DIPLOMARBEIT

Feature Generation and Selection in Hyperspectral Imaging

ausgeführt am
Institut für Chemische Technologien und Analytik
der Technischen Universität Wien

unter der Leitung von
Ao.Univ.Prof. Mag.rer.nat. Dr.rer.nat Johann Lohninger
durch

Wolfgang Ganglberger
Novaragasse 13/8
A-1020 Wien

Mai 2018

Wolfgang Ganglberger

Kurzfassung

Diese Diplomarbeit präsentiert AutoFeature, einen neuen Algorithmus, der materialspezifische spektroskopische Charakteristika aus annotierten Infrarotspektroskopie-Daten völlig automatisch zu extrahieren vermag. Mithilfe dieser Charakteristika können anschließend die jeweiligen Materialien in hyperspektralen Bildern identifiziert werden. Eine Expertise in spektroskopischen Eigenschaften der Materialien ist demnach für den Anwender nicht nötig.

Der AutoFeature Algorithmus generiert einerseits tausende Features mittels Template Matching und wählt andererseits, basierend auf statistischen Methoden und maschinellem Lernen, die vielversprechendsten Features aus. Für das Template Matching wurden vier Arten von Templates konzipiert: Dreiecke, Gauß'sche Glockenkurven, allgemeine Gauß'sche Glockenkurven und Geraden. Das Template Matching erfolgt an allen Positionen des Infrarotspektrums und beruht auf dem Pearson Korrelationskoeffizienten. Die anschließende Auswahl der relevanten Features erfolgt methodisch entweder durch Fast Function Extraction, Embedded Random Forest Modelling oder durch eine der drei Filtermethoden ReliefF, Fisher Score und HSIC Lasso.

Die Studie untersucht zunächst das Verhalten des AutoFeature Algorithmus hinsichtlich Datensatzgröße und Rauschen mithilfe künstlicher Daten. Anschließend werden Features aus drei realen Datensätzen aus Mikroplastik- und Hautgewebeproben automatisch extrahiert. Diese werden für das Erstellen von Random Forest Modellen verwendet, anhand derer im ersten Experiment fünf Polymere, im zweiten Experiment Melanoma und Nicht-Melanoma und im dritten Experiment Bindegewebe und Nicht-Bindegewebe klassifiziert werden.

Bei den künstlichen Datensätzen mit Samplegröße 16 konnte der Algorithmus die korrekten Features bis zu einem Rauschniveau von 10% erkennen, bei Samplegröße 100 bis zu einem Rauschniveau von 25%. Für reale Daten wurden Features aller vier Templates extrahiert, die sich ausschließlich in charakteristischen Absorptionsbändern befinden. Die genauen Positionen und Breiten mancher Features fallen dennoch unerwartet aus. Die Validierung der Random Forest Modelle mit Testdaten resultierte in einer Klassifikationsgenauigkeit von mindestens 99.6% im Fall der Polymere und in perfekten Klassifikationen bei den Melanoma- und Bindegewebsdaten. Mittels unterschiedlicher Selektionsmethoden wurden Features mit variablen Dichteeigenschaften ausgewählt, die jedoch alle eine überzeugende Unterscheidbarkeit der Klassen aufweisen.

Insgesamt konnten mithilfe des AutoFeature Algorithmus sowohl bei künstlichen als auch bei realen Daten Features automatisch extrahiert werden, die nicht nur chemisch sinnvoll, sondern auch für Klassifikationen geeignet sind. Um das Potential des AutoFeature Algorithmus festzustellen, bedarf es weiterer Untersuchungen mit vielfältigeren Datensätzen. Durch das Erstellen zusätzlicher Templates und die Anpassung der Selektionsparameter ist eine algorithmische Weiterentwicklung möglich.

Abstract

This master's thesis presents Autofeature, a novel algorithm that enables the automatic extraction of material specific spectroscopic characteristics from an annotated infrared spectroscopy dataset. With these characteristics the material can then be identified in hyperspectral images. Accordingly, no expertise of the user in the spectroscopic properties of the material is necessary.

On the one hand, the AutoFeature algorithm generates thousands of features based on template matching and on the other hand, selects the most promising features based on statistical and machine learning methods. Four types of templates are designed: triangles, Gaussian bells, general Gaussian bells and straight lines. The matching is performed at all possible infrared spectrum positions by employing the Pearson correlation coefficient. The subsequent feature selection is carried out with fast function extraction, embedded random forest modelling or with one of the following three filter selection methods ReliefF, Fisher score and HSIC lasso.

The study first investigates the properties of the AutoFeature algorithm concerning sample size and noise. Next, features are automatically extracted from three real-world data sets containing microplastic and skin tissue specimens. These features are then used to train random forest classification models for class predictions of five polymers in the first experiment, melanoma and non-melanoma in the second experiment, and connective tissue and non-connective tissue in the third experiment.

For artificial data, the algorithm was able to extract correct features for noise levels of 10% for a sample size of 16 respectively 25% for sample size 100. For real-world data, features of all four types are extracted and the features are only located at characteristic absorption bands of the substances being investigated. The exact positions and widths of some features are unexpected though. The validation of the random forest models with unseen test data yielded classification accuracies of 99.6% or higher for the polymer predictions and a perfect classification for the melanoma and connective tissue predictions. While the different selection methods result in features with different probability density functions, they all yield features with convincing class discrimination properties.

Overall, the AutoFeature algorithm was able to automatically extract features that were chemically meaningful and suited for prediction tasks for both artificial and real-world data. To evaluate further potential of the algorithm, examinations with datasets of greater variety need to be performed. We believe, by designing additional templates and adapting parameters of the selection methods, further algorithmic progress can be made.

Acknowledgements

I would like to express my deep sense of gratitude to my supervising Prof. Hans Lohninger for not only involving me in thrilling discourses in the fields of statistics, physics and chemistry, but also for his sociable, genuine and supportive nature. Working on this thesis was made enjoyable throughout and was a valued part of learning.

I wish to thank all the people who have contributed to this thesis. Special thanks are extended to my colleagues Benedikt Steindl and Andreas Cremer who have shaped this work with their opinions and suggestions.

Furthermore, I would like to thank all of those great-minds I have been fortunate enough to meet and learn from during my academic journey. Thank you, dear colleagues, professors and friends – I would not be where I am today without all of you.

Particular thanks, love and appreciation to Sandra Voser. She is continuously supportive, encourages me to pursue my goals and most of all, enriches my life.

Lastly, I would like to thank my family for their long lasting support. I dedicate this thesis to my parents, Anita and Herbert, without whom all of this would not have been possible. I cannot thank them enough for their liberal education and for their belief in me.

Thank you.

Contents

Chapter 1

Introduction

Spectroscopy is the science of the interaction between matter and any part of the electromagnetic spectrum [1]. Acquired spectroscopic data is commonly presented as a *spectrum* that shows the magnitude of the measured interaction as a function of either frequency or wavelength. As different materials interact differently with electromagnetic radiation, knowledge of a spectrum enables the identification of a material being investigated [1], the task we are mainly interested in in this thesis. Categorization of spectra is difficult not only because there are overlapping characteristics for different materials but also because materials being investigated can consist of an assortment of substances. The latter is especially true for biological components.

If not only one spectrum but any number of spectra in a spatial context is analysed, we speak of *hyperspectral imaging*. This can yield vast amounts of information that can hardly be processed by humans. Hence, computational methods to analyse hyperspectral images are beneficial in today's world of big data. However, for many computational material identification tools, one still needs to know about the characteristics of the substances that are being investigated. These characteristics are often unknown or only known to experts in this field. In this work we will introduce a novel algorithm that is designed to automatically extract valuable characteristics of different materials from a dataset of annotated spectra. We will call these valuable characteristics *features* and they will facilitate the identification and detection of materials in new, unseen hyperspectral images. With an automated way of finding features, we believe in opening up the tools of spectroscopy to those who are not experts in this field. Consequently, challenges where spectroscopy may play a role in the solutions could be addressed by more people, potentially leading to a faster pace of finding a greater variety of solutions.

Many researchers in the fields of chemistry, physics, signal processing, geoscience and remote sensing have been working towards an automatic feature extraction in the last decades. Different approaches including refined wavelength selections, principal component analysis based data transformations, and linear and non-linear feature extraction methods have been used [2] [3] [4] [5] [6]. We will combine a form of template matching with statistical and machine learning-based methods. In particular, we will design four classes of generic templates. The various shapes of each class are used to find matches and statistical information in the spectra. This will typically result in thousands of feature candidates. To select the most promising features of these thousands of candidates, five variants of linear and non-linear machine learning methods based on embedded random forest modelling, fast function extraction, Fisher score, ReliefF algorithm and HSIC lasso are used. All steps can be done without the assistance of a user and constitute an automatic feature extraction and selection algorithm that we name *AutoFeature*.

The spectroscopic and statistical methodologies used in the automatic feature engineering will be discussed before introducing the AutoFeature algorithm. We will conduct experiments with artificial data and real-world microplastic and skin tissue data. In these experiments, we investigate on the one hand the properties of the different AutoFeature variants, and on the other hand, we analyse the resulting, automatically extracted features and examine if they are suitable for tasks such as class prediction of different polymers, melanoma and non-melanoma, and connective tissue and non-connective tissue.

Chapter 2

Infrared Spectroscopy

As a part of analytical chemistry, infrared (IR) spectroscopy is a method that enables the identification of the chemical composition of substances. In particular, infrared spectroscopy is a versatile tool for qualitative and quantitative determination of molecular bonds.

The infrared part of the electromagnetic spectrum covers electromagnetic waves with wavelengths λ ranging from 0.78 μm to 1000 μm. The measurement units of spectra in this energy region are usually *wavenumbers* $\hat{\nu}$ instead of wavelengths. The speed of light in vacuum c and the frequency ν link these two entities [1]:

$$\hat{\nu}(\mathrm{cm}^{-1}) = \frac{1}{\lambda(\mu\mathrm{m})} \cdot 10^4(\mu\mathrm{m/cm}) = \frac{\nu(\mathrm{Hz})}{c(\mathrm{cm/s})} \tag{2.1}$$

Due to application and instrumentation reasons, the infrared spectrum is commonly divided into near- ($0.78 - 2.5$ μm), mid- ($2.5 - 15$ μm) and far-infrared ($15 -$ 1000 μm).

We will use the IR spectrum from approximately 3600 cm^{-1} = 2.8 μm to 1250 cm^{-1} = 8 μm for an investigation of chemical specimens.

Infrared spectroscopy measures the absorption, emission and reflection of infrared light in its interaction with chemical specimens. We will work with absorption spectroscopy, the underlying theory of it will be presented below and follows the discussion in Skoog et al.'s *Principles of Instrumental Analysis* (7th edition, 2017, [1]).

The main underlying physical effects of infrared spectroscopy are different vibrational and rotational states of molecules. These different states are separated by only a small difference in energy that is within the bandwidth of infrared radiation.

To be able to absorb infrared radiation, molecules need to change their dipole moment. As a first consequence of this, *mononuclear* molecules such as O_2, which do not show any change in the dipole moment during vibrations and rotations, are unsuited for infrared spectroscopy. *Polar* molecules' dipole moments are characterized by the magnitude of the difference in the charges and the distance of the charge centers. When a polar molecule vibrates, the dipole moment changes in a periodic manner inducing an electric field. This electric field interacts with the electric field of the infrared radiation. If the radiation frequency matches the molecule's vibrational frequency, energy is transmitted, changing the amplitude of the vibration — the infrared radiation gets *absorbed*. Similarly, molecules' rotational movements can interact with radiation.

The molecule's rotational state can be altered by relatively low energetic radiation with wavelengths $\lambda > 100\mu$m. Vibrational states are changed by radiations with wavelengths in the mid-infrared segment. Both rotational and vibrational energy

states are quantized causing energy absorption lines for gas molecules. For liquids and solids, due to intra- and inter-molecular interactions, absorption signals broaden to a continuum.

Measurement of the absorption of infrared radiation can be done by three types of instruments:

1. Dispersive devices consisting of wavelength-dependent spectral photometer and a grid monochromator.
2. Fourier-transform spectral photometer with interferometer.
3. Non-dispersive photometers.

Today infrared spectroscopy is usually carried out with a Fourier-transform spectrometer [1] which is also used for this study.

2.1 Fourier Transform Spectroscopy

Contrary to other types of spectroscopy, the Fourier-transform spectroscopy measures the power of the signal (and its changes) in the *time-domain*. The obtained time-domain signal is then transformed to the *frequency-domain* by the Fourier-transform [1].

Fourier-transform spectroscopy has several advantages:

1. Because it has very few optical elements and no slits, a great portion of the radiation's energy reaches the detector. This leads to an increased signal-to-noise ratio.
2. High resolution and reproducibility leads to the feasibility of analyzing complex spectra.
3. All elements of radiation reach the detection sites at the same time. This allows one to rapidly obtain a spectrum, i.e. in less than a second.

Infrared rays that impinge on a sample are scattered, transmitted, absorbed and reflected. The intensity of the sum of these altered rays is equal to the intensity of the incident ray. In Fourier-transform infrared (FTIR) spectroscopy any of the altered rays can be measured. Favourably, the sample preparation procedures are matched with the choice of radiation measurement [7].

We use thin slides of samples together with transmission spectroscopy that is observed at a 90° angle of incidence.

2.2 Spectroscopy of Polymers

A polymer is a macromolecule consisting of repeated subunits. Polymers, both natural and synthetic, exhibit a broad range of characteristics [8] and have become indispensable and ubiquitous parts of daily life [9]. Because of polymers' manifold appearances, there have been multiple spectroscopic investigations of these materials in the past decades. These investigations are usually carried out with FTIR spectroscopy and can serve very different purposes such as the structural characterization of (co-)polymers, the analysis of polymerization processes and much more [10].

In this work, we are interested in the *detection* and *identification* of microplastic particles. This is an important issue in assessing the state of the environment and especially aquatic ecosystems.

As the US agency *National Oceanic and Atmospheric Administration* points out, plastic in its various sizes and shapes is the most prevalent waste in oceans and larger lakes [11]. Particles that have a diameter of less than 5mm are referred to as *microplastic*. These small plastic particles get into the aquatic environment because they pass the filtration systems. They can either be formed from the debris of larger particles or are products of industry, e.g. small polyethylene particles in health and beauty goods. Neither volume and distribution nor the impact of microplastic is well understood today. However, standardized field methods for collecting and analyzing microplastic have been specified over the last few years. This enables the possibility of a global assessment of the amount and effects of microplastic particles [11]. Because of growing scientific and social concerns regarding the amount and distribution of microplastic in the environment, many studies covering this topic have been conducted recently. For identification of different types of polymeres, FTIR is commonly utilized [12].

Part of this work aims to detect and identify the polymers polyethylene, polypropylene, polystyrol, polymethylmethacrylate and polyacrylonitrile. Therefore, these polymers are introduced briefly in the next part.

Polyethylene (PE)

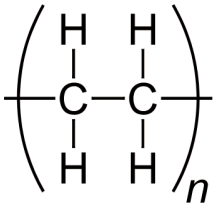


FIGURE 2.1: Structural formula of polyethylene $(C_2H_4)_n$, displaying simple molecular composition. It is the most frequently used plastic.

Polyethylene, with an estimated production of approximately 100 million tons per year (2018) is the most widely used plastic in today's world. It is most commonly used as a packaging material and for the manufacturing of pipes [13].

Polyethylene exists in different variants, predominantly having the chemical formula $(C_2H_4)_n$, see Figure 2.1. Its relatively simple chemical structure is reflected in the infrared spectrum, see Figure 2.2. There are three strong absorption bands in PE's spectrum, resulting from CH_2 asymmetric stretching ($\approx$ 2915-2920 cm^{-1}), CH_2 symmetric stretching ($\approx$ 2851-2843 cm^{-1}) and bending deformation ($\approx$ 1475-1450 cm^{-1}) [14].

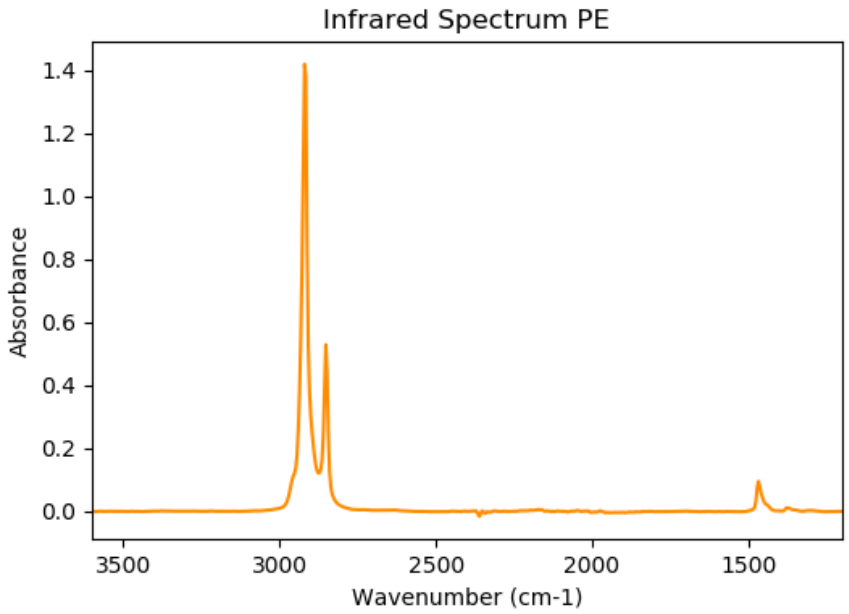


FIGURE 2.2: Infrared spectrum of polyethylene. The three absorption bands from left to right correspond to CH_2 asymmetric stretching ($\approx$ 2915-2920 cm^{-1}), CH_2 symmetric stretching ($\approx$ 2851-2843 cm^{-1}) and bending deformation ($\approx$ 1475-1450 cm^{-1}) [14].

Polypropylene (PP)

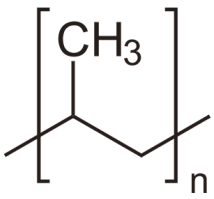


FIGURE 2.3: Structural formula of polypropylene $(C_3H_6)_n$. It is the second most frequently used plastic.

Polypropylene,$(C_3H_6)_n$ (Figure 2.3) is the world's second most used plastic. Its usages are versatile including flexible and rigid packaging, material for everyday products, clothes and vehicles [15]. Polypropylene's main absorption bands (Figure 2.4) in the infrared spectrum are due to [16]:

- symmetrical CH_3 stretching vibration (≈2925 and 2868 cm^{-1})
- asymmetrical CH_2 stretching vibrations (≈ 2915-2920 cm^{-1})
- symmetrical CH_2 stretching (≈ 2851-2843 cm^{-1})
- CH stretching vibrations (≈ 2808 cm^{-1})
- C-H deformation vibrations (≈ 1475-1450 cm^{-1} and 1377 cm^{-1})

,

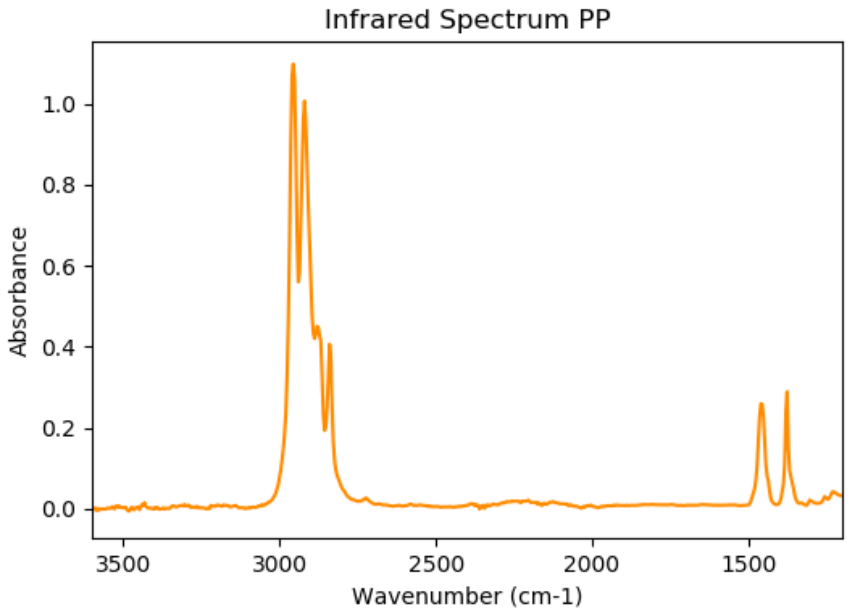


FIGURE 2.4: Infrared spectrum of polypropylene. The absorption bands from left to right correspond to symmetrical CH_3 stretching vibration (≈2925 and 2868 cm^{-1}), asymmetrical CH_2 stretching vibrations (≈ 2915-2920 cm^{-1}), symmetrical CH_2 stretching (≈ 2851-2843 cm^{-1}), CH stretching vibrations (≈ 2808 cm^{-1}) and C-H deformation vibrations (≈ 1475-1450 cm^{-1} and 1377 cm^{-1}) [16].

Polystyrene (PS)

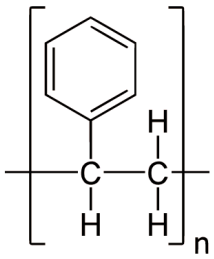


FIGURE 2.5: Structural formula of polystyrene, $(C_8H_8)_n$.

Polystyrene, a polymer with molecular composition $(C_8H_8)_n$, is used in differing fields such as consumer products, food packaging, laboratory ware, electronics, toys and much more [17]. Polystyrene's infrared spectrum (Figure 2.6) reveals the following strong bands:

- =C-H stretching vibration (between 3100 and 3000 cm^{-1})
- aromatic ring streching vibrations (1600-1430 cm^{-1})
- monosubstituted aromatic group vibrations (2000 and 1660 cm^{-1})
- C-H stretching vibration due to aliphatic group (3000-2800 cm^{-1})

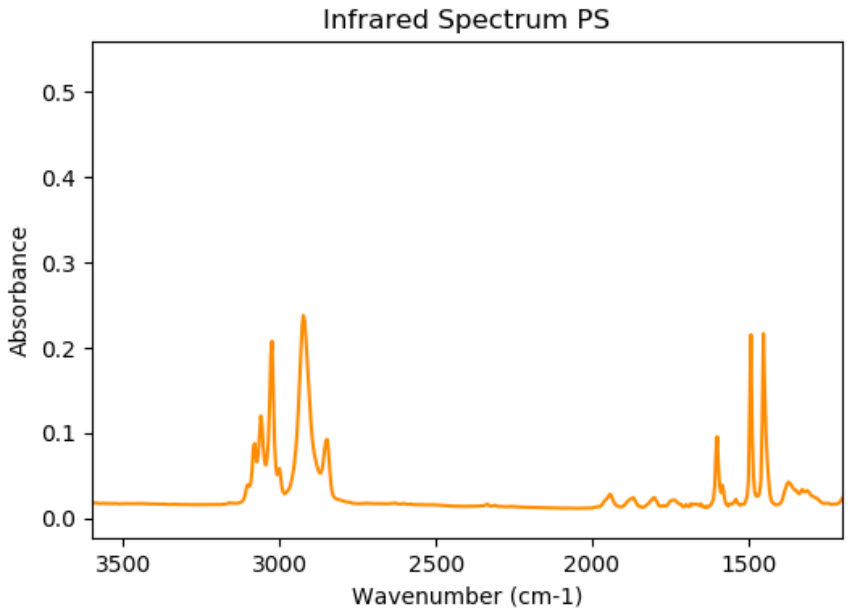


FIGURE 2.6: Infrared spectrum of polystyrene. The absorption bands from left to right correspond to =C-H stretching vibration (between 3100 and 3000 cm^{-1}), aromatic ring streching vibrations (1600-1430 cm^{-1}), monosubstituted aromatic group vibrations (2000 and 1660 cm^{-1}) and C-H stretching vibration due to aliphatic group (3000-2800 cm^{-1}).

Polymethylmethacrylate (PMMA)

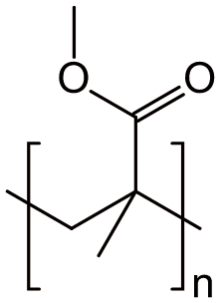


FIGURE 2.7: Structural formula of polymethylmethacrylate, $C_5H_8O_2$.

PMMA's repetitive unit is $C_5H_8O_2$, see Figure 2.7. PMMA is a transparent thermoplastic with favourable characteristics such as high impact strength, shatter-resistant and lightweight. That is why it is commonly used as a substitute for inorganic glass [18]. PMMA's infrared spectrum (2.8) consists of the following important absorption bands [19]:

- C-O-C stretching vibration (1250 - 1150 cm^{-1})
- α-methyl group vibrations ($\approx$ 1388 cm^{-1})
- absorption vibration ($\approx$ 1062 cm^{-1}, 987 cm^{-1}, 843 cm^{-1})
- C-H bond stretching vibrations of -CH_3 ($\approx$ 2997 cm^{-1}) and CH_2 ($\approx$ 2952cm^{-1}) groups

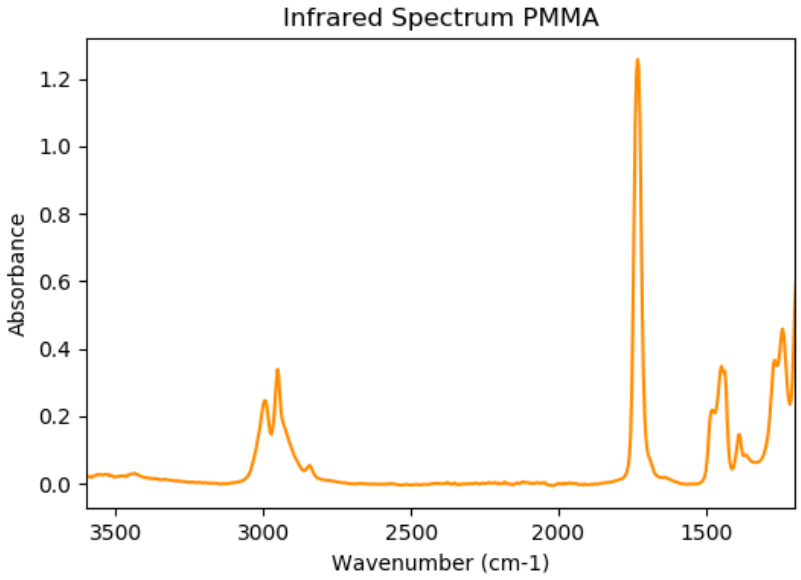


FIGURE 2.8: Infrared spectrum of polymethylmethacrylate. The absorption bands from left to right correspond to C-O-C stretching vibration (1250 - 1150 cm^{-1}), α-methyl group vibrations ($\approx$ 1388 cm^{-1}), absorption vibration ($\approx$ 1062 cm^{-1}, 987 cm^{-1}, 843 cm^{-1}), C-H bond stretching vibrations of -CH_3 ($\approx$ 2997 cm^{-1}) and CH_2 ($\approx$ 2952cm^{-1}) groups [19].

Polyacrylonitrile (PAN)

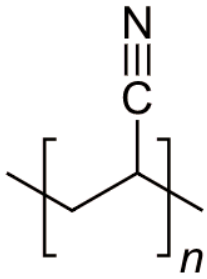


FIGURE 2.9: Structural formula of polyacrylonitrile, C_3H_3N.

Polyacrylnitril (C_3H_3N, Figure 2.9) is mainly used for fibers, both in homo- and copolymer forms. PAN-based copolymer fibers are primarily used in textiles.
PAN fiber's infrared spectrum (Figure 2.2) exhibits the following absorption bands [20]:

- -CH stretch (3000-2850 cm^{-1})
- C≡N stretch (2260-2240 cm^{-1})
- -CH_2 stretch (1465 cm^{-1})
- C=O (part of copolymers) stretching (1740-1705 cm^{-1})

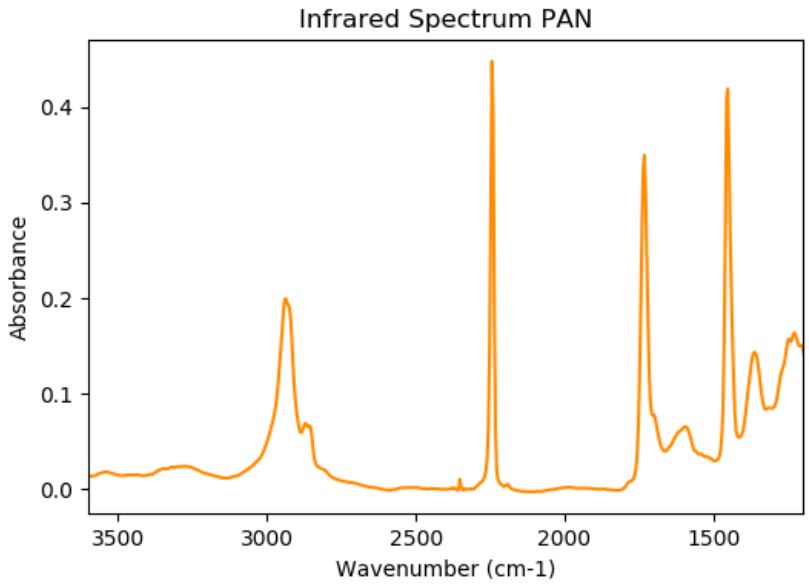


FIGURE 2.10: Infrared spectrum of polyacrylnitril. The absorption bands from left to right correspond to -CH stretch (3000-2850 cm^{-1}), C≡N stretch (2260-2240 cm^{-1}) and -CH_2 stretch (1465 cm^{-1}), C=O (part of copolymers) stretching (1740-1705 cm^{-1}) [20].

2.3 Spectroscopy of Skin Tissues and Malignant Melanoma

The human skin is commonly divided into three primary layers: the epidermis, the dermis and hypodermis.

The epidermis forms the outermost layer and serves as a protection barrier. The most important and most prevalent cells in the epidermis are keratinocytes, Merkel cells, melanocytes, and Langerhans cells.

The skin layer beneath the epidermis is the dermis. It primarily consists of connective tissue protecting the body from stress and strain. The dermis contains different functional units such as blood vessels, lymphatic vessels, glands and hair follicles.

Figure 2.11 shows a cross section of the skin.

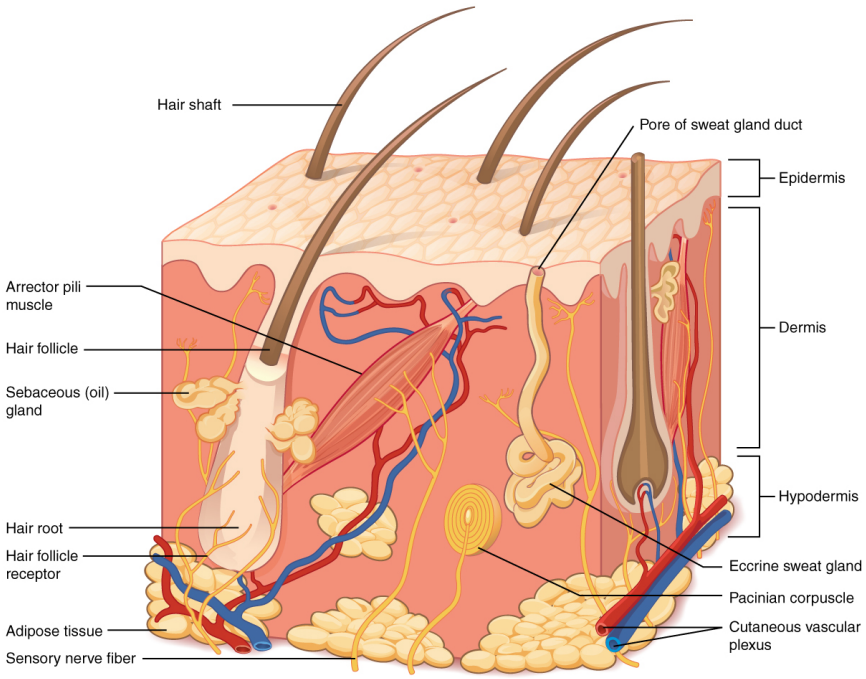


FIGURE 2.11: Cross section of the skin. 501 Structure of the skin (c) OpenStax College, Rice University (`https://cnx.org/contents/FPtK1zmh@6.27:RxywCGkA@5/Layers-of-the-Skin`), CC BY 3.0.

Melanoma is a malignant tumor of melanocytes. Melanocytes are melanin-producing cells present in several organs including the skin. In the skin they are called *cutaneous melanoma* and are situated in the bottom layer of the epidermis. The dark pigmented melanin is responsible for the skin colors.

While melanomas mostly occur in the skin, they have been found in other human body sites such as the mouth, the intestines or the eye [21], [22]. The cancerous expansion is caused by unrepaired DNA damage to skin cells (e.g. from ultraviolet radiation) that leads to mutations and skin cells that reproduce rapidly [23]. If diagnosed early, melanoma is usually curable. If not, the cancer is able to form metastases in distant parts of the body where it is hard to medicate. Compared to other types of skin cancer, melanoma is by far the most lethal one. Early diagnosis is therefore crucial for a cure and there are standardized practices to distinguish melignant melanoma from benign clusters of melanocytes (melanocytic nevus, commonly called a mole). One of these practices is to assess the ABCDE warning signs (Asymmetry, Border, Color, Diameter, Evolving) of atypical moles. For further histopathologic diagnosis and microstaging, a biopsy is necessary. [21]

Analysing and assessing the extracted tissues is challenging and studies have shown considerable variability among the agreement between experienced dermatopathologists [24], [25]. To improve the quality of histopathologic diagnosis in modern medicine, spectroscopic methods are investigated in ongoing research [26], [27], [28], [29].

Spectral Characteristics of Biological Samples in Mid-IR

A critical challenge in the analysis of biological tissue with spectroscopy is the characterisation of tissue-specific absorption bands.

Because biological samples often consist of an assembly of various biomolecules, they yield spectra with overlapping absorption bands. Therefore, differences in the magnitude of specific absorption bands may also indicate different classes. Biological units influencing the absorption bands are among others proteins, carbohydrates, lipids and DNA and RNA [27].

Chapter 3

Statistics, Machine Learning and Data Mining

This chapter introduces methods from the fields of statistics, machine learning and data mining that are essential for this thesis and the automatic feature extraction algorithm presented below. The focus is on presenting the underlying concepts of those methods, so one can understand the problem definitions, aims and solution processes. For details, the reader is referred to extensive literature in these fields (see among others Bishop [30] and Hastie, Tibshirani, and Friedman [31]).

In section 3.1 a short summary of the broad topics of supervised and unsupervised learning is given and the notions of regression, classification and clustering are introduced. The challenge of building a model that fits available data but is also general enough for new, unseen cases is described in part 3.2. Methods such as linear regression and least squares are presented in more detail in section 3.3. The need of the generalization of simple regression models will lead to part 3.4, illustrating regularization. Section 3.5 will introduce a simple, yet very effective optimization technique called coordinate descent. The field of symbolic regression and one of its methods called fast function extraction is presented in 3.6. Section 3.7 presents a classification model building algorithm called random forest. As spectroscopic hyperspectral image analysis deals with a substantial number of variables, we will discuss high dimensional data and modelling in high dimensional spaces. 3.8 will highlight the challenges that high dimensional spaces involve, referred to as *the curse of dimensionality*. To overcome those challenges, we are curious about ways to create features that carry *important* information (*feature extraction*, 3.9) and how to choose the best subset of those features for model design (*feature selection*, 3.10).

3.1 Supervised and Unsupervised Learning

Learning methods can typically be divided into two classes: supervised and unsupervised learning. Before expressing the differences of these two classes, the similarities are considered.

In both cases, we assume that there is an unknown, latent function f which a machine shall learn. The function learned by the machine which best resembles f is denoted by $\hat{f}$. Furthermore, a learner has to learn from something. In machine learning, this is data and often referred to as training set T. The training set T consists of m training examples $\mathbf{X_i}$. Every $\mathbf{X_i} \in T$ is a p-dimensional vector, meaning that each training example consists of p different variables x_j. These x_j are also called inputs, predictors or independent variables.

In supervised learning, for each $\mathbf{X_i}$ there is a variable Y_i in the training set called the output, target, response or dependent variable. We are usually interested in predicting the output variable Y on the basis of the input example $\mathbf{X}$. We can do this by inferring a suitable function $\hat{f}$ from the training set $(\mathbf{X}, Y)_i$, $i \in \{1, \dots m\}$. The function $\hat{f}$ can then be used to predict new, unseen test data $\mathbf{X}$ for which the label Y is not known. In our case Y will be a scalar and the nature of the variable can vary. It can either be quantitative (i.e. $Y \in [0, 1]$) or qualitative (i.e. Y is an element of a finite set of categories). In statistics, regression analysis is the method of estimating the relationships among variables. In machine learning, the term regression is also used for prediction tasks with a continuous output variable. Prediction tasks with a discrete, qualitative output is termed classification.

In unsupervised learning there are no labeled output variables. Instead, the output variables or labels of training examples shall be learned. Therefore, a function $\hat{f}$ that meaningfully labels the training examples (and therefore discovers some structure in the data) should be derived. Clustering is a typical example of unsupervised learning [31] [32].

3.2 Variance and Bias

The way that a machine infers a function $\hat{f}$ from the data $(\mathbf{X}, \mathbf{Y})$ can differ. In general, algorithms that yield accurate *and* stable models are favoured. Usually, there is a trade-off between accuracy and stability during the learning phase. Figure 3.2 shows how the task of finding a function $\hat{f}$ that separates two classes (orange circles and blue diamonds are training data) was solved in three different ways: In Figure 3.2a, the separating function $\hat{f_A}$ is a straight line. Depending on which side of the line an input example is, it gets classified either as a member of the orange or blue

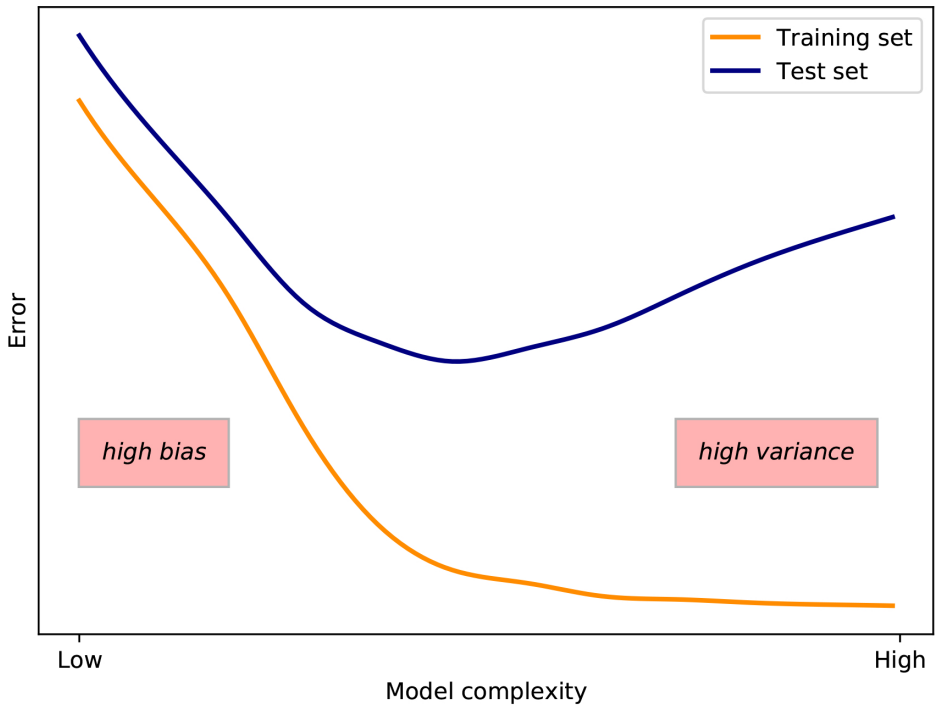
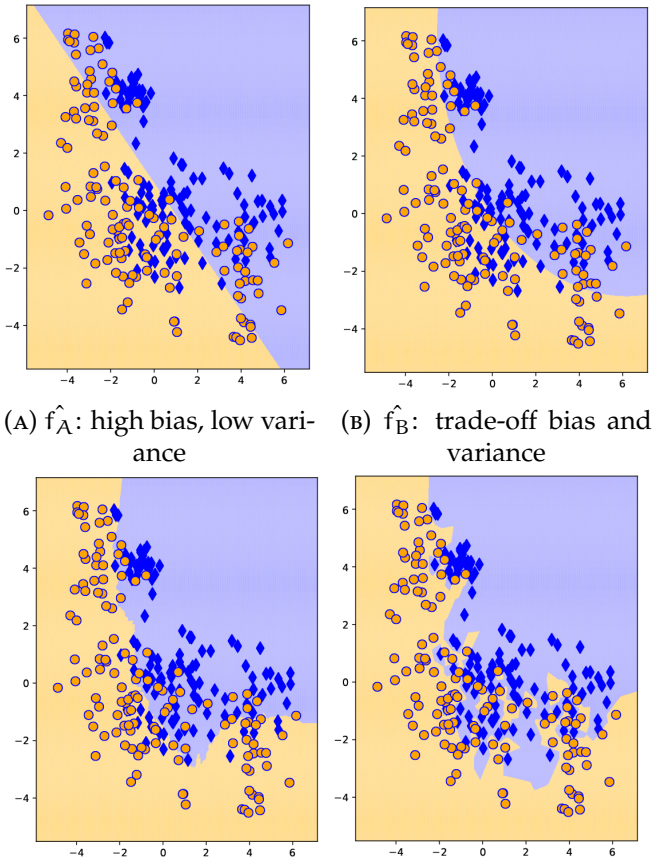


FIGURE 3.1: Typical relationship between model complexity and prediction errors on training and test set.

class. All the training data of each class which is on the other class' side of the function (e.g. orange circles in the blue classification side of function) are misclassified using this function $\hat{f_A}$. Figure 3.2d shows a more complex separating function $\hat{f_D}$ that consists of many isolated parts. Here, no training data gets misclassified. On the other hand, we have strong reasons to believe that this function is fitting the training data too narrowly and does not yield stable predictions for new, unseen data. In machine learning,$\hat{f_A}$ is said to have high *bias* and low *variance* while $\hat{f_D}$ has low bias and high variance. Figure 3.2b and Figure 3.2c show separating functions $\hat{f_B}$ and $\hat{f_C}$ that trade off bias and variance: While there are misclassifications of the training data, it still separates the two classes quite well and yields stable predictions for new data. Function $\hat{f_B}$ has higher bias and lower variance than $\hat{f_C}$ — depending on the use case, one function may be preferred over the other.

Figure 3.1 gives another visual insight into the relationship between model complexity and prediction errors on training and test samples. Increasing the model complexity yields functions that can better fit the training samples. However, at some point they *overfit* them, leading to an increasing prediction error in test samples.

In the following sections, we will outline the path to various algorithms that are able to compute functions $\hat{f}$ that have some trade-off between bias and variance, depending on the parameters used in the algorithms.



(A) $\hat{f_A}$: high bias, low variance

(B) $\hat{f_B}$: trade-off bias and variance

(C) $\hat{f_C}$: trade-off bias and variance

(D) $\hat{f_D}$: low bias, high variance

FIGURE 3.2: Different separating functions for two classes. Usually, a trade-off between bias and variance is preferred.

3.3 Linear Regression Models and Least Squares

In a linear regression model, given the vector of inputs $\mathbf{X_i} = (x_1, x_2, ..x_p)$, the prediction of the output Y is done by the following model

$$\hat{Y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j. \tag{3.1}$$

The term β_0 is called the bias or intercept while β_j serve as weights for the coordinates of input vector $\mathbf{X}$. Considering the input-output space as $(p+1)$-dimensional, equation 3.1 represents a hyperplane in this space. The model is called *linear* because it is linear in its parameters. The nature of the variables x_j is not restrained.

Given a set of training data $(\mathbf{X}, \mathbf{Y})$, we are interested in a method that places the hyperplane *as close as possible* to the data. One way of defining *as close as possible* is the minimization of the sum of squared distances of the training target variables Y_i and the model's predicted response variables $\hat{Y}_i$. This minimization of the residual sum of squares is called *least squares* [31] :

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{m} (Y_i - \mathbf{X_i}^T \boldsymbol{\beta})^2 \tag{3.2}$$

Equation 3.2 can be written in matrix form:

$$\mathrm{RSS}(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \tag{3.3}$$

Since $\mathrm{RSS}(\boldsymbol{\beta})$ is a quadratic function of its parameters, there has to exist at least one minimum. If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then there is a unique solution [33] given by:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{3.4}$$

Using the least squares algorithm for finding parameters β_i in model 3.1 leads to derivation of a predictive regression function $\hat{f}$ for which two problems are probable to occur:

- Since the aim is fitting the training data, the resulting models are prone to have high variance and low bias. Specifically, one can observe parameters getting large magnitudes if the model complexity is high or if the number of training samples is low. These large positive and negative coefficients are often the reason for high variance [30].
- The models can be complex, i.e. many variables $x_j, j \in \{1, ...p\}$ may be included in the model (i.e. have a coefficient $|\beta_j| > 0$) which limits interpretability.

Solving these two issues is regularly done with a simple extension of least squares called *regularization*.

3.4 Regularization

As mentioned above, high coefficients β_i in a linear regression model can be an indication of overfitting and high variance. One method to avoid these unpleasant effects is to add a penalty term for large coefficients to the least squares equation that is to be minimized, leading to the following equation:

$$\boldsymbol{\beta} = \arg\min_{\hat{\boldsymbol{\beta}}} \left\{ \sum_{i=1}^{m} \left(Y_i - \hat{\beta_0} - \sum_{j=1}^{p} X_{ij} \hat{\beta_j} \right)^2 + \lambda \sum_{j=1}^{p} |\hat{\beta_j}|^q \right\} \tag{3.5}$$

where $\lambda \in \mathbb{R}$ is the regularization coefficient and $q \in \mathbb{R}$ is the regularization exponent. Depending on the choice of the regularization parameter and exponent, the penalty term $\lambda \cdot |\boldsymbol{\beta}|^q$ influences the magnitude of the coefficients $\boldsymbol{\beta}$.

Ridge regression

For $q = 2$, the regularization is called *ridge regression*, L_2 *regularization*, *Tikhonov regularization* or *weight decay*. The choice of $q = 2$ makes the regularization error term quadratic and hence the minimization equation remains quadratic, ensuring a unique solution.

In ridge regression, increasing the regularization parameter λ shrinks the coefficients towards zero but not to exactly zero [30] [34] [35]. If there are co-linear variables in linear regression, they often increase variance and make the model unstable. The shrinkage of coefficients lessens this phenomenon [31].

Lasso regression

The case $q = 1$ is called *lasso* (*least absolute shrinkage and selection operator*) or L_1 *regularization*. The lasso constraint $\lambda \cdot \sum_{j=1}^{p} |\beta_j|$ is not differentiable at zero which can be a limitation in certain cases. An interesting property however is that solutions of lasso tend to be sparse, i.e. as the regularization parameter λ increases, coefficients β become exactly zero. Thus, lasso can serve as a *variable selection* tool. However, in the case $p > n$, at most n variables can be selected because of the nature of the optimization problem [30] [36] [37] [38].

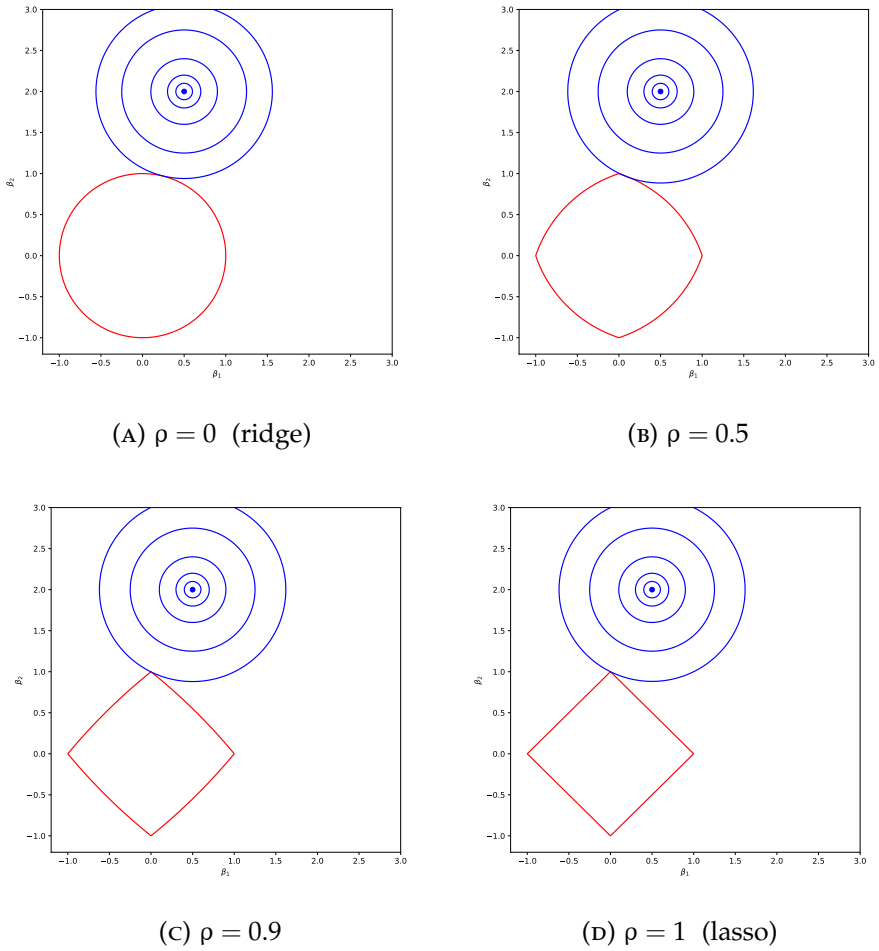
Figure 3.3a and 3.3d show the contours of the ridge and lasso regularization terms respectively, as well as the circular contours of an unregularized error function in a 2-d example (with two parameters β_1 and β_2). While the ridge equation $\beta_1^2 + \beta_2^2 \leqslant t$ yields a circular constraint, the lasso equation $\beta_1 + \beta_2 \leqslant t$ manifests itself as an equilateral parallelogram. Ordinary, unregularized least squares can lead to some optimum (β_1, β_2) that is not within the regularization constraint. The point where the error function contour first touches the regularization constraint is a solution to the regularized minimization task. For the diamond-like shape of the lasso constraint, compared to the circular shape of the ridge constraint, there is a higher chance that this point is at a position where one coefficient β_i is zero [31].

Elastic Net

As mentioned above, ridge and lasso regularization have different effects on the resulting model coefficients. *Elastic net* is a penalty regularization that attempts to unite beneficial properties of ridge and lasso. Rather than choosing a regularization exponent $1 < q < 2$, both L_1 and L_2 regularization terms get added to build the penalty term:

$$\lambda \cdot \sum_{j=1}^{p} \left(\rho|\beta_j| + (1-\rho)\beta_j^2\right) \tag{3.6}$$

The second penalty term, like ridge, aims to shrink correlated predictors. At the same time, the first term reinforces a sparse solution in the coefficients [31] [36]. Moreover, in a $p > m$ setting, models with more than m variables can be constituted. With the *mixing parameter* ρ, we can direct the elastic net to prioritize particular properties while the regularization weight λ controls the degree of regularization. The contours of the elastic net for $\rho = 0.5$ and $\rho = 0.9$ are depicted in Figures 3.3b and 3.3c respectively.



(A) $\rho = 0$ (ridge) (B) $\rho = 0.5$

(C) $\rho = 0.9$ (D) $\rho = 1$ (lasso)

FIGURE 3.3: The red contour illustrates the elastic net constraint for parameters β_1 and β_2 for different mixing parameters ρ. The blue, circular contours represent isolines of residual sum of squares for an unregularized minimization (with the minimum at β = (0.5, 2)).

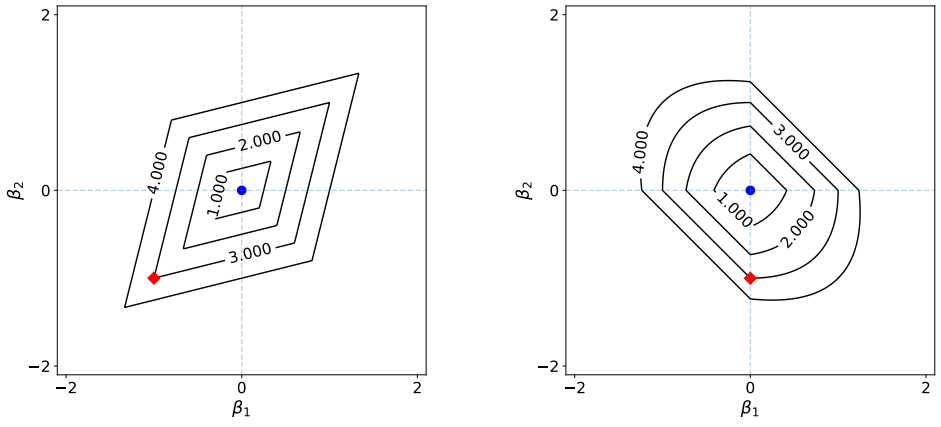
3.5 Coordinate Descent Optimization

While an analytical solution is possible for ridge regression because of the differentiable, convex nature of the problem (quadratic function), this is not the case for non-differentiable lasso. Hence, a numerical method that minimizes an L_1-regularized least squares equation is required.

A simple and fast algorithm that became popular for lasso in the past few years is *coordinate descent*. To apply the algorithm for fixed regularization parameter λ, the following procedure is conducted. In each step, coordinate descent optimizes one β_i, retaining all other β_j $(\forall j \in \{1, ...p\} : j \neq i)$ at their present values. Accordingly, the minimization problem gets solved by iteratively finding the minimum along one coordinate. In consequence of efficiency considerations, there are different ways to choose the next coordinate to be minimized. A simple approach that we use is cyclic coordinate descent in which there is an arbitrary fixed order [39] [40] [41].

In general, finding the minimum of a convex, non-differentiable function is not guaranteed by coordinate descent as Figure 3.4a points out. In this example, the minimum (blue dot) can not be found because coordinate descent is stuck at a non-smooth part (red diamond). A move in any direction would increase the error function.

However, coordinate descent is able to find the minimum in lasso problems. That is because the non-differentiable part $\lambda\|\beta\|_1 = \sum_{i=1}^{p} \lambda|\beta_i|$ is separable and each addend $\lambda|\beta_i|$ is convex. Figure 3.4b visualizes this condition, the non-smooth parts of the error function lie along coordinates, enabling the coordinate descent algorithm to move to the minimum when it is temporarily on a non-differentiable spot [42].



(A) Non-separable (B) Separable

FIGURE 3.4: Two error functions (with the blue circles as the minimum) that both have discontinuities. Coordinate descent algorithm is stuck on the left side (at the red diamond position) because moving along any coordinate increases the error. On the right side the function is separable, enabling coordinate descent to move along an axis.

3.6 Fast Function Extraction

Fast function extraction (FFX) is a method to obtain a white box model for given training data $(\mathbf{X}, \mathbf{Y})$ and was introduced by McConaghy in 2011 [43]. It is part of the larger field of symbolic regression which intends to fit a model to given data. Unlike traditional regression, symbolic regression also aims to find the symbolic form of a function. As an introductory showcase, solutions for a given data set $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{y})$ in symbolic regression could look like:

$$\begin{aligned} y &= \sin(x_1) + 4 \cdot x_2 \\ &\text{or} \\ y &= 2 \cdot x_1 + x_2 + 5 \cdot x1 \cdot x2 \end{aligned} \tag{3.7}$$

The main aspect of equations 3.7 is the symbolic forms of the models which were not fixed a priori but were derived in the symbolic regression process. The roots of symbolic regression lie in genetic programming [44] whereas fast function extraction uses a non-evolutionary approach.

The goal of FFX is to find pareto-optimal models in terms of model error and model complexity for given data $(\mathbf{X}, \mathbf{Y})$. The models are *generalized linear*, meaning they consist of a linear combination of N_B basis functions B_i, $i \in \{1, ..., N_B\}$:

$$\hat{y} = m(\mathbf{x}) = a_0 + \sum_{i=1}^{N_B} a_i \cdot B_i(\mathbf{x}) \tag{3.8}$$

The mean squared error $\mathbf{y} - m(\mathbf{x})$ is used to assess the model's error and the number of basis functions N_B serves as a measure of complexity.
To derive models from the training set, FFX employs pathwise regularized learning [36] based on elastic net, which was demonstrated in 3.3. The elastic net minimization formulation is posed in the following way:

$$\hat{\beta} = \min_{\beta} ||\mathbf{Y} - \mathbf{X} \cdot \beta||^2 + (1 - \rho) \cdot \lambda \cdot ||\beta||^2 + \rho \cdot \lambda \cdot ||\beta||_1 \tag{3.9}$$

By varying λ from 0 to 1, the regularization penalty is altered, facilitating the formulation of different models. Usually along the *path* of decreasing λ, models with a larger number of basis functions will emerge.

Fast function extraction comes in different flavours. In the following, a selected algorithm that will be used later is illustrated. By no means are all the features of FFX needed for this work, therefore the reader is referred to the original publication for the full FFX algorithm [43].

The FFX design which we use is depicted subsequently:

1. **Generate univariate bases**. The design matrix $\mathbf{X} \in \mathbb{R}^{m,p}$ is read in.
2. **Pathwise regularized learning**. For a user defined range of λ, models for the univariate bases are computed.
3. **Generate bivariate bases**. By assessing the coefficients of the models in step 2, k most important basis functions are used to create interacting, bivariate bases $x_i \cdot x_j \quad \forall i, j \in \{1, ...k\}$. Add these bivariate bases to the univariate ones.
4. **Pathwise regularized learning**. For a user-defined range of λ, models for the uni- and bivariate bases are computed.

5. **Non-dominating filtering**. Pareto-optimal models concerning error and complexity are selected. Therefore, a model m_i with N_i basis functions and error e_i is selected if it is not dominated by another model, i.e. if there is no model m_j with less complexity ($N_j < N_i$ basis functions) and smaller error ($e_j < e_i$).

The limitation of using k instead of all univariate bases for building bivariate bases is made because of algorithm complexity reasons. While the complexity without limitation is $O(m \cdot p^4)$, it is reduced to $O(m \cdot p^2)$ with limitation [43].

3.7 Random Forest

Random forest, first published by Breiman in 2001 [45], is one of the most successful learning algorithms today. It is suited to handle large data sets and scales with the volume of information. Random forest is a supervised learning method that can be used for classification and regression. It is popular because it can be applied to a wide range of prediction problems, it has few parameters to tune and it can successfully handle cases in which the number of variables is much larger than the number of observations.

A random forest is an ensemble of individual decision trees, commonly called CART [46]. Every decision tree is trained in its own way based on some randomly selected variables and casts a unit vote in a prediction task. Hence, the random forest prediction result is the average of the decision tree predictions. Random forest utilizes a divide-and-conquer approach meaning it is recursively breaks down a problem into sub-problems until the sub-problems are simple enough to solve. This is done by two essential components: Bagging [47] and CART-split criterion [46]. Bagging is a procedure that generates bootstrap samples from the original data set, trains a predictor from each sample and uses the average of those predictors as the overall prediction for an example. The CART-split criterion is used to construct individual decision trees. Based on this criterion, at each node of the tree suitable splits can be performed. The criterion is mostly based on residual sum of squares for regression and Gini criterion for classification [45] [48].

In the following, an excerpt of the **random forest algorithm**, which follows Breiman's original publication [45], is presented:

First, M_{trees} bootstrapped samples $\mathbf{X}^*_{iM}$, $iM \in \{1, ..., M_{trees}\}$ are drawn (with or without replacement) from the original dataset $(\mathbf{X_{orig}}, \mathbf{Y_{orig}})$ so that the probability of drawing an example from the dataset is uniform. The size of each bootstrapped sample is a fraction of the number of the original dataset.

For each bootstrapped sample $\mathbf{X}^*_{iM}$, a decision tree is built in the following way:

1. Select an unprocessed node and uniformly choose a subset $\mathbf{F_{sub}}$ with cardinality P_{sub} from all features f_i.

2. Perform a split at the selected node in one of the features $f_i \in F_{sub}$ maximizing the CART-criterion.

3. Recursively repeat step 2 until there are $leafsize$ or less observations at a node.

Consequently, user adjustable parameters are [45] [48]:

- $r = \frac{N_{bootstrap\ sample}}{N_{original\ dataset}}$: The size of each bootstrapped sample as a fraction of the number of the total training sample. For larger r, the individual trees tend to be more robust but also more similar to each other.

- The number of trees M_{trees}. Every tree has relatively low bias but can have substantial variance. By growing more decision trees, random forest's variance decreases and predictive performance advances up to a certain number of trees at which performance stabilizes. For real world problems, this number of trees is often around 50. Because of computational costs, one is interested in not choosing unnecessarily many trees.

- $leafsize$: The maximum number of observations in each terminal node, also called leaf. Default values are often 5 for regression and 1 for classification [49].

- The number of features P_{sub} being used to find an optimal split at each node. With little P_{sub}, individual trees tend to become more different which does not necessarily imply an increase in performance.

Even though random forests are relatively simple to utilize, it is important to note that mathematical analysis of the algorithm is not. Therefore, the properties of the random forest algorithm and the impacts of its parameters are still an active research topic [48][50].

A very useful and beneficial by-product when creating a random forest model is the possibility to estimate variable importances. This can be done in two variants. *Mean decrease impurity* measures the reduction of impurity of nodes for splits using one variable, averaged over all trees. The second variable importance estimate utilizes the *out-of-bag-error*, which is the average error for every $(\mathbf{X_i}, Y_i)$ calculated using the prediction trees that did not contain the individual $(\mathbf{X_i}, Y_i)$ in the bootstrapped training set. To estimate the importance of a variable f_i, we calculate the out-of-bag-error twice. Once with the *true* data and once for data in which the values of the variable f_i get randomly permuted. The difference in the two error measurements is called *mean decrease accuracy* (MDA) and reflects the importance of the variable for the prediction task [45] [48].

3.8 The Curse of Dimensionality

In many of today's pattern recognition and machine learning tasks, we are confronted with datasets with an immense number of examples (samples) and variables (dimensions). Considerations about the interaction of these two elements are critical for solving such tasks.

For example, based on Hastie et al's considerations [31] let us consider uniformly distributed observations on a line segment with length 1. After placing a new observation on the line segment, we are interested in the length of the sub-segment containing the new observation and 10% of the data. Since the observations are distributed uniformly, the expected length of the sub-segment is 0.1. Expanding the example to two dimensions yields a square with edge length 1 with uniformly distributed observations. We are now interested in the edge length of the sub-square that contains the new observation and 10% of the data. Since the observations are uniformly distributed, the expected area of the sub-square is 0.1, which means the

edge length is $\sqrt{10} \approx 0.316$.

To capture the closest 10% of the data, we need a fraction of 0.1 of the only variable in the 1-dimensional case but a fraction of 0.316 of both variables in the 2-dimensional case. If we do this experiment with ten dimensions, we need a fraction of $\sqrt[10]{0.1} \approx 0.794$ of each variable. Therefore, in high dimensions, we have to be careful about relying on methods that use *close* data points to predict new observations. The phenomenon of sparseness in high dimensions is often referred to as *the curse of dimensionality*, introduced by Bellman in 1961 [51] and 1964 [52].

A result of the sparseness in high dimensions is that all sample points are close to a boundary of the sample space. This is problematic because predictions are more difficult in such regions — rather than *just* interpolation between training samples, the prediction method needs to extrapolate from neighboring training data [31].

Breaking the curse of dimensionality can be done with an alignment of the number of data samples and dimensions. The sampling density is proportional to $m^{1/p}$. Therefore, a 1-dimensional problem with $m_1 = 100$ samples is as dense as a 10-dimensional problem with $m_{10} = 100^{10}$ samples. Reducing the number of dimensions and increasing data samples naturally densifies the problem space [31].

While the number of data samples will be treated as fixed in our experiments, the idea of dimensionality reduction will be examined thoroughly. Dimensionality reduction can be accomplished by feature extraction and feature selection. Both concepts will be covered in the succeeding sections.

3.9 Feature Extraction

As mentioned above, because of the *curse of dimensionality*, we are often interested in reducing the dimensions of a high dimensional variable space. However, we would like to retain as much information which is valuable for the prediction task as possible. An approach to achieve this goal is *feature extraction*, which transforms the variable space by some functional mapping [53]. By using a linear or non-linear function M_i, the original m variables can be mapped to $k < m$ new variables, called *features* f_i, $i \in \{1, ..k\}$:

$$\begin{aligned} f_1 &= M_1(v_1, v_2, ..., v_m) \\ &\dots \\ f_k &= M_k(v_1, v_2, ..., v_m) \end{aligned} \tag{3.10}$$

Obviously, the selection of suited mapping functions is crucial for the desired goal of extracting the maximum valuable information from the variables v_i. There are many methods that are commonly used today for computing those mapping functions. One of the most popular methods is *principal component analysis*, invented by Karl Pearson in 1901 [54] .

We will use mapping functions based on Pearson's correlation coefficient described in chapter 3.11. However, we will use *lots of* mapping functions as a first step, violating the $k < m$ premise[1]. In a subsequent step, the number of features will be limited to get $k < m$.

[1] For the case $k \geqslant m$, the term *feature construction* is also used in the literature.

3.10 Feature Selection

Feature selection is the task of choosing an *optimal* subset of $k < m$ features from an initial set of m features. This task is essential when many features are either redundant or irrelevant features but is non-trivial since the number of subsets is 2^m [55]. The optimality criterion is defined by an evaluation function. Feature selection methods are divided into three categories according to Guyon and Elisseeff [56]:

- **Wrappers** use the concurrent classifier's prediction as an evaluation metric. While they certainly are able to select a subset that is optimal for a given prediction task, the need to train a classifier for each investigated subset makes this variant highly time expensive.
- **Filters** are general and classifier-independent methods. They solely use the structure of the data and the class labels to come up with subsets. They are usually less time consuming. Yet commonly used evaluation metrics such as mutual information are not necessarily optimal for a given prediction task.
- **Embedded methods** perform variable selection or ranking in the training stage of a classifier. Like wrapping methods, these methods are specific to a given classifier and prediction task.

Earlier in chapter 3.3, we saw that lasso inherently performs feature selection. Chapter 3.7 introduced random forest's variable importance measure that can be used to (iteratively) select features. Both are examples of embedded feature selection methods.

In the following, selected filter feature selection methods are introduced that will be used later.

ReliefF

Relief algorithms were introduced by Kira and Rendell in 1992 [57] and have been steadily improved since. The first algorithm of the Relief family used the Euclidean distance as a distance metric and a quadratic error. Further advancements by Kononenko et al. in 1997 [10] introduced the Manhattan distance and the absolute error term (*ReliefF*). We use a publically available implementation [58] that is based on the version of Kononenko et al.

For a dataset with m instances, p variables and n_{cl} class labels, the algorithm starts with initializing a $1 \times p$ zero weight vector. Then, for $r < m$ iterations, the following three steps with the goal of weight updates are performed [59]:

1. Select a random instance R_j , $j \in \{1, ...m\}$.
2. From each class, select the k instances that are closest (by some distance metric) to instance R_j. The k instances from the same class as R_j are labeled H_l, the ones from every other class c as $M_l(c)$.

3. For all features $x_i,\ i \in \{1, ...p\}$ do:

$$w(x_i) = w(x_i) - \sum_{j=1}^{k} \text{diff}(x_i, R_j, H_l)/(m \cdot k) + \sum_{c \neq \text{class}(R_j)} \left[\frac{P(c)}{1 - P(\text{class}(R_j))} \sum_{j=1}^{k} \text{diff}(x_i, R_j, M_l(c)) \right] /(m \cdot k)$$

with:

$$\text{diff}(x_i, R_j, I_l) = \frac{|R_j(x_i) - I_l(x_i)|}{\max(\mathbf{x_i}) - \min(\mathbf{x_i})} \ .$$

Fisher Score

Fisher score is based on the idea that the distance of data, when using an optimal subset of features, is minimal within each class and maximal between each class compared to distances when using non-optimal feature subsets. Because of the great complexity, the algorithm is usually limited by considering each feature separately.

While more detailed descriptions can be found in the literature [60], the main part of the algorithm is presented as follows:

1. Select feature $f_j, j \in \{1, ...p\}$.
2. Compute the mean μ_k^j and standard deviation σ_k^j for feature j and class k.
3. Compute the mean μ^j for feature j for all classes.
4. Compute fisher score for feature j: $F(x^j) = \frac{\sum_{k=1}^{c} n_k \left(\mu_k^j - \mu^j\right)^2}{\sum_{k=1}^{c} (\sigma_k^j)^2}$ with n_k as the number of data points in class k.

For this work, a Python implementation of the scikit-feature selection repository [61] was used.

HSIC Lasso

In 2014, Yamada et al. proposed a method [62] inspired by lasso feature selection but designed to be able to detect non-linear input-output dependencies. This is done by using the Hilbert-Schmidt independence criterion (HSIC) which is based on kernel methods. Optimal features using this method are found by solving the following minimization problem, as shown in the original publication [62]:

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \quad & \frac{1}{2} \| \bar{\mathbf{L}} - \sum_{k=1}^{p} \alpha_k \bar{\mathbf{K}}^{(k)} \|_{Frob}^2 + \lambda \|\boldsymbol{\alpha}\|_1, \\ \text{so that} \quad & \alpha_1, \cdots \alpha_p \geqslant 0 \end{aligned} \tag{3.11}$$

where $\|.\|_{Frob}$ is the Frobenius norm, $\bar{\mathbf{K}}^{(k)} = \Gamma \mathbf{K}^{(k)} | \Gamma$ and $\hat{\mathbf{L}} = \Gamma \mathbf{L} \Gamma$ are centered Gram matrices, $K_{i,j}^{(k)} = K(x_{k,i}, x_{k,j})$ and $L_{i,j} = L(y_i, y_j)$ are Gram matrices, $K(x, x')$ and $L(y, y')$ are kernel functions, $\Gamma = \mathbf{I}_n - \frac{1}{p} \mathbf{1}_p \mathbf{1}_p^T$ is the centering matrix, $\mathbf{I}_p$ is the p-dimensional identity matrix and $\mathbf{1}_p$ is the p-dimensional vector with all ones.

3.11 Pearson Correlation Coefficient

Features used in this work are based on the *Pearson correlation coefficient*. For two continuous variables X and Y, the Pearson correlation coefficient ρ is defined as:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{3.12}$$

where:

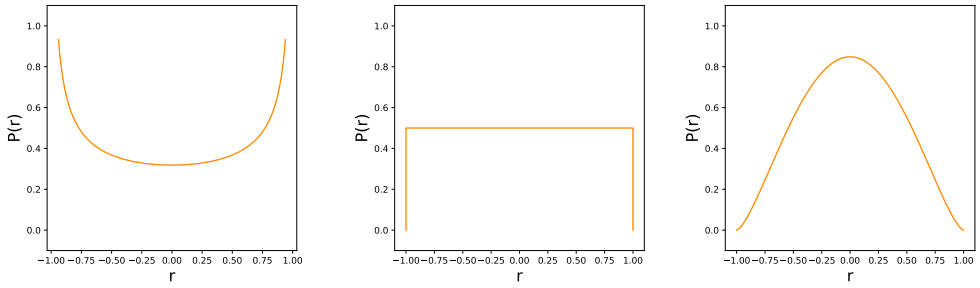
- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

While ρ is commonly used for the population correlation coefficient, r denotes the sample correlation coefficient. For two continuous samples with size n, $(x_i), (y_i)$ $i \in \{1, ...n\}$ denote the single samples and $\bar{x}$ and $\bar{y}$ stand for the sample means. The sample correlation coefficient r can then be computed by:

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}. \tag{3.13}$$

The correlation coefficient is a measure of linear relationship of the two samples.

The sampling correlation coefficient r is commonly used as an estimate for the population coefficient ρ. It's important to note that r follows a sampling distribution that depends on ρ and the sample size n. For a bivariate normal distribution, an analytical solution for the sampling density distribution was developed by Fisher [63].



(A) $n = 3$ (B) $n = 4$ (C) $n = 7$

FIGURE 3.5: The density function $P(r)$ of the sample correlation coefficient r for a normal bivariate distribution and a population correlation coefficient $\rho = 0$ depends on the sample size n.

Figure 3.5 shows the density distribution for samples with different sample sizes, drawn from two variables that are uncorrelated (which means $\rho = 0$) and both normally distributed. One can see for a very small sample size ($n = 3$), how the probability of $r = 0$ is the minimum of the density function. For $n = 4$, all values of r are equally likely. For increasing n, the probability that r is the true value $\rho = 0$ increases.

For non-normally distributed samples, one has to be aware of unpleasant effects when working with the Pearson correlation coefficient. Firstly, the distribution of r is different than it is for the normality assumption. Secondly, the choice of the correlation coefficient can be inappropriate to determine the linear relationship for particularly non-normal data [64].

Section 4.2 will introduce the features used for the experiments in detail. They all depend on the Pearson correlation coefficient although normality can not be guaranteed for all feasible subsets of the real-world data we use. As a consequence of the density distribution of r, the minimum sample size is set to 5.

3.12 Model Validation

Model validation is the process of evaluating the predictive power of a model. It is necessary to better understand and assess future predictions. Section 3.2 already revealed that model validation is often non trivial. There are different techniques for performance estimation including a popular method called k-fold cross validation [65].

We will focus on a *training and test set paradigm* (3.2), i.e. we use training data for building a model and a separate, independent test set for validating the model. For validation, every sample $\mathbf{X_i} \in \mathbf{X}_{test} \quad i \in \{1, ...m_{test}\}$ is passed to the random forest model and its class $Y_{predict}$ is predicted.

Binary classification

For a binary classification task comparing the predicted class $Y_{predict}$ with the true class Y_{true} can lead to one of the following outcomes:

- *True positive (TP)*: The classifier's prediction $Y_{predict}$ = 1 is correct ($Y_{true} = 1$).
- *False positive (FP)*: The classifier's prediction $Y_{predict}$ = 1 is false ($Y_{true} = 0$).
- *True negative (TN)*: The classifier's prediction $Y_{predict}$ = 0 is correct ($Y_{true} = 0$).
- *False negative (FN)*: The classifier's prediction $Y_{predict}$ = 0 is false ($Y_{true} = 1$).

Common metrics in machine learning are:

$$\text{Recall or sensitivity} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}}$$

$$\text{Specificity} = \frac{\sum \text{TN}}{\sum \text{TN} + \sum \text{FP}}$$

$$\text{Accuracy} = \frac{\sum \text{TP} + \sum \text{TN}}{m_{test}}$$

Usually, there is a trade-off between sensitivity and specificity: the more positive class instances that are correctly classified, the more negative class instances that are falsely classified as positives. Accuracy is a simple measure that computes the ratio of how many of all instances are classified correctly, independent of its class.

Multi-class classification

The methods of binary classification can be extended to meet multi-class classification requirements. Designating a correctly classified sample from class i as TP_i, accuracy for multi-class problem is simply defined as:

$$Accuracy = \frac{\sum_i \sum TP_i}{m_{test}}$$

While accuracy is not a reliable metric in all classification tasks (class skewness, different importance of classes) [30], it is suitable for the prediction tasks we are aiming at in this thesis.

More details about the model performance can be derived from a *confusion matrix*, a table that provides a quantitative overview of the predictions for each class. In such a matrix, index (i, j) displays how many samples of class i (true class) were classified as class j. Therefore, the diagonal of the tables shows how many instances from each class are correctly classified.

Chapter 4

Methods

In this work, we would like to find out if automatically extracted features are a useful foundation for predictive classification tasks in the spectroscopic domain. As a starting point of automated feature extraction, we always use a labeled dataset, i.e. m infrared spectra S_j, $j \in \{1, ...m\}$ of k different classes c_i, $i \in \{1, ...k\}$ as well as information about which spectrum belongs to which class (*ground truth*).

Chapter 3 presented different concepts tightly and loosely connected to feature extraction and prediction tasks. In this chapter, we are keen to find out which of these concepts are beneficial for the aim of automated feature extraction in spectroscopy.

In particular, we will present three different approaches we designed and analysed: *FFX approach*, *filter approach* and *embedded random forest approach*. The overall design of the algorithm is similar for all three of them and consists of the following steps:

1. Generation of a large number of features and creation of a design matrix.
2. Data split into training and test set.
3. Feature selection by three different approaches. Only the training set is used.
4. Build a random forest model with selected features. Only the training set is used.
5. Validation of the random forest model by applying it to a test set.

We call the overall algorithm *AutoFeature*, section 4.3 introduces the algorithm as well as its three versions of feature selection approaches.

4.1 FTIR Data Preprocessing

Processing and analysis of the hyperspectral data is done in the multisensor imaging tool ImageLab [66]. In ImageLab, a hyperspectral image is stored as a *hypercube*. We use three-dimensional hypercubes that consist of spatial coordinates (two dimensions) and wave numbers (third dimension). A pixel of the hyperspectral image refers to a spatial location of the hypercube (with x and y coordinates) and consists of a spectrum.

In ImageLab, pixels of hyperspectral images are selected that belong to a substance that we are interested in and the ground truth of these samples is assigned (see sections 5.2 and 5.3 for detailed selection procedures and label annotation for the different datasets used).

After having exported the spectral data of the selected samples as a *.csv* file, further operations and experiments are performed in Python [67].

4.2 Generic Features

As the discussion of the curse of dimensionality (chapter 3.8) has shown, an important issue in building prediction models is the potential sparseness of high dimensional spaces. We are therefore interested in a non-excessive number of features that carry a significant amount of information helpful for the prediction task.

In this work, we build generic features that are likely valuable in any modelling task with continuous spectra. Although these features are able to carry meaningful information, they still represent only a small fraction of possible feature designs. The generic features that will be presented below are part of a proof-of-concept approach that may be augmented by additional features in the future.

As discussed in chapter 2, different atoms and bonds that form molecules absorb infrared radiation of different energy. The resulting molecule-specific absorption bands have certain shapes and peaks. We want to build features that are able to *capture* the information about these specific shapes at different locations. We do this by creating generic shapes (see below) and computing the Pearson correlation coefficient of these template shapes and parts of the spectrum. With this approach, we aim to achieve dimensionality reduction since information about a part of the spectrum can be summed up with just the correlation coefficient, which is a single scalar.

Features based on the correlation coefficient have some interesting properties (see section 3.11 for a review of the Pearson correlation coefficient). Firstly, they only capture the information about the linear similarity of template and spectrum data. They are not influenced by linear transformations of the type $X \rightarrow a + bX$, $a, b \in \mathbb{R}$. This for example means that the features carry information about the shape of the spectrum independent of a linear baseline shift. Also, the features are independent of the magnitude of a linear slope of the spectrum. This property can be advantageous since absorption bands in spectra can be linearly scaled due to reflectance effects. However, a possible disadvantage is that some low-amplitude noise in the spectrum by chance takes on some shape that resembles a certain molecule's absorption band.

We designed 4 groups of template shapes that are presented subsequently.

Triangle-shaped Peaks

The first group of templates are triangle-shaped peaks. We define the width of a peak as the number of sample points (layers) at the triangle's base edge. This easy solution is good enough as a proof-of-concept that we are aiming for but would need to be improved in order to gain widths independent of spectral resolution. The width of these shapes is defined to be uneven (in order for the center of the peak to be an integer) and varies from length 5 to 59, see Figure 4.1 for triangle shapes with minimum and maximum width. We will denote a triangle-shaped peak with its center at position x_{center} and width w as $T(x_{center}, w)$.

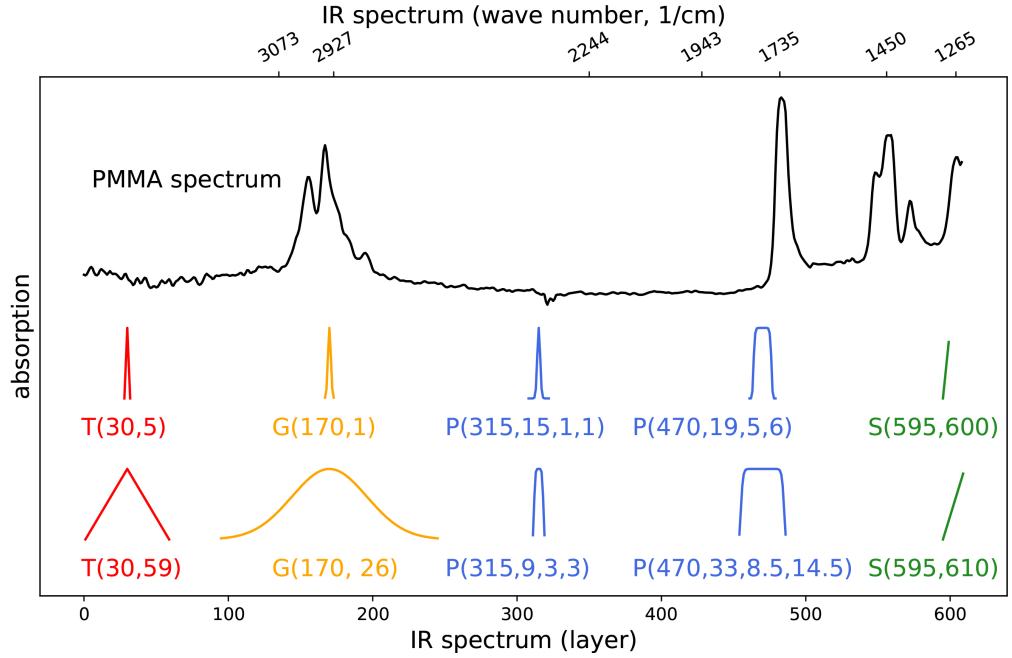


FIGURE 4.1: Selected features and a sample PMMA spectrum are depicted. For triangle- (red), Gaussian bell- (orange) and straight line features (green) the minimum and maximum width for each feature class is shown. For the general Gaussian bell feature class (blue), four shapes are selected for illustration.

Gaussian Bell

The second type of feature is Gaussian bell shaped. The Gaussian bell is defined by two parameters, the center position x_{center} of the symmetrical shape and the standard deviation σ. The shape is computed in two steps. Firstly, a preliminary version is computed with:

$$G(\sigma) = e^{-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2}. \tag{4.1}$$

where n is the number of points desired for the preliminary version. This, by default, is set to 151 data points, limiting the size of the preliminary version to some *large enough* length. The preliminary version is normalized in a way that its maximum value is 1. The final version of the Gaussian bell shape is defined by using the subset of $w(n)$ where $w(n) > 0.01$.

Center position x_{center} and σ (influencing the shape's width) are alterable parameters, setting the notion $G(x_{center}, \sigma)$ for a specific Gaussian bell feature. Figure 4.1 depicts Gaussian bell features with minimum and maximum width.

General Gaussian Bell

The next feature type is a generalization of the Gaussian bell. It is able to extend the ordinary Gaussian bell feature so that the bell curve can have broader or sharper peaks, steeper or flatter slopes and longer tails. The feature takes three input pa-

rameters:

- n ... number of data points of the feature.
- p ... shape parameter
- σ ... standard deviation

The general Gaussian bell feature, denoted by G(n,p,σ), is computed by:

$$G(n, p, \sigma) = e^{-\frac{1}{2}|\frac{n}{\sigma}|^{2p}}. \tag{4.2}$$

In Figure 4.1, four types of the general Gaussian bell feature are illustrated.

Straight Line

The fourth kind of template is a straight line. The feature's single input parameter is the number of output data points n. The parameter n is odd and $5 \leqslant n \leqslant 25$, see Figure 4.1 for the straight line features with minimum and maximum length.

The line is computed so that the straight line's first value is 0 and the last value is 1. Lines of different length have therefore different slope which is irrelevant since the correlation coefficient is independent for such kind of linear transformations. The underlying reason for using this feature is that we sense there might be statistical differences of the correlation with a straight line in parts of the spectrum with lateral peaks and in other parts without them.

4.3 AutoFeature Algorithm

This section presents AutoFeature, the algorithm designed in this thesis for automated feature generation and selection. Figure 4.2 illustrates the workflow of the algorithm.

Feature Generation and Creation of Design Matrix

Firstly, features are generated which, as presented above, are based on the correlation of parts of the spectrum and templates. Four different types of shapes and different forms and widths of each type constitute a total of 84 templates. These templates can be used at any position of the spectrum, yielding thousands of features for a typical infrared spectrum. Applying all n_f features to all m spectra samples yields the design matrix $\mathbf{X} \in \mathbb{R}^{m \times n_f}$. Class information is saved in ground truth vector $\mathbf{Y} \in \mathbb{R}^{m \times 1}$.

The design matrix together with the ground truth vector $(\mathbf{X}, \mathbf{Y})$ is split into the *training* and *test set*. The split is done in a way so that 80% of each class' data is contained in the training set, the remaining 20% in the test set. Hence, the proportions of the number of samples of each class are the same in full data, training data and test data.

The training data is used for all of the following steps until building a (final) random forest model. The test set is not used in any step of model creation but for validating the random forest model ultimately.

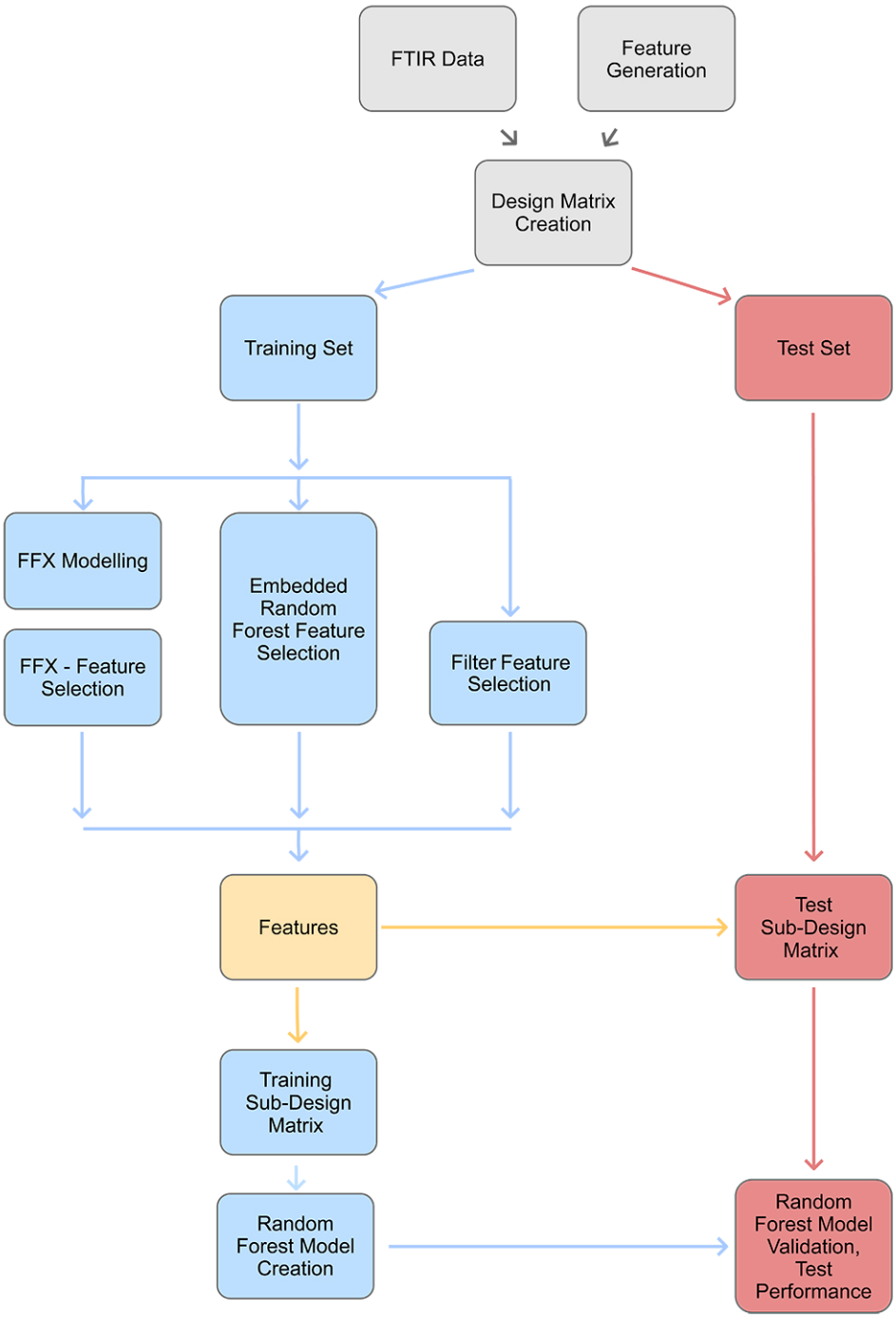


FIGURE 4.2: Illustration of the AutoFeature algorithm. The algorithm starts with the generic feature generation. After employing the generic features on spectroscopic data, a design matrix is obtained that is split into a training and a test set. One out of three feature selection approaches can be selected by the user. The resulting features from the feature selection process are used to form sub-design matrices, both for the training and the test set. A random forest model is trained with the training sub-design matrix and is then evaluated with the test samples.

Feature Selection

In the next step, the training data is used to determine the most promising features for prediction out of the thousands features generated in step 1. This can be done in three different ways:

FFX approach

User adjustable parameters are:

- n_{features}: maximum number of features that are selected.
- ρ: elastic net regularization mixing parameter.
- $n_{\text{univariate}}$: maximum number of univariate basis functions for pathwise regularized learning (FFX step 2).
- $n_{\text{main univariate}}$: number of univariate basis functions that is used to compute the bivariate basis functions (FFX step 3).
- $n_{\text{bivariate}}$: maximum number of bivariate basis functions for pathwise regularized learning (FFX step 4).

Fast function extraction (see section 3.6) is utilized for feature selection. This is done by FFX's modelling method so that generalized linear models are computed with elastic net regularization. Features are selected by choosing appropriate basis functions of these models.

Firstly, the design matrix computed in the previous step is used to perform pathwise regularized learning with fixed elastic net regularization parameter ρ. During pathwise regularized learning, the regularization parameter λ is decreased stepwise until a resulting model consists of $n_{\text{univariate}}$ univariate basis functions. $n_{\text{main univariate}}$ most important basis functions are used to generate bivariate basis functions of the form $x_i \cdot x_j \quad \forall i, j \in \{1, ...n_{\text{main univariate}}\}$. After merging the univariate and bivariate basis functions, pathwise regularized learning is performed again until a model results with maximum $n_{\text{bivariate}}$ basis function.

Subsequently, a non-dominating filtering of the resulting models yields a subset of models that are pareto-optimal concerning error and complexity. Eventually, the basis functions of the model with complexity n_{features} are selected as features. If there is no model with the chosen complexity level n_{features} in the non-dominated subset, the complexity level is decreased to the next possible (part of non-dominated subset) model complexity.

The FFX feature selection approach is carried out in a binary *one-vs-all* fashion. This means, for k different classes, k runs have to be performed. In each run, one class c_i is selected as the positive class ($Y_{binary} = 1$) and all other classes $c_j, \quad j \neq i$ are selected as negative classes ($Y_{binary} = 0$). Then, with the temporary Y_{binary} labels, the feature selection procedure is carried out. In each run, we therefore obtain the most important features for distinguishing class c_i from the other classes. After all runs, the most important features are merged. If a maximum number of n_{features} features is desired for a multi-class problem, in each run $\frac{n_{\text{features}}}{k}$ features have to be selected.

Filter approach

User adjustable parameters:

- n_{features}: number of features selected.
- $\texttt{ykernel}$ [HSIC lasso]: type of kernel method.
- $n_{\text{neighbors}}$ [ReliefF]: number of neighbors for weight updates.

In the filter approach, features get selected by three filter methods *HSIC lasso*, *ReliefF* and *FisherScore*, see section 3.10. The outcome is three sets of features.

For each of the filter algorithms, the training design matrix $\mathbf{X}_{\texttt{train}}$ and ground truth vector $\mathbf{Y}_{\texttt{train}}$ are used as input. Every filter method computes a ranking of all input features and n_{features} most important features are selected.

Like the FFX approach, all filter approaches are carried out in a binary one-vs-all fashion.

Embedded random forest approach

User adjustable parameters:

- n_{features}: number of features selected.
- $q_{\texttt{discard}} \in [0, 1]$: fraction of features that is discarded after each run.
- random forest parameters, see section 3.7

In the embedded random forest approach, the random forest's variable importance measure is used, see section 3.7. Features selection is done by the following iterative scheme:

1. Use training set $(\mathbf{X}_{\texttt{train}}, \mathbf{Y}_{\texttt{train}})$ to build a random forest model.
2. Select a fraction of $1 - q_{\texttt{discard}}$ features that have the largest variable importance measured by the random forest's mean decrease impurity. Discard the rest.
3. Repeat step 1 (with remaining features) and 2 until the number of remaining features is less than or equal to n_{features}.

Feature Sub-Design Matrix and Random Forest Model Creation

User adjustable parameters:

- random forest parameters, see section 3.7

The resulting n_{features} selected features in the previous step are used to create *sub-design matrices*. A sub-design matrix $\mathbf{X}_{\texttt{sub}} \in \mathbb{R}^{m \times n_{\text{features}}}$ is a submatrix of the full design matrix $\mathbf{X} \in \mathbb{R}^{m \times p}$ so that all samples but the selected features are used.
In this way, a training sub-design matrix and a test sub-design matrix can be constructed from the full training design matrix and full test design matrix respectively.

The sub-design matrix is used to create a random forest model aiming to build a valid predictive statistical model. Depending on the feature selection method used, the random forest model is labeled as $\text{RF}_{\texttt{FFX}}$, $\text{RF}_{\texttt{HSIC}}$, $\text{RF}_{\texttt{Fisher}}$, $\text{RF}_{\texttt{ReliefF}}$ or $\text{RF}_{\texttt{embedded}}$.

Validation of Random Forest Model

To validate the built random forest models, the prediction performance is assessed on the test sub-design matrix. In particular, the accuracy and confusion matrices (see section 3.12) are computed and presented. By assessing these metrics of the different models $RF_{\texttt{selection method}}$, we can analyse which of the approaches might be more suitable for the spectroscopic domain than others.

Chapter 5

Experiments

In this chapter, the set-up of the conducted experiments is presented. First, we will investigate how the AutoFeature FFX-approach is affected by noise and the sample size of the dataset. Therefore, we will create artificial hyperspectral data.

In the next experiment, we will use a real-world microplastic dataset that contains different types of polymers. All AutoFeature variants will be carried out — the resulting features will be discussed as well as model performances evaluated and compared.

Finally, we will attempt to solve two skin tissue classification tasks with the AutoFeature embedded random forest approach. While in the first task a discrimination of tumorous melanoma and non-tumorous epidermis cells is desired, the second task is about differentiating connective tissue and non-connective tissue in the dermis.

5.1 AutoFeature Investigation with Artificial Data

To investigate the functionality of FFX in the feature selection domain, we create artificial data. The data is designed to emulate real-world spectra. Each artificial spectrum consists of 609 data points, equivalent to the microplastic spectra used in section 5.2.

We are particularly interested in the question of how well FFX works with noisy data. Therefore, we add different levels of noise to the artificial spectra and observe the results of FFX.

The artificial data emulates three different substances. Each substance has just one, triangle-shaped absorption band. Moreover, the width of the triangles is set to 15 datapoints for all three classes and every spectra's baselines are arbitrarily shifted vertically by 0.5. The center positions of the classes are placed on three consecutive data points. The spectra with center locations on the outside are labeled as *class 0* while the ones in the middle are labeled as *class 1*, as shown in Figure 5.1.

Two artificial datasets with different sample sizes are created. Dataset *D1* contains 8 class 1 samples and 4 samples from each type of class 0 while dataset *D2* contains 50 class 1 samples and 25 samples for each class 0 type. Table 5.1 summarizes the noise-free specifications of the spectra.

As a next step, we add normally distributed homoscedastic noise to the samples of each dataset. The mean of the noise is set to zero while the standard deviation is increased stepwise, taking one of the values $\mathtt{std}_{set} = \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25\}$. After adding noise with a chosen standard deviation $s \in \mathtt{std}_{set}$ to all samples, the FFX approach is followed. The elastic net mixing parameter is set to $\rho = 0.9$ and $n_{features} = 10$ features are selected (see Table 5.2). For each dataset and

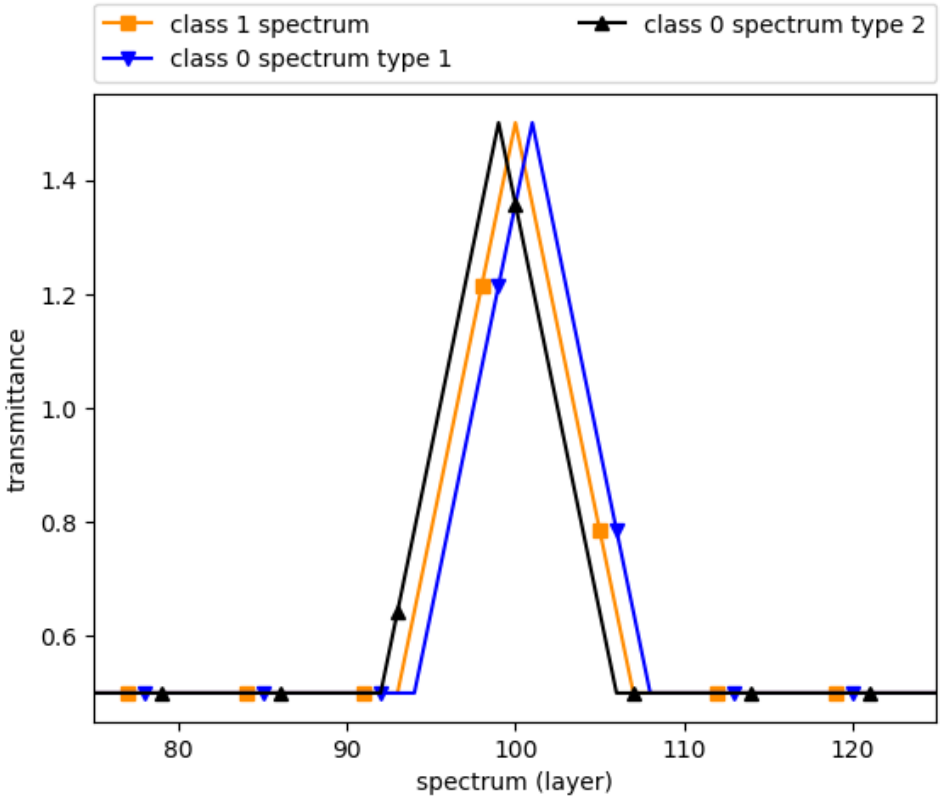


FIGURE 5.1: Segment of artificial data. Three classes are designed; two of them are treated as negative class (class 0), one as positive class (class 1). All spectra consist of 609 data points which are all constant except for the segment shown in this figure. The width of the triangle shaped absorption bands is 15 for all classes, the central positions are 99 and 101 for the two types of class 0 and 100 for class 1.

TABLE 5.1: Specifications of artificial data that is used to investigate properties of fast function extraction in feature selection. Three types of spectra are created, one of them is treated as positive class (class 1), the others as negative class (class 0).

	Class 1	Class 0, type 1	Class 0, type 2
Position	100	99	101
Width	15	15	15
Amplitude	1	1	1
Baseline shift	0.5	0.5	0.5
samples (D1)	8	4	4
samples (D2)	50	25	25

for each level of noise the resulting FFX model is observed with ten features. We further visualize *which* features are selected by plotting the width of the features against their central positions. For different datasets and different levels of noise, we observe the patterns and distinctions of these 2D distributions of the selected features. Convincing FFX models should consist of features that at least partly cover the spectral region of interest, i.e. $[92, 108]$, preferably being close to the specification of the positive class $\mathrm{T}(100, 15)$. We suspect that increased noise will

result in random-based, meaningless features and want to check how strongly this aspect is influenced by the standard deviation of the noise and the size of the dataset.

To keep the analysis simple, only triangle-shaped features with a width between 5 and 80 are permitted in the FFX runs.

TABLE 5.2: FFX parameters used for experiments with artificial data.

Parameter	Value
Regularization ratio ρ	0.9
n_{features}	10
$n_{\text{univariate}}$	20
$n_{\text{main univariate}}$	20
$n_{\text{bivariate}}$	no limit

5.2 AutoFeature Experiment with Microplastic Data

Microplastic Data Collection

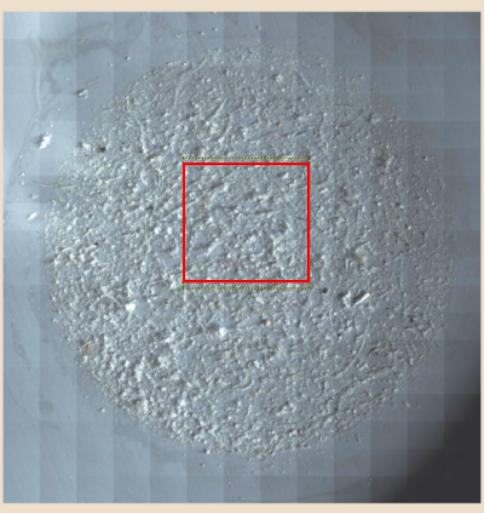


FIGURE 5.2: Aluminium oxide filter containing the microplastic photographed in visible light. The area within the red rectangle is selected for all further analysis.

The microplastic dataset was collected by a research group at the Faculty for Biology, Chemistry and Earth Sciences at the University of Bayreuth. The dataset was acquired by first taking water from natural, running water with all contained organic substances. Then, microplastic particles of different polymers were mixed into the water. The water, containing organic and polymer substances was filtered through an aluminium oxide filter with a meshsize of 100 nm.

By using a FPA-based micro-FTIR spectroscopic microscope [68], the filter and the microplastic-containing specimens were measured in transmission mode. Further properties of the measurement are a wave number range of 3600 - 1250 cm^{-1}, a resolution of 8 cm^{-1} and a coaddition of six scans.

We select an arbitrary subregion of the full data sample that we use for all further

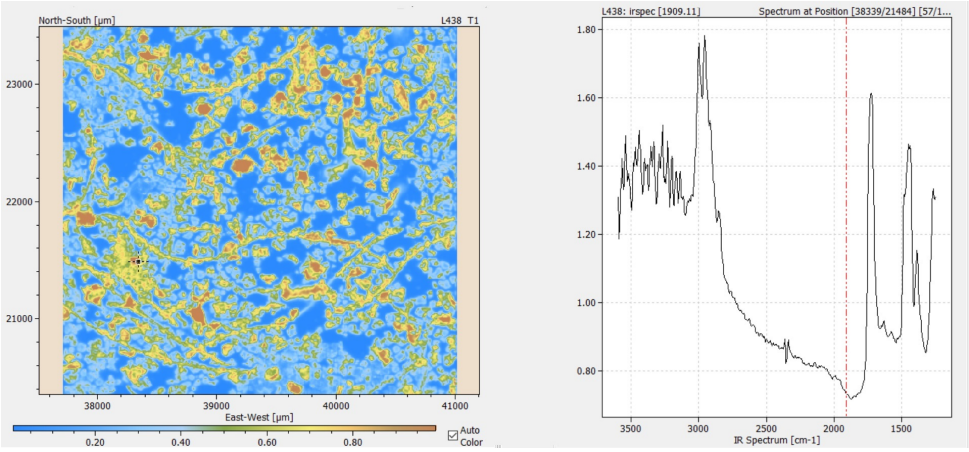


FIGURE 5.3: The left side of the figure displays a hyperspectral image of the microplastic sample. The absorbance at wavenumber 1901 cm^{-1} is shown. For each pixel of the hyperspectral image, there is a spectrum. The spectrum at the cursor's position is shown on the right side of the figure.

analysis, see Figure 5.2. This subregion is depicted in Figure 5.3 in the hyperspectral domain. In particular, the figure shows the absorbance at wavenumber 1901 cm^{-1} (arbitrary choice) for each pixel.

Microplastic Dataset for AutoFeature Experiments

We make a selection of datapoints for further analysis and for automatic feature generation experiments that is described in the following paragraph.
The aim of the selection is to obtain a dataset with real-world spectra for different polymers present in the specimen. Having many examples of real-world spectra increases the chance that the dataset will contain important variations of spectra for the same chemical substance. These variations occur due to different physical effects in spectroscopy.

In previous inquiries, spectral features for polymer detection and classification were manually designed using ImageLab. The features contain important information about intensities, integrals, intensity ratios, peak positions and shapes. Also, a classification model based on random forest was built in this study. These features together with the classification model are used to classify each pixel in the selected region of the microplastic data. All pixels classified as any kind of polymer are visually inspected. The spectra are compared to a reference spectrum and classification decisions are manually changed if they do not seem plausible. One special case is pixels that contain two or more different spectra. Those are annotated as 'mixed spectra' but are not used any further in the experiments.

Moreover, we used some features together with principal component analysis to detect different classes in the dataset. The scores for different principal components are used to identify candidates for the final polymer dataset. After visual inspection of the spectra of the candidates, pixels are annotated.
Merging the annotated pixels from the feature-classification and feature-PCA method gives the final database used for feature generation experiments.

Microplastic AutoFeature Experiment

The microplastic dataset, collected and selected as described above, is used for an AutoFeature (see section 4.3) experiment. Settings and parameters used for all steps in the AutoFeature workflow are presented in this section.

Feature Generation

All four feature shapes including triangle, Gaussian bell, general Gaussian bell and straight line, presented in section 4.2, are used at all possible spectral positions.

Training and Test Set

Standard settings of the AutoFeature algorithm are used, i.e. 80% of the data is used for training and the remaining 20% is used for testing.

Feature Selection

For feature selection, for each class c_i we randomly select $\min(m_{c_i}, 100)$ spectra from the training set, where m_{c_i} is the number of spectra for class c_i. These spectra are used for all feature selection approaches.

Tables 5.3 (FFX approach), 5.4 (embedded approach) and 5.5 (filter approach) list the parameters used for the different feature selection approaches.

TABLE 5.3: FFX feature selection approach parameters used for experiments with microplastic data.

Parameter	Value
Regularization ratio ρ	0.9
n_{features}	50
$n_{\text{univariate}}$	400
$n_{\text{main univariate}}$	400
$n_{\text{bivariate}}$	no limit

TABLE 5.4: Embedded feature selection approach parameters used for AutoFeature experiments with microplastic data.

Parameter	Value
n_{features}	50
q_{discard}	0.995
sample fraction r	1
number of trees M_{trees}	50
leafsize	1
features at split P_{sub}	$\sqrt{n_{\text{features in design matrix}}}$

Feature Sub-Design Matrix and Random Forest Model Creation

For creating the training and test sub-design matrices, we use all available spectra in the training and test sets.

The parameters used for the final random forest model in this experiment are shown in Table 5.6.

TABLE 5.5: Filter feature selection approach parameters used for AutoFeature experiments with microplastic data.

Parameter	Value
n_{features}	50
ykernel [HSIC lasso]	'Delta'
$n_{\text{neighbors}}$ [ReliefF]	0.8

TABLE 5.6: Random forest model building parameters used for AutoFeature experiments with microplastic data.

Parameter	Value
sample fraction r	1
number of trees M_{trees}	50
leafsize	1
features at split P_{sub}	$\sqrt{n_{\text{features in design matrix}}}$

Firstly, we are interested in the resulting features from the FFX approach and will have a closer look at them. For each of the five binary one-vs-all FFX runs, we select the model with complexity $n_{\text{features}} = 10$. We manually select some features of these models that we consider *distinct* (i.e. we would rather consider features $T(234, 14)$ and $T(511, 13)$ to be distinct than features $T(234, 14)$ and $T(236, 12)$). These distinct features are illustrated together with microplastic spectra. So, a visual evaluation of the reasonability and meaning of the automatically selected features is possible.

To better understand why some features are selected by the FFX approach, we look at the histograms of these features. These histograms show the feature value distributions for the positive and the negative class samples. We expect some differences in the two distributions to occur since we expect the algorithm to select features that are different for the positive and the negative class. By inspecting the histograms, the question of *how* those distributions differ can be answered. Moreover, we manually choose a feature that an expert in this field would find appropriate and that is not among the selected features by FFX. For this feature, the histograms for the positive and the negative classes are illustrated again. Distribution differences for the positive and negative class of the manually selected feature are compared to the distribution differences of the FFX selected features.

Similarly, we will examine four features selected by the embedded random forest approach together with a manually chosen feature that an expert in this field would find interesting. Since random forest can handle multiclass problems, the histograms for every class is displayed.

To compare the results of all five feature selection methods, the widths of the selected features are plotted against the positions of the features for each method. While the information about the shape of the features is not included in this approach, differences and similarities in which features are selected by different methods are easily perceived.

Eventually, random forest models are built with the automatically extracted features and validated on the test set that has not been used in any step before.

FFX Stability Examination

We regard an automatic feature generation and selection algorithm as *stable* if similar features are selected for similar real-world data. If similar data results in very different features, we call the algorithm *unstable*. In the latter case, we would consider the underlying models to have low bias and high variance which would make the resulting features unconvincing.

We consider the spectra within each polymer class in the microplastic dataset to be relatively similar. To assess the stability of FFX when dealing with real-world data, we design the following experiments. Firstly, four samples (named A, B, C and D) with an equal number of spectra and equal class distributions are drawn from the microplastic dataset in the following way:

1. Set a number of spectra per class $n_{\text{spectra per class}}$ that is to be included in every sample.
2. For all five polymer classes c_i do the following:
   - if the number of spectra of class c_i is larger than $5 \cdot n_{\text{spectra per class}}$: Randomly draw $n_{\text{spectra per class}}$ spectra for each sample without replacement.
   - if the number of spectra of class c_i is smaller than $5 \cdot n_{\text{spectra per class}}$ but larger than $n_{\text{spectra per class}}$: Randomly draw $n_{\text{spectra per class}}$ spectra for each sample with replacement.
   - if the number of spectra of class c_i is smaller than $n_{\text{spectra per class}}$: Add all available samples of class c_i to all samples. If this condition is fulfilled, the class distribution in the samples will be imbalanced.
3. For each of these samples, a full FFX AutoFeature algorithm is followed.

Three sets of four samples with $n_{\text{spectra per class}} \in \{32, 64, 200\}$ are created.

To examine the similarities and differences in the resulting features for the four different samples, we plot the feature-width against the feature-position as described above. We do this for all three chosen $n_{\text{spectra per class}}$ cases and observe the effect of the sample size on the results.

5.3 AutoFeature Experiment with Skin Tissue Data

In the experiments described below, real-world skin tissue data is used for two important classification tasks in medicine. We will use the AutoFeature algorithm to automatically find features for the following tasks:

- Classification of tumorous (melanoma) and non-tumourous (not melanoma) cells in the epidermis.
- Classification of connective tissue and non-connective tissues in the dermis.

Melanoma and Connective Tissue Dataset

The skin tissue specimens used in this thesis stem from the Department of Pathophysiology and Allergy Research at the Medical University of Vienna [69]. The specimens are formalin fixed and paraffin embedded (FFPE), a common method for conservation and stabilisation of biological tissue before microsections and

examinations with a microscope. $CaFl_2$ is used as sample carriers.

For each tissue section that is extracted for spectroscopic analysis, neighboring tissue sections are extracted as well, FFPE processed and H&E stained. The H&E stained samples enable a conventional analysis of the tissues and a ground truth annotation by an expert in histopathology.

We expect to obtain spectra with some characteristic properties for each type of tissue that occurs in the epidermis and dermis because of the following reasons:

Epidermis: The most prevalent cell type in the epidermis is keratinocyte, consisting of a number of structural proteins, enzymes, lipids and antimicrobial peptides. One of these structural proteins, keratin is expected to change absorption bands.

Melanoma: Because tumorous cells have abnormal and rapid reproduction cycles, more DNA and RNA is present in these cells than in non-tumorous cells, altering the spectrum.

Connective tissue: Structural proteins collagen and elastin, among others, influence the spectrum.

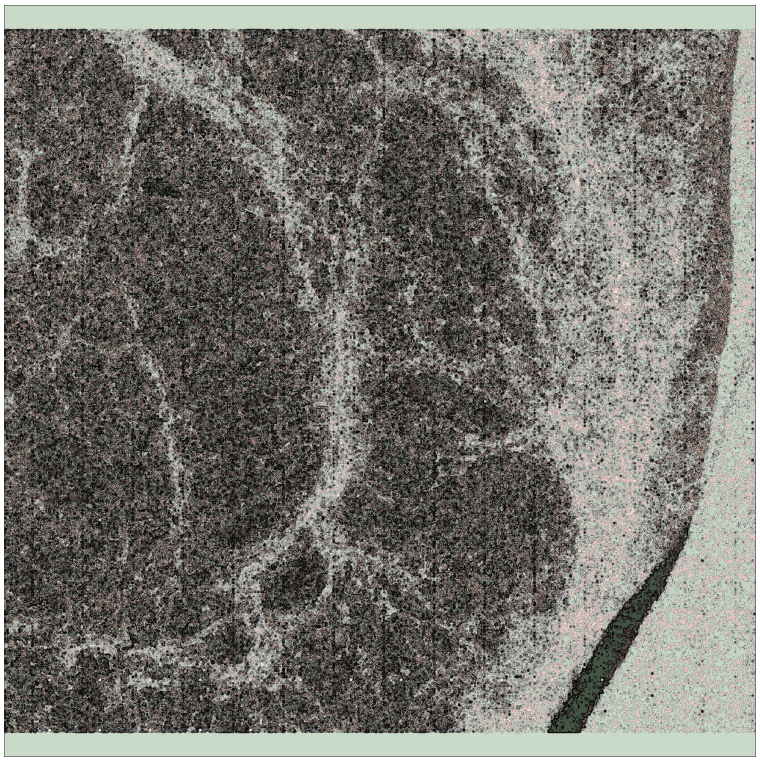


FIGURE 5.4: Picture in visible light taken from the dermis specimen containing the connective tissue.

Hyperspectral Image Acquisition

Hyperspectral images are recorded in transmission mode on an FTIR-microscope Bruker Hyperion 3000 with a liquid nitrogen cooled 64×64 pixel FPA detector and a sample area of 175×175 µm. Using a 15-fold objective with a pixel resolution of 2.7 µm together with a 4×4 binning produce a final resolution of $2.7 \cdot 4 = 10.8$ µm.

To increase the signal to noise ratio, four scans for each pixel are accumulated. The spectra of the tissues are obtained between $\hat{\nu} = 3845$ and 879 cm^{-1} with a spectral resolution of 2 cm^{-1}. The spectra of the datasets are then resampled by a factor of 2.

Melanoma Dataset for AutoFeature Experiments

Pixel areas of the hyperspectral image are labeled as *melanoma* or *non-melanoma* if the H&E stains provide enough information about their ground truth, i.e. if areas that belong to melanoma cells or to usual epidermis cells can be recognized. Outside of these pixel areas, 40 pixels are randomly selected for each class. These overall 80 pixels with their corresponding tumorous or non-tumorous spectra form the dataset for the following automatic feature generation and selection experiments.

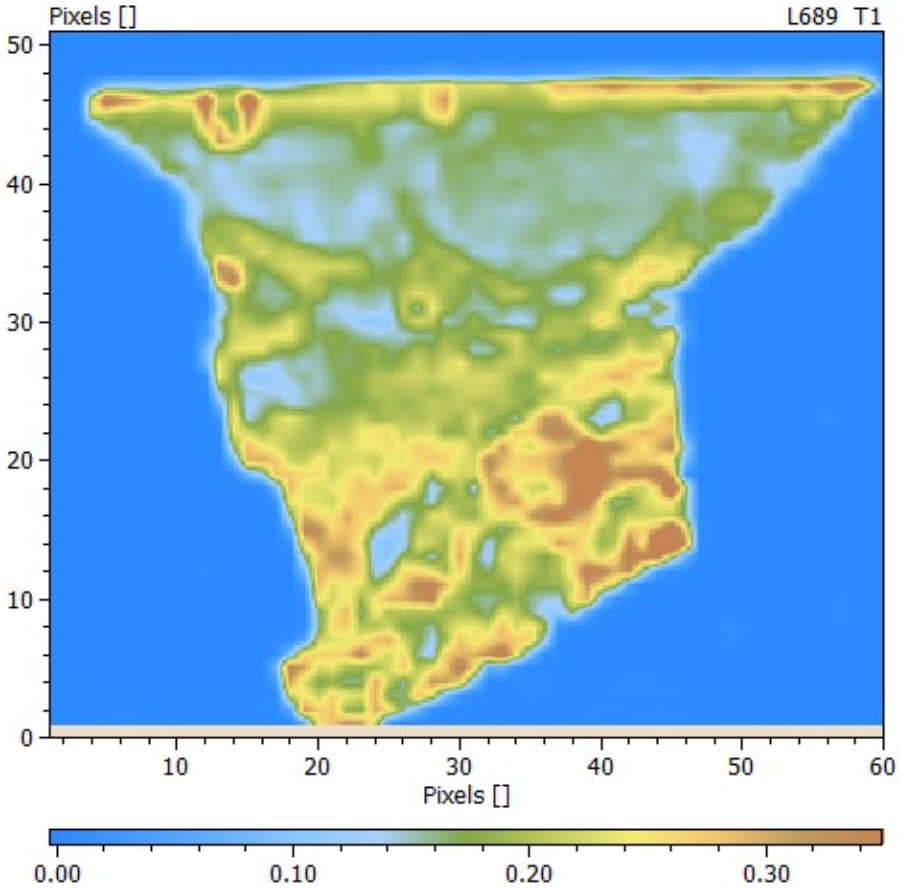


FIGURE 5.5: Image of epidermis containing melanoma cells showing the absorbance for wavenumber 1307 cm^{-1}.

Connective Dataset for AutoFeature Experiments

The selection of the connective tissue dataset for the AutoFeature experiments is done analogously to the melanoma dataset. 40 pixels for both classes *connective tissue* and *non-connective tissue* are chosen to be included in the dataset. Figure 5.4 shows an image of the specimen in visual light, Figure 5.6 displays it at a single wavenumber in the infrared spectrum.

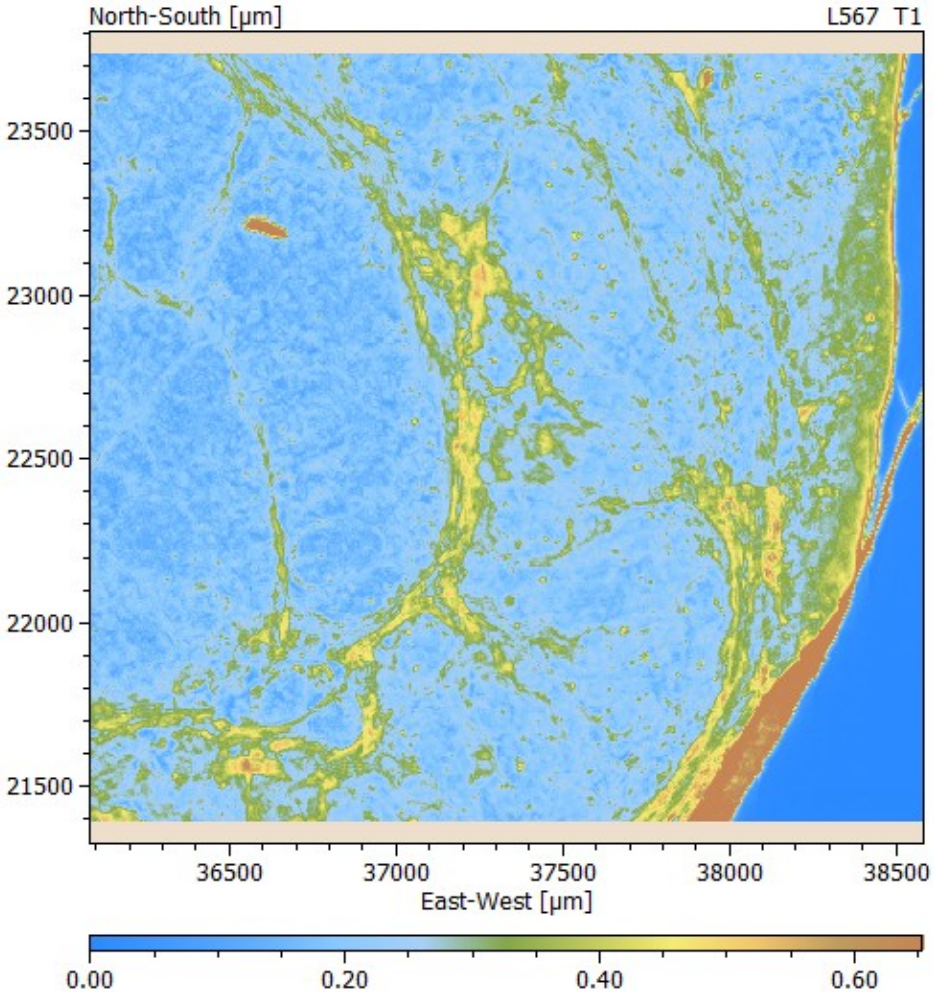


FIGURE 5.6: Image of dermis containing connective tissue showing the absorbance for wavenumber 1664 cm^{-1}.

AutoFeature Algorithm Settings for Skin Tissue Experiments

For melanoma and connective tissue classification tasks, the same settings are used in the AutoFeature algorithm and are described in the following.

Feature Generation

All four feature shapes (see section 4.2) are used with standard settings.

Training and Test Set

Standard settings as described in section 4.3 are used, i.e. 80% of the data is used for training and the remaining 20% is used for testing.

Feature Selection

For feature selection, the full training dataset together with the embedded random forest approach is used. Table 5.7 lists the parameters used for the chosen AutoFeature version.

TABLE 5.7: Embedded feature selection approach parameters used for experiments with skin tissue data.

Parameter	Value
n_{features}	25
q_{discard}	0.995
sample fraction r	1
number of trees M_{trees}	50
leafsize	1
features at split P_{sub}	$\sqrt{n_{\text{features in design matrix}}}$

Feature Sub-Design Matrix and Random Forest Model Creation

The full training and test sets are used to create the sub-design matrices for the random forest modelling and validation.

The random forest parameters used for the final random forest model in this experiment are listed in Table 5.8.

TABLE 5.8: Random forest model building parameters used for experiments with skin tissue data.

Parameter	Value
sample fraction r	1
number of trees M_{trees}	50
leafsize	1
features at split P_{sub}	$\sqrt{n_{\text{features in design matrix}}}$

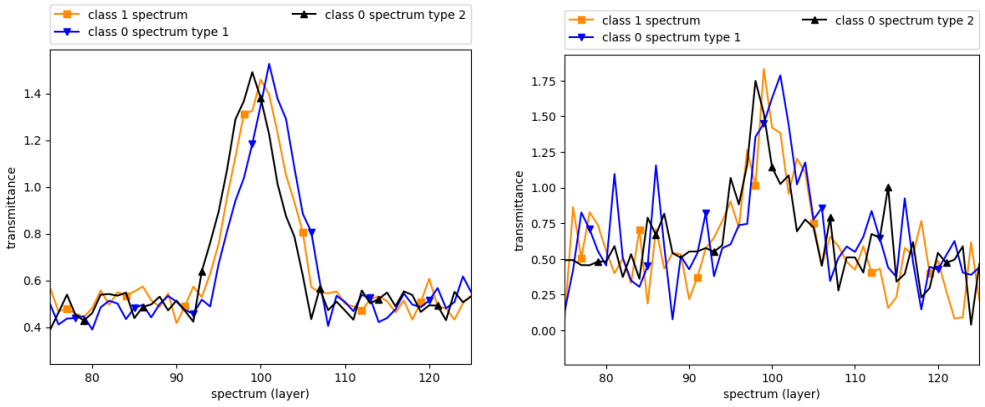
After building a random forest model with 25 selected features, we validate it on the test set that consists of $40 \cdot 0.2 = 8$ samples per class.

Chapter 6

Results

6.1 Results of AutoFeature Investigation with Artificial Data

As described in section 5.1, we increase stepwise the standard deviation of noise added to artificially generated spectra. Figure 6.1 shows segments of randomly chosen spectra for each class after adding noise with standard deviations $\sigma_{noise} = 0.05$ and $\sigma_{noise} = 0.2$. For noise $\sigma_{noise} = 0.05$, a clear distinction between the peaks and the baseline can be made by visual inspection. One can also anticipate the underlying denoised spectra for the different classes. For $\sigma_{noise} = 0.2$, both the distinction of the triangles from the baseline and the class discrimination is much harder to make.



(A) $\sigma_{noise} = 0.05$. (B) $\sigma_{noise} = 0.2$.

FIGURE 6.1: Randomly chosen spectra for each class after adding noise with standard deviation $\sigma_{noise} = 0.05$ (A) and $\sigma_{noise} = 0.2$ (B). (A) Clear distinctions between classes and between peaks and baseline can be made by visual inspection. (B) Distinctions between different classes and between baseline and triangle shaped absorption bands are difficult to make by visual inspection.

The resulting FFX models for the different noise levels are depicted in Table 6.1 for dataset D1 and in Table 6.2 for dataset D2. Figure 6.2 (D1) and Figure 6.3 (D2) provide a visual presentation of the features' position and width.

TABLE 6.1: Results of fast function extraction model building for artificial data (dataset D1) with sample size $m = 16$. The central position of the noise free class 1 spectrum is at layer 100, the width is 15 layers. FFX is able to extract features located at layer 100 for low amplitude noise $\sigma_{noise} \leqslant 0.1$ and extracts features based solely on random effects for noise $\sigma_{noise} \geqslant 0.15$.

σ_{noise}	FFX model
0.01	-1.80 + 0.511·**T(100,21)** + 0.504·**T(100,17)** + 0.390·**T(100,19)** + 0.298·T**(100,15)** + 0.296·**T(100,23)** + 0.208·**T(100,13)** + 0.136·**T(100,11)** + 0.0914·**T(100,9)** + 0.0399·**T(100,7)** + 0.0157·**T(100,5)**
0.05	-4.93 + 2.93·**T(100,15)** + 1.73·**T(100,13)** + 1.06·**T(100,11)** + 0.146·**T(100,7)** + 0.0834·T(261,61) - 0.0662·T(240,57) + 0.0407·T(43,67) + 0.0222·T(359,9) - 0.000890·T(237,49)
0.1	-0.786 - 2.04·**T(100,9)** · T(87,27) - 0.682·**T(100,11)** · T(87,27) - 0.185·T(407,35) + 0.183·**T(100,9)** · **T(100,17)** + 0.0795·T(281,7) - 0.0737·T(547,9) + 0.0687·T(523,7) - 0.0277·T(296,9) - 0.0233·T(254,5) - 0.0204·T(207,5)
0.15	0.527 + 0.327·T(485,45) + 0.280·T(454,31) + 0.280·T(565,21) - 0.148·T(391,7) + 0.123·T(488,7) + 0.0830·T(167,27) - 0.0792·T(126,9) + 0.0477·T(166,25) + 0.0437·T(168,27) + 0.0322·T(347,9)
0.2	0.506 - 0.739·T(406,19) + 0.368·T(179,77) - 0.167·T(33,5) + 0.123·T(458,13) + 0.0937·T(288,9) + 0.0571·T(200,11) + 0.0416·T(397,5) + 0.0355·T(346,59) + 0.0305·T(346,51) - 0.0118·T(490,5)
0.25	0.370 + 0.880·T(114,71) - 0.348·T(339,11) - 0.319·T(402,9) - 0.305·T(36,21) - 0.178·T(509,17) + 0.103·T(502,13) - 0.0405·T(456,47) + 0.0194·T(311,15) - 0.0191·T(339,9) + 0.0190·T(34,5)

AutoFeature-FFX results for the D1 (m = 16) dataset

For m = 16 and $\sigma_{noise} = 0.01$, all central positions of the features are at the *true* layer = 100. The set of widths of the ten different features is the closest possible to the *true* width of 15. For noise $\sigma_{noise} = 0.05$, four features are positioned at layer 100 and have the largest coefficients in the FFX model. Because of random patterns generated by noise, four features far off from layer 100 and with very different widths (from 9 to 61) are part of the model. The FFX approach is still able to detect features with noise = $\sigma_{noise} = 0.1$, although they are partly combined with random-based features in bivariate basis functions. By adding noise with a standard deviation of 0.15 or more, no features at layer 100 are found in this setting.

AutoFeature-FFX results for the D2 (m = 100) dataset

For a larger sample size $m = 100$, the outcome for $\sigma_{noise} = 0.01$ is similar to the $m = 16$ case. Interestingly however, two random-based features (with very small coefficients in the model) do appear. With only one exception for noise $\sigma_{noise} = 0.05$, all features' central positions are located at layer = 100. Hence, the amount of noise-based features is reduced compared to the smaller sample size.

TABLE 6.2: Results of fast function extraction model building for artificial data (dataset D2) with sample size $m = 100$. The central position of the denoised class 1 spectrum is at layer 100, the width is 15 layers. FFX is able to extract features located at layer 100 for all noise levels $0.01 \leqslant \sigma_{noise} \leqslant 0.25$. For noise level $\sigma_{noise} = 0.25$, the layer 100 feature only has the third largest coefficient. The improved FFX results compared to the $m = 16$ case stem from the law of large numbers.

σ_{noise}	FFX model
0.01	-5.17 + 1.95 ·**T(100,15)** + 1.65 ·**T(100,13)** + 1.12 ·**T(100,11)** + 0.608 ·**T(100,17)** + 0.414 ·**T(100,7)** + 0.250 ·**T(100,5)** + 0.176 ·**T(100,9)** - 0.000983 ·T(135,13) - 0.000478 ·T(136,15)
0.05	-2.21 + 0.778 ·**T(100,13)**^{2} + 0.732 ·**T(100,15)** · **T(100,13)** + 0.573 ·**T(100,15)**^{2} + 0.452 ·**T(100,19)** · **T(100,13)** + 0.277 ·**T(100,5)**^{2} + 0.237 ·**T(100,19)** · **T(100,15)** + 0.152 ·**T(100,7)**^{2} + 0.0518 ·**T(100,5)** · **T(100,7)** + 0.0240 ·**T(100,7)** · **T(100,13)** - 0.000938 ·T(252,41) · **T(100,19)**
0.1	-4.51 + 2.17 ·**T(100,17)** + 1.33 ·**T(100,15)** + 1.29 ·**T(100,13)** + 0.327 ·**T(100,19)** + 0.294 ·**T(100,11)** - 0.165 ·T(123,57) - 0.115 ·T(124,55) - 0.0390 ·T(193,13) + 0.0267 ·T(265,13) - 0.00649 ·T(244,5)
0.15	-1.06 + 1.40 ·**T(100,15)**^{2} + 0.457 ·**T(100,11)** · **T(100,15)** + 0.257 ·**T(100,19)** · **T(100,15)** + 0.104 ·T(182,35) · **T(100,19)** - 0.0376 ·T(341,9) - 0.0342 ·T(502,27) · **T(100,15)** - 0.0264 ·T(413,15) · **T(100,15)** + 0.0181 ·T(528,9) · **T(100,15)** - 0.0180 ·T(247,9) + 0.00656 ·T(565,9)
0.2	-0.212 + 1.09 ·**T(100,15)**^{2} - 0.207 ·T(20,35) · **T(100,15)** - 0.0802 ·T(20,23) · **T(100,15)** - 0.0507 ·T(465,27) · **T(100,15)** + 0.0316 ·T(47,7) - 0.0270 ·T(25,9) - 0.0153 ·T(262,5) + 0.00781 ·T(48,9) - 0.00568 ·T(482,11) + 0.00129 ·T(172,7)
0.25	0.483 + 0.152 ·T(436,35) - 0.106 ·T(330,19) +0.0812 ·**T(100,5)** - 0.0538 ·T(536,49) + 0.0461 ·T(554,45) - 0.0450 ·T(465,7) + 0.0363 ·T(169,13) - 0.0206 ·T(244,9) - 0.0107 ·T(83,7) + 0.000406 ·T(559,7)

For $m = 100$, even for noise = 0.2, a feature located at layer 100 is detected by the FFX approach.

The tendency of detecting *better* features with an increased sample size is not surprising. According to the law of large numbers, the average of results obtained by a random experiment, converge to the expected value as the number of experiments increases. Therefore, in this case, the FFX approach is more likely to find the true discrimination features with larger sample sizes.

For noise level $\sigma_{noise} = 0.25$, only one feature at layer 100 (with width 5) is detected. Hence, when dealing with real-world datasets that do contain such high levels of noise, a sample size of approximately 100 is necessary to be able to extract

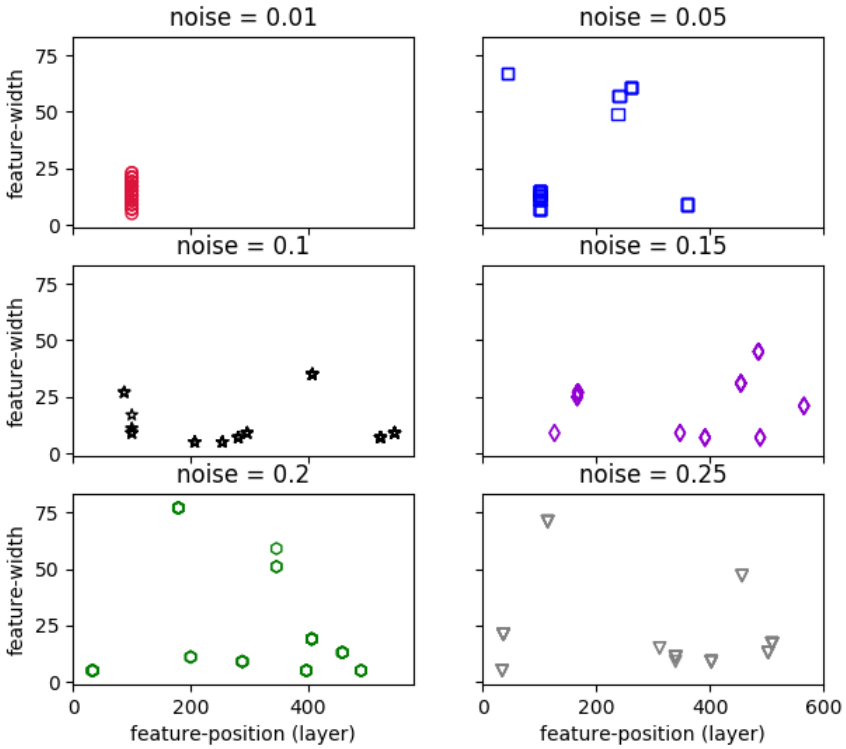


FIGURE 6.2: Visualization of the results of AutoFeature-FFX for artificial data (dataset D2) with sample size $m = 16$. For each noise level, the positions and widths of features from FFX models with complexity $n_{\text{features}} = 10$ are depicted. The central position of the noise free class 1 spectrum is at layer 100, the width is 15 layers. FFX is able to extract features located at layer 100 for low amplitude noise $\sigma_{noise} \leqslant 0.1$ and finds features based solely on random effects for noise $\sigma_{noise} \geqslant 0.15$.

a meaningful feature. The interpretation of an AutoFeature-FFX result without having a ground truth would still be difficult in such a setting, since the result shows that the correct layer 100 feature only has the third largest coefficient after two random based features.

As the extraction of the underlying *T(100,15)* feature demonstrates for many noise levels for both datasets D1 and D2, FFX can be a suitable choice for automatic feature generation. Interestingly, in all dataset noise combinations, FFX either extracted a layer 100 feature or a completely random based feature. Other features with center locations close to 100 that carry information about all three classes were never extracted.

As expected, increasing noise results in a declining number of meaningful features. Also as anticipated, a dataset with a larger number of samples enabled FFX to extract better features for a fixed level of noise. The ratio of noise and samples in real world data sets is therefore an important parameter in assessing the possibility of automatic feature generation with FFX.

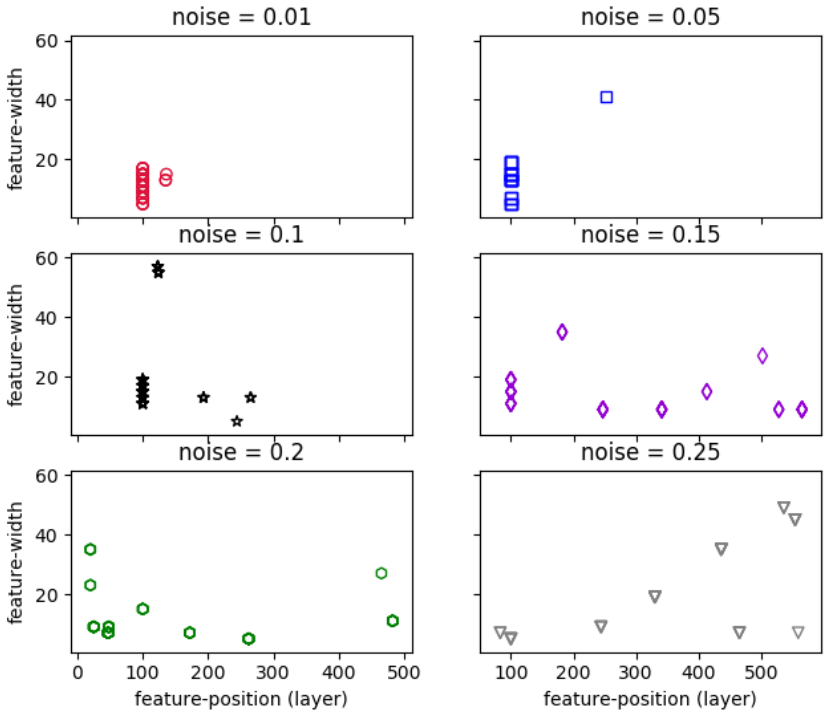


FIGURE 6.3: Visualization of the results of AutoFeature-FFX for artificial data (dataset D2) with sample size $m = 100$. For each noise level, the positions and widths of features from FFX models with complexity $n_{\text{features}} = 10$ are depicted. The central position of the noise free class 1 spectrum is at layer 100, the width is 15 layers. FFX is able to extract features located at layer 100 for all noise levels $0.01 \leqslant \sigma_{noise} \leqslant 0.25$. For noise level $\sigma_{noise} = 0.25$, the layer 100 feature only has the third largest coefficient. The improved FFX results compared to the $m = 16$ case stem from the law of large numbers.

6.2 Results of AutoFeature Experiment with Microplastic Data

Dataset for Feature Generation Experiments

The data selection and annotation procedure described in section 5.2 resulted in a dataset with the following number of samples per polymer class:

TABLE 6.3: Number of samples for each class in the microplastic dataset used for automatic feature generation.

Class	number of samples
PE	64
PP	39
PS	364
PAN	639
PMMA	1545

The polymer classes in the dataset are rather imbalanced, as the class with the most samples (PMMA, 1545) and fewest samples (PP, 39) point out.

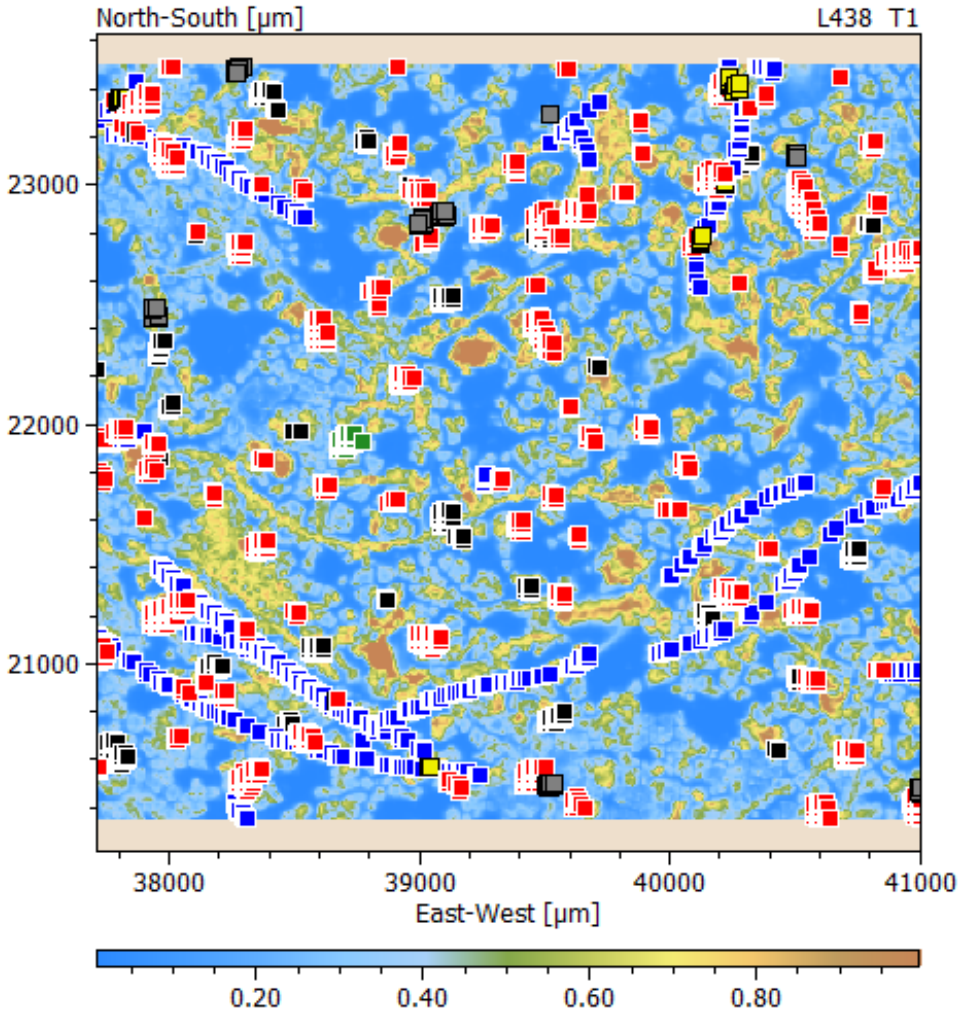


FIGURE 6.4: Selected region of interest of microplastic data. For each pixel, an infrared spectrum is available. The absorbance at wavenumber 1901 cm^{-1} is shown. With manually designed features, polymers are detected in the data (see text). The detected polymers are shown as colored squares in the figure with the following color code: green = polypropylene, black = polystyrene, blue = polyacrylonitrile, red = polymethylmethacrylate, grey = polyethylene, yellow = mixed spectra (not used for further analysis).

Figure 6.4 displays the spatial location and class information of the annotated pixels. One can observe the PAN fiber structure and, for example, that there is only a single polypropylene particle (with an area of 39 pixels).

Results of AutoFeature Experiment with Microplastic Data

Figures 6.5 to 6.9 depict the main resulting features for the five binary FFX approaches for the microplastic dataset. As the vertical position does not matter for the features, a baseline shift is freely chosen for the plots. Besides the features, for each polymer class a selected spectrum is plotted. The positive class polymer in a binary one-vs-all run, is plotted in orange, all negative classes are in black.

The full FFX models are presented in equations 6.1 (PE) , 6.2 (PP), 6.3 (PS), 6.4 (PMMA) and 6.5 (PAN).

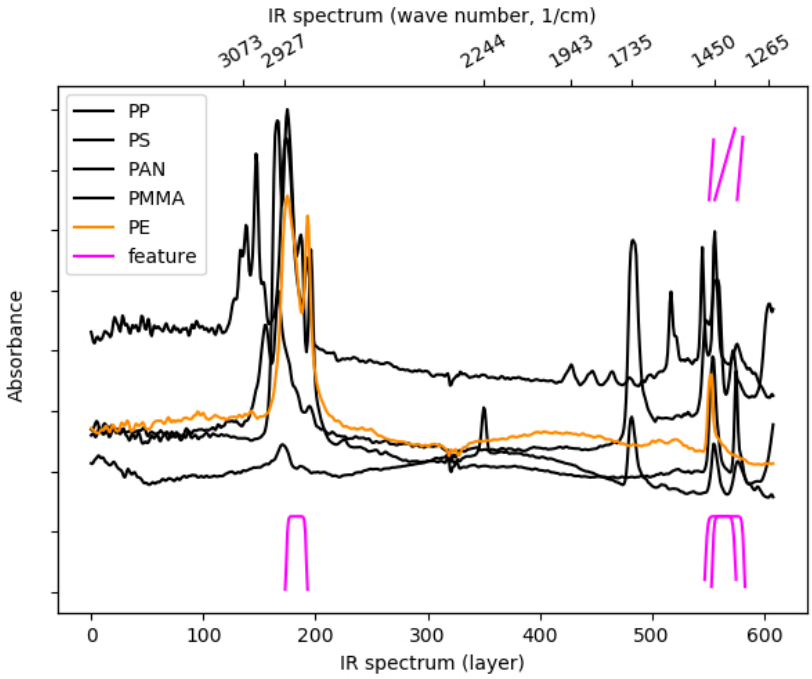


FIGURE 6.5: Part of the resulting features for FFX AutoFeature algorithm with microplastic dataset, positive class: polyethylene (PE), negative classes: polypropylene (PP), polystyrol (PS), polymethylmethacrylate (PMMA) and polyacrylonitrile (PAN). A maximum of 100 samples for each class is used for feature selection and the resulting features of the FFX model with ten basis functions are illustrated in magenta. Vertical positions of features are freely shifted from the baseline to improve readability. Most features cover the range from 1500 to 1300 cm^{-1}, capturing information about the PE characteristic absorption band and additional information from other classes' absorption bands. See equation 6.1 for the full FFX model.

In all five cases, features selected by FFX are solely in the region where characteristic absorption bands occur. However, the central position of the features does often not coincide with the central position of an absorption band. In many cases the features width is also not equal to the width of the absorption bands. The fact that the features occur only in absorption band regions support the conclusion that the features are extracted because of some *real* differences among the classes and not because of random noise. Otherwise, features would appear at all possible positions, which is not the case. This result is a first indication that the automatic feature engineering is feasible for real-world spectra.

In the following discussion, we will loosely denote the spectral region from 3200 to 2700 cm^{-}1 (layers 103 to 233) as *left side* and the spectral region from 1800 to 1250 cm^{-}1 (layers 466 to 609) as *right side of the spectrum*.

In the PE case (short for 'In the binary run where PE is the positive class'), there is a single feature, P(183,21,6,8.5), selected on the left side of the spectrum while all others are selected on the right side. Except one feature, S(551,556), that is located entirely at the PE characteristic absorption band, all other features at the right hand side cover more than the characteristic absorption band. A possible explanation is the need for distinction between the PE and PP spectra. As they are quite similar but PP has another absorption band right of the absorption band of PE, the spectral

information in this range is of great value for classification.

$$\begin{aligned}\hat{y}_{\mathrm{PE}} = {} & 0.207 - 0.972 * \mathrm{P}(561, 29, 5.7, 12.5) * \mathrm{P}(568, 31, 8, 13.5) + 0.896 * \\ & \mathrm{P}(565, 31, 8, 13.5) * \mathrm{P}(568, 31, 8, 13.5) + 0.385 * \mathrm{P}(183, 21, 6, 8.5) * \\ & \mathrm{P}(568, 31, 8, 13.5) + 0.169 * \mathrm{P}(568, 31, 8, 13.5)^2 + 0.151 * \mathrm{P}(568, 31, 8, 13.5) * \\ & \mathrm{S}(551, 556) - 0.149 * \mathrm{S}(551, 556) + 0.128 * \mathrm{S}(556, 575) * \\ & \mathrm{P}(568, 31, 8, 13.5) + 0.0817 * \mathrm{S}(576, 582) * \mathrm{S}(551, 556) + \\ & 0.0408 * \mathrm{S}(551, 556)^2 + 0.0261 * \mathrm{S}(576, 583) * \mathrm{S}(551, 556)\end{aligned} \tag{6.1}$$

Interestingly, the result for the PP case looks quite different. Here, all selected features are on the left side of the spectrum. All four categories of feature shapes with different positions and widths are selected.

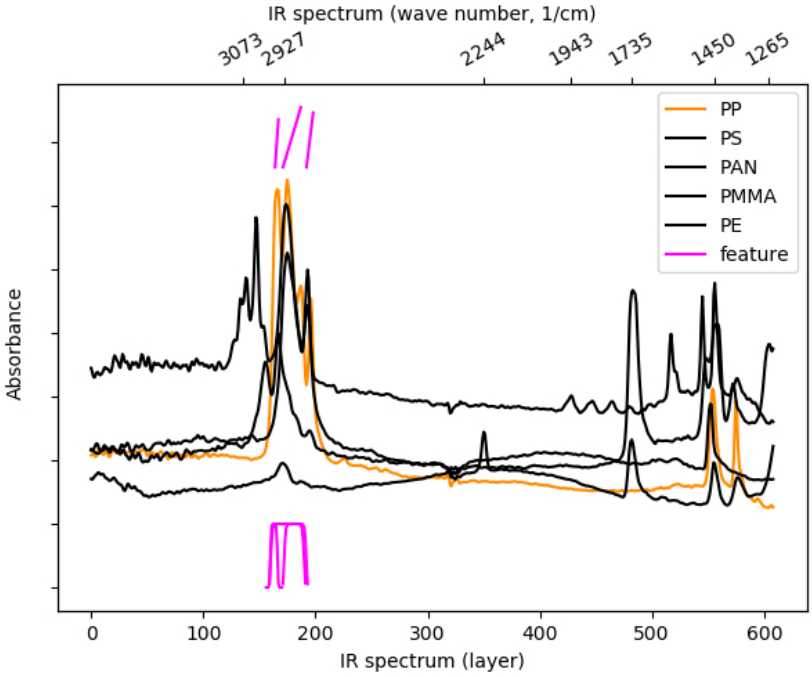


FIGURE 6.6: Part of the resulting features for FFX AutoFeature algorithm with microplastic dataset, positive class: polypropylene (PP), negative classes: polyethylene (PE), polystyrol (PS), polymethylmethacrylate (PMMA) and polyacrylonitrile (PAN). A maximum of 100 samples for each class is used for feature selection and the resulting features of the FFX model with ten basis functions are illustrated in magenta. Vertical positions of features are freely shifted from the baseline to improve readability. All features are positioned in the range from 3100 to 2700 cm^{-1}, containing information about characteristic absorption bands of the polymers. See equation 6.2 for the full FFX model.

$$\begin{aligned}\hat{y}_{PP} = {} & 0.569 + 0.278 * P(182, 23, 6, 9.5) + 0.258 * S(171, 188) + \\ & 0.166 * S(192, 199) + 0.149 * P(176, 33, 8.5, 14.5) + \\ & 0.104 * P(175, 33, 8.5, 14.5) + 0.0835 * P(163, 15, 3, 3) + \\ & 0.0637 * S(192, 198) - 0.0591 * S(164, 168) + \\ & 0.0344 * S(193, 198) + 0.000101 * T(163, 15)\end{aligned} \tag{6.2}$$

The resulting features in the PS case consist only of S features. These features can be clearly categorized into two positions. Seven of the ten straight line features all have nearly the same width and range from layer 142 to 168. The remaining three features that are all based on the right side have larger coefficients than on the left side, compensating the smaller quantity. These straight lines differ minimally, ranging from layers 543-544 to 553-554.

$$\begin{aligned}\hat{y}_{PS} = {} & 0.334 - 0.0205 * S(544, 554) - 0.0176 * S(543, 554) - \\ & 0.0158 * S(544, 553) - 0.0144 * S(144, 167) - \\ & 0.0131 * S(143, 166) - 0.00748 * S(142, 165) - \\ & 0.00688 * S(145, 168) - 0.00436 * S(144, 166) - \\ & 0.00277 * S(143, 165) - 0.000510 * S(145, 167)\end{aligned} \tag{6.3}$$

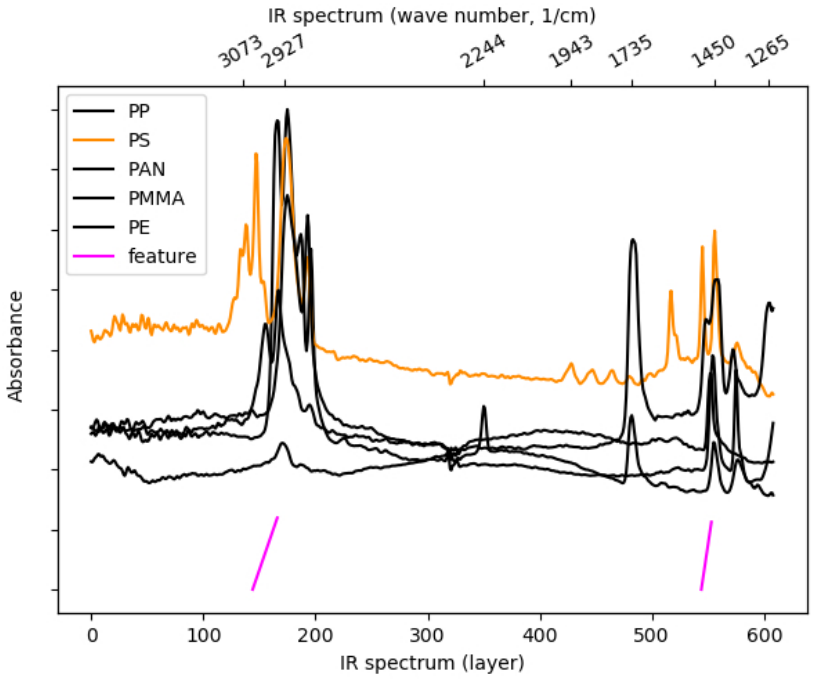


FIGURE 6.7: Part of the resulting features for FFX AutoFeature algorithm with microplastic dataset, positive class: polystyrol (PS), negative classes: polyethylene (PE),polypropylene (PP), polymethylmethacrylate (PMMA) and polyacrylonitrile (PAN). A maximum of 100 samples for each class is used for feature selection and the resulting features of the FFX model with ten basis functions are illustrated in magenta. Vertical positions of features are freely shifted from the baseline to improve readability. For the PS model, only straight line features located in the polymer absorption bands are selected. The features are very similar, covering either the range from 3040 cm^{-1} to 2950 cm^{-1} or 1500 cm^{-1} to 1465 cm^{-1}. See equation 6.3 for the full FFX model.

In the PMMA case there is a single feature, P(554,27,7,11.5), having the largest coefficient on the right side though. Most of the left side features are again quite alike as triangle and Gaussian shape features have their center at 153 and 154 and as straight line features ranging from 153-155 to 176-177.

$$\begin{aligned}\hat{y}_{PMMA} = {} & 0.312 + 0.0745 * P(554, 27, 7, 11.5) - 0.0287 * S(154, 177) - \\ & 0.0248 * S(154, 176) - 0.0187 * S(153, 176) - \\ & 0.0133 * S(155, 176) + 0.0132 * T(154, 39) + \\ & 0.0129 * T(154, 37) + 0.00964 * T(153, 43) + \\ & 0.00351 * G(154, 7) + 0.00328 * G(154, 6)\end{aligned} \tag{6.4}$$

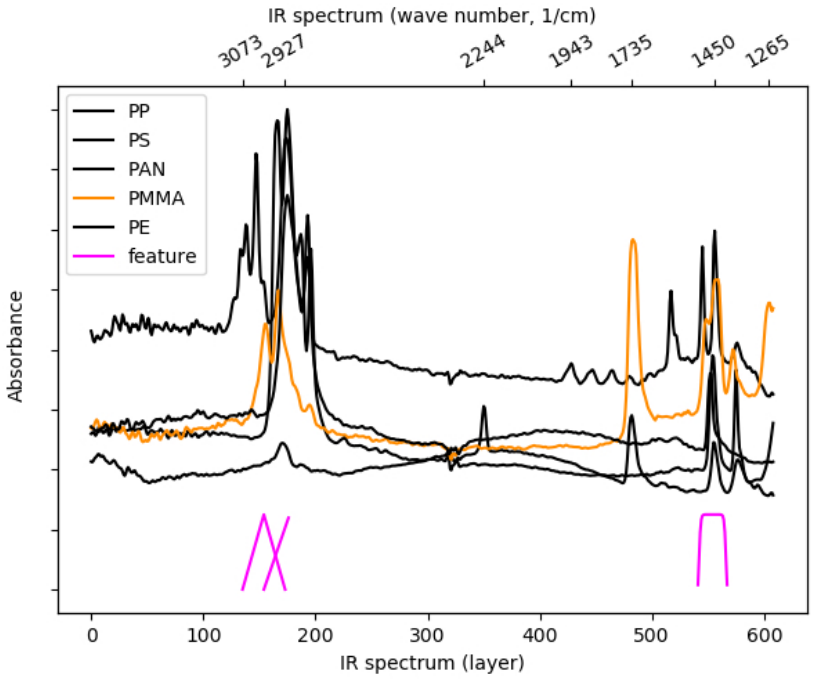


FIGURE 6.8: Part of the resulting features for FFX AutoFeature algorithm with microplastic dataset, positive class: polymethylmethacrylate (PMMA) , negative classes: polyethylene (PE), polypropylene (PP), polystyrol (PS) and polyacrylonitrile (PAN). A maximum of 100 samples for each class is used for feature selection and the resulting features of the FFX model with ten basis functions are illustrated in magenta. Vertical positions of features are freely shifted from the baseline to improve readability. For the PMMA model, one feature is selected from the absorption band range on the right side while the remaining feature cover the absorption bands on the left side. See equation 6.4 for the full FFX model.

Interestingly, only P features are selected in the PAN case. While some features are alike, there is still a large variety as four distinct spectral positions are covered by features. All of these features at least partly cover the PAN specific absorption bands. In the PAN case, many product bases are formed in the FFX procedure, some of them consisting of features from the left and right side of the spectrum.

$$
\begin{aligned}
\hat{y}_{PAN} = {} & 0.189 - 0.224 * P(171, 15, 4, 6) * P(547, 15, 4, 6) - \\
& 0.132 * P(171, 11, 3, 3.5) * P(497, 29, 5.7, 12.5) + \\
& 0.0881 * P(171, 15, 4, 6) * P(171, 11, 3, 3.5) + \\
& 0.0557 * P(171, 17, 5, 6.5) * P(168, 31, 8, 13.5) - \\
& 0.0487 * P(171, 9, 3, 3) * P(497, 29, 5.7, 12.5) - \\
& 0.0345 * P(171, 11, 3, 3.5) * P(594, 23, 6, 9.5) + \\
& 0.0265 * P(497, 29, 5.7, 12.5) * P(547, 15, 4, 6) - \\
& 0.0198 * P(171, 15, 4, 6) * P(538, 33, 8.5, 14.5) + \\
& 0.0197 * P(538, 33, 8.5, 14.5) * P(497, 29, 5.7, 12.5) - \\
& 0.00300 * P(594, 23, 6, 9.5) * P(171, 9, 3, 3)
\end{aligned}
\tag{6.5}
$$

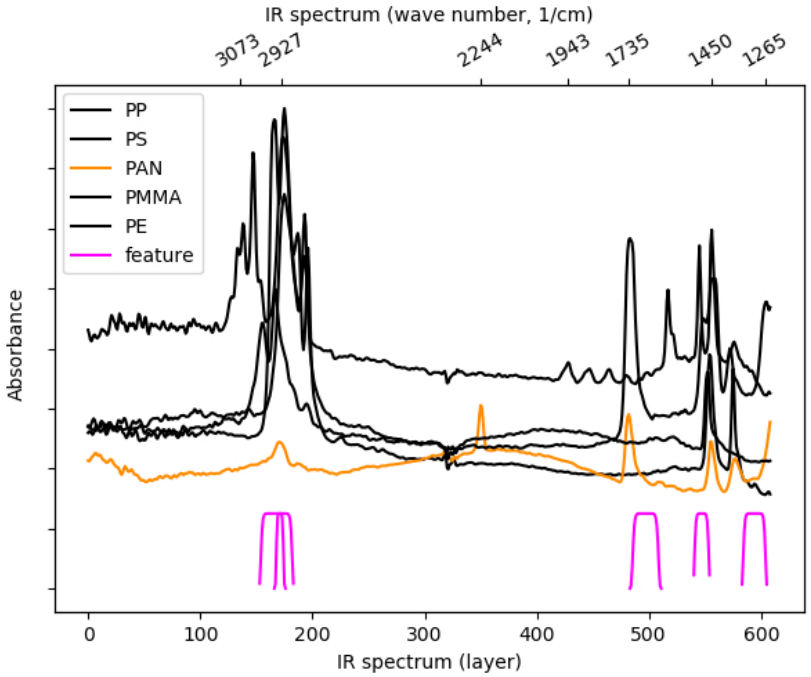


FIGURE 6.9: Part of the resulting features for FFX AutoFeature algorithm with microplastic dataset, positive class: polyacrylonitrile (PAN), negative classes: polyethylene (PE), polypropylene (PP), polystyrol (PS) and polymethylmethacrylate (PMMA). A maximum of 100 samples for each class is used for feature selection and the resulting features of the FFX model with ten basis functions are illustrated in magenta. See equation 6.5 for the full FFX model. Vertical positions of features are freely shifted from the baseline to improve readability. For the PMMA model, only general Gaussian shape features are selected. The features cover most of PAN's characteristic absorption bands but not the absorption band at 2250 cm^{-1} that has no overlapping absorption bands from other polymer classes. Why FFX might not select this feature is discussed in the text and further illustrated in Figure 6.10.

In the five binary FFX runs with microplastic data, features with all four designed shapes are used. Features are only selected in the absorption band regions of the polymers. However, in many binary runs, only a few polymer characteristic absorption bands are covered by features. Some characteristic absorption bands are

not covered by any feature in any run. For example, the PAN absorption band around layer 350 was not used in any of the extracted models.

Often, position and width of features tightly overlap with absorption bands while in other cases they do not. The center location of many features does not match the peak location of an absorption band but is shifted by several layers. Mostly, these features still cover a wide range of an absorption band but have additional information contained as well, e.g. from another absorption band of another class. These results may be surprising at first glance but in fact are an expected and beneficial outcome of the approach that is conducted. The goal of automatic feature generation is to find features that carry information about classes and their distinctions that are statistically stable and predictive. While there are cases for which features tightly overlapping with an absorption peak *may* be valuable predictors with the aforementioned properties, there are cases for which they *may not*. Moreover, there are certainly cases for which they are not the *best* discriminators between classes. For building features that just perfectly match some absorption bands that humans are able to spot easily, certainly no automatic approach is needed. However, situations are harder to handle for humans when there are overlapping absorption bands for different classes and when features that tightly mirror a specific absorption band do not provide the best class discrimination. Then, automatic approaches can help to find features that are statistically reliable and do carry valuable information for class distinction.

Still, there are results that are unexpected. One of these unexpected results is that the PAN absorption band around layer 350 is not selected by the FFX approach. The absorption peak is solely present in PAN and no absorption band from another class is overlapping with this peak. Hence, we would expect a feature that depicts the information about the presence of this peak to be helpful for class discrimination. As there is no abnormal variation of this peak among the different PAN spectra, we would also assume such a feature to be statistically stable. To better understand why FFX does not select the absorption band, three feature value distributions for PAN and non-PAN classes are plotted as histograms in Figure 6.10. In the upper and middle plot, the histograms show features P(171,15,4,6) and P(547,15,4,6). These features are selected by FFX and together form a product base with the largest coefficient in the PAN model. In the lower plot, feature T(350,11) is depicted. This feature is chosen because it overlaps well with the PAN absorption band.

For both FFX-selected features, the histograms for the PAN and non-PAN feature values are easy to differentiate and certainly provide a good distinction between the classes. Additionally, no histogram is *flat* but they have clear mounds and are unimodal. For both features, there is a slight overlap in the class histograms. For the manually selected feature, there is very slight overlap between the class histograms as well, enabling class separability. However, the histograms look very different compared to the previous two cases. The feature has a value of around 0.95 for all PAN spectra, confirming statistical relevance and stability. For the non-PAN class, the distribution is very flat along nearly all possible correlation coefficient values and exhibits several mounds.

The reason why FFX results in models that contain features with mound-like distributions rather than flat distributions is the minimization task in regularized least squares. Rather than optimizing class separability, least squares' aim is the minimization of the sum of squared residuals. Because features with flat class distributions have larger variances than features with mound-like distributions, they will

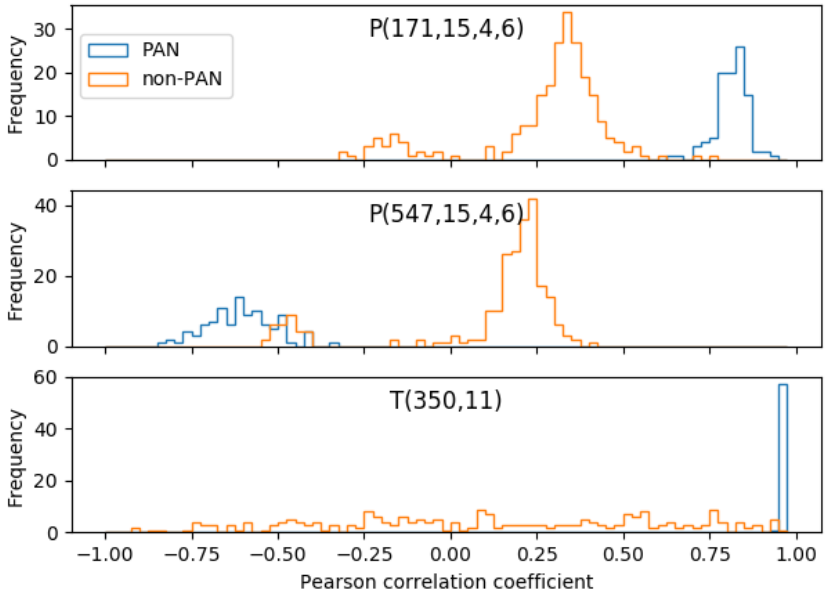


FIGURE 6.10: In the upper and middle plot, the histograms show features P(171,15,4,6) and P(547,15,4,6). These two feature are selected by FFX AutoFeature algorithm with 10 basis functions. Furthermore, they together form a product base with the largest coefficient in the PAN model (see equation 6.5 for the full FFX model). In the lower plot, the feature T(350,11) is depicted. This feature is manually chosen because it overlaps well with the PAN absorption band at 2250 cm^{-1}. Distributions of both PAN and non-PAN classes differ between the AutoFeature-selected features and the one manually selected. AutoFeature-FFX results in models that contain features with mound-like distributions rather than flat distributions because of the minimization task in regularized least squares. Because features with flat class distributions have larger variances than features with mound-like distributions, they will result in models with a larger error if the target variables are binary. Even though nearly all values for the PAN class are equal, the feature does not get selected in the elastic net model because of the high variance of the class 0 values.

result in models with a larger error if the target variables are binary. Even though nearly all values for the PAN class are around 0.95 and the class 1 distribution has small variance, the feature does not get selected in the elastic net model because of the high variance of the class 0 values.

Figures 6.11 (left side of the spectrum) and 6.12 (right side of the spectrum) display the widths and positions of the resulting features from the AutoFeature experiments done with five different feature selection methods, see section 5.2. Besides these two ranges no feature is selected by any selection method. This fact is an indication that, in principle, all methods are suitable for automatic feature generation since they at least select features in ranges that contain relevant chemical information.

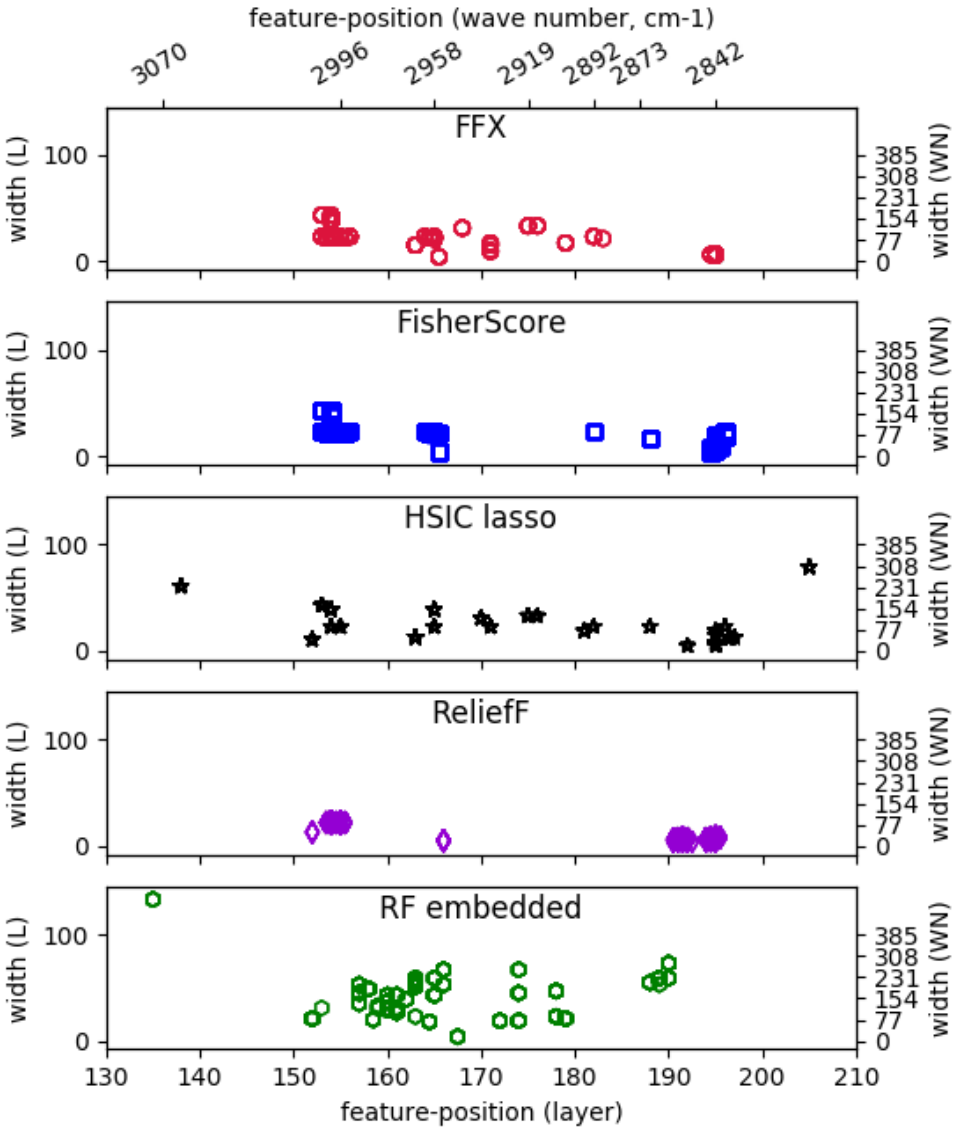


FIGURE 6.11: Parts of the resulting features from the five AutoFeature selection methods FFX, FisherScore, HSIC lasso, ReliefF and embedded random forest. While the embedded random forest uses a multiclass approach, the other methods are carried out in a one-vs-all fashion. All selected features cover one or more polymer absorption bands. FisherScore and ReliefF result in features that show less variability than FFX and HSIC lasso. Embedded RF's features have the greatest variability in this part of the spectrum.

While the features selected by the different methods are roughly in the same spectral range, the results still show some clear differences.

Firstly, it seems that both FisherScore and ReliefF select many similar features. In the width-position plots, distinct and tight clusters can be observed. This property does not seem advantageous as features containing similar information as an already existing feature usually do not improve models. FisherScore and ReliefF show similar results concerning feature widths as well as they select features that

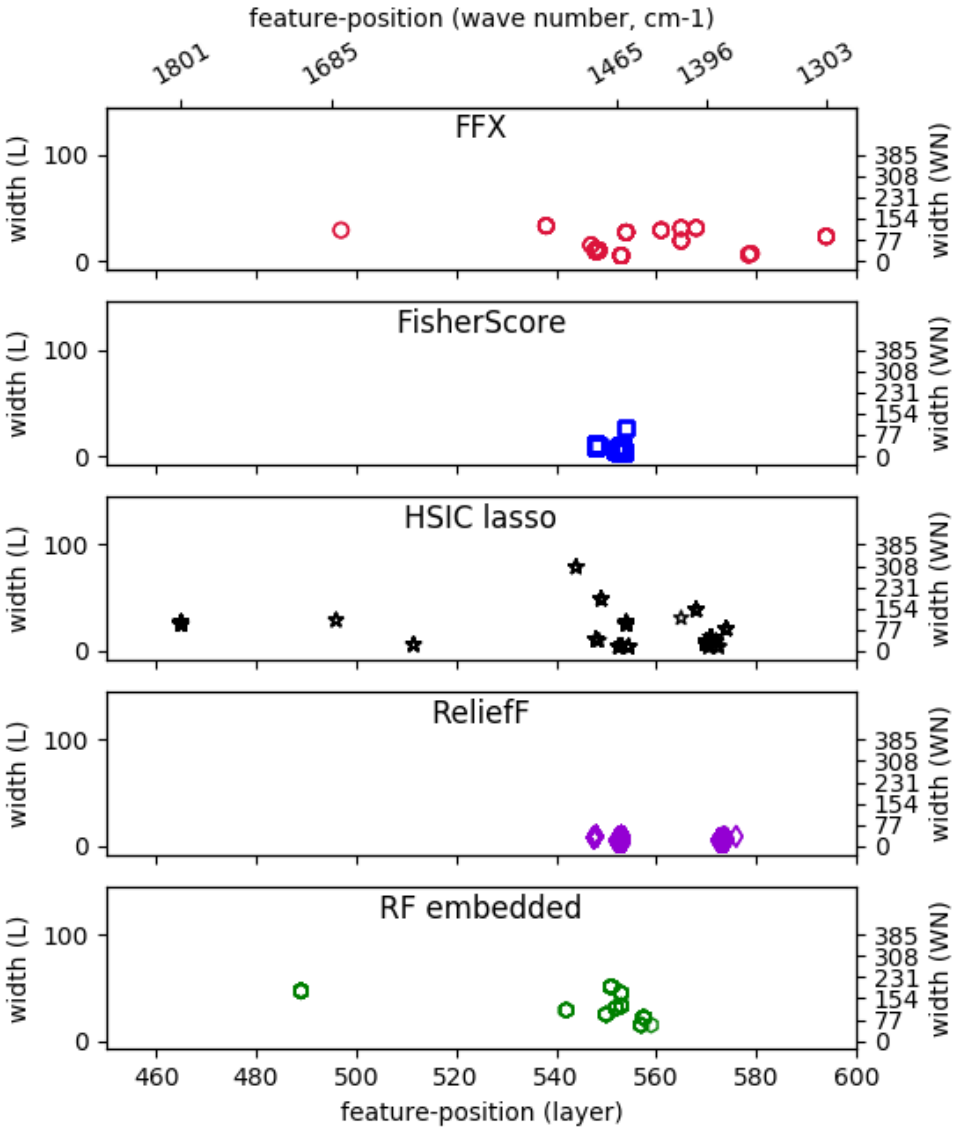


FIGURE 6.12: Parts of the resulting features from the five AutoFeature selection methods FFX, FisherScore, HSIC lasso, ReliefF and embedded random forest. While embedded random forest uses a multiclass approach, the other methods are carried out in a one-vs-all fashion. All selected features cover one or more polymer absorption bands. FisherScore and ReliefF result in features that show fewer variability than FFX, HSIC lasso and embedded forest. FFX and HSIC lasso's features have the greatest variability in this part of the spectrum.

are, on average, more narrow than features from other methods.

Variability among FFX's and HSIC lasso's features is larger. Although there are clusters of features as well, they are more spread out. HSIC lasso also selects some features at positions where no other method selected any feature.

The embedded random forest approach extracts a variety of features on the left side of the spectrum. A possible explanation is that random forest is the only feature selection method used that is carried out in a multiclass fashion in contrast

to the one-vs-all fashion for the other methods. However, on the right side of the spectrum, the variability of FFX and HSIC lasso features seem to be larger than the variability of the embedded random forest features.

Interestingly, no feature covers the range around 2250 cm^{-}1 in which PAN exhibits a characteristic absorption band that is also discussed above. Before, we have discussed reasons for FFX with its elastic net model to not select features in this range. These reasons do not hold for embedded random forest since random forest is a tree-based non-linear method that is also often used solely for class discrimination. Apparently, other features are viewed to be more valuable for classification during random forest modelling. The ten most important features, ranked by random forest's internal variable importance (mean decrease impurity) are T(166,53), T(161,43), T(158,49), G(165,7), T(163,51), T(551,51), P(152,21,6,8.5), G(552,5), T(157,53) and G(166,11). The empirical distributions of four of these features for the five polymer classes are shown in Figure 6.13. Moreover, the same information is plotted for feature T(350,11) that tightly matches the PAN absorption peak at 2250 cm^{-1}. All four features selected by the embedded random forest show distinctions among all or almost all classes. These features are certainly valuable for a future prediction (if the model validation does not show otherwise). For the T(350,11) feature we have already seen above that the distribution for the PAN class is very narrow and nearly all PAN spectra show a value around 0.95 for this feature. However, the feature values for all other classes are distributed along a wide range from -1 to 1, making a class differentiation among the non-PAN classes impossible.

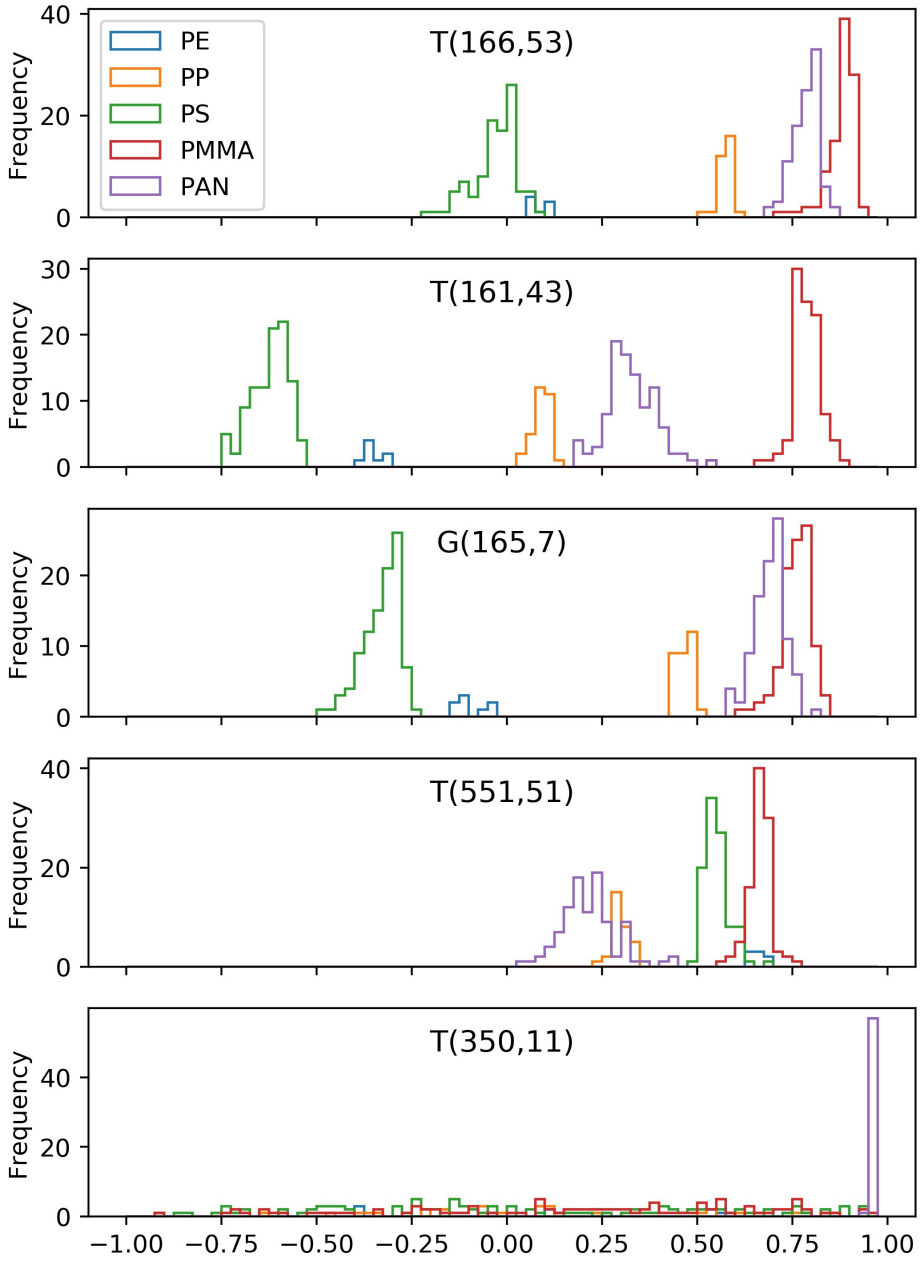


FIGURE 6.13: Empirical feature value distributions are shown for the five polymer classes. Features T(166,53), T(161,43), G(165,7) and T(551,51) are among the ten most important features of embedded random forest feature selection, ranked by random forest's mean decrease impurity measure. T(350,11) is not selected by random forest but manually selected since it overlaps tightly with the PAN absorption band at 2250 cm^{-1}. All four features selected by embedded random forest show remarkable distinctions among all or almost all classes. The T(350,11) feature shows excellent distinction between PAN and non-PAN classes but poor distinction within the non-PAN classes.

FFX stability experiment

Figure 6.14 shows the results of the FFX stability experiments described in 5.2. For a class sample size = 32, some features are found for only a single sample. For

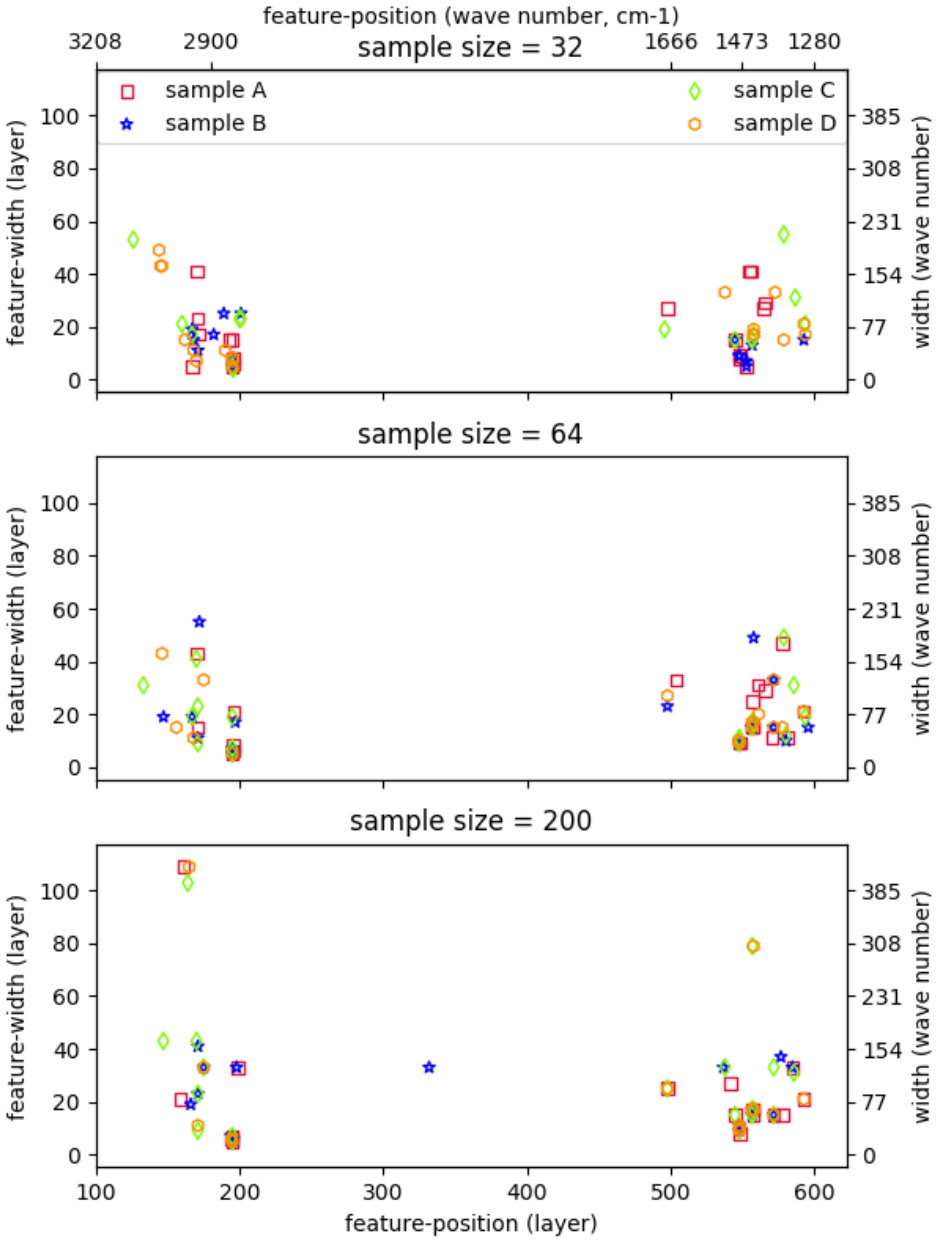


FIGURE 6.14: After drawing four sub-samples from the microplastic dataset for three different sample sizes (per class), the FFX feature generation and selection algorithm is performed for all sub-samples. When sample size for each class is 32, some features are found for only a single sample. In terms of the sample sizes of 64 and 200, there are similar features from another sample for most features from any sample. These results indicate good stability of the FFX feature selection approach, making it feasible for real-world spectroscopic data. Table 7.1 (Appendix) lists the features shown in this figure.

example, in the right side of the spectrum there are unique features for samples A,C and D. This effects are hardly visible in the sample size = 64 case. Here, for most features from any sample there are similar features from another sample. This observation is also true for the sample size = 200 case. For the sample size = 200 case, features with positions of around 160 and widths of around 107 appear for three out of the four samples. These features have not been selected in any sample for the sample size 32 and 64 case.

This result is a strong indicator that FFX feature selection is robust enough to handle natural variability among real-world spectroscopic datasets, at least for a fair dataset size.

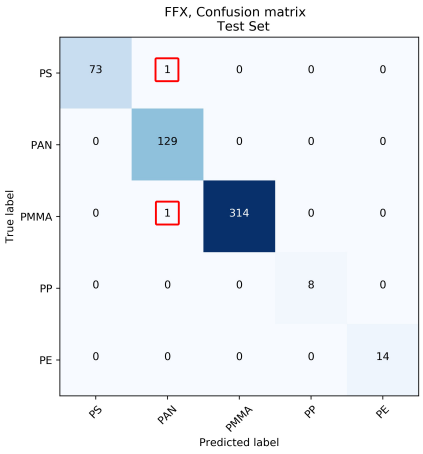
Validation

In the section above we analysed histograms of feature value distributions for different classes. For the features that are selected by any of the five approaches tested, different classes show distinct, mostly non-overlapping feature value distributions. Such situations are generally preferred in classification tasks. However, it's possible that the automatic feature generation processes yield features that do provide such favourable class distinction properties but only by chance. If this was the case, the features would not provide any class distinction properties for unseen data (i.e. not included in the training phase) of the same classes.

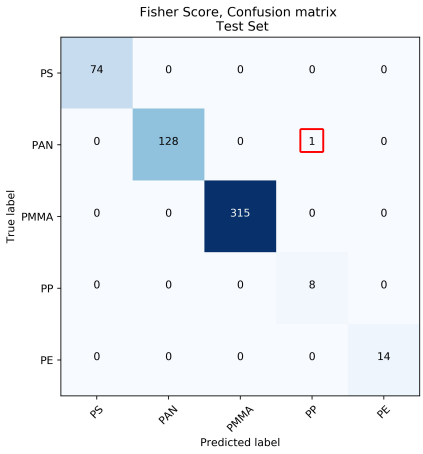
After creating a random forest model for each set of features that was selected by the five feature selection methods, these models are evaluated with the test data that has not been used in any step in feature extraction or model creation before. Figure 6.15 shows the resulting confusion matrices for the five different approaches. The following errors occur when classifying the test data:

- Embedded RF approach: two PMMA spectra are classified as PAN.
- FFX: one PMMA and one PS spectrum are classified as PAN.
- Fisher Score: one PAN spectrum is classified as PP.
- ReliefF: one PE spectrum is classified as PMMA.

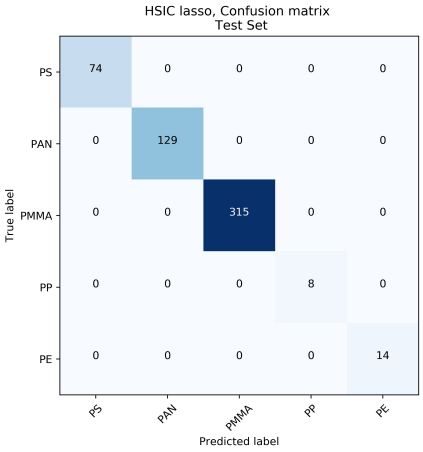
Apart from these few misclassifications, all samples are classified correctly. This is a strong validation of the final models. Hence, it is a strong validation for all steps done before - implying that the AutoFeature algorithm works well for this data.



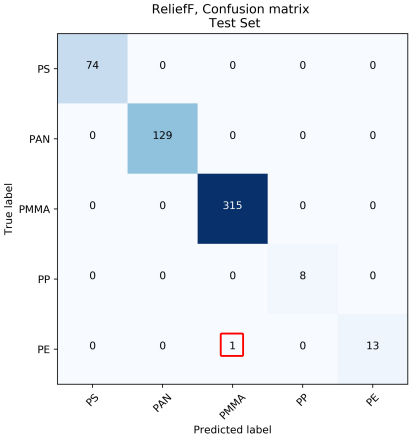
(A) FFX



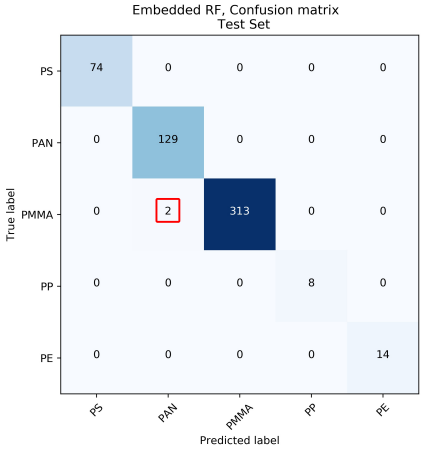
(B) Fisher Score



(C) HSIC lasso



(D) ReliefF



(E) Embedded RF

FIGURE 6.15: Validation of five random forest models that are created with 50 features each. The features are created automatically by the AutoFeature algorithm presented in this work using five different feature selection approaches. Test data used for validation stems from the same data collection origin as training data but is not used in any stage of AutoFeature or random forest modelling before. Apart from a few misclassifications, all new, unseen spectra are classified correctly.

6.3 Results of AutoFeature Experiment with Melanoma Data

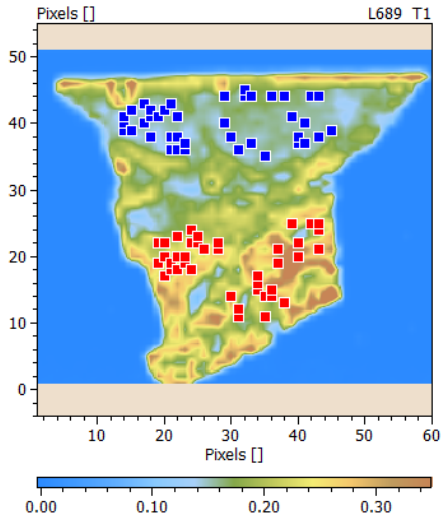


FIGURE 6.16: Image of epidermis showing the absorbance for wavenumber 1307 cm^{-1}. Full spectra of marked locations are used for AutoFeature experiments, red indicating the melanoma and blue the non-melanoma class.

Figure 6.16 shows a chemical picture (the absorbance for wavenumber 1307 cm^{-1}) of the melanoma sample. It also illustrates the spatial locations of pixels selected for the melanoma dataset for the AutoFeature algorithm.

Resulting features from the embedded random forest AutoFeature approach are listed in Table 6.4 and sorted by their variable importance in descending order. The variable importance of feature S(714,719) is largest with a value of 0.1 while four features show an importance of 0, indicating that further dimensionality reduction is possible.

Representative melanoma and non-melanoma spectra are depicted together with the ten features showing the largest variable importances in Figure 6.17. The spectra of the two classes are very similar. Differences can be perceived in small shifts of some absorption bands as well as intensity differences at certain wave numbers. Even though the automatically selected features are based on the correlation coefficient and are not designed to capture information about intensity differences *directly*, they are located both at locations where primarily band shifts as well as intensity differences are perceived. This is because different intensity changes also have to cause different shapes. Considering the small differences in the spectra, the results of the feature selection is satisfying.

The validation of the random forest model that is built with the 25 automatically selected features resulted in a perfect classification of the test set. All eight samples from the tumor and non-tumor classes are classified correctly, also shown in the normalized confusion matrix in Figure 6.15. Despite the small size of the test set and the fact that the training and test spectra stem from the same biological specimen, this result proves the ability of the AutoFeature algorithm to pick up features suitable for classifications of hyperspectral images of biological samples.

Boxplots of features S(714,719) (largest importance), P(173,17,4,5) (second largest importance) and T(163,17) (rank 11) for non-tumor and tumor as well as training and test sets are shown in Figure 6.19. Feature S(714,719) shows excellent statistical

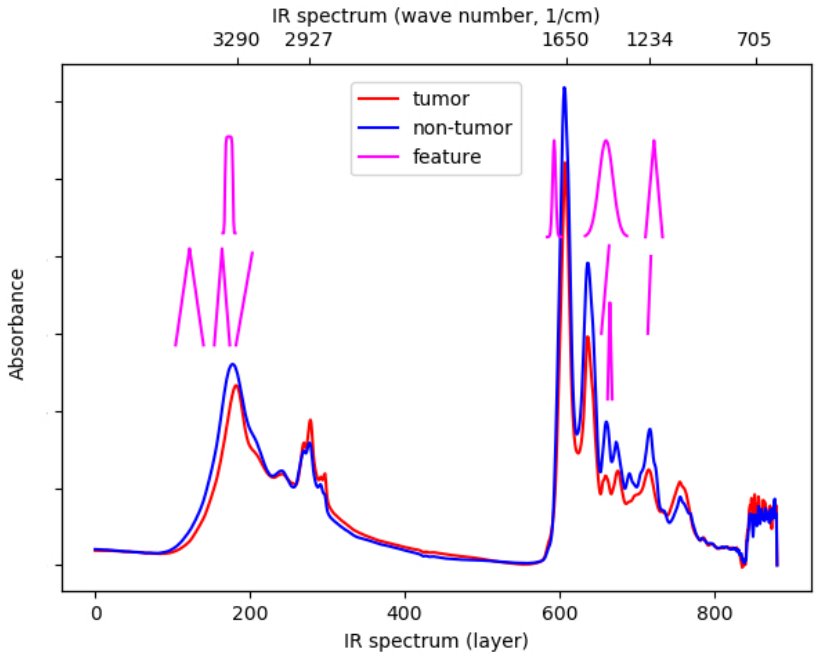


FIGURE 6.17: The tumor (melanoma) and non-tumor (not melanoma) spectra are selected from the full specimen dataset. The two classes are very similar, differences can be perceived in small shifts of some absorption bands as well as intensity differences at certain wave number ranges. The ten features illustrated in magenta resulted from the embedded random forest AutoFeature algorithm and obtain largest variable importances.

properties for classification. Considering the small size of the test set, the distributions of the two classes are similar for the training and the test set. Also, a clear distinction between the feature value distribution of the two classes can be made, manifesting medians of around 0 for non-tumour and -0.9 for tumour. Feature P(173,17,4,5) shows the same tendency of positive properties but to a smaller degree. As the medians 0.05 and 0.35 of the two different classes indicate, the class distributions are less distinct. Although feature T(163,17) provides perfect class separability for our data, its usefulness should be questioned. The differences in the medians of the two different classes is less than 0.1, resulting in the need for small variance within each class. This is apparently true for our dataset but is likely not to be fulfilled if samples from different biological sites or patients are taken or if steps in the data acquisition process are handled differently.

TABLE 6.4: Resulting features and their random forest variable importance from the embedded random forest AutoFeature approach (see text) for the melanoma dataset. Four features with an importance of 0 indicate the possibility of further dimensionality reduction.

Importance	Feature
0.1	S(714,719)
0.08	P(173,17,4,5)
0.08	T(665,7)
0.08	P(593,19,1,2)
0.06	S(654,665)
0.06	T(164,21)
0.06	T(122,37)
0.06	T(722,23)
0.06	S(182,204)
0.06	G(660,9)
0.04	T(163,17)
0.04	T(178,49)
0.04	T(703,13)

Importance	Feature
0.04	T(113,57)
0.02	T(744,13)
0.02	S(242,262)
0.02	T(350,41)
0.02	G(665,4)
0.02	T(115,45)
0.02	S(763,772)
0.02	T(772,39)
0.0	T(781,43)
0.0	T(775,9)
0.0	T(730,51)
0.0	S(739,751)

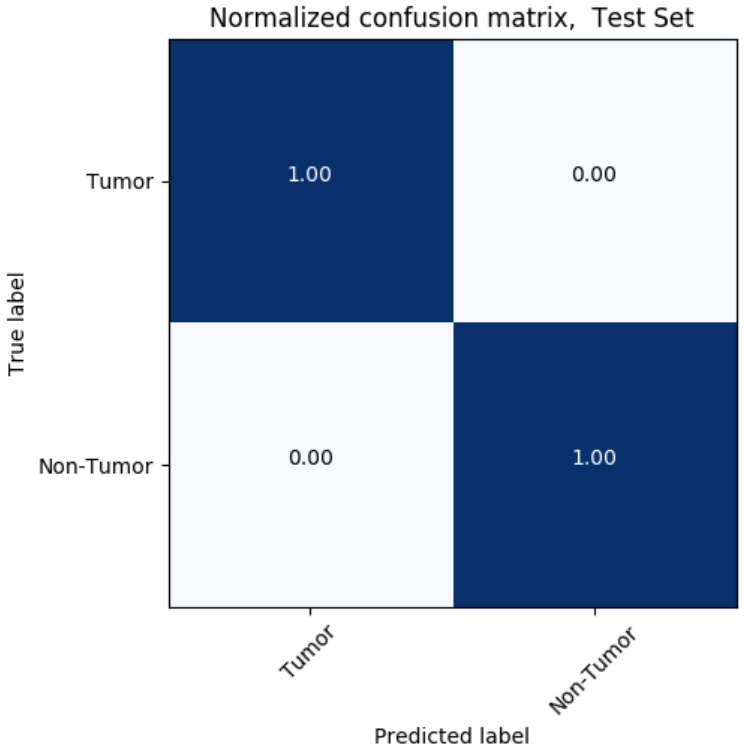
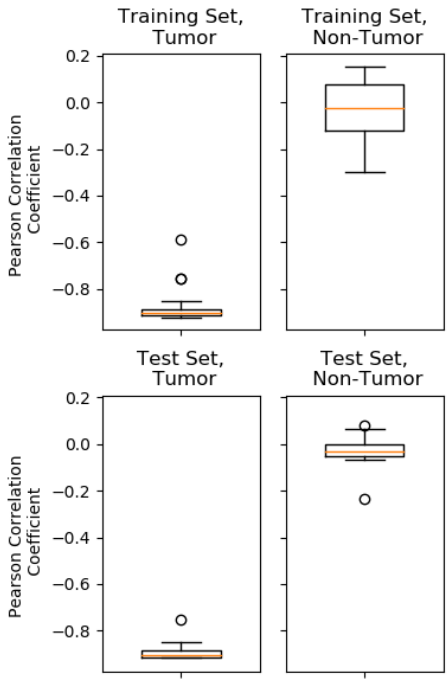
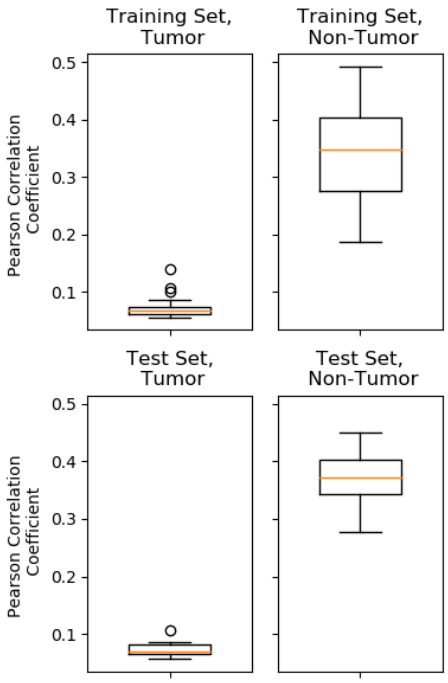


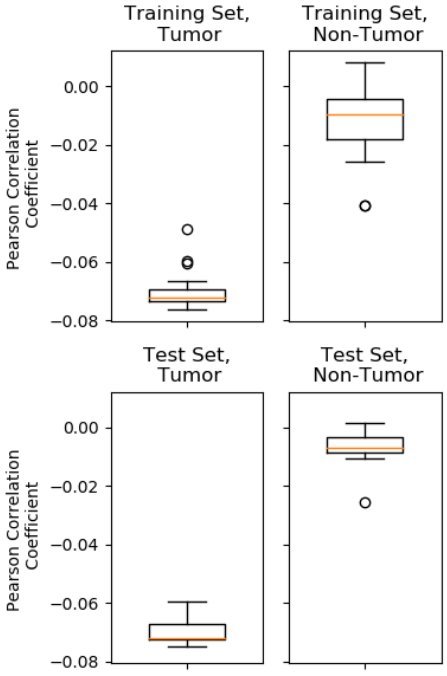
FIGURE 6.18: Validation of random forest model for melanoma classification, created with automatically selected features, on an unseen test set. All eight test samples for each class are classified correctly. Despite the small size of the test set and the fact that the training and test spectra stem from the same biological specimen, this result proves the ability of the AutoFeature algorithm to generate and select features suitable for classifications of hyperspectral images of biological samples.



(A) Feature S(714,719), import. rank 1.



(B) Feature P(173,17,4,5), import. rank 2.



(C) Feature T(163,17), import. rank 11.

FIGURE 6.19: Features S(714,719), P(173,17,4,5) and T(163,17) are automatically generated and selected by the embedded random forest AutoFeature algorithm presented. Feature S(714,719) shows excellent statistical properties for classification of tumorous and non-tumurous spectra. The distributions of the two classes are similar for the training and the test set and a clear distinction between the feature value distribution of the two classes can be made. Feature P(173,17,4,5) shows the same tendency of positive properties but to a smaller degree. Feature T(163,17) provides perfect class separability for our data, while absolute value of median difference is small. The latter finding might cause less predictive power when dealing with greater variabilities in biological samples and specimen preparations.

6.4 Results of AutoFeature Experiment with Connective Tissue Data

Connective Tissue Data Set

Figure 6.21a shows the absorbance for wavenumber 1664 cm^{-1} of a section of the dermis that contains connective tissues. It also illustrates the spatial locations of pixels selected for AutoFeature experiments.

Representative connective tissue and non-connective tissue spectra are depicted together with the ten features with largest variable importances in Figure 6.20. Similar to the melanoma case, the spectra of the two classes are very similar. Still, the features do seem to be located at locations with some sort of difference.

The random forest model built with the 25 automatically selected features results in a correct classification of all eight test samples for both classes. The feature value distribution for features with variable importance ranks 1, 2 and 5 are illustrated as boxplots in Figure 6.22. These distributions show the same trends as the analysed features in the melanoma case.

Because the results of the AutoFeature algorithm for the connective tissue dataset are analogous to the melanoma case, all analysis and interpretations are equivalent to the ones presented in the previous section 6.3.

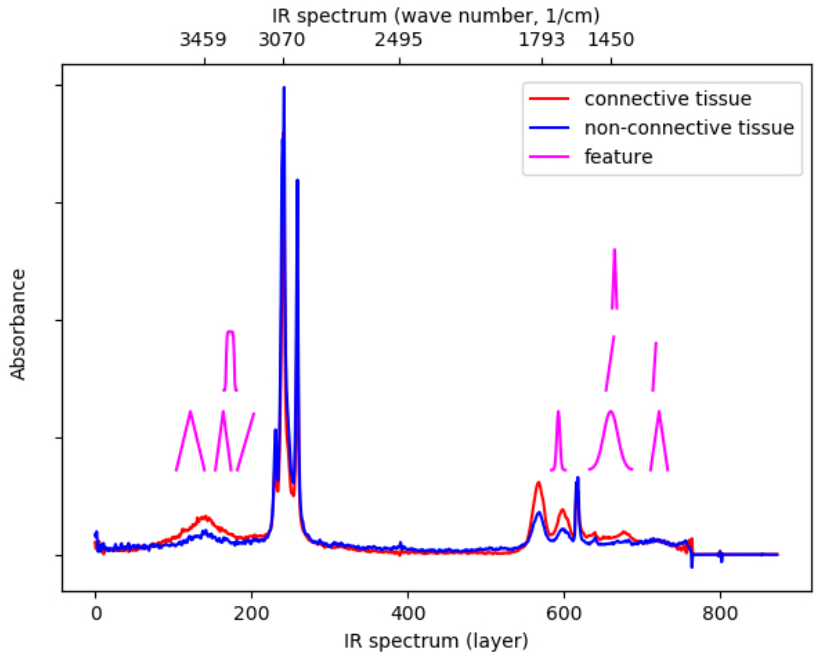
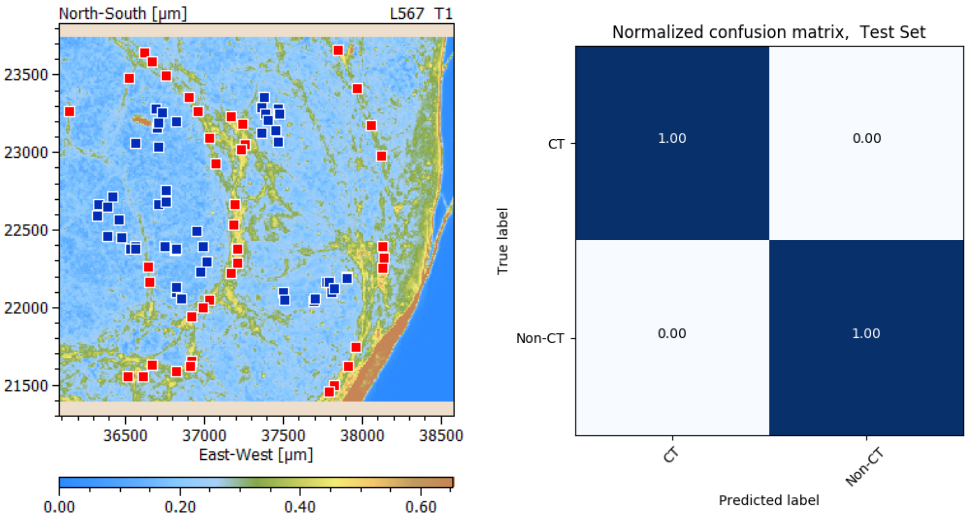


FIGURE 6.20: The connective tissue and non-connective tissue spectra are selected from the full specimen dataset. The two classes are very similar, differences can be perceived in small shifts of some absorption bands as well as intensity differences at certain wave number ranges. The ten features illustrated in magenta resulted from the embedded random forest AutoFeature algorithm.

TABLE 6.5: Resulting features and their random forest variable importance from the embedded random forest AutoFeature approach (see text) for the connective tissue dataset. Three features with an importance of 0 indicate the possibility of further dimensionality reduction.

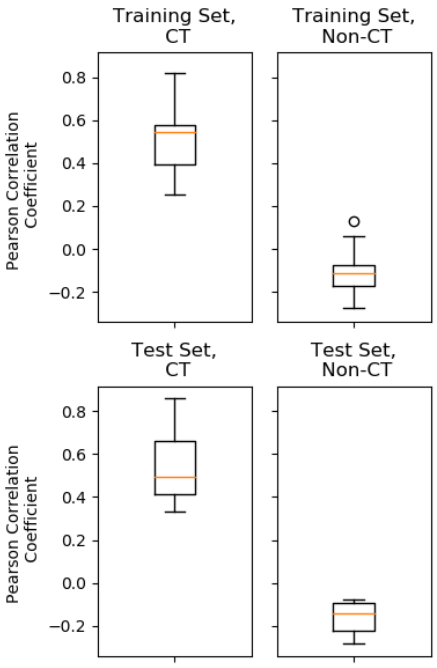
Importance	Feature	Importance	Feature
0.12	T(601,33)	0.02	G(603,4)
0.1	G(622,20)	0.02	G(622,12)
0.1	P(615,29,5.7,12.5)	0.02	G(623,20)
0.08	S(597,617)	0.02	G(626,16)
0.06	G(197,13)	0.02	G(626,21)
0.06	G(579,16)	0.02	G(628,24)
0.06	G(602,6)	0.02	P(613,29,5.7,12.5)
0.06	S(606,616)	0.02	S(595,615)
0.04	G(596,11)	0.02	S(635,644)
0.04	G(625,14)	0.0	P(612,29,5.7,12.5)
0.04	G(629,23)	0.0	S(619,628)
0.04	G(641,16)	0.0	T(599,45)
0.02	G(591,18)		



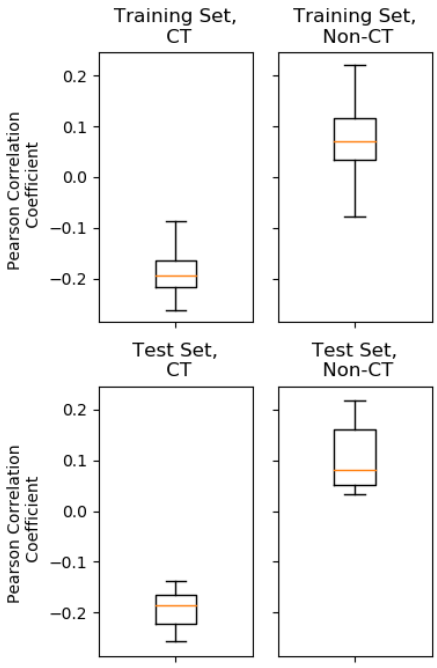
(A) (B)

FIGURE 6.21: A) The absorbance for wavenumber 1664 cm^{-1} is shown of a section of the dermis that contains the connective tissues. The markers indicate the pixels selected for AutoFeature experiments, red is the connective tissue class while blue is the non-connective tissue class.

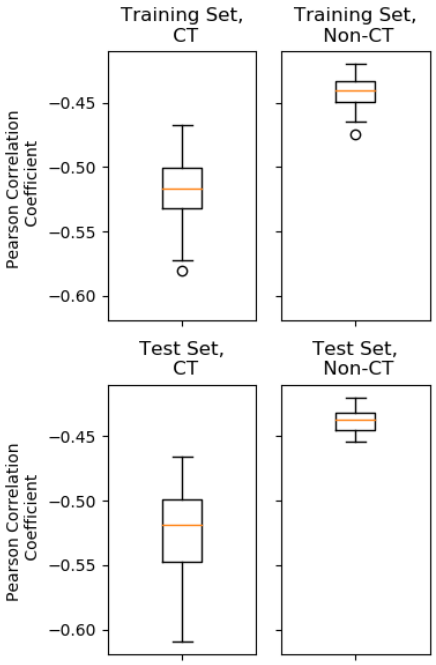
B) Validation of random forest model for connective tissue classification, created with automatically selected features on an unseen test set. All eight test samples for each class are classified correctly. Despite the small size of the test set and the fact that the training and test spectra stem from the same biological specimen, this result proves the ability of the AutoFeature algorithm to generate and select features suitable for classifications of hyperspectral images of biological samples.



(A) T(601,33), import. rank 1.



(B) G(622,20), import. rank 2.



(C) Feature G(197,13), import. rank 5

FIGURE 6.22: Features T(601,33) and G(622,20) are automatically generated and selected by the embedded random forest AutoFeature algorithm presented. Feature T(601,33) shows excellent statistical properties for classification of connective tissue and non-connective tissue spectra. The distributions of the two classes are similar for the training and the test set and a clear distinction between the feature value distribution of the two classes can be made. Feature G(622,20) shows the same tendency of positive properties but to a smaller degree. Feature G(197,13) nearly provides perfect class separability for the available data, while absolute value of median difference is small. The latter finding might cause less predictive power when dealing with greater variabilities in biological samples and specimen preparations.

Chapter 7

Conclusion

We have presented *AutoFeature*, an algorithm that is able to automatically generate and select valuable features for spectroscopic class prediction tasks. While we have investigated different methods for certain steps in the algorithm, the overall scheme has been unvarying. Firstly, thousands of feature candidates are generated with the help of generic templates. Secondly, the most promising features out of these candidates are selected by different statistical and machine learning methods.

We believe that the AutoFeature algorithm can be advantageous for any analysis of hyperspectral images or datasets consisting of continuous spectra. It has been demonstrated that the algorithm is able to extract meaningful features from annotated real-world polymer and skin tissue datasets. Notably, the algorithm has been shown to be suitable for datasets with small sample size.

In addition, AutoFeature is not only able to give experts in the field new insights but it may also open up the powerful tool of spectroscopy for non-experts. As we have shown for polymers, with the foundation of an annotated dataset, all further classification can then be done automatically. The challenge of detecting and identifying microplastic particles in aquatic environments can then be addressed with spectroscopy by many more people, which can lead to progress at a faster pace. For biological specimens, the algorithm may help to select robust features among the typically heterogeneous samples. The results in the class prediction tasks of melanoma and non-melanoma and connective tissue and non-connective tissue are promising. Certainly, assessments on larger datasets, originating from a variety of patients, have to be done in the future to evaluate the algorithm's full capabilities in this field.

Further, both steps of the algorithm, feature generation and selection, can clearly be improved in the future. Because of the insights we got from these conducted experiments, the generic template shapes may be altered and new templates may be designed. Also, refinement of feature selection methods and parameter studies may further enhance the algorithm's potential of automatic feature generation and selection in spectroscopy.

Appendix

TABLE 7.1: List of features extracted for FFX stability experiments, see Figure 6.14.

Features	Datasets											
	A 200	A 32	A 64	B 200	B 32	B 64	C 200	C 32	C 64	D 200	D 32	D 64
G(144|8)	0	0	0	0	0	0	0	0	0	0	1	0
G(145|7)	0	0	0	0	0	0	0	0	0	0	1	0
G(146|7)	0	0	0	0	0	0	0	0	0	0	1	1
G(147|7)	0	0	0	0	0	0	1	0	0	0	0	0
G(161|18)	1	0	0	0	0	0	0	0	0	0	0	0
G(164|17)	0	0	0	0	0	0	1	0	0	0	0	0
G(165|18)	0	0	0	0	0	0	0	0	0	1	0	0
G(170|7)	0	0	1	0	0	0	1	0	0	0	0	0
G(172|9)	0	0	0	0	0	1	0	0	0	0	0	0
G(189|4)	0	0	0	0	1	0	0	0	0	0	0	0
G(557|13)	0	0	0	0	0	0	1	0	0	0	0	0
G(558|13)	0	0	0	0	0	0	0	0	0	1	0	0
G(558|8)	0	0	0	0	0	1	0	0	0	0	0	0
G(577|6)	0	0	0	1	0	0	0	0	0	0	0	0
G(579|9)	0	0	0	0	0	0	0	1	0	0	0	0
P(133|31|8|13.5)	0	0	0	0	0	0	0	0	1	0	0	0
P(147|19|1|2)	0	0	0	0	0	1	0	0	0	0	0	0
P(156|15|3|3)	0	0	0	0	0	0	0	0	0	0	0	1
P(159|21|6|8.5)	1	0	0	0	0	0	0	0	0	0	0	0
P(160|21|6|8.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(162|15|1|1)	0	0	0	0	0	0	0	0	0	0	1	0
P(166|19|1|2)	0	0	0	1	0	0	0	0	0	0	0	0
P(167|17|4|5)	0	0	0	0	0	0	0	1	0	0	0	0
P(167|19|1|2)	0	0	0	0	1	1	0	0	1	0	0	0
P(168|11|3|3.5)	0	0	0	0	0	0	0	0	0	0	1	1
P(168|15|3|4)	0	0	0	0	1	0	0	0	0	0	0	0
P(171|11|3|3)	0	0	0	0	0	0	0	0	0	1	0	0
P(171|11|3|3.5)	0	0	0	0	1	1	0	0	0	0	0	0
P(171|15|4|6)	0	0	1	0	0	0	0	0	0	0	0	0
P(171|23|1|3)	0	1	0	1	0	0	1	0	1	0	0	0
P(171|9|3|3)	0	0	0	0	0	0	1	0	1	0	0	0
P(172|17|5|6.5)	0	1	0	0	0	0	0	0	0	0	0	0
P(175|33|8.5|14.5)	0	0	0	1	0	0	1	0	0	1	0	1
P(182|17|5|6.5)	0	0	0	0	1	0	0	0	0	0	0	0
P(190|11|3|3.5)	0	0	0	0	0	0	0	0	0	0	1	0
P(193|15|1|1)	0	1	0	0	0	0	0	0	0	0	0	0
P(195|15|1|1)	0	1	0	0	0	0	0	0	0	0	0	0
P(195|19|1|2)	0	0	0	0	0	0	0	0	1	0	0	0
P(197|17|5|6.5)	0	0	0	0	0	1	0	0	0	0	0	0
P(198|33|8.5|14.5)	0	0	0	1	0	0	0	0	0	0	0	0
P(199|33|8.5|14.5)	1	0	0	0	0	0	0	0	0	0	0	0
P(200|23|6|9.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(201|23|6|9.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(201|25|6.5|10.5)	0	0	0	0	1	0	0	0	0	0	0	0
P(332|33|8.5|14.5)	0	0	0	1	0	0	0	0	0	0	0	0
P(496|19|5.5|7.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(498|23|6|9.5)	0	0	0	0	0	1	0	0	0	0	0	0
P(498|25|6.5|10.5)	1	0	0	0	0	0	1	0	0	1	0	0
P(498|27|7|11.5)	0	1	0	0	0	0	0	0	0	0	0	1
P(504|33|8.5|14.5)	0	0	1	0	0	0	0	0	0	0	0	0
P(537|33|8.5|14.5)	0	0	0	1	0	0	0	0	0	0	0	0
P(538|33|8.5|14.5)	0	0	0	0	0	0	1	0	0	0	1	0
P(542|27|7|11.5)	1	0	0	0	0	0	0	0	0	0	0	0
P(545|15|1|1)	1	1	0	0	1	0	1	1	0	0	0	0
P(548|9|3|3)	0	1	0	0	0	0	0	0	0	0	0	0
P(557|13|3|3.5)	0	0	0	0	1	0	0	0	0	0	0	0
P(557|15|3|4)	0	0	1	1	0	1	1	1	0	0	0	1

Continuation of Table 7.1.

Features	Datasets											
	A 200	A 32	A 64	B 200	B 32	B 64	C 200	C 32	C 64	D 200	D 32	D 64
P(557|15|4|6)	0	0	0	0	0	0	0	1	1	0	0	0
P(557|17|4|5)	1	0	0	1	0	0	1	0	1	1	0	1
P(557|25|6.5|10.5)	0	0	1	0	0	0	0	0	0	0	0	0
P(558|15|3|4)	1	0	0	0	0	0	0	0	0	0	0	0
P(558|17|4|5)	1	0	0	1	0	1	1	1	1	1	1	1
P(558|17|5|6.5)	0	0	0	0	0	0	0	0	0	1	1	0
P(558|19|5|6)	0	0	0	0	0	0	0	0	0	0	1	0
P(561|31|8|13.5)	0	0	1	0	0	0	0	0	0	0	0	0
P(565|27|7|11.5)	0	1	0	0	0	0	0	0	0	0	0	0
P(566|29|5.7|12.5)	0	1	1	0	0	0	0	0	0	0	0	0
P(571|11|3|3.5)	0	0	1	0	0	0	0	0	0	0	0	0
P(572|15|1|1)	1	0	0	1	0	1	0	0	0	1	0	0
P(572|15|3|3)	0	0	0	0	0	0	1	0	0	0	0	1
P(572|33|8.5|14.5)	0	0	0	0	0	1	1	0	0	0	0	1
P(573|33|8.5|14.5)	0	0	0	0	0	0	0	0	0	0	1	0
P(578|15|1|1)	1	0	0	0	0	0	0	0	0	0	0	1
P(579|15|1|1)	0	0	0	0	0	0	0	0	0	0	1	0
P(585|33|8.5|14.5)	1	0	0	1	0	0	0	0	0	0	0	0
P(586|31|8|13.5)	0	0	0	0	0	0	1	0	1	0	0	0
P(587|31|8|13.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(593|15|4|6)	0	0	0	0	1	0	0	0	0	0	0	0
P(593|21|6|8.5)	1	0	1	0	0	0	0	0	0	1	1	1
P(594|17|5|6.5)	0	0	0	0	0	0	0	0	0	0	1	0
P(594|19|5.5|7.5)	0	0	0	0	0	0	0	0	1	0	0	0
P(594|21|6|8.5)	0	0	0	0	0	0	0	1	0	0	0	0
P(596|15|4|6)	0	0	0	0	0	1	0	0	0	0	0	0
S(192|198)	1	0	0	0	0	0	0	0	0	1	0	0
S(192|199)	1	0	0	0	0	1	1	0	1	0	1	0
S(192|200)	0	1	1	0	1	0	0	0	0	0	1	0
S(193|198)	1	1	1	1	1	1	1	1	1	1	1	1
S(193|199)	0	1	1	1	1	0	0	0	0	0	0	0
S(194|198)	0	0	0	0	0	0	0	1	0	0	0	0
S(543|553)	0	0	0	0	0	0	0	0	0	0	0	1
S(543|554)	0	0	0	1	0	0	0	0	1	1	0	0
S(544|553)	0	0	0	1	1	1	1	0	1	1	0	1
S(544|554)	0	0	0	1	0	1	0	0	1	1	0	0
S(545|553)	1	1	0	0	0	0	0	0	0	0	0	0
S(545|554)	0	1	1	0	0	0	0	0	0	0	0	0
S(550|557)	0	0	0	0	1	0	0	0	0	0	0	0
S(551|556)	0	1	0	0	1	0	0	0	0	0	0	0
S(552|572)	0	0	0	0	0	0	0	0	0	0	0	1
S(575|587)	0	0	0	0	0	0	0	0	1	0	0	0
S(576|586)	0	0	0	0	0	1	0	0	0	0	0	0
S(577|588)	0	0	1	0	0	0	0	0	0	0	0	0
T(126|53)	0	0	0	0	0	0	0	1	0	0	0	0
T(167|5)	0	1	0	0	0	0	0	0	0	0	0	0
T(170|41)	0	1	0	0	0	0	0	0	1	0	0	0
T(170|7)	0	0	0	0	0	0	0	0	0	0	1	0
T(171|41)	0	0	0	1	0	0	0	0	0	0	0	0
T(194|7)	0	0	0	1	0	0	0	0	0	0	0	0
T(196|21)	0	0	1	0	0	0	0	0	0	0	0	0
T(548|9)	0	0	1	0	1	0	0	0	0	0	0	0
T(555|41)	0	1	0	0	0	0	0	0	0	0	0	0
T(556|41)	0	1	0	0	0	0	0	0	0	0	0	0
T(557|15)	0	0	1	0	0	1	0	0	0	0	0	0
T(578|47)	0	0	1	0	0	0	0	0	0	0	0	0
T(579|49)	0	0	0	0	0	0	0	0	1	0	0	0

Bibliography

[1] Douglas A. Skoog, F. James Holler, and Stanley R. Crouch. *Principles of Instrumental Analysis*. en. Cengage Learning, Jan. 2017. ISBN: 978-1-305-57721-3.

[2] X. Cheng et al. "A NOVEL INTEGRATED PCA AND FLD METHOD ON HYPERSPECTRAL IMAGE FEATURE EXTRACTION FOR CUCUMBER CHILLING DAMAGE INSPECTION". en. In: *Transactions of the ASAE* 47.4 (2004), pp. 1313–1320. ISSN: 2151-0059. DOI: 10.13031/2013.16565.

[3] J. Ren et al. "Effective Feature Extraction and Data Reduction in Remote Sensing Using Hyperspectral Imaging [Applications Corner]". In: *IEEE Signal Processing Magazine* 31.4 (July 2014), pp. 149–154. ISSN: 1053-5888. DOI: 10.1109/MSP.2014.2312071.

[4] S. Kumar, J. Ghosh, and M. M. Crawford. "Best-bases feature extraction algorithms for classification of hyperspectral data". In: *IEEE Transactions on Geoscience and Remote Sensing* 39.7 (July 2001), pp. 1368–1379. ISSN: 0196-2892. DOI: 10.1109/36.934070.

[5] Zoltán Kovács and Szilárd Szabó. "An interactive tool for semi-automatic feature extraction of hyperspectral data". In: *Open Geosciences* 8 (Sept. 2016), p. 40. ISSN: 2391-5447. DOI: 10.1515/geo-2016-0040.

[6] Lindi J. Quackenbush. "A Review of Techniques for Extracting Linear Features from Imagery". en. In: *Photogrammetric Engineering & Remote Sensing* 70.12 (Dec. 2004), pp. 1383–1392. ISSN: 00991112. DOI: 10.14358/PERS.70.12.1383.

[7] J. L. Koenig. *Spectroscopy of Polymers, Second Edition*. English. 2 edition. Amsterdam: Elsevier Science, Sept. 1999. ISBN: 978-0-444-10031-3.

[8] Michael M. Coleman and Paul C. Painter. *Fundamentals of Polymer Science*. en. Taylor & Francis, July 1996. ISBN: 978-1-56676-152-9.

[9] N. G. McCrum, C. P. Buckley, and C. B. Bucknall. *Principles of Polymer Engineering*. Second Edition. Oxford, New York: Oxford University Press, Aug. 1997. ISBN: 978-0-19-856526-0.

[10] Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. "Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF". en. In: *Applied Intelligence* 7.1 (Jan. 1997), pp. 39–55. ISSN: 0924-669X, 1573-7497. DOI: 10.1023/A:1008280620621.

[11] National Oceanic and Atmospheric Administration US Department of Commerce. *What are microplastics?* EN-US. URL: https://oceanservice.noaa.gov/facts/microplastics.html (visited on 02/14/2018).

[12] Martin G. J. Löder and Gunnar Gerdts. "Methodology Used for the Detection and Identification of Microplastics—A Critical Appraisal". en. In: *Marine Anthropogenic Litter*. Springer, Cham, 2015, pp. 201–227. ISBN: 978-3-319-16509-7 978-3-319-16510-3. DOI: 10.1007/978-3-319-16510-3_8.

[13] Mushabeb Z. Alqahtani, Arifusalam Shaikh, and Malick M. Ndiaye. "Focused Plant Optimization Strategy for Polyethylene Multi-grades and Multi-sites Production". en. In: *Arabian Journal for Science and Engineering* (Nov. 2017), pp. 1–13. ISSN: 2193-567X, 2191-4281. DOI: 10.1007/s13369-017-2882-7.

[14] J.V Gulmine et al. "Polyethylene characterization by FTIR". en. In: *Polymer Testing* 21.5 (Jan. 2002), pp. 557–563. ISSN: 01429418. DOI: 10.1016/S0142-9418(01)00124-6.

[15] *Polypropylene (PP) - Study: Market, Analysis | Ceresana*. URL: http://www.ceresana.com/en/market-studies/plastics/polypropylene/ (visited on 02/15/2018).

[16] M. P. McDonald and I. M. Ward. "The assignment of the infra-red absorption bands and the measurement of tacticity in polypropylene". In: *Polymer* 2 (Jan. 1961), pp. 341–355. ISSN: 0032-3861. DOI: 10.1016/0032-3861(61)90037-4.

[17] *What is Polystyrene? | Uses, Benefits, and Safety Facts*. en-US. May 2014. URL: https://www.chemicalsafetyfacts.org/polystyrene-post/ (visited on 02/15/2018).

[18] Umar Ali, Khairil Juhanni Bt Abd Karim, and Nor Aziah Buang. "A Review of the Properties and Applications of Poly (Methyl Methacrylate) (PMMA)". In: *Polymer Reviews* 55.4 (Oct. 2015), pp. 678–705. ISSN: 1558-3724. DOI: 10.1080/15583724.2015.1031377.

[19] Guorong Duan et al. "Preparation and Characterization of Mesoporous Zirconia Made by Using a Poly (methyl methacrylate) Template". In: *Nanoscale Research Letters* 3.3 (Feb. 2008), pp. 118–122. ISSN: 1931-7573. DOI: 10.1007/s11671-008-9123-7.

[20] Robson Fleming Ribeiro et al. "Thermal Stabilization study of polyacrylonitrile fiber obtained by extrusion". In: *Polímeros* 25.6 (Dec. 2015), pp. 523–530. ISSN: 0104-1428. DOI: 10.1590/0104-1428.1938.

[21] *Melanoma Treatment*. en. pdqCancerInfoSummary. URL: https://www.cancer.gov/types/skin/hp/melanoma-treatment-pdq (visited on 05/02/2018).

[22] Bernard W. Stewart et al., eds. *World cancer report*. en. OCLC: 249562218. Lyon: IARC Press, 2003. ISBN: 978-92-832-0411-4.

[23] *Melanoma - SkinCancer.org*. URL: https://www.skincancer.org/skin-cancer-information/melanoma (visited on 05/02/2018).

[24] E. R. Farmer, R. Gonin, and M. P. Hanna. "Discordance in the histopathologic diagnosis of melanoma and melanocytic nevi between expert pathologists". eng. In: *Human Pathology* 27.6 (June 1996), pp. 528–531. ISSN: 0046-8177.

[25] R. Corona et al. "Interobserver variability on the histopathologic diagnosis of cutaneous melanoma and other pigmented skin lesions". eng. In: *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 14.4 (Apr. 1996), pp. 1218–1223. ISSN: 0732-183X. DOI: 10.1200/JCO.1996.14.4.1218.

[26] Richard M. Levenson. "Spectral Imaging and Pathology: Seeing More". en. In: *Laboratory Medicine* 35.4 (Sept. 2015), pp. 244–251. ISSN: 0007-5027, 1943-7730. DOI: 10.1309/KRNFWQQEUPLQL76L.

[27] Matthew J. Baker et al. "Using Fourier transform IR spectroscopy to analyze biological materials". eng. In: *Nature Protocols* 9.8 (Aug. 2014), pp. 1771–1791. ISSN: 1750-2799. DOI: 10.1038/nprot.2014.110.

[28] Ali Tfayli et al. "Discriminating nevus and melanoma on paraffin-embedded skin biopsies using FTIR microspectroscopy". eng. In: *Biochimica Et Biophysica Acta* 1724.3 (Aug. 2005), pp. 262–269. ISSN: 0006-3002. DOI: 10.1016/j.bbagen.2005.04.020.

[29] Robert H. Wilson et al. "Review of short-wave infrared spectroscopy and imaging methods for biological tissue characterization". In: *Journal of Biomedical Optics* 20.3 (Mar. 2015). ISSN: 1083-3668. DOI: 10.1117/1.JBO.20.3.030901.

[30] Christopher Bishop. *Pattern Recognition and Machine Learning*. en. Information Science and Statistics. New York: Springer-Verlag, 2006. ISBN: 978-0-387-31073-2.

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. en. 2nd ed. Springer Series in Statistics. New York: Springer-Verlag, 2009. ISBN: 978-0-387-84857-0.

[32] Nils J. Nilsson. *Introduction to Machine Learning, a early draft of a proposed textbook*. 1998. URL: stanford.edu/people/nilsson/MLBOOK.pdf (visited on 02/07/2018).

[33] James Burke. *Optimization of Quadratic Functions*. Notes Course 408. 2014.

[34] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1 (Feb. 1970), pp. 55–67. ISSN: 0040-1706. DOI: 10.1080/00401706.1970.10488634.

[35] L. E. Melkumova and S. Ya. Shatskikh. "Comparing Ridge and LASSO estimators for data analysis". In: *Procedia Engineering*. 3rd International Conference "Information Technology and Nanotechnology", ITNT-2017, 25-27 April 2017, Samara, Russia 201 (Jan. 2017), pp. 746–755. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2017.09.615.

[36] Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic net". en. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (Apr. 2005), pp. 301–320. ISSN: 1369-7412, 1467-9868. DOI: 10.1111/j.1467-9868.2005.00503.x.

[37] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 0035-9246.

[38] Hamed Haselimashhadi and Veronica Vinciotti. "A Differentiable Alternative to the Lasso Penalty". In: *arXiv:1609.04985 [stat]* (Sept. 2016). arXiv: 1609.04985.

[39] Jerome Friedman et al. "Pathwise coordinate optimization". EN. In: *The Annals of Applied Statistics* 1.2 (Dec. 2007), pp. 302–332. ISSN: 1932-6157, 1941-7330. DOI: 10.1214/07-AOAS131.

[40] Tong Tong Wu and Kenneth Lange. "Coordinate descent algorithms for lasso penalized regression". In: *The Annals of Applied Statistics* 2.1 (Mar. 2008). arXiv: 0803.3876, pp. 224–244. ISSN: 1932-6157. DOI: 10.1214/07-AOAS147.

[41] Mark Schmidt. "Least Squares Optimization with L1-Norm Regularization". In: *CS542B Project Report* (2005), pp. 14–18.

[42] Ryan Tibshirani. *Coordinate Descent*. Course Convex Optimization 10-725/36-725. 2015. URL: http://www.stat.cmu.edu/~ryantibs/convexopt-S15/lectures/.

[43] Trent McConaghy. "FFX: Fast, Scalable, Deterministic Symbolic Regression Technology". en. In: *Genetic Programming Theory and Practice IX*. Genetic and Evolutionary Computation. Springer, New York, NY, 2011, pp. 235–260. ISBN: 978-1-4614-1769-9 978-1-4614-1770-5. DOI: 10.1007/978-1-4614-1770-5_13.

[44] John R. Koza. "Genetic programming as a means for programming computers by natural selection". en. In: *Statistics and Computing* 4.2 (June 1994), pp. 87–112. ISSN: 0960-3174, 1573-1375. DOI: 10.1007/BF00175355.

[45] Leo Breiman. "Random Forests". en. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1010933404324.

[46] Leo Breiman et al. *Classification and Regression Trees*. en. Taylor & Francis, Jan. 1984. ISBN: 978-0-412-04841-8.

[47] Leo Breiman. "Bagging Predictors". en. In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1018054314350.

[48] Gérard Biau and Erwan Scornet. "A Random Forest Guided Tour". In: *Test* 25.2 (Nov. 2016). arXiv: 1511.05741, pp. 197–227.

[49] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. "Gene selection and classification of microarray data using random forest". In: *BMC Bioinformatics* 7 (Jan. 2006), p. 3. ISSN: 1471-2105. DOI: 10.1186/1471-2105-7-3.

[50] Erwan Scornet. "Tuning parameters in random forests". en. In: *ESAIM: Proceedings and Surveys* 60 (2017), pp. 144–162. ISSN: 2267-3059. DOI: 10.1051/proc/201760144.

[51] Richard Bellman and Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. en. Princeton University Press, 1961.

[52] Richard Bellman. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 2010. ISBN: 978-0-691-14668-3.

[53] N. Wyse, R. Dubes, and A.K. Jain. "Pattern recognition in practice". In: Pattern recognition in practice (1980), pp. 415–425.

[54] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901), pp. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720.

[55] M. Dash and H. Liu. "Feature selection for classification". In: *Intelligent Data Analysis* 1.1 (Jan. 1997), pp. 131–156. ISSN: 1088-467X. DOI: 10.1016/S1088-467X(97)00008-5.

[56] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3.Mar (2003), pp. 1157–1182. ISSN: ISSN 1533-7928.

[57] Kenji Kira and Larry A. Rendell. "The Feature Selection Problem: Traditional Methods and a New Algorithm". In: *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI'92. San Jose, California: AAAI Press, 1992, pp. 129–134. ISBN: 978-0-262-51063-9.

[58] Ryan J. Urbanowicz et al. "Benchmarking Relief-Based Feature Selection Methods". In: *arXiv:1711.08477 [cs]* (Nov. 2017). arXiv: 1711.08477.

[59] Marko Robnik-Šikonja and Igor Kononenko. "Theoretical and Empirical Analysis of ReliefF and RReliefF". en. In: *Machine Learning* 53.1-2 (Oct. 2003), pp. 23–69. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1025667309714.

[60] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000. ISBN: 978-0-471-05669-0.

[61] Jundong Li et al. "Feature Selection: A Data Perspective". In: *arXiv:1601.07996 [cs]* (Jan. 2016). arXiv: 1601.07996.

[62] Makoto Yamada et al. "High-Dimensional Feature Selection by Feature-Wise Kernelized Lasso". In: *Neural Computation* 26.1 (Oct. 2013), pp. 185–207. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00537.

[63] R. A. Fisher. "FREQUENCY DISTRIBUTION OF THE VALUES OF THE CORRELATION COEFFIENTS IN SAMPLES FROM AN INDEFINITELY LARGE POPU;ATION". en. In: *Biometrika* 10.4 (May 1915), pp. 507–521. ISSN: 0006-3444. DOI: 10.1093/biomet/10.4.507.

[64] Charles J. Kowalski. "On the Effects of Non-Normality on the Distribution of the Sample Product-Moment Correlation Coefficient". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 21.1 (1972), pp. 1–12. ISSN: 0035-9254. DOI: 10.2307/2346598.

[65] F. Mosteller and John W. Tukey. *Data Analysis. Including Statistics. .(G. Lindzey and E. Aronson, eds.)* Vol. 2. Handbook of Social Psychology, Chapter 10. Addison-Wesley, Reading. MA, 1968.

[66] Hans Lohninger. *ImageLab*. Retz, Austria. URL: http://www.imagelab.at/ (visited on 04/18/2018).

[67] *Python 3.6*. URL: https://www.python.org/ (visited on 04/18/2018).

[68] *HYPERION - Overview*. en. URL: https://www.bruker.com/products/infrared-near-infrared-and-raman-spectroscopy/ft-ir-microscopes-raman-microscopes/hyperion/overview.html (visited on 02/20/2018).

[69] Dozent Dr. Christine Hafner. *Private Communications*. Department of Pathophysiology and Allergy Research at the Medical University of Vienna.