian Classifiers.” In: International Symposium on Biomedical Imaging (ISBI), 125–128.

T. Yokoo, W. Bae, G. Hamilton, et al. (2010). “A Quantitative Approach to Sequence and Image Weighting.” Journal of Computer-Assisted Tomography, 34, 317–331.

C. Zanella, M. Campana, B. Rizzi, et al. (2010). "Cells Segmentation from 3-D Confocal Images of Early Zebrafish Embryogenesis." IEEE Transactions on Image Processing, 19(3), 770–781.

C. Zechmann, B. Menze, B. Kelm, et al. (2011). "How much spatial context do we need? Automated versus manual pattern recognition of 3D MRSI data of prostate cancer patients." NMR in Biomedicine, submitted.

J. Zhang (1992). "The mean field theory in EM procedures for Markov random fields." IEEE Transactions on Signal Processing, 40(10), 2570–2583.

J. Zhou, K. Chan, V. Chong, et al. (2005). "Extraction of Brain Tumor from MR Images Using One-Class Support Vector Machine." In: IEEE Engineering in Medicine and Biology 27th Annual Conference.

Y. Zhu, Q. Liao, W. Dou, et al. (2005). "Brain tumor segmentation in MRI based on fuzzy aggregators." In: Visual Communications and Image Processing 2005, *Proceedings of SPIE*, vol. 5960, 1704–1711.