Adaptive Conjoint Wavelet–Support Vector Classifiers

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Julia Neumann
aus Idar-Oberstein

Mannheim, 2004

Dekan: Professor Dr. Jürgen Potthoff, Universität Mannheim
Referent: Professor Dr. Gabriele Steidl, Universität Mannheim
Korreferent: Professor Dr. Christoph Schnörr, Universität Mannheim

Tag der mündlichen Prüfung: 15. Februar 2005

Dept. of Mathematics and Computer Science
UNIVERSITY OF MANNHEIM
68131 Mannheim, Germany

Dissertation

Adaptive Conjoint Wavelet–Support Vector Classifiers

Julia Neumann

December 16, 2004

Acknowledgements

I express my most cordial gratitude to my supervisors Christoph Schnörr and Gabriele Steidl without whose support I would not have had the courage to hold on!

The research on which this thesis is based was funded by the DFG under grant Schn 457/5.

Thanks to Ole Jakubik for providing the material from his diploma thesis.

Further thanks to Stefan Schmidt for the lively and dependable collaboration concerning the evaluation of the CT data, as well as to Dr. Pekar and Dr. Kaus, Philips Research Laboratories Hamburg, for providing the data.

Summary

Combined wavelet – large margin classifiers succeed in solving difficult signal classification problems in cases where solely using a large margin classifier like, e.g., the Support Vector Machine may fail. This thesis investigates the problem of conjointly designing both classifier stages to achieve a most effective classifier architecture. Particularly, the wavelet features should be adapted to the Support Vector classifier and the specific classification problem. Three different approaches to achieve this goal are considered:

The classifier performance is seriously affected by the wavelet or filter used for feature extraction. To optimally choose this wavelet with respect to the subsequent Support Vector classification, appropriate criteria may be used. The radius – margin Support Vector Machine error bound is proven to be computable by two standard Support Vector problems. Criteria which are computationally still more efficient may be sufficient for filter adaptation. For the classification by a Support Vector Machine, several criteria are examined rating feature sets obtained from various orthogonal filter banks. An adaptive search algorithm is devised that, once the criterion is fixed, efficiently finds the optimal wavelet filter.

To extract shift invariant wavelet features, Kingsbury's dual–tree complex wavelet transform is examined. The dual–tree filter bank construction leads to wavelets with vanishing negative frequency parts. An enhanced transform is established in the frequency domain for standard wavelet filters without special filter design. The translation and rotational invariance is improved compared with the common wavelet transform as shown for various standard wavelet filters. So the framework well applies to adapted signal classification.

Wavelet adaptation for signal classification is a special case of feature selection. Feature selection is an important combinatorial optimisation problem in the context of supervised pattern classification. Four novel continuous feature selection approaches directly minimising the classifier performance are presented. In particular, they include linear and nonlinear Support Vector classifiers. The key ideas of the approaches are additional regularisation and embedded nonlinear feature selection. To solve the optimisation problems, difference of convex functions programming which is a general framework for non-convex continuous optimisation is applied. This optimisation framework may also be interesting for other applications and succeeds in robustly solving the problems, and hence, building more powerful feature selection methods.

Zusammenfassung

Kombinierten Wavelet – Supportvektor-Klassifikatoren gelingt es, schwierige Signalklassifikationsprobleme in Fällen zu lösen, in denen die alleinige Anwendung einer Supportvektor Maschine fehlschlägt. Die vorliegende Arbeit untersucht den gemeinsamen Entwurf beider Klassifikatorstufen, um eine möglichst effiziente Klassifikationsarchitektur zu erreichen. Insbesondere sollten die Waveletmerkmale an den Supportvektor-Klassifikator und das spezielle Klassifikationsproblem angepasst werden. Drei verschiedene dieses Ziel verfolgende Ansätze werden betrachtet:

Die Klassifikationsgenauigkeit hängt stark von der Wahl des Wavelets oder Filters für die Merkmalsextraktion ab. Um dieses Wavelet optimal im Hinblick auf die nachfolgende Supportvektor-Klassifikation auszuwählen, können geeignete Kriterien herangezogen werden. Es wird gezeigt, dass die "Radius – Margin" Fehlerschranke für Supportvektor Maschinen von zwei Standard–Supportvektor Problemen berechenbar ist. Noch effizientere Kriterien können für die Filteranpassung ausreichen. Für die Supportvektor-Klassifikation werden einige Kriterien verglichen, die von verschiedenen orthogonalen Filterbänken erzeugte Merkmalsmengen bewerten. Es wird ein adaptives Suchverfahren entworfen, das, gegeben ein Kriterium, effizient das optimalen Waveletfilter findet.

Um translationsinvariante Waveletmerkmale zu extrahieren wird Kingsburys komplexe Wavelettransformation betrachtet. Diese Filterbankkonstruktion führt zu Wavelets ohne negative Frequenzanteile. Eine erweiterte Transformation für Standard–Waveletfilter ohne speziellen Filterentwurf wird im Frequenzbereich eingeführt. Die Translations- und Rotationsinvarianz wird dadurch gegenüber der gewöhnlichen Wavelettransformation verbessert, wie für vielfältige Standard–Waveletfilter gezeigt wird. Damit lässt sich diese Konstruktion vorteilhaft in der angepassten Signalklassifikation anwenden.

Waveletanpassung für die Signalklassifikation ist ein Spezialfall der Merkmalsauswahl, einem wichtigen kombinatorischen Optimierungsproblem im Problembereich der überwachten Mustererkennung. Vier neuartige stetige Merkmalsauswahlansätze werden vorgestellt, die direkt die Klassifikationsgenauigkeit minimieren. Insbesondere berücksichtigen diese lineare und nichtlineare Supportvektor-Klassifikatoren. Die Kernideen der Ansätze sind zusätzliche Regularisierung und eingebettete nichtlineare Merkmalsauswahl. Um die Optimierungsprobleme zu lösen, wird die Zielfunktion als Differenz konvexer Funktionen dargestellt. Zur Lösung derartiger nicht-konvexer, stetiger Optimierungsprobleme wird ein allgemeines Verfahren, der DCA, angewendet. Dieser könnte ebenso für andere Anwendungen interessant sein. Damit gelingt es, die Probleme robust zu lösen und somit leistungsfähigere Merkmalsauswahlmethoden zu konstruieren.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

General:

e	Euler's number $e \approx 2.718$
ln	natural logarithm $e^{\ln x} = x$
x_+	$x_+ := \max(x, 0)$
$\oplus$	binary XOR operator
$\overline{x}$	conjugate of complex number $\overline{a + ib} = a - ib$ for $a, b \in \mathbb{R}$
$\mathbf{x}$	column vector $\mathbf{x} = (x_i)_i$
$\mathbf{0}$	vector of zeros in the respective space
$\mathbf{e}$	vector of ones in the respective space
$\mathbf{A}$	matrix $\mathbf{A} = (a_{ij})_{ij}$
$\mathbf{I}$	identity matrix in appropriate dimensions
$\mathbf{A}^\top$	transpose of matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	trace of matrix $\mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}$
$\mathrm{diag}(a_i)$	diagonal matrix with entries $a_{ii} = a_i$ and $a_{ij} = 0$ for $i \neq j$
$\lvert\mathbf{w}\rvert$	componentwise absolute value $\lvert\mathbf{w}\rvert = (\lvert w_1\rvert, \lvert w_2\rvert, \dots)^\top$
$\mathbf{x}\,\theta\,\mathbf{y}$	componentwise inequality: $x_i\,\theta\,y_i\ \forall\, i$ with $\theta \in \{<, \leq, >, \geq\}$
P	probability of an event

Sets and Spaces:

$\overline{X}$	closure of set X
$\oplus$	direct sum of subspaces
span	finite linear combinations of set of vectors
$\mathbb{N}_0$	natural numbers including zero $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$
$\mathbb{R}_+$	set of positive real numbers $\{x \in \mathbb{R} : x > 0\}$
$\mathbb{R}_{0+}$	set of nonnegative real numbers $\{x \in \mathbb{R} : x \geq 0\}$
$\overline{\mathbb{R}}$	closure of $\mathbb{R}$: $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$
ℓ_p	Banach spaces of real valued absolutely p-summable sequences $\mathbf{a} = (a_i)_{i\in\mathbb{N}}$ with norm $\|\mathbf{a}\|_p = \|\mathbf{a}\|_{\ell_p} = \left(\sum_{i\in\mathbb{N}} \lvert a_i\rvert^p\right)^{1/p}$ for $p \geq 1$; Hilbert space ℓ_2 with Euclidean norm $\|\cdot\| = \|\cdot\|_2$ has corresponding inner product $\langle \mathbf{a}, \mathbf{b}\rangle = \langle \mathbf{a}, \mathbf{b}\rangle_{\ell_2} = \sum_{i\in\mathbb{N}} a_i b_i$
$\langle \cdot, \cdot \rangle_{\mathrm{F}}$	Frobenius inner product $\langle \mathbf{A}, \mathbf{B}\rangle_{\mathrm{F}} = \mathrm{tr}(\mathbf{A}^\top \mathbf{B}) = \sum_{i,j} a_{ij} b_{ij}$
$L_2(\mathcal{X})$	Hilbert space of real valued square integrable functions on $\mathcal{X}$ with inner product $\langle f, g\rangle_{L_2} = \int_{\mathcal{X}} f(x) g(x)\,\mathrm{d}x$
$\mathcal{C}^m$	space of m times continuously differentiable functions on $\mathbb{R}$
$[-\mathbf{v}, \mathbf{v}]$	cuboid $\{\mathbf{w} \in \mathbb{R}^d : -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v}\}$ for $\mathbf{v} \in \mathbb{R}^d$

Functions:

sgn	sign function $\mathrm{sgn}(x) := \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{otherwise} \end{cases}$
$\mathrm{supp}\, f$	support of function $\mathrm{supp}\, f := \overline{\{x : f(x) \neq 0\}}$
$\mathrm{dom}\, f$	domain of convex function f (see Def. 7)
f^*	conjugate of function f (see Def. 8)
$\nabla f(\mathbf{x})$	gradient $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_i}(\mathbf{x})\right)_i$
$Hf(\mathbf{x})$	Hessian matrix $Hf(\mathbf{x}) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x})\right)_{ij}$
$\partial f(\mathbf{x})$	subgradient of f at $\mathbf{x}$ (see Def. 6)
δ	unit impulse $\delta(x) := \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{otherwise} \end{cases}$
χ_C	indicator function of a feasible convex set C: $\chi_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{otherwise} \end{cases}$
$\mathcal{L}$	Lagrangian function

Pattern Recognition:

m	input signal dimension (often $m = 1$)
l	input signal length
$\mathbf{s}$	input signal $\mathbf{s} \in \mathbb{R}^l$
$\mathbf{S}$	2D input signal $\mathbf{S} \in \mathbb{R}^{l \cdot l}$
d	number of features, dimension of pattern vectors
$\mathcal{X}$	compact pattern space $\mathcal{X} \subset \mathbb{R}^d$
$\mathbf{x}$	pattern (vectors) $\mathbf{x} \in \mathcal{X}$
y	target class label $y \in \{-1, 1\}$
$\mathbf{y}$	vector of class labels
n	number of training data
$n_{\pm 1}$	class cardinality $n_{\pm 1} = \lvert\{i : y_i = \pm 1\}\rvert$
$\mathcal{Z}$	training set $\mathcal{Z} := \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\}$
t	target function $t : \mathcal{X} \to \{-1, 1\}$
$\mathbf{X}$	matrix of pattern vectors $\mathbf{X} \in \mathbb{R}^{n \times d}$, each row is a pattern
$\mathbf{S}_w, \mathbf{S}_b, \mathbf{S}_m$	scatter matrices, see Sec. 4.2

SVMs:

K	kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
$\mathcal{H}_K$	reproducing kernel Hilbert space according to kernel K
$\mathcal{F}_K$	feature space $\mathcal{F}_K \subset \ell_2$ according to kernel K
$\boldsymbol{\phi}$	feature map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{F}_K$
$\mathbf{K}$	kernel matrix $\mathbf{K} = (K(\mathbf{x_i}, \mathbf{x}_j))_{i,j=1}^n$
$\mathbf{w}$	normal of separating hyperplane $\mathbf{w} \in \mathcal{F}_K$
b	bias term, offset of hyperplane $\{\mathbf{x} \in \mathcal{F}_K : \mathbf{w}^\top \mathbf{x} + b = 0\}$
$\boldsymbol{\xi}$	slack variables
C	weight parameter for SVM
D	weight parameter for feature selection methods
$\mathbf{Y}$	diagonal matrix of class labels
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	Lagrange multipliers, dual SVM variables

1. Introduction

What happens in your brain when you read this sentence? Assuming you are already in a reading mood, you identify the single letters and then try to match the corresponding black and white areas to known letter symbols. The human visual system does the same with other, possibly more complex objects.

And how do you recognise the single letters? It is the result of a learning process. Of course the visual system develops by itself, but the knowledge about patterns in the real world has to be learnt by examples. This is an example of a supervised classification problem as considered in this thesis. We will mainly deal with two-class classification problems where we assume we are given two sets of samples belonging to different classes. We intend to construct a machine, that is a learning system, that assigns class labels for new samples based on the known training set. A classifier represents the samples by vectors and deduces a decision rule from the training set that maps new sample vectors to possible class labels. There are many possible classifier choices. A popular, general, competitive algorithm is the Support Vector Machine that generates statistically well generalising decision rules supported by few training samples only.

Classification problems often emerge in the real world: More complex examples, also to the human observer, are the analysis of gene expressions and in the continuous case medical applications and acoustic signals in one dimension and texture images in two dimensions. Sample signals for the detection of ventricular tachycardia as a medical application are shown in Fig. 1.1. Texture are pseudo-regular patterns with inherent structure such as images of forest ground, a crowd of people or simply fire. One-dimensionally structured samples are shown in Fig. 1.2.

Now how do we sensibly "feed" our learning machine with the data? A fundamental principle handling complex problems is to split them into subproblems. Reconsider the reading example: A child does not only learn patterns, but also learns to represent them in a suitable way, e.g. characters by single line segments. In the examples of texture and heartbeat signals, we are provided with a vast amount of grey value/colour pixels or momentary measured frequencies, respectively. A suitable representation for these signals are the overall shape combined with smaller patterns capturing the detailed structure. Wavelets provide such a representation.

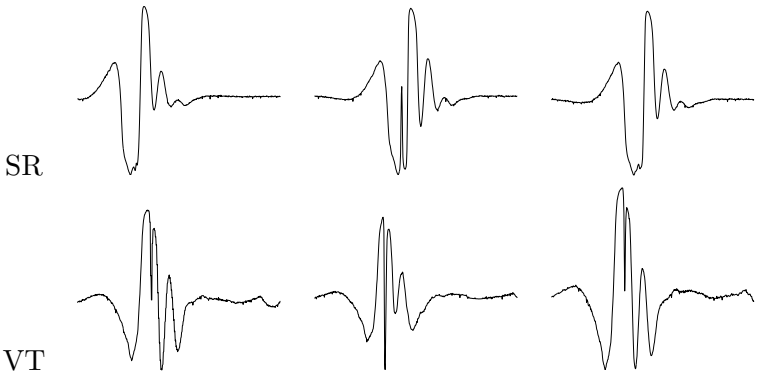


Figure 1.1.: Two-class problem (heartbeats: sinus rhythm (SR) and ventricular tachycardia(VT))

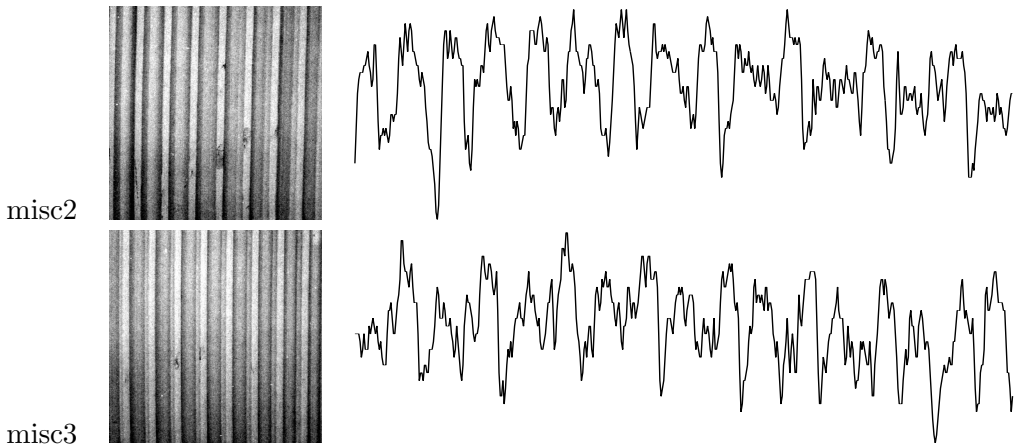


Figure 1.2.: Texture sample: linearly rescaled images and exemplary rows

Contribution

When splitting a problem into subproblems, one should not lose track of the overall goal! The fundamental idea in this thesis is to *jointly* design the wavelet representation and the classification step to solve the pattern recognition problem most effectively.

Wavelets [Mallat, 1999, Strang and Nguyen, 1996] are popular tools for signal processing and Support Vector Machines [Vapnik, 1998] for classification. Both are successfully applied to signal classification, but mostly examined independently. A combined signal classification architecture is proposed by [Strauss and Steidl, 2002]. Using a simple strategy, they show that adapting the wavelet representation to the problem may considerably improve the classification. Beside selecting frequency bands as done by [Coifman and Wickerhauser, 1992, Strauss et al., 2003], adapting the shape of the wavelet is particularly effective. In view of a joint design, we address the problem of optimally preprocessing signals for two-class classification.

Wavelet Adaptation Firstly, we select orthogonal compactly supported wavelets adapted to the problem to represent. Our wavelet adaptation approach improves the classification accuracy compared with commonly used algorithms. We study manifold adaptation criteria. Beside the distance of the class centres used by [Strauss and Steidl, 2002], we examine common measures in pattern recognition such as classification error bounds and scatter measures. Particularly, we prove that the most popular Support Vector Machine error bound, the radius – margin bound, can be evaluated more simply by a standard Support Vector Machine. We show empirically that simple measures well approximate this error bound for our application. So criteria which are computationally still more efficient are sufficient for our filter adaptation and, hence, feature selection. The resemblance of the distances induced by the Gaussian kernel to the original Euclidean distances depending on the kernel parameter are another aspect that we study. Further, we formulate wavelet adaptation as a concise optimisation problem. To solve this problem, we devise an adaptive search algorithm that, once the criterion is fixed, efficiently finds the optimal wavelet filter.

Feature Selection Secondly, the selection of optimal wavelets is a special case of the selection of optimal features for classification. We study the established feature selection approach for a simple linear classifier "feature selection concave" introduced by [Bradley and Mangasarian, 1998]. In order to apply the method to the wavelet adaptation problem, we extend it to the selection of feature sets in the diploma thesis [Jakubik, 2003] supervised during this research. We further develop new feature selection approaches for more effective nonlinear and Support Vector classifiers based on the powerful difference of convex functions optimisation framework presented by [Pham Dinh and Hoai An, 1998]. The linear approaches on the one hand achieve improved generalisation. On the other hand, for the first time we perform embedded fea-

ture selection for nonlinear classifiers. By directly optimising the classifier performance, our methods accomplish the desired feature selection and classification performance simultaneously. We demonstrate their favourable performance for well designed artificial and various real-world problems. For organ segmentation in computed tomography scans examined in [Schmidt, 2004], we are able to improve the segmentation performance for real patient data.

Invariance to Signal Shifts Thirdly, we want to recognise also shifted input signals, that is, in the examples above, displaced heartbeats or texture photos taken at different positions.

The dual–tree complex wavelet transform was proposed by [Kingsbury, 2001] to be robust to signal shifts. As Kingsbury only applies heuristic arguments, we show how the transform achieves invariance. In contrast to [Selesnick, 2001], we prove the shift invariance in a finite setting where the transform was introduced and is applied. We also derive the directional properties of the complex transform in multiple dimensions. By our analysis, we are able to generalise the dual–tree wavelet transform to the frequency domain. This yields a flexible framework with a library of wavelet transforms that provides the desired approximate shift invariance. We examine the performance of the common dual–tree and the dual–tree transform in the frequency domain in multifaceted experiments. We apply the constructed wavelets to signal classification with improved invariance.

Outline

Chapter 2 introduces our two-stage signal classifier architecture in detail. After sketching the setup in the first section, we examine both stages in the following sections. In particular, Sec. 2.2.2 establishes the connection between filter banks and discrete wavelets. The theory of reproducing kernel Hilbert spaces on which nonlinear Support Vector Machines are based is reviewed in Sec. 2.3.2, and Support Vector Machines for multi-class problems are sketched in Sec. 2.3.4. We reconsider the conjoint classifier design in Sec. 2.4.

We generalise the wavelet transform used in the feature extraction step to shift-invariant complex wavelets in Chap. 3. After illustrating the deficiency of the common transform, we analyse the cause for the sensitivity in Sec. 3.2. As a remedy, we construct filter pairs in Sec. 3.3 whose shift invariance is established in Theorem 3. After introducing the extension to multiple dimensions in Sec. 3.4, we elaborate on the transform in the frequency domain in Secs. 3.6 and 3.7 and evaluate both transforms in Secs. 3.5 and 3.8. Their application to signal classification is documented in Sec. 3.9.

The wavelet adaptation process is addressed in Chap. 4. In Sec. 4.3, the criteria proposed in Sec. 4.2 are compared. The solution of the adaptation problem devised in

Sec. 4.4 by means of common optimisation algorithms and our search algorithm Algorithm 4.5.1 is studied in Sec. 4.5.

Chapter 5 is dedicated to more general feature selection methods. After a structured presentation of known methods in Sec. 5.2, we apply "feature selection concave" to wavelet adaptation. We develop our new approaches in Sec. 5.4 and handle them by difference of convex functions programming with Algorithm 5.5.1 in Sec. 5.5. The evaluation is given in Sec. 5.6. Again, extensions to multi-class problems are discussed in Sec. 5.7.

Appendix A gives a useful relation between single class Support Vector Machines and Support Vector problems for novelty detection and clustering that also proves Theorem 8 on the computation of the radius of a set of vectors. This establishes the convenient evaluation of the radius – margin bound.

Appendix B reviews theory of convex functions and convex optimisation which are particularly useful for difference of convex functions programming studied in Chap. 5, and also for the wavelet optimisation in Chap. 4. In particular, we derive the dual difference of convex functions program in Example 5 in Sec. B.4.

2. Conjoint Wavelet–Support Vector Classifiers

2.1. Signal Classification Setup

The task we are dealing with is to assign an unknown signal to one of two classes based on classified training samples, as already described in the introduction. More formally, we assume we are given high-dimensional non-stationary quasi-periodic signals $\mathbf{s} \in \mathbb{R}^l$. The two classes are coded by labels ± 1 so that we assume a training set

$$\{(\mathbf{s}_i, y_i) \in \mathbb{R}^l \times \{-1, 1\} : i = 1, \ldots, n\}$$

of n associations. We intend to map a new signal $\mathbf{s} \in \mathbb{R}^l$ to the label of the most likely class. To achieve this, instead of trying to classify signals directly, we first reduce the signal dimension by extracting relevant numbers — the so-called *features* — from the signals and then classify the signals according to their feature values. Figure 2.1 gives an overview over the global classification setup.

The feature extraction step commonly relies on a multiresolution representation often obtained by filter banks (see [Arivazhagan and Ganesan, 2003, Azencott et al., 1997, Dunn et al., 1994, Ojala et al., 2002, Randen and Husøy, 1999, Reed and du Buf, 1993, Scheunders et al., 1998]). Those filtering approaches are closely related to wavelet decomposition used for feature extraction by [Portilla and Simoncelli, 2000, Li et al., 2003, Unser, 1995, Jones et al., 2001, Arivazhagan and Ganesan, 2003]. To generate low-dimensional feature vectors, we propose to use the norm of the coefficients of the different frequency bands for classification as done in [Unser, 1995, Li et al., 2003].

As to the final classification, we use Support Vector Machines (SVMs) to solve the two-class classification problems. The SVM as a relatively new tool is described in [Vapnik, 1998, Cortes and Vapnik, 1995, Schölkopf, 1997] and is already widely accepted due to its simplicity and high flexibility. It ensures high generalisation performance without the need of a priori knowledge about the problem. The approach is based on *Structural Risk Minimisation* (see [Vapnik, 1995]) and is a generalised linear classifier that tries to maximise the margin between the two classes. The classifier has two variants concerning its invariance to noise. The 'hard margin SVM' claims that all training points are separated by the hyperplane with maximal margin. The noise insensitive variant, the 'soft margin SVM' allows some outliers falling within the margin (see[Cortes and Vapnik, 1995]).

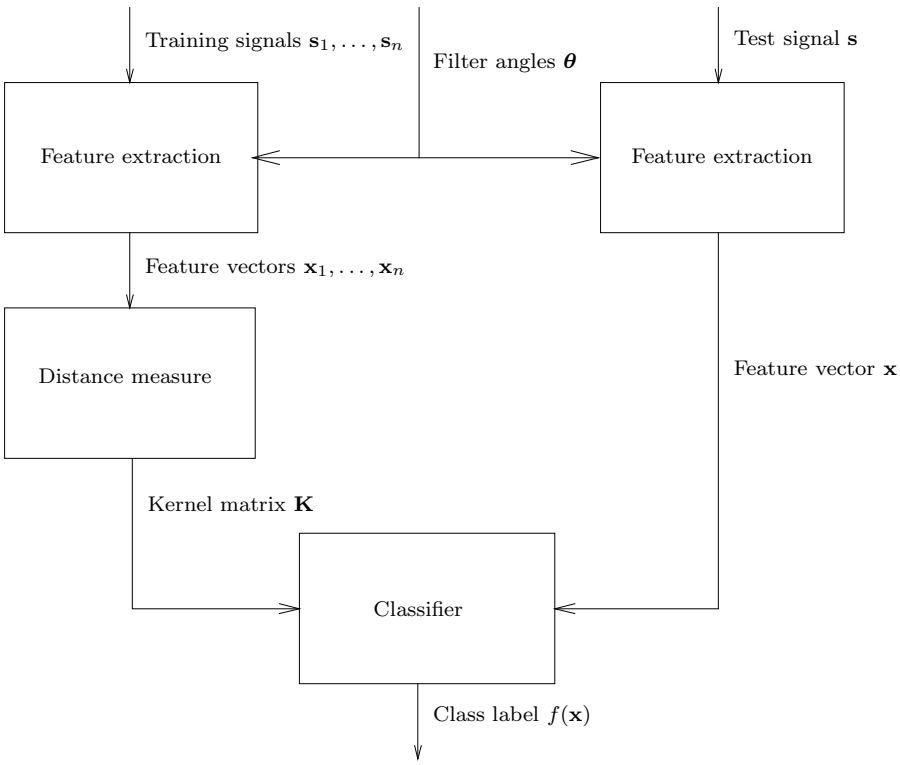


Figure 2.1.: Signal classification setup

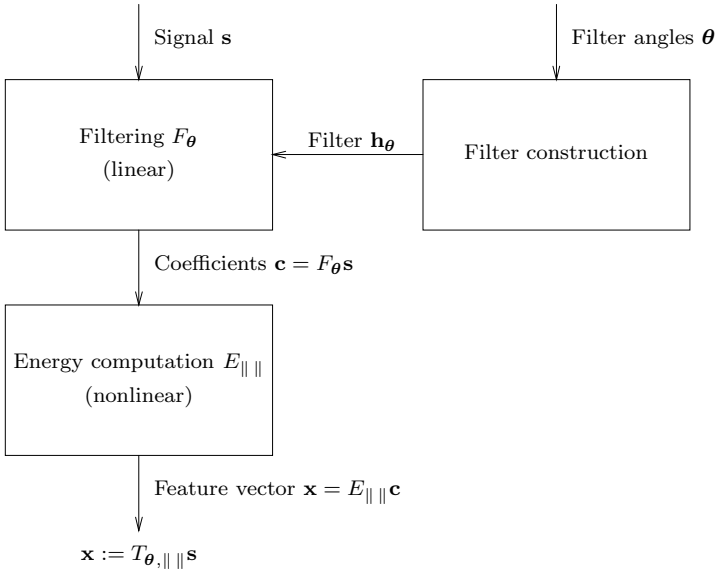


Figure 2.2.: Feature extraction: from the signal to the feature vector

The two classification stages are examined in the following two sections. We consider combining both steps to design a conjoint classifier in Sec. 2.4.

2.2. Feature Extraction by the Discrete Wavelet Transform

This section describes the feature extraction process for our application of signal classification. We summarise the mathematical definition of 'feature vectors' used in this document, and a possible parameterisation. This provides the basis for classification and feature adaptation.

Figure 2.2 illustrates the *feature extraction* process from an input signal $\mathbf{s} \in \mathbb{R}^l$ to its corresponding feature vector $\mathbf{x} \in \mathbb{R}^d$, where $d \ll l$. This feature vector is a possible input for many classification algorithms. The feature extraction process consists of two successive steps, namely filtering and energy computation of the band–pass coefficients. For the filtering we use orthogonal filter banks. As illustrated on the right hand side of the diagram, these filters can be determined by filter angles $\boldsymbol{\theta}$, which are the main parameters of our feature extraction process. Therefore the filter operator is denoted by $F_{\boldsymbol{\theta}}$. Then the features are generated using the norm of the resulting coefficients at each decomposition level. As different norms $\|\,\|$ are used, the corresponding operator is denoted by $E_{\|\,\|}$. In summary, the feature extraction operator is given by $T_{\boldsymbol{\theta},\|\,\|} := E_{\|\,\|}F_{\boldsymbol{\theta}}$. The single steps are more closely looked at in the following.

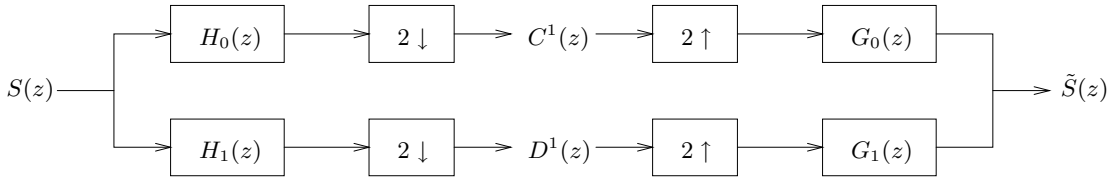


Figure 2.3.: Two–channel filter bank

2.2.1. Filter Design

For filtering, we apply the concept of filter banks. A *filter bank* is a system of filters, linked by operations as up- and downsampling to analyse a signal or synthesise it again. We extract the essential information from the resulting subband signals of an analysis filter bank. Our notion of filter banks is mainly based upon the book [Strang and Nguyen, 1996]. We only use two–channel filter banks illustrated in Fig. 2.3 whose analysis filters normally consist of a low–pass and a high–pass filter. Thereby, $\boxed{2\uparrow}$ and $\boxed{2\downarrow}$ symbolise up- and downsampling by two, respectively. Let $H_0(z) := \sum_{k\in\mathbb{Z}} h_0[k]z^{-k}$ resp. $H_1(z) := \sum_{k\in\mathbb{Z}} h_1[k]z^{-k}$ be the z–transform of these two filters. For signal decomposition, we are interested in the filter coefficient sequences $(h_0[k])_{k\in\mathbb{Z}}$, $(h_1[k])_{k\in\mathbb{Z}} \in \ell_2$.

The *modulation matrix* of a filter bank with analysis filters H_0 and H_1 is defined as

$$\mathbf{H}_{\mathrm{mod}}(z) := \begin{pmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{pmatrix} \ .$$

The filter bank is called *paraunitary* (also referred to as *orthogonal*) if

$$\mathbf{H}_{\mathrm{mod}}{}^{\top}(z^{-1})\mathbf{H}_{\mathrm{mod}}(z) = \mathbf{H}_{\mathrm{mod}}(z)\mathbf{H}_{\mathrm{mod}}{}^{\top}(z^{-1}) = 2\mathbf{I} \ . \tag{2.1}$$

Then the corresponding synthesis filters are given by

$$G_0(z) = H_0(z^{-1}) \ , \quad G_1(z) = H_1(z^{-1}) \tag{2.2}$$

and the orthogonality property ensures that $S(z) = \widetilde{S}(z)$. The *polyphase matrix* of a filter bank is defined as

$$\mathbf{H}_{\mathrm{pol}}(z) := \begin{pmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{pmatrix}$$

with entries from the polyphase decomposition

$$H_i(z) =: H_{i0}(z^2) + z^{-1}H_{i1}(z^2) \ , \quad i = 0, 1 \ .$$

As $\mathbf{H}_{\mathrm{mod}}(z)\mathbf{H}_{\mathrm{mod}}{}^{\top}(z^{-1}) = 2\mathbf{H}_{\mathrm{pol}}(z^2)\mathbf{H}_{\mathrm{pol}}{}^{\top}(z^{-2})$, the filter bank is orthogonal if and only if $\mathbf{H}_{\mathrm{pol}}(z)$ is unitary.

To split the signals into different frequency bands, often high–pass filters with at least one *vanishing moment* are considered. This means that the filter bank satisfies the low–pass condition

$$H_0(1) = \sqrt{2} \tag{2.3}$$

or, equivalently,

$$H_1(1) = 0 \ .$$

For our practical purposes, we are interested in Finite Impulse Response (FIR) filters. The analysis filters of length $2L + 2$ then read

$$H_i(z) = \sum_{k=0}^{2L+1} h_i[k] z^{-k} \ , \quad i = 0, 1 \ .$$

Furthermore, we concentrate on orthogonal filter banks. The reasons are the following:

- Perfect reconstruction prevents information loss by filtering, while retaining a minimal amount of data (for critical subsampling).
- Energy preservation and many other important properties hold.
- Restricting to orthogonal ones leads to fewer different filters. This is easier to handle, especially for the adaptation studied in Chap. 4.

Fundamental for the parameterisation of our feature extraction process is the representation of orthogonal filter banks in a lattice structure composed of rotations and delays. According to the *lattice factorisation* [Vaidyanathan, 1993, Theorem 14.3.1], [Strang and Nguyen, 1996, Theorem 4.7], a two–channel FIR filter bank with filter length $2L + 2$ is orthogonal (paraunitary) if and only if, up to filter translation and the sign of the high–pass filter, the corresponding polyphase matrix can be decomposed as

$$\mathbf{H}_{\mathrm{pol}}(z) = \left[\prod_{l=0}^{L-1} \begin{pmatrix} \cos\theta_l & \sin\theta_l \\ -\sin\theta_l & \cos\theta_l \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z^{-1} \end{pmatrix} \right] \begin{pmatrix} \cos\theta_L & \sin\theta_L \\ -\sin\theta_L & \cos\theta_L \end{pmatrix} \ , \tag{2.4}$$

where $\theta_L \in [0, 2\pi)$ and $\theta_l \in [0, \pi)$ for $l = 0, \ldots, L-1$. If the filter bank's high–pass filter has at least one vanishing moment , i.e., $H_1(1) = 0$, the filter angles add up to $\pi/4$ (see also [Strang and Nguyen, 1996, Theorem 4.6]):

$$\sum_{l=0}^{L} \theta_l = \frac{\pi}{4} \ (\mathrm{mod}\ 2\pi) \ . \tag{2.5}$$

The resulting parameter space

$$\mathcal{P}_L := \{ \boldsymbol{\theta} = (\theta_0, \ldots, \theta_{L-1}) : \theta_l \in [0, \pi) \,,\ l = 0, \ldots, L-1 \}$$

is π–periodic as the angles θ_l can be interpreted as rotation angles: A rotation of $\theta_l + \pi$ implies half a rotation extra because the corresponding matrix just alters sign. As the last angle is computed as $\theta_L = \pi/4 - \sum_{l=0}^{L-1} \theta_l$, it is that amount smaller. Hence, the final filter output is just the same as before.

Beside the lattice factorisation, there also exist other constructive factorisations of paraunitary FIR filter banks' polyphase matrices: The Householder factorisation (see [Vaidyanathan, 1993, p.314], [Strang and Nguyen, 1996, p.312]) completely parameterising this space is also used by [Moulin and Mihçak, 1998] to design signal adapted filter banks. Instead of rotations, it is based on successive reflections also leading to a fast filter bank implementation. Alternative applicable parameterisations are lattices for linear phase filters [Strang and Nguyen, 1996, Theorem 4.8] or those generated by the lifting scheme [Sweldens, 1998, Daubechies and Sweldens, 1998, Jones et al., 2001] although it is not clear how lifting can be used to obtain all filter banks of a given length constructively. Other factorisations also for biorthogonal filter banks have been studied by M. Vetterli.

In most of our experiments we assume a filter length of six corresponding to $L = 2$, i.e., a two-dimensional parameter space. According to our experiments, filters of length six are sufficient to analyse the signals in most instances, and they have the advantage that they are conveniently depicted in 2D for comparison.

We have to remark here that not all orthogonal filter banks covered by the parameter space here are related to continuous wavelets. Usually, the scaling function and wavelet belonging to a filter bank can be obtained by iterating the low–pass filter on some signal, e.g. the δ impulse. That means the scaling function is generated by inverse wavelet transform of an atomic impulse. For example for the simple orthogonal filter $h_0 = (1, 0, 0, 1)^\top$, the resulting limit function $\varphi(x) = (x \in [0, 3])$ is not an orthonormal scaling function (see also [Daubechies, 1992, p. 177]). To cope with this mismatch, the notion of discrete–time wavelets as used by [Vetterli and Kovačević, 1995] is introduced.

2.2.2. Discrete–time Wavelets

To justify the term 'wavelet decomposition' for our feature extraction process, we note that filter banks are connected to wavelets. Every orthogonal continuous wavelet corresponds to a paraunitary filter bank in that the discrete wavelet transform yields the same as filtering with the corresponding filter bank. But not all orthogonal filter banks covered by the parameter space here are related to continuous wavelets. To cope with this mismatch, in the style of the books [Vetterli and Kovačević, 1995, Chap. 3.3.2] and [Vaidyanathan, 1993, Chap. 11.4], we introduce 'discrete–time scaling sequences' and 'discrete–time wavelets'.

Given a possibly infinite signal $\mathbf{s} = (s_i)_{i\in\mathbb{Z}} \in \ell_2$ and a paraunitary filter bank with analysis filter coefficients $(h_0[k])_{k\in\mathbb{Z}}, (h_1[k])_{k\in\mathbb{Z}} \in \ell_2$ and synthesis filter coefficients $(g_0[k])_{k\in\mathbb{Z}} = (h_0[-k])_{k\in\mathbb{Z}}$, $(g_1[k])_{k\in\mathbb{Z}} = (h_1[-k])_{k\in\mathbb{Z}} \in \ell_2$ due to (2.2), we want to analyse

the signal with the filter bank. With $\mathbf{g}_{jk} := (g_j[i-2k])_{i\in\mathbb{Z}} \in \ell_2$ for $j = 0, 1$, $k \in \mathbb{Z}$, the orthogonality conditions for the z–transforms (2.1) and (2.2) imply the orthogonality conditions

$$\begin{aligned} \langle \mathbf{g}_{ji}, \mathbf{g}_{jk} \rangle_{\ell_2} &= \delta(i-k) \ , \quad j = 0, 1,\ i, k \in \mathbb{Z} \ , \\ \langle \mathbf{g}_{0i}, \mathbf{g}_{1k} \rangle_{\ell_2} &= 0 \ , \qquad\qquad\qquad\quad i, k \in \mathbb{Z} \end{aligned}$$

for the filter coefficients, where

$$\delta(x) := \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Due to the perfect reconstruction property of paraunitary filter banks, the set

$$\{\mathbf{g}_{jk} : j = 0, 1,\ k \in \mathbb{Z}\}$$

forms an orthonormal basis of ℓ_2. Hence, we can represent the signal as

$$\mathbf{s} = \sum_{k\in\mathbb{Z}} c_k^1 \mathbf{g}_{0k} + \sum_{k\in\mathbb{Z}} d_k^1 \mathbf{g}_{1k} \tag{2.6}$$

with

$$\begin{aligned} c_k^1 &= \langle \mathbf{s}, \mathbf{g}_{0k} \rangle_{\ell_2} = \langle \mathbf{s}, (h_0[2k-i])_{i\in\mathbb{Z}} \rangle_{\ell_2} = \sum_{i\in\mathbb{Z}} s_i h_0[2k-i] \ , \\ d_k^1 &= \langle \mathbf{s}, \mathbf{g}_{1k} \rangle_{\ell_2} = \langle \mathbf{s}, (h_1[2k-i])_{i\in\mathbb{Z}} \rangle_{\ell_2} = \sum_{i\in\mathbb{Z}} s_i h_1[2k-i] \end{aligned}$$

for $k \in \mathbb{Z}$. Equivalently, in the z–domain, this reads

$$S(z) = C^1(z^2) G_0(z) + D^1(z^2) G_1(z) \tag{2.7}$$

with

$$\begin{aligned} C^1(z^2) &= \frac{1}{2}(H_0(z)S(z) + H_0(-z)S(-z)) \ , \\ D^1(z^2) &= \frac{1}{2}(H_1(z)S(z) + H_1(-z)S(-z)) \ , \end{aligned} \tag{2.8}$$

which is also sketched in Fig. 2.3.

If we want to perform several decomposition steps, we refine the signal representations (2.6) or (2.7) further and obtain

$$\mathbf{c}^1 = (c_k^1)_{k\in\mathbb{Z}} = \sum_{k\in\mathbb{Z}} c_k^2 \mathbf{g}_{0k} + \sum_{k\in\mathbb{Z}} d_k^2 \mathbf{g}_{1k}$$

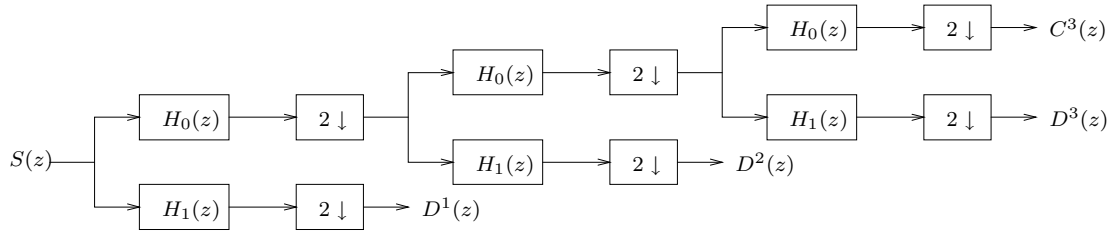


Figure 2.4.: Cascaded two–channel filter bank

with

$$c_k^2 = \sum_{i\in\mathbb{Z}} c_i^1 h_0[2k-i] = \langle \mathbf{c}^1, \mathbf{g}_{0k}\rangle_{\ell_2} = \langle (\langle \mathbf{s}, \mathbf{g}_{0i}\rangle_{\ell_2})_{i\in\mathbb{Z}}, \mathbf{g}_{0k}\rangle_{\ell_2} \ ,$$
$$d_k^2 = \sum_{i\in\mathbb{Z}} c_i^1 h_1[2k-i] = \langle \mathbf{c}^1, \mathbf{g}_{1k}\rangle_{\ell_2} = \langle (\langle \mathbf{s}, \mathbf{g}_{0i}\rangle_{\ell_2})_{i\in\mathbb{Z}}, \mathbf{g}_{1k}\rangle_{\ell_2}$$

for $k \in \mathbb{Z}$ or, equivalently, in the z–domain

$$\begin{aligned} S(z) &= C^1(z^2)G_0(z) + D^1(z^2)G_1(z) \\ &= \left(C^2(z^4)G_0(z^2) + D^2(z^4)G_1(z^2)\right) G_0(z) + D^1(z^2)G_1(z) \\ &= C^2(z^4)G_0(z^2)G_0(z) + D^2(z^4)G_1(z^2)G_0(z) + D^1(z^2)G_1(z) \end{aligned}$$

with

$$C^2(z^4) = \frac{1}{2}(H_0(z^2)C^1(z^2) + H_0(-z^2)C^1(-z^2)) \ , \tag{2.9}$$
$$D^2(z^4) = \frac{1}{2}(H_1(z^2)C^1(z^2) + H_1(-z^2)C^1(-z^2)) \ ,$$

which is sketched in Fig. 2.4.

We are looking for the filter coefficients corresponding to these iterated filters that produce the full decomposition. We therefore define the coefficient sequences $\mathbf{v}_0^j, \mathbf{w}_0^j$ of the z–transforms

$$V^j(z) := \prod_{m=0}^{j-1} G_0(z^{2^m}) \ , \qquad j \in \mathbb{N}_0 \ , \tag{2.10}$$

$$W^j(z) := G_1(z^{2^{j-1}}) \prod_{m=0}^{j-2} G_0(z^{2^m}) \ , \quad j \in \mathbb{N} \tag{2.11}$$

as *discrete–time scaling sequences* and *discrete–time wavelets*, respectively. Let $\mathbf{v}_k^j := v_0^j[\cdot - 2^j k]$ and $\mathbf{w}_k^j := w_0^j[\cdot - 2^j k]$ for $k \in \mathbb{Z}$ denote the translates of the sequences by

multiples of their sample length 2^j. The following orthogonality relations then hold for all $i, k \in \mathbb{Z}$, $j, m \in \mathbb{N}$:

$$\langle \mathbf{v}_i^j, \mathbf{v}_k^j \rangle_{\ell_2} = \delta(k-i) \ , \tag{2.12}$$

$$\langle \mathbf{w}_i^j, \mathbf{w}_k^m \rangle_{\ell_2} = \delta(k-i)\delta(m-j) \ , \tag{2.13}$$

$$\langle \mathbf{v}_i^j, \mathbf{w}_k^j \rangle_{\ell_2} = 0 \ . \tag{2.14}$$

As in the case of continuous wavelets, the sequences $\mathbf{v}_k^j$ and $\mathbf{w}_k^j$ for $j \in \mathbb{N}$, $k \in \mathbb{Z}$ have widths scaled by two and lie in different resolution subspaces j, and the wavelets and scaling sequences on each level j form a basis of the space spanned by the scaling sequences on the above level $j-1$. Hence, in analogy to the continuous case the discrete–time scaling sequences span a *multiresolution analysis* of ℓ_2, with the major difference that they may not be scaled smaller, which would require a negative j. To resume this, we define the sequences of spaces

$$V^j := \overline{\operatorname{span}\{\mathbf{v}_k^j : k \in \mathbb{Z}\}} \subset \ell_2 \ , \quad j \in \mathbb{N}_0 \ ,$$
$$W^j := \overline{\operatorname{span}\{\mathbf{w}_k^j : k \in \mathbb{Z}\}} \subset \ell_2 \ , \quad j \in \mathbb{N} \ .$$

The multiresolution properties

$$V^0 \supset V^1 \supset V^2 \supset \cdots \ ,$$
$$\bigcup_{j \in \mathbb{N}_0} V^j = V^0 = \ell_2 \ ,$$
$$\bigcap_{j \in \mathbb{N}_0} V^j = \{\mathbf{0}\}$$

then hold due to the definition of the scaling sequence filters (2.10). And due to (2.12), the set $\{\mathbf{v}_k^j : k \in \mathbb{Z}\}$ forms an orthonormal basis of V^j. Further, (2.11) and (2.14) imply that the detail spaces W^j form the orthogonal complement to the approximation spaces V^j in the next larger spaces V^{j-1}

$$V^{j-1} = V^j \oplus W^j \ , \quad j \in \mathbb{N} \ .$$

As a consequence, the whole space of sequences can be decomposed as $\ell_2 = \bigoplus_{j \in \mathbb{N}} W^j = V^J \oplus \bigoplus_{j=1}^J W^j$ with orthonormal basis

$$\{\mathbf{v}_k^J, \mathbf{w}_k^j : j = 1, \ldots, J, \, k \in \mathbb{Z}\} \ .$$

The representation of a signal $\mathbf{s} \in \ell_2$ in terms of this basis corresponds to a wavelet decomposition with J steps:

$$\mathbf{s} = \sum_{k\in\mathbb{Z}} c_k^J \mathbf{v}_k^J + \sum_{j=1}^{J} \sum_{k\in\mathbb{Z}} d_k^j \mathbf{w}_k^j$$

with *wavelet coefficients*

$$\begin{aligned} \mathbf{c}^j &:= (c_k^j)_{k\in\mathbb{Z}} = (\langle \mathbf{s}, \mathbf{v}_k^j \rangle_{\ell_2})_{k\in\mathbb{Z}} \ , \quad j \in \mathbb{N} \ , \\ \mathbf{d}^j &:= (d_k^j)_{k\in\mathbb{Z}} = (\langle \mathbf{s}, \mathbf{w}_k^j \rangle_{\ell_2})_{k\in\mathbb{Z}} \ , \quad j \in \mathbb{N} \ . \end{aligned}$$

Beside the orthogonality, another property that is important for our feature extraction holds for the filter banks, and so for the discrete wavelets generated by the lattice structure (2.4) with the constraint (2.5). The average signal value is always preserved in the low–pass channel:

Lemma 1 (average signal value). *Given a paraunitary filter bank that satisfies the low–pass condition* (2.3)*, the low–pass coefficients satisfy*

$$C^j(1) = \left(\frac{1}{\sqrt{2}}\right)^j S(1) \ , \quad j \in \mathbb{N}$$

for all signals S(z).

Proof. Consider the decomposition (2.7) with low–pass coefficients (2.8). The orthogonality condition (2.1) equivalently reads

$$\begin{aligned} H_0(z^{-1})H_0(z) + H_1(z^{-1})H_1(z) &= 2 \ , \\ H_0(z^{-1})H_0(-z) + H_1(z^{-1})H_1(-z) &= 0 \ . \end{aligned}$$

From the low–pass condition $H_0(1) = \sqrt{2}$, it follows by the first equation for $z = 1$ that

$$H_1(1) = 0 \ ,$$

and further, by the second equation,

$$H_0(-1) = 0 \ .$$

With these relations, (2.8) reads

$$C^1(1) = \frac{1}{2}(H_0(1)S(1) + H_0(-1)S(-1)) = \frac{1}{\sqrt{2}}S(1) \ .$$

The property for higher levels j follows by induction by iterating the decomposition on the low–pass coefficients as indicated by (2.9). □

The terms $S(1)$ and $C^j(1)$ for $j \in \mathbb{N}$ are the sums of the coefficients s_k and c_k^j for $k \in \mathbb{Z}$, respectively, so the lemma states that for a finite signal $\mathbf{s} \in \mathbb{R}^l$ the average signal value $S(1)/l$ is directly related to the (finite) sum of the low–pass coefficients $C^j(1)$. Especially for the choice $S(1) = 0$, this implies that the coefficients also have mean zero by $C^j(1) = 0$.

Note that for non-subsampled decomposition, by $C^1(z) = S(z)H_0(z)$, the analogous property

$$C^j(1) = \left(\sqrt{2}\right)^j S(1) \ , \quad j \in \mathbb{N}$$

holds.

Given the finite length analysis filter coefficients $(h_0[k])_{k=0,\dots,2L+1}$, $(h_1[k])_{k=0,\dots,2L+1}$, the decomposition of a signal $\mathbf{s}$ with length $l = n2^J$ for $n \in \mathbb{N}$ in J steps should be done easily. But since we are not able to calculate infinite coefficient sequences, we restrict the wavelets and scaling sequences to the finite-dimensional space $\mathbb{R}^l$. Due to this restriction, the question what to do at the boundary is coming up. We propose to continue the wavelets and scaling sequences l–periodically to preserve the orthogonality of the transform with the finite orthonormal basis

$$\left\{ \widetilde{\mathbf{v}}_k^J = \left(\sum_{z\in\mathbb{Z}} \mathbf{v}_k^J[i+zl] \right)_{i=0,\dots,l-1}, \widetilde{\mathbf{w}}_k^j = \left(\sum_{z\in\mathbb{Z}} \mathbf{w}_k^j[i+zl] \right)_{i=0,\dots,l-1} : j = 1,\dots,J,\ k = 1,\dots,l/2^j \right\} \ .$$

2.2.3. Filtering

In the following, we are mostly interested in input signals $\mathbf{s} \in \mathbb{R}^l$ of length $l = n2^J$ for $n, J \in \mathbb{N}$, $2 \nmid n$ that are normalised with respect to the ℓ_2-norm, i.e., $\|\mathbf{s}\|_{\ell_2} = \text{constant}$.

The filter operator $F_{\boldsymbol{\theta}}$ only needs the successively applied analysis filter bank. It filters by the J-level wavelet filter bank generated by $\boldsymbol{\theta}$:

$$F_{\boldsymbol{\theta}} : \mathbb{R}^l \to \mathbb{R}^l \ , \quad \mathbf{s} \mapsto \begin{pmatrix} \mathbf{c}^J \\ \mathbf{d}^J \\ \vdots \\ \mathbf{d}^1 \end{pmatrix} := \begin{pmatrix} (\langle \mathbf{s}, \widetilde{\mathbf{v}}_k^J \rangle_{\ell_2})_{k=1,\dots,l/2^J} \\ (\langle \mathbf{s}, \widetilde{\mathbf{w}}_k^J \rangle_{\ell_2})_{k=1,\dots,l/2^J} \\ \vdots \\ (\langle \mathbf{s}, \widetilde{\mathbf{w}}_k^1 \rangle_{\ell_2})_{k=1,\dots,l/2} \end{pmatrix} = \mathbf{F}_{\boldsymbol{\theta}}\mathbf{s} \ . \tag{2.15}$$

The matrix $\mathbf{F}_{\boldsymbol{\theta}} \in \mathbb{R}^{l\times l}$ is orthogonal and consequently preserves the ℓ_2-norm of the signals which reads

$$\|F_{\boldsymbol{\theta}}\mathbf{s}\|_{\ell_2} = \|\mathbf{s}\|_{\ell_2} \ . \tag{2.16}$$

In two dimensions, the simplest way to construct wavelets is by tensor products of one-dimensional wavelets in one direction and wavelets or scaling functions in the other direction. In successive steps of a multi-level transform, then usually only the scaling function coefficients are decomposed further (*non-standard transform*). Alternatively, also the low–pass components of the wavelet coefficients are decomposed again (*standard transform*). This results in $3J + 1$ or $J(J + 2) + 1$ channels for J levels, respectively.

2.2.4. Energy Computation

To generate a handy number of features that still make the signals well distinguishable, we introduce the energy operator

$$E_{\|\,\|} : \mathbb{R}^l \to \mathbb{R}^d \,, \quad \begin{pmatrix} \mathbf{c}^d \\ \mathbf{d}^d \\ \vdots \\ \mathbf{d}^1 \end{pmatrix} \mapsto \begin{pmatrix} \|\mathbf{d}^d\| \\ \vdots \\ \|\mathbf{d}^1\| \end{pmatrix} \,. \tag{2.17}$$

Here d is the number of wavelet channels, which equals the number of decomposition levels $d = q$ in one dimension. Note that in our later experiments we always deal with input signals $\mathbf{s}$ having average value zero so that $\mathbf{e}^\top \mathbf{c}^d = 0$ by Lemma 1. As possible norms for $E_{\|\,\|}$ we consider beside the ℓ_2-norm the weighted ℓ_2-norm

$$\mathbf{c} \mapsto \frac{1}{\sqrt{n}} \|\mathbf{c}\|_{\ell_2} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} |c_i|^2} \,, \qquad \mathbf{c} \in \mathbb{R}^n \,,$$

which was proposed by [Unser, 1995] to represent the channel variance. Apart from the two proposed norms, ℓ_p-norms for $p \geq 1$ may also be chosen. Especially the ℓ_1-norm behaves robustly with respect to different signals in [Strauss and Steidl, 2002]. In the two-dimensional case, the following analogous matrix norms are considered: the *Frobenius norm* $\mathbf{C} \mapsto \|\mathbf{C}\|_{\mathrm{F}} = (\mathrm{tr}(\mathbf{C}^\top \mathbf{C}))^{1/2}$ and the weighted Frobenius norm $\mathbf{C} \mapsto (st)^{-1/2} \|\mathbf{C}\|_{\mathrm{F}} = ((st)^{-1} \mathrm{tr}(\mathbf{C}^\top \mathbf{C}))^{1/2}$ for $\mathbf{C} \in \mathbb{R}^{s \times t}$.

The weighted norms seize the average power or channel variance as proposed by [Unser, 1995]. The normalisation balances the contribution of the different channels and, as its expectation is independent of the coefficient vector length, makes the feature values for different size signals comparable. In the translation invariant case, if the applied high–pass filter has at least one vanishing moment, the expectation of the coefficients for each high–pass channels is zero, so that the above norm effectively represents the channel variance. For sub–sampled decomposition, this choice of norm still provides a variance estimate.

Note that the weighted ℓ_2-norm corresponds to the subband energies in the Besov norm corresponding to $B_{2,\gamma}^{-1/2}$ (see [Mallat, 1999, Chap. 9.2.3]) which is a Banach space

for $1 \leq \gamma \leq \infty$. According to [Osher et al., 2003], these spaces are well suited to describe texture. Thus, other Besov norms may be useful for energy extraction as well. The particular spaces $B_{2,2}^{-\alpha}$ with negative order $-\alpha < 0$ also characterise smoothing operators according to [Daubechies et al., 2004].

Altogether, for a signal $\mathbf{s} \in \mathbb{R}^l$, the corresponding *feature vector* $\mathbf{x}$ is determined by

$$\mathbf{x} := T_{\boldsymbol{\theta},\|\,\|}\mathbf{s} = E_{\|\,\|}F_{\boldsymbol{\theta}}\mathbf{s} \ ,$$

which depends on $\boldsymbol{\theta}$ and the chosen norm in $E_{\|\,\|}$ as illustrated in Fig. 2.2. The norm preserving property (2.16) of the orthogonal wavelet transform implies

$$\left\|T_{\boldsymbol{\theta},\|\,\|}\mathbf{s}\right\|_{\ell_2} \leq \|\mathbf{s}\|_{\ell_2} \quad . \tag{2.18}$$

As a consequence, if we deal with ℓ_2-normalised input signals $\mathbf{s}$, then the feature vectors lie within or on a sphere in $\mathbb{R}^d$ centred at the origin. In our experiments we deal with signals having average value zero and apply the full wavelet decomposition, i.e., $l/2^d = 1$. Then $\mathbf{c}^d = 0$ according to Lemma 1. If we further use the ℓ_2-norm in $E_{\|\,\|}$, then we have equality in (2.18).

In the rest of the document, we will drop the subscript ℓ_2 for norms and inner products if that does not cause confusion.

The norm relation just discussed reveals some important structure on the set of feature vectors. To define appropriate feature vectors for classification, it is essential to take into account the classifier in use. The Support Vector Machine, which is described next, intends to maximise the 'margin' between the feature vectors of both classes in some 'feature space'. The classifier's target term, the margin as well as potential classification error bounds may motivate possible feature adaptation criteria.

2.3. Support Vector Machines

2.3.1. Classification problem

In this section we provide the tools concerning Support Vector classification with respect to the applications we have in mind. Our approach is based on the pioneering work of [Vapnik, 1995, Vapnik, 1998] and the book [Cristianini and Shawe-Taylor, 2000], where the reader can find a detailed introduction in terms of statistical learning theory.

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$ containing the feature vectors to be classified. We suppose that there exists an underlying unknown function t, the so-called *target function* which maps $\mathcal{X}$ to the binary set $\{-1, 1\}$. Given a training set

$$\mathcal{Z} := \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\} \tag{2.19}$$

of n associations similar as in Sec. 2.1, we are interested in constructing a real valued function f defined on $\mathcal{X}$ such that $\operatorname{sgn}(f)$ is a 'good approximation' of t. f classifies the training data correctly if $\operatorname{sgn}(f(\mathbf{x}_i)) = t(\mathbf{x}_i) = y_i$ for all $i = 1, \ldots, n$.

SVM classification combines the simplicity of a linear learning machine with the high generalisation ability only found in nonlinear classifiers. To this end, we introduce a so-called *feature map* $\boldsymbol{\phi} : \mathcal{X} \to \ell_2$ nonlinearly mapping the input vectors into some generally higher-dimensional space. We then search for f as a linear function in the mapped feature vectors.

It is possible to state linear learning machines only in terms of inner products between the input vectors. As we would like to have fast computation, one can directly compute the inner products instead of explicitly carrying out the feature map. This is done by means of a kernel function. The kernel function K induces a 'reproducing kernel Hilbert space' $\mathcal{H}_K$ which is defined next. In our applications we use Gaussian kernels

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)} \ , \tag{2.20}$$

which are known to have reasonable performance (see [Schölkopf et al., 1997]) and are of course positive definite as shown, e.g., in [Roussos, 1995, Lemma 1.1]. To choose the parameter σ, a good heuristic is to use the variance of the training data $\sigma = (\sum_{i,j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|^2)/(n(n-1))$ as also turns out later in the feature selection tests in Sec. 5.6.2.

The actual SVM problem and its solution are introduced in Sec. 2.3.3. We then indicate possible extensions to multi-class problems in Sec. 2.3.4 where $y \in \{1, \ldots, c\}$ instead of $y \in \{\pm 1\}$.

2.3.2. Mathematical Background: Reproducing Kernel Hilbert Spaces

We now give the details concerning the feature map and kernel functions.

A *kernel* is a positive definite symmetric function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ in $L_2(\mathcal{X} \times \mathcal{X})$. Following [Schaback, 1995], we call a function $K \in L_2(\mathcal{X} \times \mathcal{X})$ positive definite if for any finite set of elements $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$, the matrix $(K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ is positive definite. The kernel K defines the matrix of inner products $(K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ and so embodies the mapping into feature space. This definition of a kernel does not apply to the linear mapping $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$. One could also generalise the definition to conditionally positive definite functions, and in practice, even indefinite kernels have proven to be useful with SVMs [Haasdonk and Bahlmann, 2004]. But, in this document we are mainly interested in functions K arising from radial basis functions. In other words, we assume that there exists a real valued function k on $\mathbb{R}$ so that

$$K(\mathbf{x}, \mathbf{y}) = k(\|\mathbf{x} - \mathbf{y}\|) \ .$$

For a given kernel K, there exists a *reproducing kernel Hilbert space*

$$\mathcal{H}_K := \overline{\text{span}\,\{K(\widetilde{\mathbf{x}}, \cdot) : \widetilde{\mathbf{x}} \in \mathcal{X}\}}$$

of real valued functions on $\mathcal{X}$ with inner product determined by

$$\langle K(\widetilde{\mathbf{x}}, \cdot), K(\bar{\mathbf{x}}, \cdot) \rangle_{\mathcal{H}_K} := K(\widetilde{\mathbf{x}}, \bar{\mathbf{x}}) \ , \tag{2.21}$$

which has reproducing kernel K, i.e.,

$$\langle f(\cdot), K(\widetilde{\mathbf{x}}, \cdot) \rangle_{\mathcal{H}_K} = f(\widetilde{\mathbf{x}}) \quad \forall\, f \in \mathcal{H}_K \ .$$

The space $\mathcal{H}_K$ is also called *native space* in the mathematical nomenclature. By *Mercer's Theorem*, K can be expanded in a uniformly convergent series on $\mathcal{X} \times \mathcal{X}$

$$K(\mathbf{x}, \mathbf{y}) = \sum_{j \in \mathbb{N}} \lambda_j \varphi_j(\mathbf{x}) \varphi_j(\mathbf{y}) \ , \tag{2.22}$$

where $\lambda_j \geq 0$ are the eigenvalues of the integral operator $T_K : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ with $T_K f(\cdot) := \int_{\mathcal{X}} K(\mathbf{x}, \cdot) f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ and where $\{\varphi_j\}_{j \in \mathbb{N}}$ are the corresponding $L_2(\mathcal{X})$-orthonormalised eigenfunctions.

The feature map then reads

$$\boldsymbol{\phi}(\cdot) := \left(\sqrt{\lambda_j} \varphi_j(\cdot) \right)_{j \in \mathbb{N}} \ .$$

By (2.22), we have for $\mathbf{x} \in \mathcal{X}$ that $\boldsymbol{\phi}(\mathbf{x})$ is an element in ℓ_2 with

$$\|\boldsymbol{\phi}(\mathbf{x})\|^2 = \sum_{j \in \mathbb{N}} \lambda_j \varphi_j^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) = k(0)$$

and that

$$\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y}) \rangle = K(\mathbf{x}, \mathbf{y}) \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X} \ . \tag{2.23}$$

We define the *feature space* $\mathcal{F}_K \subset \ell_2$ by the ℓ_2-closure of all finite linear combinations of elements $\boldsymbol{\phi}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}$

$$\mathcal{F}_K := \overline{\operatorname{span} \{ \boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \}} \ .$$

Then $\mathcal{F}_K$ is a Hilbert space with $\|\cdot\|_{\mathcal{F}_K} = \|\cdot\|_{\ell_2}$. The feature space $\mathcal{F}_K$ and the reproducing kernel Hilbert space $\mathcal{H}_K$ are isometrically isomorphic with isometry $\iota : \mathcal{F}_K \to \mathcal{H}_K$ defined by

$$\iota(\mathbf{w}) := f_{\mathbf{w}}(\cdot) = \langle \mathbf{w}, \boldsymbol{\phi}(\cdot) \rangle_{\mathcal{F}_K} = \sum_{j \in \mathbb{N}} w_j \sqrt{\lambda_j} \varphi_j(\cdot) \ . \tag{2.24}$$

In particular, we have that

$$\|f_{\mathbf{w}}\|_{\mathcal{H}_K} = \|\mathbf{w}\|_{\mathcal{F}_K} \ . \tag{2.25}$$

Note that from another point of view $\mathcal{F}_K$ is the space of sequences of Fourier coefficients of the functions in $\mathcal{H}_K$ with respect to the orthonormal basis $\{\sqrt{\lambda_j} \varphi_j\}_{j \in \mathbb{N}}$ of $\mathcal{H}_K$.

2.3.3. SVM classification

Let us turn to our classification task. For a given training set (2.19) we intend to construct a function $f \in \mathcal{H}_K$ that minimises

$$C \sum_{i=1}^{n} (1 - y_i f(\mathbf{x}_i))_+ + \frac{1}{2} \|f\|^2_{\mathcal{H}_K} \quad , \tag{2.26}$$

where $x_+ := \max(x, 0)$ for some constant $C \in \mathbb{R}_+$ controlling the trade-off between the approximation error and the regularisation term. For the choice $C = \infty$, the resulting classifier is called *hard margin SVM*, otherwise *soft margin SVM*. The 'margin' is the minimal distance of a training point $\mathbf{x}_i$ to the hyperplane separating both classes for a linear classifier. Note that we can also look for functions of the form $f = h + b$ ($h \in \mathcal{H}_K$) with a so-called *bias term* $b \in \mathbb{R}$. We omit the bias term b here, because its explicit consideration is only needed for kernel functions that are only positive semidefinite (see [Girosi, 1998]). With our definition of a kernel, the bias is always included implicitly as the set of eigenfunctions $\{\varphi_j\}_{j \in \mathbb{N}}$ always contains a constant function, implying $1 \in \mathcal{H}_K$. As a consequence, $f = h + b \in \mathcal{H}_K$ for $h \in \mathcal{H}_K$ and $b \in \mathbb{R}$, so the minimisation already takes into account all functions of this form.

The unconstrained optimisation problem (2.26) is equivalent to the constrained optimisation problem

$$\begin{aligned} \min_{f \in \mathcal{H}_K, \xi_i \in \mathbb{R}, i=1,\ldots,n} \; & C \sum_{i=1}^{n} \xi_i + \frac{1}{2} \|f\|^2_{\mathcal{H}_K} \\ \text{subject to} \quad & y_i f(\mathbf{x}_i) \geq 1 - \xi_i \; , \qquad i = 1, \ldots, n \; , \\ & \xi_i \geq 0 \; , \qquad\qquad i = 1, \ldots, n \; . \end{aligned} \tag{2.27}$$

Every function $f \in \mathcal{H}_K$ corresponds uniquely to a sequence $\mathbf{w} \in \mathcal{F}_K$. Thus, by (2.24) and (2.25), the optimisation problem (2.27) can be rewritten as follows:

$$\min_{\mathbf{w} \in \mathcal{F}_K, \xi_i \in \mathbb{R}, i=1,\ldots,n} \; C \sum_{i=1}^{n} \xi_i + \frac{1}{2} \|\mathbf{w}\|^2_{\mathcal{F}_K} \tag{2.28a}$$

$$\begin{aligned} \text{subject to} \quad & y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle_{\mathcal{F}_K} \quad \geq 1 - \xi_i \; , \quad i = 1, \ldots, n \; , \\ & \xi_i \qquad\qquad\qquad\qquad \geq 0 \; , \qquad i = 1, \ldots, n \; . \end{aligned} \tag{2.28b}$$

To use the isometry between the reproducing kernel Hilbert space $\mathcal{H}_K$ and the feature space $\mathcal{F}_K$, usually to transfer the problem from $\mathcal{F}_K$ to $\mathcal{H}_K$ is known in the pattern recognition community as the "kernel trick" [Schölkopf and Smola, 2002]. In general the feature space $\mathcal{F}_K \subset \ell_2$ is infinite-dimensional. For a better illustration of (2.28a) we assume for a moment that $\mathcal{F}_K \subset \mathbb{R}^n$. Then the function $\widetilde{f}_{\mathbf{w}}(\cdot) := \langle \mathbf{w}, \cdot \rangle_{\mathcal{F}_K}$ defines a hyperplane $H_{\mathbf{w}} := \{\mathbf{v} \in \mathcal{F}_K : \widetilde{f}_{\mathbf{w}}(\mathbf{v}) = 0\}$ in $\mathbb{R}^n$ through the origin, and an arbitrary

point $\widetilde{\mathbf{v}} \in \mathcal{F}_K$ has the distance $|\langle \mathbf{w}, \widetilde{\mathbf{v}} \rangle_{\mathcal{F}_K}| / \|\mathbf{w}\|_{\mathcal{F}_K}$ from $H_{\mathbf{w}}$. Note that $\widetilde{f}_{\mathbf{w}}(\boldsymbol{\phi}(\mathbf{x})) = f_{\mathbf{w}}(\mathbf{x})$. Thus, the constraints $y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle_{\mathcal{F}_K} / \|\mathbf{w}\|_{\mathcal{F}_K} \geq (1 - \xi_i) / \|\mathbf{w}\|_{\mathcal{F}_K}$ for $i = 1, \ldots, n$ in (2.28b) require that every $\boldsymbol{\phi}(\mathbf{x}_i)$ must at least have the distance $(1 - \xi_i) / \|\mathbf{w}\|_{\mathcal{F}_K}$ from $H_{\mathbf{w}}$.

If there exists $\mathbf{w} \in \mathcal{F}_K$ so that (2.28b) can be fulfilled with $\xi_i = 0$ for $i = 1, \ldots, n$, then we say that our training set is *linearly separable* in $\mathcal{F}_K$. For Gaussian kernels like all positive kernels, due to the regularity of the kernel matrix, every finite training set is linearly separable in $\mathcal{F}_K$, see, e.g., [Steinwart, 2001]. Then the optimisation problem (2.28) can be further simplified to

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{F}_K} \ & \frac{1}{2} \|\mathbf{w}\|_{\mathcal{F}_K}^2 \\ \text{subject to} \quad & y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle_{\mathcal{F}_K} \geq 1 \ , \quad i = 1, \ldots, n \ . \end{aligned} \tag{2.29}$$

Given $\mathcal{H}_K$ and $\mathcal{Z}$, the optimisation problem above has a unique solution $f_{\mathbf{w}^*}$. In our hyperplane context $H_{\mathbf{w}^*}$ is exactly the hyperplane that has maximal distance ρ from the training data, where

$$\rho := \frac{1}{\|\mathbf{w}^*\|_{\mathcal{F}_K}} = \frac{1}{\|f_{\mathbf{w}^*}\|_{\mathcal{H}_K}} = \max_{\mathbf{w} \in \mathcal{F}_K} \min_{i=1,\ldots,n} \left\{ \frac{|\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle_{\mathcal{F}_K}|}{\|\mathbf{w}\|_{\mathcal{F}_K}} \right\} \ . \tag{2.30}$$

This is visualised in Fig. 2.5. The value ρ is called the *margin* of $f_{\mathbf{w}^*}$ with respect to the training set $\mathcal{Z}$. In this context, the solutions of the optimisation problems (2.28) and (2.29) are called *soft margin* and *hard margin SV classifiers*, respectively.

Next we consider the solution of the SVM problem (2.27), where we follow mainly the notation of [Wahba, 1999]. Here the notion 'support vector' comes into play.

By the *Representer Theorem* (see [Kimeldorf and Wahba, 1971, Wahba, 1999]), the minimiser of (2.27) has the form

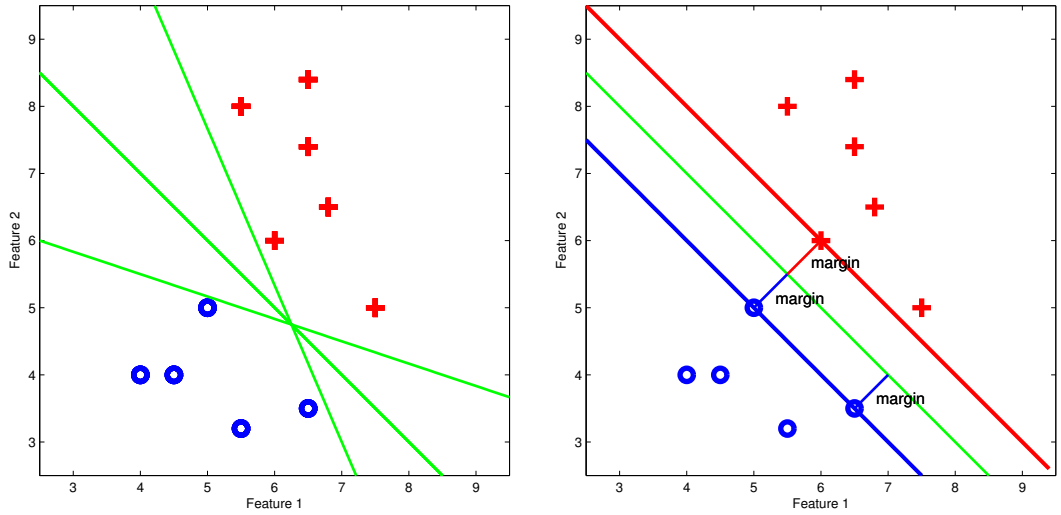
$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i K(\mathbf{x}, \mathbf{x}_i) \ . \tag{2.31}$$

In particular, the sum incorporates only training vectors $\mathbf{x}_i$. We obtain setting $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^\top$ and $\mathbf{K} := (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ that

$$\mathbf{f} = \mathbf{K}\mathbf{c} \ .$$

Note that $\mathbf{K}$ is positive definite. Further, define $\mathbf{Y} := \operatorname{diag}(y_1, \ldots, y_n)$. Then the optimisation problem (2.27) can be rewritten as

$$\begin{aligned} \min_{\mathbf{c} \in \mathbb{R}^n, \, \boldsymbol{\xi} \in \mathbb{R}^n} \ & C \mathbf{e}^\top \boldsymbol{\xi} + \frac{1}{2} \mathbf{c}^\top \mathbf{K} \mathbf{c} \\ \text{subject to} \quad & \mathbf{Y}\mathbf{K}\mathbf{c} \geq \mathbf{e} - \boldsymbol{\xi} \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ . \end{aligned} \tag{2.32}$$



(a) multiple separating lines (b) separating line with maximal margin

Figure 2.5.: Separating lines in $\mathbb{R}^2$

The dual problem with Lagrange multipliers $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$ reads

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n,\, \boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\mathbf{c}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}),$$

where

$$\mathcal{L}(\mathbf{c}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) := C\mathbf{e}^\top \boldsymbol{\xi} + \frac{1}{2}\mathbf{c}^\top \mathbf{K}\mathbf{c} - \boldsymbol{\beta}^\top \boldsymbol{\xi} - \boldsymbol{\alpha}^\top(\mathbf{Y}\mathbf{K}\mathbf{c} - \mathbf{e} + \boldsymbol{\xi})$$

subject to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{c}} = \mathbf{0} \ , \ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = \mathbf{0} \ , \ \boldsymbol{\alpha} \geq \mathbf{0} \ , \ \boldsymbol{\beta} \geq \mathbf{0} \ .$$

Now $\mathbf{0} = \partial\mathcal{L}/\partial\mathbf{c} = \mathbf{K}\mathbf{c} - \mathbf{K}\mathbf{Y}\boldsymbol{\alpha}$ is equivalent to

$$\mathbf{c} = \mathbf{Y}\boldsymbol{\alpha} \ . \tag{2.33}$$

Further we have by $\partial\mathcal{L}/\partial\boldsymbol{\xi} = \mathbf{0}$ that $\boldsymbol{\beta} = C\mathbf{e} - \boldsymbol{\alpha}$. Thus, our optimisation problem becomes

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{Y}\mathbf{K}\mathbf{Y}\boldsymbol{\alpha} + \mathbf{e}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{e} \ . \end{aligned} \tag{2.34}$$

This concave Quadratic Program (QP) is usually solved in the SVM literature. For a moderate number of associations some standard QP routines can be used and for a large

number of associations, e.g., $|\mathcal{Z}| > 4000$, specifically designed large scale algorithms should be applied, e.g., SVM*light* [Joachims, 1999].

The *Support Vectors* (SVs) are those training patterns $\mathbf{x}_i$ for which the coefficients α_i in the solution of (2.34) do not vanish. Let I denote the index set of SVs $I := \{i \in \{1, \ldots, n\} : \alpha_i \neq 0\}$. By the Kuhn–Tucker complementarity conditions summarised in Def. 4 in Appendix B.1 the solution f of the QP (2.32) has to fulfil

$$\alpha_i(y_i f(\mathbf{x}_i) - 1 + \xi_i) = 0 \;, \quad i = 1, \ldots, n \;\; . \tag{2.35}$$

This implies $y_i f(\mathbf{x}_i) \leq 1$ for $i \in I$. Then by (2.31) and (2.33), the function f has the sparse representation

$$f(\mathbf{x}) = \sum_{i \in I} c_i K(\mathbf{x}_i, \mathbf{x}) = \sum_{i \in I} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) \;\; , \tag{2.36}$$

which depends only on the SVs. Now f is a linear function in $\mathcal{F}_K$, but it may take a nonlinear shape in $\mathcal{X}$ as shown in Fig. 2.6 for the same training data as for the linear classifiers in Fig. 2.5. With respect to the margin we obtain by (2.30) and (2.21) that

$$\rho = (\|f\|_{\mathcal{H}_K})^{-1} = (\mathbf{c}^\top \mathbf{K} \mathbf{c})^{-1/2} = \left(\sum_{i \in I} y_i \alpha_i f(\mathbf{x}_i) \right)^{-1/2} \;\; .$$

In case of hard margin classification, $\xi_i = 0$ implies by (2.35) that $y_i f(\mathbf{x}_i) = 1$ for $i \in I$ so that we obtain for the margin the simple expression

$$\rho = \left(\sum_{i \in I} \alpha_i \right)^{-1/2} \;\; . \tag{2.37}$$

Apart from classification, it is also possible to apply the "kernel trick" to other methods that can be formulated in terms of inner products. For *SV regression* with real-valued targets y, the objective is modified as to penalise outliers in both directions. The common characteristic with the SV classification problem above is that only few training data are required to represent the solution. For particular choices of kernels, SV regression is equivalent to interpolating or approximating discrete splines and also to *total variation regularisation* by [Steidl et al., 2005].

Besides, the mapping $\boldsymbol{\phi}$ may also be applied to the features instead of the feature vectors [Shashua and Wolf, 2004].

Common simplified variants of the SVM are the linear programming SVM (LP SVM, see [Vapnik, 1998, Chap. 10.6], [Schölkopf and Smola, 2002, Chap. 7.7]) and the least squares SVM (see [Suykens et al., 2002]). The *LP SVM* relies on the representation (2.36) of the decision function with dual variables $\boldsymbol{\alpha} \geq \mathbf{0}$ and together with the primal variables $\boldsymbol{\xi}$ subject to the constraints of problem (2.27) minimises $\mathbf{e}^\top \boldsymbol{\alpha} + C \mathbf{e}^\top \boldsymbol{\xi}$. This

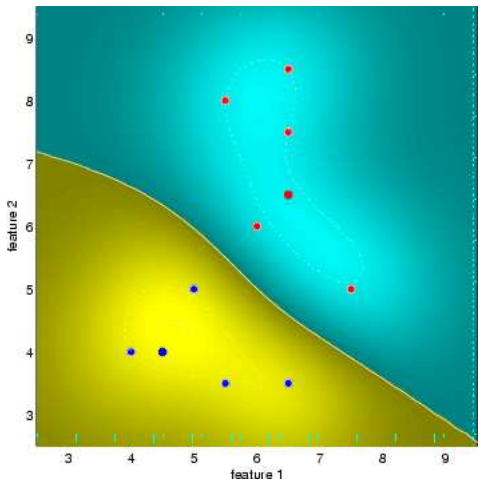


Figure 2.6.: Gaussian SVM decision function in $\mathbb{R}^2$

formulation does not use the "kernel trick" or allow for a primal-dual interpretation as for the common SVM. Instead only a Linear Program (LP) is solved that leads to a sparse solution due to the ℓ_1-norm penalty on $\boldsymbol{\alpha}$ (cf. Sec. 5.2.2). The *least squares SVM*, in contrast, allows for less sparse solutions with more SVs. Starting from problem (2.28), equality is required in (2.28b) and the ℓ_1-norm of $\boldsymbol{\xi}$ is replaced also by its ℓ_2-norm in (2.28a). Then again only a linear system has to be solved.

2.3.4. Multi-class SVMs

So far, we addressed two-class classification. To solve multi-class classification problems, SVMs are also used in practice. This is often done effectively by using a sequence of binary SV classifiers to find the likeliest class.. For the reduction of the multi-class problem to such a sequence there exist several more or less costly and reliable ways [Heiler, 2001, Hsu and Lin, 2002]. An extension to that procedure is to determine all binary classifiers at once as in [Weston and Watkins, 1999, Weston and Watkins, 1998]. This leads to fewer SVs total allowing faster classification with the potential drawback of higher training times. Further approaches include inherently vector valued decision functions, for example.

2.4. Conjoint Classifier Architecture

Now that we have defined the essential classification steps, we can reconsider the overall classifier architecture. Designing the classification process illustrated in Fig. 2.1, we jointly consider both stages to optimally coordinate both steps:

Firstly, of course the classifier should be designed regarding the training data. As mentioned in Secs. 2.3.1 and 2.3.3, the Gaussian kernel SVM is suited for any classification problem, only the kernel parameter σ should be adapted to the feature vectors.

Secondly, additionally, when extracting features, we already take the SV classifier into account. We do this by adapting the features to the concrete classifier and also, by maximising the classification accuracy, to the given training data. This is studied in several manners: As we want to ensure correct classification of translated signals, we examine a means to obtain translation invariant features in Chap. 3. A major parameter of the feature extraction is the wavelet or the filter angles as depicted in Fig. 2.1. We maximise class separability by selecting the best wavelets in Chap. 4. In a more general setting, we jointly select arbitrary features and design the SV classifier in Chap. 5.

3. Shift Invariant Multiscale Feature Extraction

3.1. Sensitivity of the Common Wavelet Transform

As described in the previous chapter, we intend to conjointly design a two-stage signal classifier. A requirement on the classifier is that it should be invariant to shifts of the input signals. As the signals are represented by feature vectors, this establishes the need for shift invariant features. Our multiscale feature extraction relies on the wavelet transform. A major problem of the common decimated discrete wavelet transform is its lack of *shift invariance* . More precisely, this means that on shifts of the input signal $\mathbf{s}$, the wavelet coefficients $\mathbf{d}^j$ from (2.15) vary substantially. The signal information may even not be stationary in the subbands so that the energy distribution across the subbands may change [Simoncelli et al., 1992, Kingsbury, 2001]. The cause for the shift dependence is the critical subsampling by the $2\downarrow$ operations that is necessary to obtain an orthogonal transform that avoids redundancy. For our classification process, this property of the transform implies that the features change when the input signal is shifted only which is of course prohibitive.

The shift dependence of the fully decimated discrete wavelet transform is demonstrated in Fig. 3.1 as also done by [Simoncelli et al., 1992]. For presentation purposes, we choose a dilated Daubechies wavelet with three vanishing moments as signal in Fig. 3.1 (a). Making a wavelet transform with itself, the result is clearly a single non-zero coefficient resulting in a single subband with positive energy in Fig. 3.1 (c). For later comparison purposes, we only plot the coefficients' absolute value. Now on a signal shift of one sample to the right (Fig. 3.1 (e)), the other subbands in (f) and (h) also contain a significant portion of the signal energy. This shows that the orthogonal discrete wavelet transform is highly sensible to the signal alignment relative to the subsampling points.

To overcome the problem of shift dependence, one possible approach is to simply omit the responsible subsampling. In m dimensions this introduces a redundancy of at least $1 + d(2^m - 1) : 1$ for d decomposition levels (in the non-standard transform. As a result, the coefficients are completely shift invariant in that they undergo the same shift as the input signal, but under a high cost that is often not desirable in signal processing — as for our classification problem. Techniques that omit or partially omit subsampling are known as cycle spinning [Coifman and Donoho, 1995], oversampled filter banks [Cvetković and Vetterli, 1998] or non-decimated wavelet transforms [Mallat, 1999].

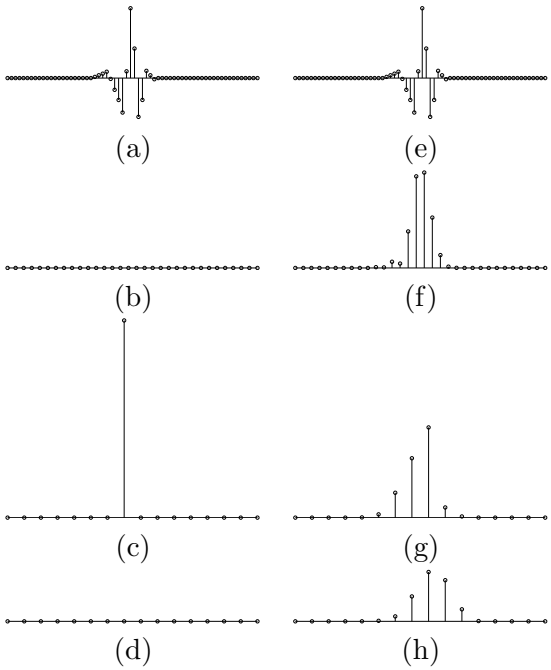


Figure 3.1.: Shift sensitivity of the common discrete wavelet transform: (a) original signal, (b)–(d) magnitude of wavelet subband coefficients, (e) signal (a) shifted by one sample, (f)–(h) magnitude of new wavelet subband coefficients

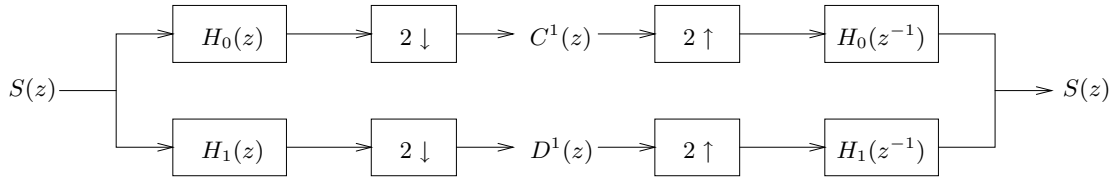


Figure 3.2.: Orthogonal filter bank

Kingsbury [Kingsbury, 2001, Kingsbury and Magarey, 1997, Kingsbury, 1998] proposes an alternative wavelet transform that achieves approximate shift invariance with a redundancy of only $2^m : 1$ and shows applications to motion estimation and denoising [Kingsbury and Magarey, 1997], texture synthesis [Kingsbury, 1998] and retrieval [de Rivaz and Kingsbury, 1999]. The transform yields complex wavelet coefficients via a 'dual–tree' of parallel real filter banks. By a phase shift of the real to the imaginary part, the coefficients incorporate a smoother filter response magnitude and shift invariance. Another advantage of this complex wavelet transform is its inherent directional selectivity in multiple dimensions that comes out without explicitly rotating a filter as in a Gabor filter bank, for example. However, a special filter design is necessary working with two real filter banks in the time domain, which may not be the best with respect to the intended application.

In Sec. 3.2 we review Kingsbury's approach to achieve translation invariant combined filter banks in a sophisticated way. We prove in Sec 3.3 that the construction proposed by Kingsbury indeed leads to wavelets with vanishing negative frequency parts and point out the generalisation to multiple dimensions in Sec. 3.4. Then we review the performance of Kingsbury's transform in Sec. 3.5. The transform is generalised in the frequency domain in Sec. 3.6. After explaining how the wavelet transform works in the frequency domain in Sec. 3.7, numerical examples illustrating the behaviour of the dual–tree complex wavelet transform in the frequency domain with respect to shift and rotational invariance are given in Sec. 3.8 for some standard wavelets. Finally, Sec. 3.9 shows how the proposed transforms are suited for our signal classification. The results in this chapter are based on work published in [Neumann and Steidl, 2003] and summarised in [Neumann and Steidl, 2005].

3.2. Translation Invariance by Parallel Filter Banks

We are interested in orthogonal two–channel filter banks with analysis low–pass filter given by the z–transform $H_0(z) = \sum_{k\in\mathbb{Z}} h_0[k]z^{-k}$, analysis high–pass filter $H_1(z) = \sum_{k\in\mathbb{Z}} h_1[k]z^{-k}$ and with synthesis filters $H_0(z^{-1})$ and $H_1(z^{-1})$. We restrict our attention to real filters, i.e. all coefficients $h_i[k] \in \mathbb{R}$ for $i = 0, 1$, $k \in \mathbb{Z}$. A corresponding filter bank is depicted in Fig. 3.2.

For an input signal $S(z)$, similarly as in Sec. 2.2.2, the analysis part of the filter bank inclusive subsequent upsampling produces the low–pass and the high–pass coefficients

$$C^1(z^2) = \frac{1}{2}[S(z)H_0(z) + S(-z)H_0(-z)] \ , \tag{3.1a}$$

$$D^1(z^2) = \frac{1}{2}[S(z)H_1(z) + S(-z)H_1(-z)] \ , \tag{3.1b}$$

respectively. The filter bank decomposes the input signal S into a low frequency part S_l^1 and a high frequency part S_h^1, more precisely

$$S(z) = S_l^1(z) + S_h^1(z) \ ,$$

where

$$S_l^1(z) = C^1(z^2)H_0(z^{-1}) = \frac{1}{2}[S(z)H_0(z)H_0(z^{-1}) + S(-z)H_0(-z)H_0(z^{-1})] \ , \tag{3.2a}$$

$$S_h^1(z) = D^1(z^2)H_1(z^{-1}) = \frac{1}{2}[S(z)H_1(z)H_1(z^{-1}) + S(-z)H_1(-z)H_1(z^{-1})] \ . \tag{3.2b}$$

Unfortunately, this decomposition is not shift invariant due to the second summands in (3.1) and (3.2), which were introduced by the down- and upsampling operators. More precisely, if the input signal is shifted, say $z^{-1}S(z)$, the application of the filter bank results in the splitting

$$z^{-1}S(z) = \widetilde{S}_l^1(z) + \widetilde{S}_h^1(z) \ ,$$

where

$$\widetilde{S}_l^1(z) = \frac{1}{2}z^{-1}[S(z)H_0(z)H_0(z^{-1}) - S(-z)H_0(-z)H_0(z^{-1})] \neq z^{-1}S_l^1(z)$$

and similarly for the high–pass part. From this calculation one can see that the shift dependence is caused by the terms not containing $S(z)$, the so-called *aliasing terms*. Note that the filter bank is of course shift invariant with respect to a double shift since by $(-1)^2 = 1$ we have that $z^{-2}S(z) = z^{-2}(S_l^1(z) + S_h^1(z))$.

One possibility to obtain a shift invariant decomposition consists in applying an additional filter bank with shifted analysis filters $z^{-1}H_0(z)$ and $z^{-1}H_1(z)$ and averaging the low–pass and the high–pass channels of both filter banks. Signify the first filter bank by index a and the second one by index b. Then this procedure implies the decomposition

$$S(z) = S_l^1(z) + S_h^1(z) \ ,$$

where

$$\begin{aligned} S_l^1(z) &= \frac{1}{2}\left(C_a^1(z^2)H_{0a}(z^{-1}) + C_b^1(z^2)H_{0b}(z^{-1})\right) \\ &= \frac{1}{4}\left[S(z)\left(H_0(z)H_0(z^{-1}) + H_0(z)H_0(z^{-1})\right)\right. \\ &\quad \left. +S(-z)\left(H_0(-z)H_0(z^{-1}) - H_0(-z)H_0(z^{-1})\right)\right] \\ &= \frac{1}{2}S(z)H_0(z)H_0(z^{-1}) \end{aligned}$$

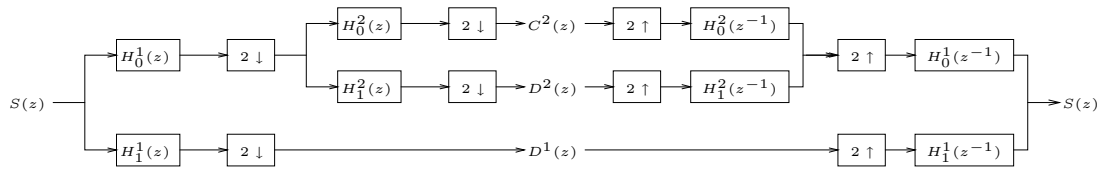


Figure 3.3.: Cascaded orthogonal filter bank

and similarly for the high–pass part. The aliasing term in $S_l^1(z)$ containing $S(-z)$ has vanished and the decomposition becomes indeed shift invariant.

Iteration of the two–channel filter bank as depicted in Fig. 3.3 for $J = 2$ levels leads to octave-band filter banks with a band–pass decomposition of the input signal. Note that the jth filter bank may use its own filters $H_\cdot^j$. We are interested in cascaded filter banks with $J \geq 2$ levels. Let $A^1(z) := H_0^1(z)$, $B^1(z) := H_1^1(z)$ and

$$A^j(z) := H_0^1(z) \cdots H_0^{j-1}(z^{2^{j-2}}) H_0^j(z^{2^{j-1}}) \ , \tag{3.3a}$$

$$B^j(z) := H_0^1(z) \cdots H_0^{j-1}(z^{2^{j-2}}) H_1^j(z^{2^{j-1}}) \tag{3.3b}$$

for $j = 2, \ldots, J$. Then the cascaded filter bank produces the low–pass coefficients

$$C^J(z^{2^J}) = \frac{1}{2^J} \sum_{k=0}^{2^J-1} S(w_{2^J}^k z) A^J(w_{2^J}^k z) \ ,$$

where $w_m := e^{-2\pi i/m}$ and for $j = 1, \ldots, J$ the band–pass coefficients

$$D^j(z^{2^j}) = \frac{1}{2^j} \sum_{k=0}^{2^j-1} S(w_{2^j}^k z) B^j(w_{2^j}^k z) = \frac{1}{2^j} \sum_{k=-2^{j-1}}^{2^{j-1}-1} S(w_{2^j}^k z) B^j(w_{2^j}^k z) \ .$$

The input signal decomposes as

$$\begin{aligned} S(z) &= S_l^J(z) + \sum_{j=1}^{J} S_h^j(z) \\ &= C^J(z^{2^J}) A^J(z^{-1}) + \sum_{j=1}^{J} D^j(z^{2^j}) B^j(z^{-1}) \\ &= S_l^J(z) + \sum_{j=1}^{J} \frac{1}{2^j} \sum_{k=-2^{j-1}}^{2^{j-1}-1} S(w_{2^j}^k z) B^j(w_{2^j}^k z) B^j(z^{-1}) \ . \end{aligned} \tag{3.4}$$

Of course this decomposition is not shift invariant because the decomposition of $z^{-r} S(z)$ with $2^j \nmid r$ does not result in a jth band–pass part $z^{-r} S_h^j(z)$ since $(w_{2^j}^k)^{-r} \neq 1$ for several k.

The ideal low–pass filter has the property $\operatorname{supp} H_0(e^{2\pi i\omega}) = [-1/4, 1/4]$ for $\omega \in [-1/2, 1/2]$. Here and in the following we write $\operatorname{supp} f$ instead of $\operatorname{supp} f \cap [-1/2, 1/2]$ for a one–periodic function f. Let us assume that $H_0^j(z)$ fulfils

$$\operatorname{supp} H_0^j(e^{2\pi i\omega}) \subseteq \left[-\frac{1}{3}, \frac{1}{3}\right] \quad \forall\, j = 1, \ldots, J \tag{3.5}$$

for $\omega \in [-1/2, 1/2]$. Then the corresponding orthogonal high–pass filters H_1^j satisfy

$$\operatorname{supp} H_1^j(e^{2\pi i\omega}) \subseteq \left[-\frac{1}{2}, -\frac{1}{6}\right] \cup \left[\frac{1}{6}, \frac{1}{2}\right] \quad \forall\, j = 1, \ldots, J \ . \tag{3.6}$$

In the following sections we restrict ourselves to the conjugate quadrature filter (CQF) setting $H_1(z) = \pm z^p H_0(-z^{-1})$ for $p \in \mathbb{Z}$ for the filter banks' low–pass and high–pass filters as it is the only setting that is valid in the orthogonal case. From (3.5) and (3.6) it follows for $j \geq 2$ that

$$\begin{aligned} \operatorname{supp} A^j(e^{\pm 2\pi i\omega}) &\subseteq \left[-\frac{1}{3 \cdot 2^{j-1}}, \frac{1}{3 \cdot 2^{j-1}}\right] \ , \\ \operatorname{supp} B^j(e^{\pm 2\pi i\omega}) &\subseteq \left[-\frac{4}{3 \cdot 2^j}, -\frac{1}{3 \cdot 2^j}\right] \cup \left[\frac{1}{3 \cdot 2^j}, \frac{4}{3 \cdot 2^j}\right] \ . \end{aligned} \tag{3.7}$$

Figure 3.4 illustrates the support properties for $J = 3$.

Let $x \bmod 1 \in [-1/2, 1/2)$ denote the symmetric residue of $x \in \mathbb{R}$ modulo one and $[\cdot, \cdot] \bmod 1$ the 'interval' with elements taken modulo one. Now $B^j(w_{2^j}^k e^{2\pi i\omega}) = B^j(e^{2\pi i(\omega - k/2^j)})$ is a (one–periodic) shift of $B^j(e^{2\pi i\omega})$ by $k/2^j$. Thus,

$$\operatorname{supp} B^j(w_{2^j}^k e^{2\pi i\omega}) \subseteq \left(\left[\frac{-4+3k}{3 \cdot 2^j}, \frac{-1+3k}{3 \cdot 2^j}\right] \cup \left[\frac{1+3k}{3 \cdot 2^j}, \frac{4+3k}{3 \cdot 2^j}\right]\right) \bmod 1$$

and consequently

$$\operatorname{supp} B^j(w_{2^j}^k e^{2\pi i\omega}) B^j(e^{-2\pi i\omega}) \subseteq \begin{cases} \left[-\frac{4}{3 \cdot 2^j}, -\frac{1}{3 \cdot 2^j}\right] \cup \left[\frac{1}{3 \cdot 2^j}, \frac{4}{3 \cdot 2^j}\right] & k = 0 \ , \\ \pm \left[\frac{1}{3 \cdot 2^j}, \frac{2}{3 \cdot 2^j}\right] & k = \pm 1 \ , \\ \pm \left[\frac{2}{3 \cdot 2^j}, \frac{4}{3 \cdot 2^j}\right] & k = \pm 2 \ , \\ \emptyset & 3 \leq |k| \leq 2^{j-1} \end{cases}$$

for $j \geq 2$. See Fig. 3.5 for an illustration. Hence decomposition (3.4) can be rewritten as

$$S(z) = S_l^J(z) + S_h^1(z) + \sum_{j=2}^{J} \frac{1}{2^j} \sum_{k=-2}^{2-\delta(2-j)} S(w_{2^j}^k z) B^j(w_{2^j}^k z) B^j(z^{-1}) \ . \tag{3.8}$$

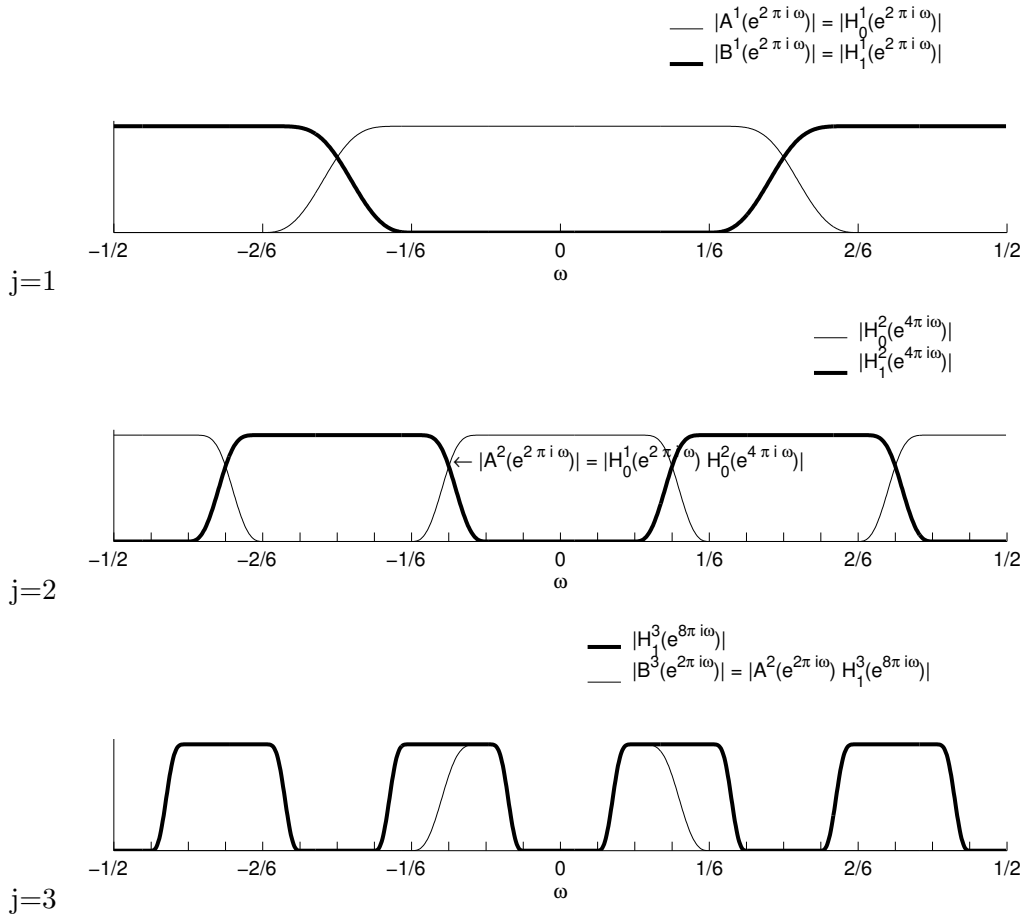


Figure 3.4.: Desired filter support at different levels j

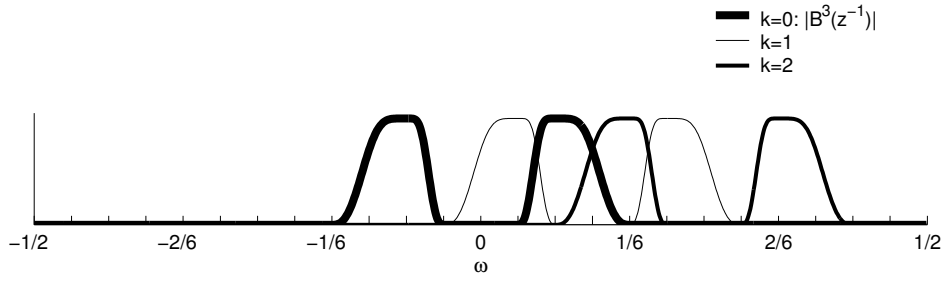


Figure 3.5.: Desired support of shifted high–pass filter $B^j(w_{2^j}^k e^{2\pi i\omega})$ at level $j = 3$

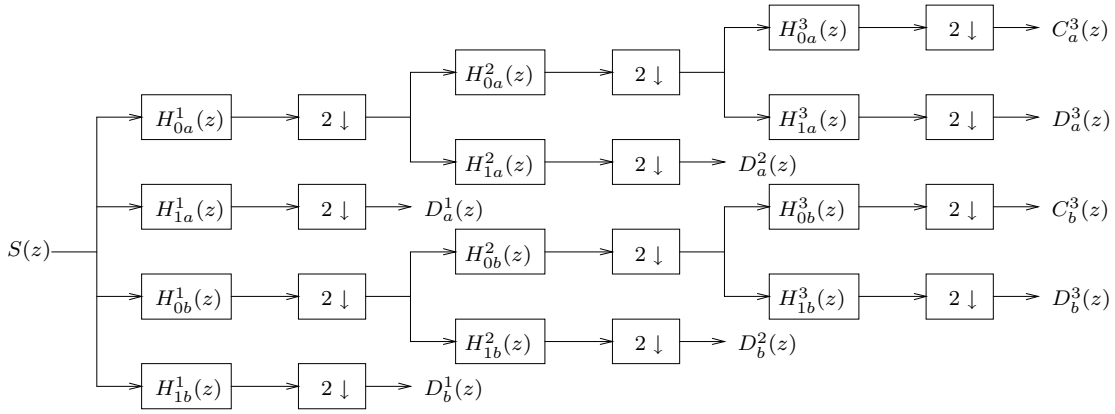


Figure 3.6.: Dual–tree filter bank

To remedy the drawback of the decomposition not being shift invariant and to be able to obtain perfect reconstruction and even apply orthogonal filter banks, Kingsbury suggests in [Kingsbury, 1998, Kingsbury, 2001] to apply a 'dual–tree' of two parallel filter banks with special properties and combine their band–pass outputs as in the non-cascaded case earlier in this section. The structure of a resulting analysis filter bank for a one-dimensional signal $\mathbf{s}$ is sketched in Fig. 3.6, where we use again the index a for the original cascaded filter bank and the index b for the additional one. Then the input signal is split as

$$\begin{aligned} S(z) = \frac{1}{2}\Big(& S^J_{la}(z) + S^J_{lb}(z) \\ & + \sum_{j=1}^{J} \frac{1}{2^j} \sum_{k=0}^{2^j-1} S(w_{2^j}^k z)\big(B^j_a(w_{2^j}^k z)B^j_a(z^{-1}) + B^j_b(w_{2^j}^k z)B^j_b(z^{-1})\big)\Big) \ , \end{aligned} \tag{3.9}$$

where the inner sum can be restricted as in (3.8) if both H_{0a} and H_{0b} satisfy property (3.5).

For $P^j(z) = \sum_{k\in\mathbb{Z}} p^j[k]z^{-k}$ with $p^j[k] \in \mathbb{C}$, let $P^j(z)^* := \sum_{k\in\mathbb{Z}} \overline{p^j[k]}z^k$. Note that then $\operatorname{supp} P^j(e^{2\pi i\omega}) = I$ implies $\operatorname{supp} P^j(e^{2\pi i\omega})^* = -I$. Let us assume that B^j_a and B^j_b further have the property that

$$B^j_a(z) + iB^j_b(z) = P^j(z) \ , \tag{3.10a}$$

$$B^j_a(z) - iB^j_b(z) = P^j(z)^* \ , \tag{3.10b}$$

where P^j is only supported on the positive frequencies $\omega \in [0, 1/2]$. More precisely, with respect to (3.7) we have

$$\operatorname{supp} P^j(e^{2\pi i\omega}) \subseteq \left[\frac{1}{3\cdot 2^j}, \frac{4}{3\cdot 2^j}\right] \ .$$

Obviously, (3.10) implies

$$\begin{aligned} B_a^j(z) &= \frac{1}{2}(P^j(z) + P^j(z)^*) \ , \\ B_b^j(z) &= -\frac{1}{2}i(P^j(z) - P^j(z)^*) \ . \end{aligned}$$

Using these relations, we obtain in (3.9) that

$$B_a^j(w_{2^j}^k z)B_a^j(z^{-1}) + B_b^j(w_{2^j}^k z)B_b^j(z^{-1}) = \frac{1}{2}(P^j(w_{2^j}^k z)P^j(z^{-1})^* + P^j(w_{2^j}^k z)^* P^j(z^{-1})) \ .$$

Since $\operatorname{supp} P^j(w_{2^j}^k e^{2\pi i\omega}) \subseteq \left[\frac{1+3k}{3\cdot 2^j}, \frac{4+3k}{3\cdot 2^j}\right] \bmod 1$ and $\operatorname{supp} P^j(e^{-2\pi i\omega})^* = \operatorname{supp} P^j(e^{2\pi i\omega}) \subseteq \left[\frac{1}{3\cdot 2^j}, \frac{4}{3\cdot 2^j}\right]$, this expression vanishes for $1 \leq |k| \leq 2^j - 1$. If we further choose the filters $H_{\cdot a}^1$ and $H_{\cdot b}^1$ as in the non-cascaded case to cancel the aliasing at the first level, (3.9) can be rewritten as

$$S(z) = \frac{1}{2}\Big(S_{la}^J(z) + S_{lb}^J(z) + \sum_{j=1}^{J} \frac{1}{2^j} S(z)\big(B_a^j(z)B_a^j(z^{-1}) + B_b^j(z)B_b^j(z^{-1})\big)\Big) \ .$$

Evidently, this band–pass decomposition is translation invariant. The complex filter P^j from (3.10a) implies that the wavelet coefficients are combined in the same manner $D^j(z) = D_a^j(z) + iD_b^j(z)$.

Similar ideas concerning the additional filter bank can be used for the alias cancellation of the low–pass filter. In this case, because of property (3.5) only the translated product filters for $k = \pm 1$ may cause aliasing. But as it is not easily attainable that the low–pass product filters A^j have a property similar to (3.10), one cancels only the odd translates by letting the b product filters be the a filters shifted by half a sample.

Unfortunately, there do not exist real orthogonal FIR filters $H_{\cdot a}^j$ and $H_{\cdot b}^j$ such that B_a^j and B_b^j fulfil property (3.10). One can only construct FIR filters so that (3.10) is satisfied approximately. Special biorthogonal and orthogonal filters of this kind were constructed by [Kingsbury, 1999, Kingsbury, 2001] and [Selesnick, 2001, Fernandes et al., 2003].

3.3. Construction of Filter Pairs

Concerning the filter design for the dual–tree transform introduced in the previous section, in order to achieve shift invariance, Kingsbury [Kingsbury, 1998, Kingsbury, 2001] claims that every filter pair H_{0a}^j and H_{0b}^j at levels $j = 2, \ldots, J$ should have a delay difference of half a sample: Suppose that we are given an orthogonal filter pair $H_0(z)$ and $H_1(z) = \pm z^p H_0(-z^{-1})$ for $p \in \mathbb{Z}$. We will see that cascaded filter banks a and b such that (3.10) is fulfilled can be constructed in the following way:

On the first level $j = 1$, as proposed in the beginning of the previous section, we attain the required delay difference by

$$H^1_{0a}(z) := H_0(z) \ , \qquad H^1_{1a}(z) := H_1(z) = \pm z^p H_0(-z^{-1}) \ , \tag{3.11a}$$

$$H^1_{0b}(z) := z^{-1} H^1_{0a}(z) = z^{-1} H_0(z) \ , \qquad H^1_{1b}(z) := z^{-1} H^1_{1a}(z) = \pm z^{p-1} H_0(-z^{-1}) \ . \tag{3.11b}$$

By the previous section, this guarantees that the combined band–pass component $S^1_h(z)$ is completely shift invariant.
At all higher levels $j = 2, \ldots, J$ we use the filters

$$H^j_{0a}(z) := H_0(z) \ , \qquad H^j_{1a}(z) := H_1(z) = \pm z^p H_0(-z^{-1}) \ . \tag{3.12}$$

The filters in bank b should differ from these filters by a shift of half a sample. Allowing only orthogonal filters implies

$$H^j_{0b}(e^{2\pi i\omega}) = e^{-\pi i\omega} H_0(e^{2\pi i\omega}) \ , \quad \omega \in \left[-\frac{1}{2}, \frac{1}{2}\right) \ . \tag{3.13}$$

This equals $z^{-1/2} H_0(z)$ with $z = e^{2\pi i\omega}$, but only for $\omega \in [-1/2, 1/2)$ since the right hand side is not one–periodic in ω. Therefore we actually use its one–periodic extension

$$H^j_{0b}(e^{2\pi i\omega}) := e^{-\pi i(\omega \bmod 1)} H_0(e^{2\pi i\omega}) \ , \quad \omega \in \mathbb{R} \ . \tag{3.14}$$

In other words, the filter coefficients of H^j_{0b} are the Fourier coefficients of the one–periodic function on the right hand side of (3.14). This function is not in $\mathcal{C}^\infty$, but in $\mathcal{C}^{m-1}$ if $H_0(z) = (1+z)^m F(z)$.

As for H_1, we define the high–pass filter by

$$\begin{aligned} H^j_{1b}(e^{2\pi i\omega}) &= \pm e^{2p\pi i\omega} H^j_{0b}\left(e^{-2\pi i\left(\omega+\frac{1}{2}\right)}\right) \\ &\overset{(3.14)}{=} \pm e^{2p\pi i\omega} e^{\pi i\left[\left(\omega+\frac{1}{2}\right) \bmod 1\right]} H_0\left(e^{-2\pi i\left(\omega+\frac{1}{2}\right)}\right) \\ &= e^{\pi i\left[\left(\omega+\frac{1}{2}\right) \bmod 1\right]} H_1(e^{2\pi i\omega}) \ , \quad \omega \in \mathbb{R} \ . \end{aligned} \tag{3.15}$$

For $\omega \in [-1/2, 1/2)$ this leads in particular to

$$H^j_{1b}(e^{2\pi i\omega}) = \begin{cases} ie^{\pi i\omega} H_1(e^{2\pi i\omega}) & \omega \in \left[-\frac{1}{2}, 0\right) \ , \\ -ie^{\pi i\omega} H_1(e^{2\pi i\omega}) & \omega \in \left[0, \frac{1}{2}\right) \ . \end{cases}$$

Note that H_{0b} and H_{1b} are supported as H_0 and H_1, respectively. It is also possible to change the orientation for all the delays of the b filters, but then the combined filter $B^j_a(z) + iB^j_b(z)$ is supported on the negative frequencies only.

To first demonstrate the filter properties, we consider the high–pass filter on level three as an example. The corresponding passband in tree a comprises filtering with a product filter

$$B_a^3(z) = H_{0a}^1(z) H_{0a}^2(z^2) H_{1a}^3(z^4) \ ,$$

and analogously for tree b. Hence we deal with the expanded filters

$$\begin{aligned} H_{0b}^2(e^{2\pi i 2\omega}) &\overset{(3.14)}{=} \begin{cases} e^{-\pi i(2\omega+1)} H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[-\frac{1}{2}, -\frac{1}{4}\right) \\ e^{-\pi i 2\omega} \quad H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[-\frac{1}{4}, \frac{1}{4}\right) \\ e^{-\pi i(2\omega-1)} H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[\frac{1}{4}, \frac{1}{2}\right) \end{cases} \ , \\ &= \begin{cases} -e^{-2\pi i\omega} H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[-\frac{1}{2}, -\frac{1}{4}\right) \\ +e^{-2\pi i\omega} H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[-\frac{1}{4}, \frac{1}{4}\right) \\ -e^{-2\pi i\omega} H_{0a}^2(e^{2\pi i 2\omega}) & \omega \in \left[\frac{1}{4}, \frac{1}{2}\right) \end{cases} \ , \end{aligned} \tag{3.16}$$

$$\begin{aligned} H_{1b}^3(e^{2\pi i 4\omega}) &\overset{(3.15)}{=} e^{\pi i\left(4\omega - \frac{1}{2} - \lfloor 4\omega \rfloor\right)} H_{1a}^3(e^{2\pi i 4\omega}) \\ &= \begin{cases} -ie^{4\pi i\omega} H_{1a}^3(e^{2\pi i 4\omega}) & \omega \in \left[-\frac{1}{2}, -\frac{1}{4}\right) \\ \ \ ie^{4\pi i\omega} H_{1a}^3(e^{2\pi i 4\omega}) & \omega \in \left[-\frac{1}{4}, 0\right) \\ -ie^{4\pi i\omega} H_{1a}^3(e^{2\pi i 4\omega}) & \omega \in \left[0, \frac{1}{4}\right) \\ \ \ ie^{4\pi i\omega} H_{1a}^3(e^{2\pi i 4\omega}) & \omega \in \left[\frac{1}{4}, \frac{1}{2}\right) \end{cases} \ . \end{aligned} \tag{3.17}$$

The combined complex filter then reads

$$\begin{aligned} P^3(z) &= H_{0a}^1(z) H_{0a}^2(z^2) H_{1a}^3(z^4) + i H_{0b}^1(z) H_{0b}^2(z^2) H_{1b}^3(z^4) \\ &\overset{(3.11b)}{=} H_{0a}^1(z) \left(H_{0a}^2(z^2) H_{1a}^3(z^4) + i z^{-1} H_{0b}^2(z^2) H_{1b}^3(z^4) \right) \ . \end{aligned}$$

Subject to the angle ω, the expression in parentheses results in

$$\begin{aligned} & H_{0a}^2(e^{2\pi i 2\omega}) H_{1a}^3(e^{2\pi i 4\omega}) + i e^{-2\pi i\omega} H_{0b}^2(e^{2\pi i 2\omega}) H_{1b}^3(e^{2\pi i 4\omega}) \\ \overset{(3.16),(3.17)}{=} \ & H_{0a}^2(e^{2\pi i 2\omega}) H_{1a}^3(e^{2\pi i 4\omega}) \\ & + i e^{-2\pi i\omega} \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{Bmatrix} e^{-2\pi i\omega} H_{0a}^2(e^{2\pi i 2\omega}) \begin{Bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{Bmatrix} i e^{4\pi i\omega} H_{1a}^3(e^{2\pi i 4\omega}) \\ = \ & H_{0a}^2(e^{2\pi i 2\omega}) H_{1a}^3(e^{2\pi i 4\omega}) \left(1 - \begin{Bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{Bmatrix} \right) \end{aligned}$$

and finally

$$H_{0a}^2(e^{2\pi i2\omega})H_{1a}^3(e^{2\pi i4\omega}) + ie^{-2\pi i\omega}H_{0b}^2(e^{2\pi i2\omega})H_{1b}^3(e^{2\pi i4\omega})$$
$$= H_{0a}^2(e^{2\pi i2\omega})H_{1a}^3(e^{2\pi i4\omega}) \cdot \begin{cases} 0 & \omega \in \left[-\frac{1}{2}, -\frac{1}{4}\right) \\ 0 & \omega \in \left[-\frac{1}{4}, 0\right) \\ 2 & \omega \in \left[0, \frac{1}{4}\right) \\ 2 & \omega \in \left[\frac{1}{4}, \frac{1}{2}\right) \end{cases} .$$

As a consequence, the combined product filters have a passband only in the positive frequency range $\omega \in [0, 1/2]$.

To prove that our parallel filter banks fulfil (3.10) on all levels j, we need the following lemma:

Lemma 2. *For $j \in \mathbb{N}$ and $\omega \in [-1/2, 1/2)$, the function*

$$f_j(\omega) := \left[\left(2^j\omega + \frac{1}{2}\right) \bmod 1\right] - 2\omega - [2\omega \bmod 1] - \cdots - \left[2^{j-1}\omega \bmod 1\right]$$

fulfils

$$f_j(\omega) = \begin{cases} \frac{1}{2} & \omega \in \left[-\frac{1}{2}, 0\right) \\ -\frac{1}{2} & \omega \in \left[0, \frac{1}{2}\right) \end{cases} .$$

Proof. We prove the relation by induction on j.

For $j = 1$ we distinguish between two cases: For $\omega \in [-1/2, 0)$ we conclude that $2\omega + 1/2 \in [-1/2, 1/2)$ so that

$$(2\omega + \frac{1}{2}) \bmod 1 - 2\omega = 2\omega + \frac{1}{2} - 2\omega = \frac{1}{2} .$$

For $\omega \in [0, 1/2)$ we obtain $2\omega + 1/2 \in [1/2, 3/2)$ so that

$$(2\omega + \frac{1}{2}) \bmod 1 - 2\omega = 2\omega - \frac{1}{2} - 2\omega = -\frac{1}{2} .$$

If the assumption holds true for $k \leq j$, then we obtain

$$\begin{aligned} & f_{j+1}(\omega) \\ = \quad & f_j(\omega) + \left[\left(2^{j+1}\omega + \frac{1}{2}\right) \bmod 1\right] - \left[\left(2^j\omega + \frac{1}{2}\right) \bmod 1\right] - [2^j\omega \bmod 1] \\ = \quad & \left[\left(2^{j+1}\omega + \frac{1}{2}\right) \bmod 1\right] - \left[\left(2^j\omega + \frac{1}{2}\right) \bmod 1\right] - [2^j\omega \bmod 1] \\ & + \begin{cases} \frac{1}{2} & \omega \in \left[-\frac{1}{2}, 0\right) \\ -\frac{1}{2} & \omega \in \left[0, \frac{1}{2}\right) \end{cases} . \end{aligned}$$

It remains to show that $\big[\big(2^{j+1}\omega + 1/2\big) \bmod 1\big] - \big[\big(2^j\omega + 1/2\big) \bmod 1\big] - [2^j\omega \bmod 1] = 0$.

Every $\omega \in \mathbb{R}$ can be written as $\omega = k2^{-(j+1)} + \omega_0$ for $k \in \mathbb{Z}$ and $\omega_0 \in \big[0, 2^{-(j+1)}\big)$. Then it follows

$$\left(2^{j+1}\omega + \frac{1}{2}\right) \bmod 1 = \left(k + 2^{j+1}\omega_0 + \frac{1}{2}\right) \bmod 1 = 2^{j+1}\omega_0 - \frac{1}{2} \ .$$

Concerning the remaining expressions, we consider again two cases:

$$\begin{aligned}\left(2^j\omega + \frac{1}{2}\right) \bmod 1 &= \left(\frac{k}{2} + 2^j\omega_0 + \frac{1}{2}\right) \bmod 1 = \begin{cases} 2^j\omega_0 - \frac{1}{2} & 2 \mid k \ , \\ 2^j\omega_0 & 2 \nmid k \ , \end{cases} \\ 2^j\omega \bmod 1 &= \left(\frac{k}{2} + 2^j\omega_0\right) \bmod 1 = \begin{cases} 2^j\omega_0 & 2 \mid k \ , \\ 2^j\omega_0 - \frac{1}{2} & 2 \nmid k \end{cases}\end{aligned}$$

so that

$$\begin{aligned}&\left[\left(2^{j+1}\omega + \frac{1}{2}\right) \bmod 1\right] - \left[\left(2^j\omega + \frac{1}{2}\right) \bmod 1\right] - [2^j\omega \bmod 1] \\ &= 2^{j+1}\omega_0 - \frac{1}{2} - 2^j\omega_0 + \frac{1}{2} - 2^j\omega_0 = 0 \ .\end{aligned}$$

□

By the following theorem, we see that our filter banks indeed fulfil (3.10):

Theorem 3. *Let the filters for two cascaded filter banks a and b be given by* (3.11), (3.12), (3.14) *and* (3.15). *For $j = 2, \dots, J$, let the corresponding product filters $B_\cdot^j$ be defined by* (3.3b). *Then it holds*

$$B_b^j(e^{2\pi i\omega}) = \begin{cases} iB_a^j(e^{2\pi i\omega}) & \omega \in \big[-\frac{1}{2}, 0\big) \ , \\ -iB_a^j(e^{2\pi i\omega}) & \omega \in \big[0, \frac{1}{2}\big) \ . \end{cases} \tag{3.18}$$

The filters B_a^j and B_b^j are real.

Thus, if B_a^j and B_b^j are nearly supported as in (3.7), $B_a^j \pm iB_b^j$ have the same support but only on the right or left hand side of the real axis, respectively. Together with the derivation in the previous section for the first level $j = 1$, this means that if the filters are well localised, then the transform is approximately free of aliasing. And still, the wavelet coefficients of both trees D_a^j and D_b^j are real.

Proof. We obtain by (3.3b), (3.11a) and (3.12) that

$$B_a^j(e^{2\pi i\omega}) = H_0(e^{2\pi i\omega})H_0(e^{2\pi i2\omega}) \cdots H_0(e^{2\pi i2^{j-2}\omega})H_1(e^{2\pi i2^{j-1}\omega}) \ .$$

On the other hand, we get by (3.3b), (3.11b), (3.14) and (3.15) that

$$\begin{aligned} B_b^j(e^{2\pi i\omega}) &= e^{-2\pi i\omega}H_0(e^{2\pi i\omega})e^{-\pi i(2\omega \bmod 1)}H_0(e^{2\pi i2\omega})\cdots e^{-\pi i(2^{j-2}\omega \bmod 1)}H_0(e^{2\pi i2^{j-2}\omega}) \\ &\quad e^{\pi i\left[\left(2^{j-1}\omega+\frac{1}{2}\right) \bmod 1\right]}H_1(e^{2\pi i2^{j-1}\omega}) \\ &= e^{\pi i\left\{\left[\left(2^{j-1}\omega+\frac{1}{2}\right) \bmod 1\right]-2\omega-[2\omega \bmod 1]-\cdots-[2^{j-2}\omega \bmod 1]\right\}} \\ &\quad H_0(e^{2\pi i\omega})H_0(e^{2\pi i2\omega})\cdots H_0(e^{2\pi i2^{j-2}\omega})H_1(e^{2\pi i2^{j-1}\omega}) \\ &= e^{\pi i\left\{\left[\left(2^{j-1}\omega+\frac{1}{2}\right) \bmod 1\right]-2\omega-[2\omega \bmod 1]-\cdots-[2^{j-2}\omega \bmod 1]\right\}}B_a^j(e^{2\pi i\omega}) \end{aligned}$$

and further by Lemma 2 that

$$\begin{aligned} B_b^j(e^{2\pi i\omega}) &= B_a^j(e^{2\pi i\omega})\begin{cases} e^{\pi i/2} & \omega \in \left[-\frac{1}{2},0\right) \\ e^{-\pi i/2} & \omega \in \left[0,\frac{1}{2}\right) \end{cases}, \\ &= B_a^j(e^{2\pi i\omega})\begin{cases} i & \omega \in \left[-\frac{1}{2},0\right) \\ -i & \omega \in \left[0,\frac{1}{2}\right) \end{cases}. \end{aligned}$$

By (3.3b), (3.11a) and (3.12) the product filter B_a^j is real. Now a filter is real if and only if its negative frequency response is the complex conjugate of its positive frequency response. Hence, (3.18) implies that B_b^j is real if B_a^j is real. This completes the proof. □

One of the main ideas of our proof, namely the careful handling of the one–periodicity of the filters we have later also found in Selesnick's article [Selesnick, 2001]. However, Selesnick considers infinite filter iterations related to wavelets, i.e., in our notation, $B_\cdot^j(z^{2^{1-j}})$ for $j \to \infty$, whereupon the b filter bank including the first step has to be shifted by half a sample. He shows that then the corresponding wavelets obtained with the CQF setting form a *Hilbert transform* pair, that is they are related as B_a^j and B_b^j in (3.18). This implies that the resulting combined complex wavelet has only positive frequency response. In contrast, we address exactly Kingsbury's approach with a finite number of filter iterations and a special design of the first filter bank pair.

As the combined filter $P^j = B_a^j + iB_b^j$ by the theorem is equivalent to the product filter B_a^j on the positive frequencies, one could also try to apply a single filter bank with filter P^j to avoid the computational overhead. But even if you accomplish to realise this filter bank efficiently, this prevents from perfect reconstruction. Besides, as we will see later on, the dual–tree representation in more than one dimension is not inefficient compared with a filter bank with positive frequency filter.

Finally, let us also have a look at the low–pass filters. If we also combine the low–pass filters as in (3.10a), the resulting filter still responds to negative frequencies. By (3.3a),

(3.11), (3.12) and (3.14) we obtain that

$$\begin{aligned} & A_a^j(e^{2\pi i\omega}) + iA_b^j(e^{2\pi i\omega}) \\ = \; & H_0(e^{2\pi i\omega}) \cdots H_0(e^{2\pi i 2^{j-1}\omega}) \\ & + ie^{-2\pi i\omega} e^{-\pi i(2\omega \bmod 1)} \cdots e^{-\pi i(2^{j-1}\omega \bmod 1)} H_0(e^{2\pi i\omega}) \cdots H_0(2\pi i e^{2^{j-1}\omega}) \\ = \; & A_a^j(e^{2\pi i\omega}) \left(1 + ie^{-\pi i(2\omega + (2\omega \bmod 1) + \cdots + (2^{j-1}\omega \bmod 1))}\right) \quad . \end{aligned}$$

For ω in the desired support of the filters $[-1/(3 \cdot 2^{j-1}), 1/(3 \cdot 2^{j-1})]$, this simplifies to

$$A_a^j(e^{2\pi i\omega}) + iA_b^j(e^{2\pi i\omega}) = A_a^j(e^{2\pi i\omega}) \left(1 + ie^{-2^j \pi i\omega}\right)$$

so that

$$|A_a^j(e^{2\pi i\omega}) + iA_b^j(e^{2\pi i\omega})| = |A_a^j(e^{2\pi i\omega})|(2 + 2\sin 2^j\pi\omega)^{1/2} \quad . \tag{3.19}$$

The second factor on the right hand side takes its minimum zero at $\omega = -2^{-(j+1)}$ and its maximum two at $\omega = 2^{-(j+1)}$. As a consequence, the frequency response of the combined low–pass filter also suppresses negative frequencies to some extent and leans to the right compared with $|A_a^j(e^{2\pi i\omega})|$.

Similarly, the first level combined complex high–pass filter $B_a^1(e^{2\pi i\omega}) + iB_b^1(e^{2\pi i\omega})$ with ideal passband of $[-1/2, -1/4] \cup [1/4, 1/2]$ has the magnitude

$$\begin{aligned} |B_a^1(z) + iB_b^1(z)|^2 &= (H_1(z) + iz^{-1}H_1(z))(H_1(z^{-1}) - izH_1(z^{-1})) \\ &= (1 + i(z^{-1} - z) - i^2 z^{-1} z) H_1(z) H_1(z^{-1}) \\ &= (2 + i(-2i\sin 2\pi\omega))|H_1(z)|^2 \\ &= 2(1 + \sin 2\pi\omega)|H_1(z)|^2 \quad . \end{aligned}$$

As a consequence, the filter also has a passband mainly in the positive frequency range $\omega \in [0, 1/2]$.

A similar alias cancelling approach is followed by [Bernard, 1999] when constructing analytic wavelets for applying them to optic flow computation. He also uses a multiplicative mask in the frequency domain to obtain only the positive frequency part of the filter. Only instead of the sine function above, he uses Deslauriers-Dubuc interpolation filters. These still achieve better suppression of negative frequencies, but have no trivial realisation in the spatial domain.

3.4. Complex Wavelet Transform in Multiple Dimensions

In order to extend the transform to multiple dimensions, a filter bank is usually applied separably in all dimensions. But for the complex filters, a further extension is necessary as also indicated by [Kingsbury and Magarey, 1997]: If we apply the Fourier transform

to a real signal, the representation is conjugate symmetric to the origin so that, in one dimension, we obtain that the negative frequency part is just the complex conjugate of the positive frequency part. Hence the signal may be recovered from just one half of its spectrum and is recoverable from the filter output of complex filter banks with a passband of just positive frequencies. But in multiple dimensions, only opposite quadrant Fourier coefficients are redundant being complex conjugates. With the separably applied complex product filter having only frequency response in the positive quadrant, the signal hence cannot be recovered. In m dimensions, the frequency response of 2^{m-1} non-opposite quadrants is necessary to recover the signal. The necessary information can be obtained by conjugate filters. All necessary quadrants are, e.g., covered by applying all positive/negative frequency tensor products of the filters in $m-1$ dimensions and leaving the filter in the remaining dimension fixed. At level j, the filter bank for $m=2$ should then produce the outputs

$$\begin{aligned}
C^j_{a/b}(z_1,z_2) &= (2^j\downarrow)\Big((A^j_a(z_1)\pm iA^j_b(z_1))(A^j_a(z_2)+iA^j_b(z_2))S(z_1,z_2)\Big)\ ,\\
D^j_{1a/b}(z_1,z_2) &= (2^j\downarrow)\Big((A^j_a(z_1)\pm iA^j_b(z_1))(B^j_a(z_2)+iB^j_b(z_2))S(z_1,z_2)\Big)\ ,\\
D^j_{2a/b}(z_1,z_2) &= (2^j\downarrow)\Big((B^j_a(z_1)\pm iB^j_b(z_1))(A^j_a(z_2)+iA^j_b(z_2))S(z_1,z_2)\Big)\ ,\\
D^j_{3a/b}(z_1,z_2) &= (2^j\downarrow)\Big((B^j_a(z_1)\pm iB^j_b(z_1))(B^j_a(z_2)+iB^j_b(z_2))S(z_1,z_2)\Big)\ ,
\end{aligned}$$

where $2^j\downarrow$ denotes downsampling by 2^j and the subscript a/b is related to the $\pm$ in the first factor, see also Fig. 3.7. Hence, the filter bank has six complex high–pass subbands at each level and two complex low–pass subbands in contrast to three real high–pass and one real low–pass subband for the real two-dimensional transform. So the complex transform has a coefficient redundancy of 4:1 or $2^m:1$ in m dimensions.

Due to the special filter construction and the dual–tree implementation in our case, the required complex product filters may be realised easily [Kingsbury, 1999]. For the filter pairs subsumed in Theorem 3, we obtain filters supported on the other half of the ω-axis by just toggling the sign in $B^j_a \pm iB^j_b$ or by combining the filter outputs of the separate filters B^j_a and B^j_b with a different sign, respectively. To implement the filter bank efficiently, [Kingsbury, 1999] proposes to apply the real and imaginary filter parts separately and combine them in the end. But one has to be careful: As in the one-dimensional case, no complex calculus is done between the real and imaginary parts as they only come out to be complex coefficient pairs in the final band. Figure 3.7 shows two levels of the resulting dual–tree filter bank for a two-dimensional input signal $\mathbf{S}$ without subsampling operations. The markers indicate real coefficient parts r and row or column imaginary parts i_1 and i_2, respectively. The output of each subband in the filter bank is a 4-tuple $(r,t,s,u) \mathrel{\widehat{=}} r+si_1+ti_2+ui_1i_2$. To obtain the usual filters corresponding to the a coefficients, one sets $i_1=i_2=i$ and obtains $(r-u)+i(s+t)$. For the conjugate row filter, one sets $-i_1=i_2=i$ and obtains $(r+u)+i(-s+t)$ for the b

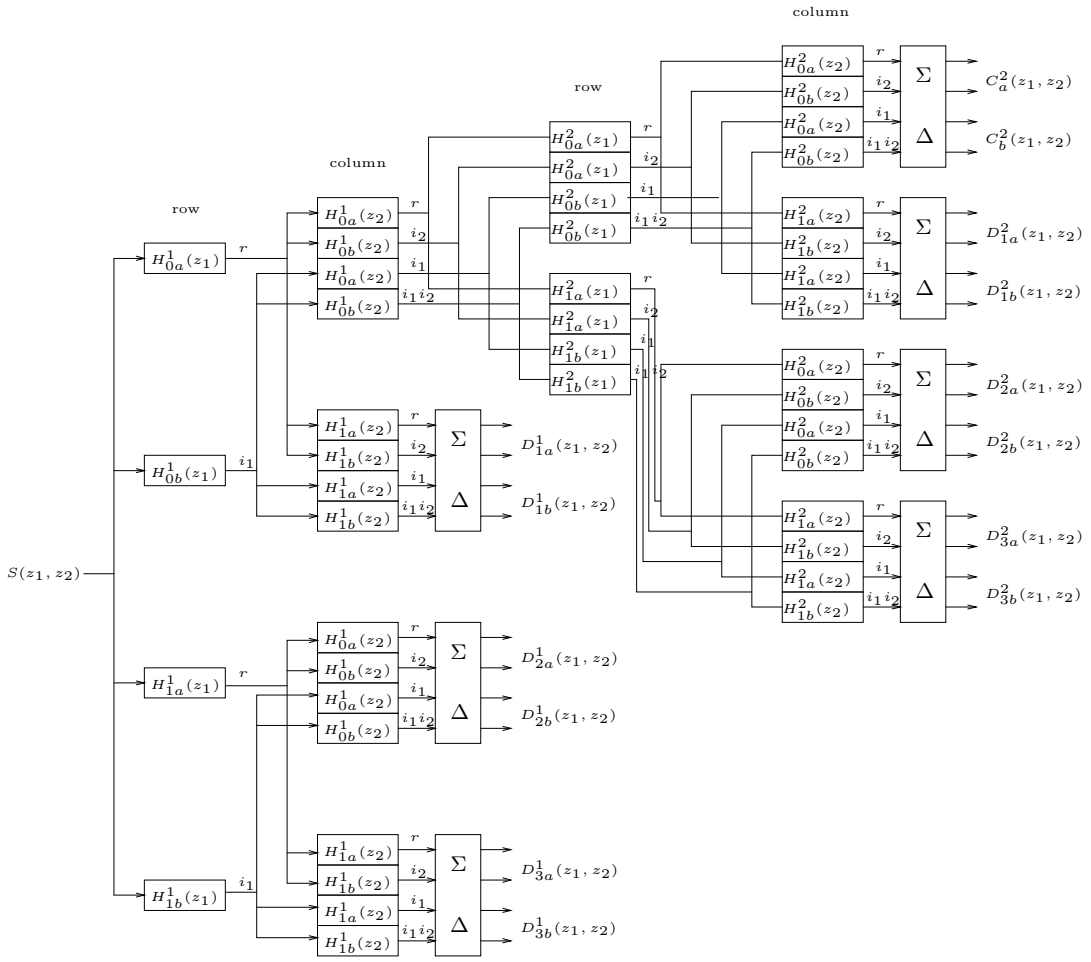


Figure 3.7.: Dual–tree filter bank for a 2D signal **S** without subsampling operations

coefficients. This operation is indicated by the Σ, Δ blocks at the end of each subband. For further decomposition steps, the row and column filtering blocks of the second level can be iterated for the low–pass channel, i.e., they replace the uppermost Σ, Δ block. One can see in Fig. 3.7 that the real coefficients marked by r then are transformed step by step by themselves without interleaving with the other coefficient bands. The same occurs for the other three components. Hence, in two dimensions, from level two on we have four parallel trees that are only combined in the end to build the complex coefficients.

In the case of real two-dimensional filter banks, the three high–pass filters have orientations of $0\,^\circ, 45\,^\circ$ and $90\,^\circ$, respectively. For the complex filters, [Kingsbury, 1998, Kingsbury, 2001] claims that the six subband filters are oriented at $\pm 15\,^\circ$, $\pm 45\,^\circ$ and

$\pm 75\,^\circ$. Indeed, if we determine the filter orientation as the angle of the maximal frequency response magnitude, by $\max |F_{\mathrm{row}}(z_1)F_{\mathrm{col}}(z_2)| = (\max |F_{\mathrm{row}}(z_1)|)(\max |F_{\mathrm{col}}(z_2)|)$ the orientation is given by

$$\arctan \frac{\arg\max_\omega |F_{\mathrm{col}}(e^{2\pi i\omega})|}{\arg\max_\omega |F_{\mathrm{row}}(e^{2\pi i\omega})|}$$

so that we have to examine the maxima of $|A_a^j + iA_b^j|$ and $|B_a^j + iB_b^j|$. By Theorem 3 and (3.19), we see that they depend on the single filters B_a^j and A_a^j whose shapes in turn depend on the basis filter and even on the level j. For example for the Haar filter we have

$$\begin{aligned} A_a^j(e^{2\pi i\omega}) &= \left(\frac{1+e^{-2\pi i\omega}}{\sqrt{2}}\right)\cdots\left(\frac{1+e^{-2\pi i 2^{j-1}\omega}}{\sqrt{2}}\right) \\ \Rightarrow |A_a^j(e^{2\pi i\omega})| &= 2^{j/2}|\cos\pi\omega\cdots\cos 2^{j-1}\pi\omega| \overset{\text{[Daubechies, 1992, p.211]}}{=} 2^{-j/2}\left|\frac{\sin 2^j\pi\omega}{\sin\pi\omega}\right| \ . \end{aligned}$$

With (3.19), one can easily check that $\arg\max_\omega |A_a^j(e^{2\pi i\omega}) + iA_b^j(e^{2\pi i\omega})| \neq 2^{-j}\cdot\mathrm{const}$, nor is the total orientation angle constant during the levels. The actual orientations for our examined complex row high–pass and column low–pass filters corresponding to subband $2a$ vary roughly from $15\,^\circ$ to $30\,^\circ$. Of course, not only the maximum frequency response is important for the filter orientation, but the filter may be a superposition of differently oriented components.

Even if the two-dimensional complex transform does not have fixed filter orientations, we expect it to be more robust against angular disturbances because of the six differently oriented subbands.

3.5. Performance Evaluation of Kingsbury's Dual–Tree Complex Wavelet Transform

In this section we examine the behaviour of the dual–tree wavelet transform with respect to translation and rotational invariance. To achieve the filter delay of half a sample and at the same time keep the filter responses approximately the same as required in Sec. 3.3, Kingsbury proposes two different approaches: One can either alternate even and odd length biorthogonal filters or use orthogonal basis filters that have a delay of a quarter of a sample by themselves. In the latter case, the total delay difference of half a sample between the trees (3.13) implies for the filters at levels $j \geq 2$

$$H_{0b}^j(z) = z^{-1}H_{0a}^j(z^{-1}) \ .$$

As an example, as proposed by [Kingsbury, 2001], we consider (5,3)-tap biorthogonal LeGall filters on the first level and a 6-tap orthogonal filter of length ten subsequently,

namely

$$\begin{aligned} H_{0a}^1(z) &= \frac{1}{8}(-z^2+2z+6+2z^{-1}-z^{-2}) \ , \\ H_{1a}^1(z) &= \frac{1}{2}z^{-1}(-z+2-z^{-1}) \ , \\ H_{0a}^j(z) &= 0.03516384z^4 - 0.08832942z^2 + 0.23389032z \\ &\quad +0.76027237 + 0.58751830z^{-1} - 0.11430184z^{-3} \ , \quad j \geq 2 \ , \end{aligned}$$

where the orthogonal filter $H_{0a}^j(z)$ for $j \geq 2$ results from the lattice structure (2.4) with angles $\boldsymbol{\theta} = \pi/4\,(-1.62, 0.81, 1.81, 0)$ and an additional delay of z^4. The imaginary filter part applied in tree b then reads

$$\begin{aligned} H_{0b}^j(z) &= z^{-1}H_{0a}^j(z^{-1}) \\ &= 0.03516384z^{-5} - 0.08832942z^{-3} + 0.23389032z^{-2} \\ &\quad +0.76027237z^{-1} + 0.58751830 - 0.11430184z^2 \ , \quad j \geq 2 \ . \end{aligned}$$

The biorthogonal filters are rescaled afterwards to obtain $H_0(1) = G_0(1) = \sqrt{2}$ although the Parseval equality is replaced by the Riesz stability condition here anyway. But this ensures that the coefficients have the same magnitude for all transforms.

Longer filters proposed by [Kingsbury, 2001] are the 14-tap orthogonal filter

$$\begin{aligned} H_{0a}^j(z) &= 0.00325314z^6 - 0.00388321z^5 + 0.03466035z^4 - 0.03887280z^3 \\ &\quad -0.11720389z^2 + 0.27529538z^1 + 0.75614564 + 0.56881042z^{-1} \\ &\quad +0.01186609z^{-2} - 0.10671180z^{-3} + 0.02382538z^{-4} + 0.01702522z^{-5} \\ &\quad -0.00543948z^{-6} - 0.00455690z^{-7} \ , \quad j \geq 2 \end{aligned}$$

with the biorthogonal (9,7)-tap Antonini filters with two vanishing moments

$$\begin{aligned} H_{0a}^1(z) &= 0.03782845550726z^4 - 0.02384946501956z^3 - 0.11062440441844z^2 \\ &\quad +0.37740285561283z + 0.85269867900889 + 0.37740285561283z^{-1} \\ &\quad -0.11062440441844z^{-2} - 0.02384946501956z^{-3} + 0.03782845550726z^{-4} \ , \\ H_{1a}^1(z) &= 0.06453888262870z^2 \\ &\quad -0.04068941760916z - 0.41809227322162 + 0.78848561640558z^{-1} \\ &\quad -0.41809227322162z^{-2} - 0.04068941760916z^{-3} + 0.06453888262870z^{-4} \end{aligned}$$

in the first step.

3.5.1. One-dimensional

The complex filter bank with the 6-tap filter just presented and the (5,3)-tap filters in the first step, has approximately the same filter lengths as the Daubechies filter from

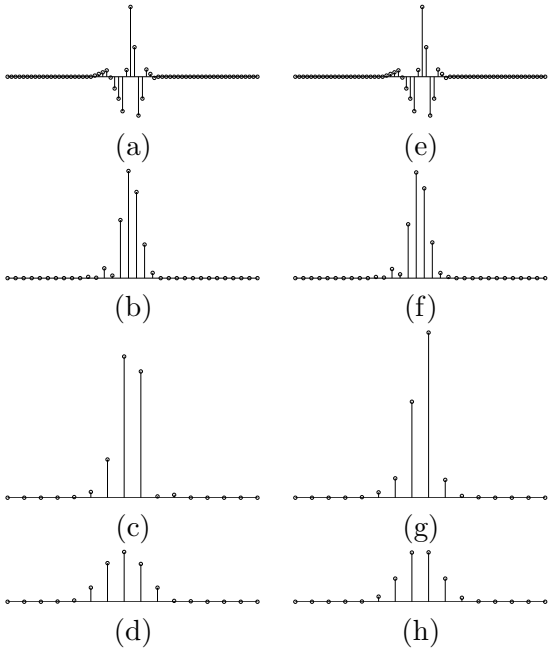


Figure 3.8.: Shift sensitivity of Kingsbury's complex wavelet transform: (a) original signal equal to Fig. 3.1 (a), (b)–(d) magnitude of wavelet subband coefficients, (e) signal from (a) shifted by one sample, (f)–(h) magnitude of new wavelet subband coefficients

Fig. 3.1 with its six coefficients. The behaviour of the complex filter bank on signal shifts is illustrated in Fig. 3.8 analogous to Fig. 3.1. We are only able to plot the magnitude of the complex coefficients here. In contrast to the discrete transform, the distribution of the coefficient energy across the subbands is almost equal for the shifted signal. Even the shape of the coefficient magnitude mainly stays the same. This indicates that the complex wavelet transform behaves more stable on signal shifts.

An illustration of the shift invariance that is a bit more founded is given in Fig. 3.9. For four levels of the same wavelet transforms as in Figs. 3.1 and 3.8, the contribution of all subbands is shown. For each subband, only the appropriate coefficients are passed to the respective inverse transform. The analysed signals are a step function and a sample row of the corrugated iron image 'Misc.0002' from the MeasTex collection [Smith, 1997]. This texture and the similar 'Misc.0003' as well as two exemplary rows are shown in Fig. 1.2. Clearly, the subband contributions in Fig. 3.9 are periodic with period 2^l at level l. Again it shows that the subband information of the complex wavelet transforms in Fig. 3.9 (b) and (d) below is much more stable across all shifts than that of the common wavelet transform.

For our application of feature extraction by the wavelet transform and subsequent

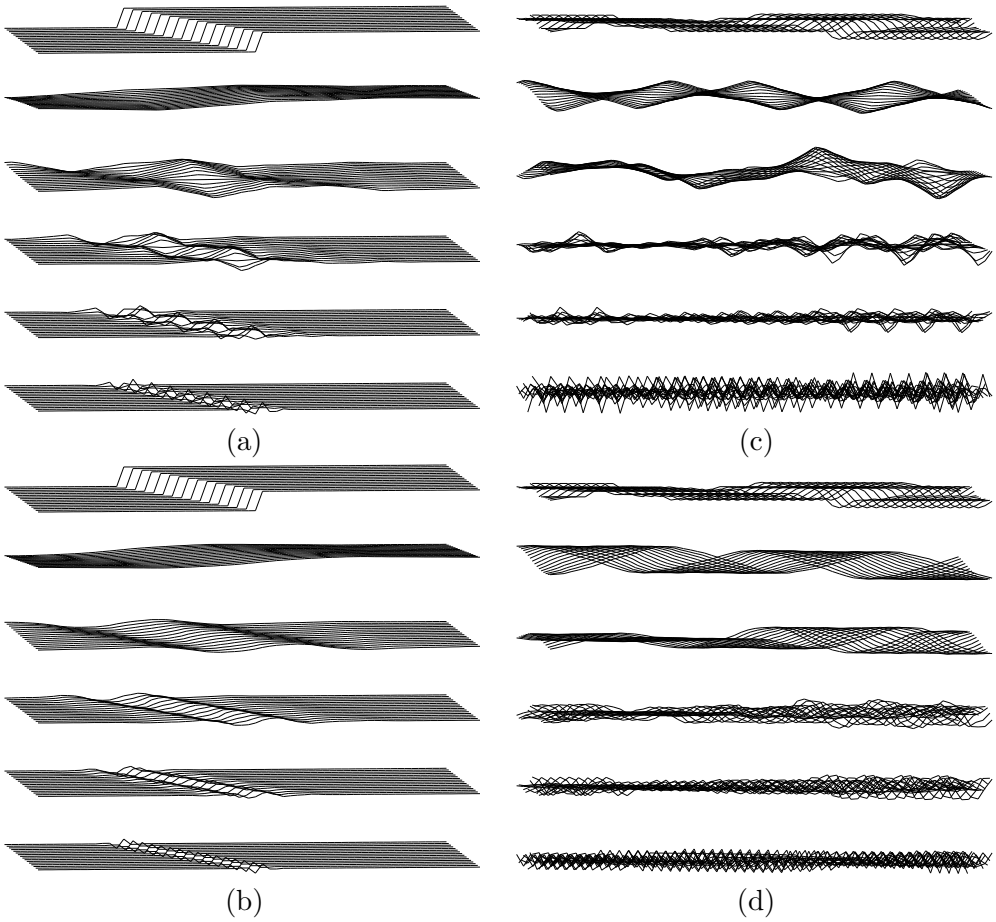


Figure 3.9.: Subband information of real and complex wavelet transforms; shifted signal and contribution of scaling function and wavelets from level four to one at the 16 distinctive shifts: (a) step function represented by Daubechies 3 wavelet, (b) step function represented by complex wavelet, (c) sample row of texture 'misc2' represented by Daubechies 3 wavelet, (d) sample row of texture 'misc2' represented by complex wavelet

		level								
signal	wavelet	1	2	3	4	5	6	7	8	9
	Haar	34	252	1988	10954	11923	12506	1305	3413	3879
misc2	Daub. 3	41	153	1211	8469	8634	4685	719	2963	3730
	complex	0	7	42	56	55	74	27	38	40
	Haar	46	332	619	1159	2147	6146	2125	5001	5325
misc3	Daub. 3	46	231	356	1165	4507	7725	1103	6217	7483
	complex	0	9	16	8	8	13	22	80	85

Table 3.1.: Average feature scatter on shifts of the input signal

classification, it is especially interesting how signal shifts affect the extracted features. Therefore, we compute the scatter of each feature for a single signal $\mathbf{s}$ with respect to signal shifts

$$\frac{1}{l}\sum_{i=1}^{l}\left(E_{\|\,\|_{\ell_2}}F\mathbf{s}_{\cdot-i}-\frac{1}{l}\sum_{j=1}^{l}E_{\|\,\|_{\ell_2}}F\mathbf{s}_{\cdot-j}\right)^2 \quad ,$$

where F denotes the filter operator $F_{\boldsymbol{\theta}}$ for orthogonal wavelet transforms or the complex transform. Using the ℓ_2-norm for the energy operator (2.17), the features are equal to the channel energies. Table 3.1 gives the feature scatter for the transforms with the Haar wavelet and again the Daubechies 3 and the 6-tap complex wavelets for both images shown in Fig. 1.2. The features for the complex and real transforms have the same magnitude as the biorthogonal filters H_{0a}, H_{1a} are normalised as well. Hence the scatter is comparable. Evidently, the scatter of the complex features is much smaller than for the real features. It even gets close to zero which is the variance of the features for the non-subsampled transform with wavelet frames. At level one of the complex transform, the variance is always zero because effectively, no subsampling is done due to the delayed filters (3.11b).

3.5.2. Two-dimensional

Concerning the rotational invariance of the wavelet features, we make a similar investigation as in the one-dimensional case. We compute the scatter of each combined feature for an image $\mathbf{S}$ with respect to image rotations

$$\frac{1}{n_r}\sum_{r=1}^{n_r}\left(E_{\|\,\|_{\mathrm{F}}}FR_{2\pi r/n_r}\mathbf{S}-\frac{1}{n_r}\sum_{s=1}^{n_r}E_{\|\,\|_{\mathrm{F}}}FR_{2\pi s/n_r}\mathbf{S}\right)^2 \quad ,$$

where F denotes the filter operator again and R_r denotes a rotation about the centre of radian angle r. The rotated image's corner values cannot be computed. Consequently, we only examine a smaller part around the image centre. By 'combined features', we

signal	wavelet		level 1	2	3	4	5	6	7	8
misc2	critically sampled	Haar	4845	2976	1638	748	279	5	1	1
		Daub. 3	13226	3506	406	2641	276	8	3	2
	complex	(5,3)/6-tap	1744	618	638	1081	162	6	5	8
	no subsampling	Haar	1456	280	50	38	1	0	1	1
		Daub. 3	1580	542	325	730	75	8	2	2
misc3	critically sampled	Haar	57260	865	7031	510	104	4	1	1
		Daub. 3	24422	18939	927	229	58	6	3	2
	complex	(5,3)/6-tap	2449	4033	2417	457	113	6	5	9
	no subsampling	Haar	1594	2793	124	33	2	0	1	1
		Daub. 3	1413	3421	1520	239	48	6	3	2

Table 3.2.: Average feature scatter on rotations of the input image

mean that the coefficients of all subbands on a certain level are combined to obtain an intended rotational invariant feature for each subband frequency. As the complex transform covers six directions roughly corresponding to $15\,^\circ + n30\,^\circ$ for $n = 0, \ldots, 5$, we choose $n_r = 24$ in our examples to cover all full and intermediate image rotations. Table 3.2 gives the feature scatter for the transforms with the Haar, the Daubechies 3 and the 6-tap complex wavelet again for both images shown in Fig. 1.2. We conclude that the scatter of the complex features is smaller than for the real features, but as in one dimension again larger than the scatter of the features for the non-subsampled transform with wavelet frames.

We now investigate the classification performance of a 2D wavelet–SV classifier with complex wavelets. We also include wavelets adapted to the problem by selecting from the parameter space $\mathcal{P}_2$ resulting of the lattice structure (2.4) the filter that maximally separates the class centres of the feature vectors, confer Chap. 4. For the same rotations as for the feature scatter investigation, we cut out quadratic image fragments of side length 64 and try to classify them according to their combined weighted Frobenius norm features at the six decomposition levels. Another way of using wavelet subband energy to classify images is to decompose the image as a whole and locally aggregate the energy features, e.g. squared coefficients, by histograms or smoothing filters as in [Randen and Husøy, 1999] instead of using the energy operator E. Other popular features for texture classification are, e.g., coefficient correlations [Portilla and Simoncelli, 2000] or filter output histograms. The latter technique is also applied in the experiments in Sec. 5.6.3. But here we simply divide the image into disjoint fragments.

As the textures 'misc2' and 'misc3' are structurally too similar, we use two normalised texture images from the Brodatz collection [Brodatz, 1966] that are visualised in Fig. 3.10. The classification results for the problem 'd16 – d84' are given in Table 3.3.

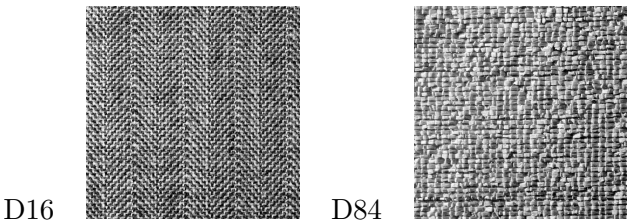


Figure 3.10.: Texture images

wavelet		d16 – d84	d16 – d84 rotated
critically sampled	Haar	23	19
	Daub. 3	14	15
	adapted	19	18
complex	(5,3)/6-tap	12	16
	(9,7)/14-tap	12	15
no subsampling	Haar	8	13
	Daub. 3	8	15
	adapted	9	14

Table 3.3.: Classification error [%] of an image classifier for different wavelet transforms

The fragments of the first 64 rows of each image are used for training, the rest is used for evaluating the classification error. In the rotated problem version, the test data instead consists of the centre fragments of the images rotated by radian angles $(2\pi r)/24$ for $r = 1, \ldots, 24$ about the centre.

The complex wavelets achieve better classification performance than the critically sampled wavelets, rather comparable to that of the transform without subsampling. Image rotations seem to have the highest impact on the overcomplete transforms. But problematic with these experiments are the small image fragments. If we only apply five decomposition steps instead of six, the classification accuracy gets much better. This may constitute a significant disadvantage for the longer complex wavelets.

3.6. Complex Wavelet Transform in the Frequency Domain

A shift invariant transform may be obtained using a pair of real filters that are delayed by half a sample as argued in Sec. 3.3. The specific delay together with an identical frequency response may be achieved only approximately. Examples are filters of odd and even length or filters with a shift of a quarter sample by themselves given in Sec. 3.5.

An alternative to this approach is to design the shifted filter in the frequency domain where the above requirements can be exactly achieved. When transforming back

the filter into the spatial domain, again an approximation has to be made if only real filters (with symmetric magnitude in the frequency domain) are allowed. But the transform can be performed in the frequency domain as well. In this section we propose to apply the dual–tree complex wavelet transform in the frequency domain. This has the advantage that it suffices to know $H_\cdot(e^{2\pi i\omega})$ for some discrete values of ω, while the explicit knowledge of the filter coefficients $h_\cdot$ is not necessary. Hence, we can start with known orthogonal, but not necessarily FIR filters that approximately fulfil the support condition (3.5) and add an appropriate second filter bank. No special filter design is necessary as the complex filter construction procedure applies to all orthogonal wavelets. In our numerical experiments we apply, e.g., Butterworth filters [Oppenheim and Schafer, 1989, Gottscheber and Steidl, 1999] and orthogonal B–spline filters (Battle–Lemarié filters [Blatter, 2003, Chap. 6.4]) of different orders.

Of course this transform requires application of the Fourier transform so that with respect to arithmetic complexity this approach can only compete with real filter banks in the time domain having not too small filter lengths.

By Theorem 3 we have an explicit construction method for cascaded filters with vanishing negative frequency parts. Their approximate shift invariance should improve the more closely condition (3.5) is fulfilled. We examine complex filter banks based on the following standard orthogonal filters H_0:

symbol	orthogonal basis filter H_0
H	Haar filter
D_3	Daubechies filter with three vanishing moments
BW_m	Butterworth filter with m vanishing moments [Oppenheim and Schafer, 1989, Gottscheber and Steidl, 1999]
BL_m	Battle–Lemarié filter with $m+1$ vanishing moments [Blatter, 2003, Chap. 6.4], [Mallat, 1999, p. 249]

The Haar and Daubechies filters are FIR filters of length two and six, respectively, which are generated by the lattice structure (2.4). The Butterworth and Battle–Lemarié orthogonal spline filters have infinite impulse response. The BW and BL filters have most notable frequency characteristics. As for the 'sinc' function in the continuous case, the resulting drawback is infinite support in the spatial domain. But when the transform is performed in the frequency domain anyway, this doesn't need to bother us. For other properties see [Oppenheim and Schafer, 1989, Blatter, 2003].

The frequency responses of the combined product filters for these basis filters are plotted in Fig. 3.11. As it was to expect, we observe that all complex filters P^j for $j \geq 2$ are basically only supported on the right half of the ω-axis, whereas the low–pass and the first level filters P^1 also respond to negative frequencies. Further, in agreement with (3.19), the low–pass filters' frequency responses all 'lean' to the right. The suppression of the side bumps improves as the order m of the filter increases and it satisfies property (3.5) more closely. But then, the filters become less concentrated in the time domain.

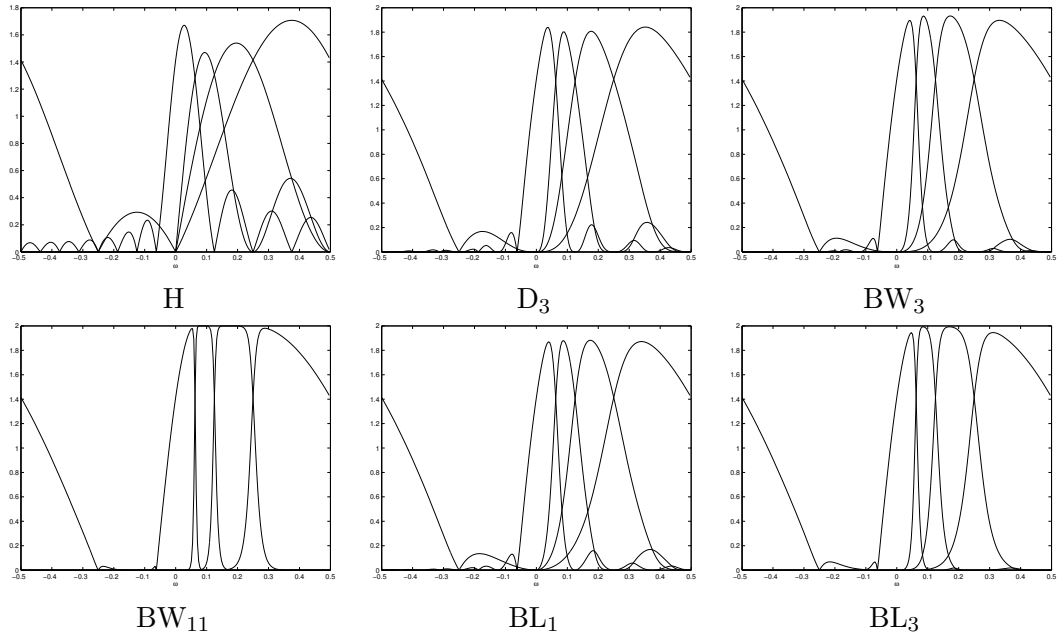


Figure 3.11.: Frequency response magnitude of complex filters $A_a^3 + iA_b^3$ and P^j for $j = 3, 2, 1$

The real impulse responses of the constructed 2D combined complex filters for BW_3 at level four of the transform are plotted in Fig. 3.12. The appropriate complex parts look similar. As for Kingsbury's specially designed filters, the six subband filters all have different dominant directions. This is still better visible in the frequency domain: Figure 3.13 shows the frequency responses of the level three combined complex Butterworth filters for all six subbands again.

The real parts of other filters' impulse responses are plotted in Fig. 3.14. As for the BW filters, the corresponding complex parts look similar and the other orientations are roughly conjugated mirrors. The filters' impulse and frequency responses look similar to those of the BW_3 filter displayed in Figs. 3.12 and 3.13, respectively.

3.7. Fast Wavelet Transform in the Frequency Domain

In the previous section we proposed a general construction method to obtain nearly shift invariant filter banks defined in the frequency domain and designed some exemplary filter banks. Of course these filter banks can in general not be applied in the time domain: even if the filters $H_{:a}^j$ have FIR, their shifted versions by half a sample $H_{:b}^j$ will not have this property due to their degraded regularity. So the runtime of a time domain transform step becomes quadratic. Hence we have to make the decompo-

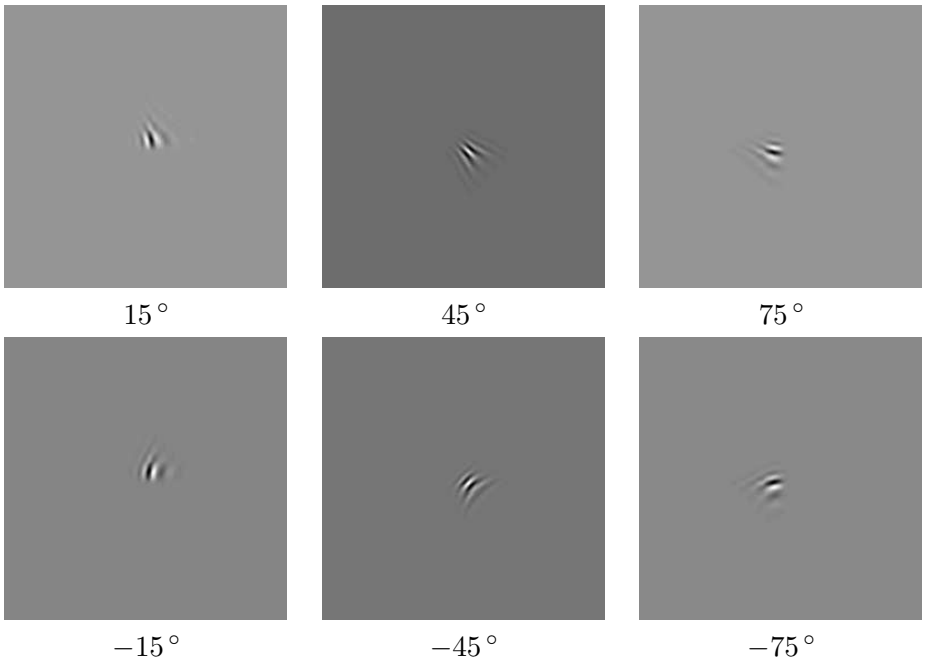


Figure 3.12.: Real impulse response of 2D complex filters for BW_3 at level four

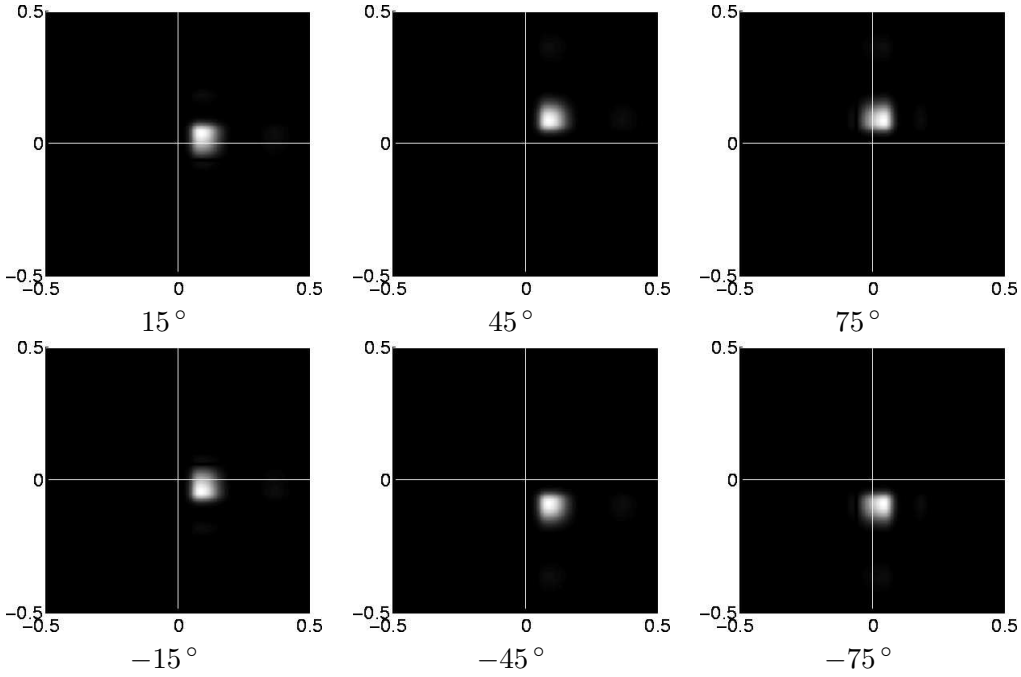


Figure 3.13.: Frequency response of 2D complex filters for BW_3 at level three

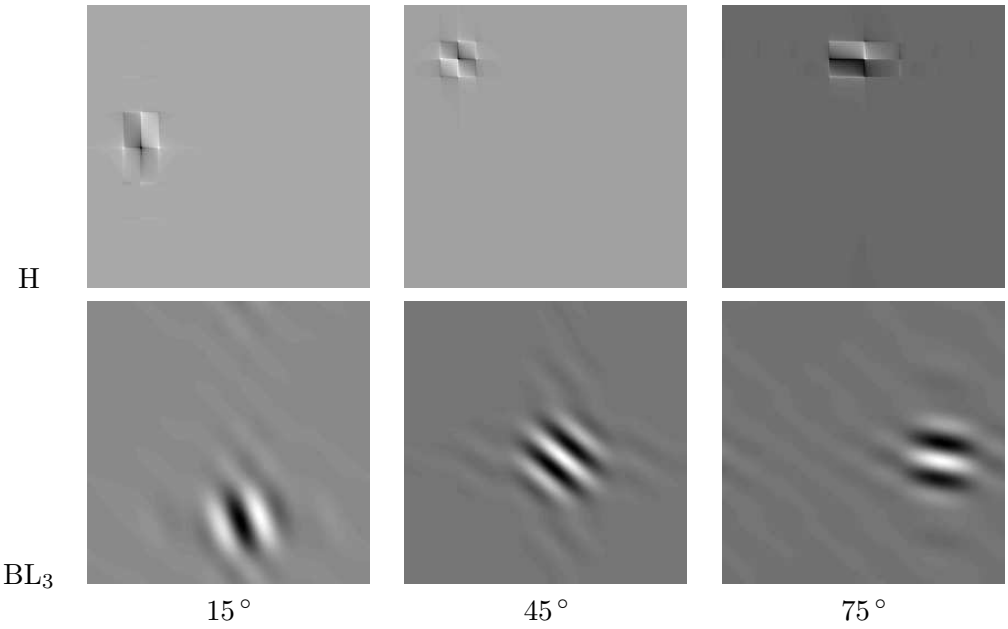


Figure 3.14.: Real impulse response of 2D complex filters at level five

sition in the frequency domain as also proposed by [Forster et al., 2003]. Note that this procedure imposes periodic boundary conditions which are very popular anyway. Moreover, we need (real) Fourier transforms for which there exist efficient implementations, e.g. [Frigo and Johnson, 1998].

We now explain the algorithm. Further information on wavelet transforms in the Fourier domain can be found, e.g., in [Plonka and Tasche, 1995]. Consider a given signal $\mathbf{s} = (s_0, s_1, \ldots, s_{N-1})$ of length N in the time domain. To express or actually circumvent the downsampling operation in the frequency domain, we rely on the polyphase representation [Strang and Nguyen, 1996, Chap. 4.2] of our quantities. After a z–transform, our signal then reads

$$S(z) = \sum_{j=0}^{N-1} s_j z^{-j} = S_0(z^2) + z^{-1} S_1(z^2) \ ,$$

where S_0 comprises the even powers of z and S_1 the odd ones by

$$S_0(z^2) = \frac{S(z) + S(-z)}{2} \ , \quad S_1(z^2) = \frac{S(z) - S(-z)}{2} z \ . \tag{3.20}$$

Similarly, we represent the filters for $i = 0, 1$ as

$$H_i(z) = \sum_{j=0}^{N-1} h_i[j] z^{-j} = H_{i0}(z^2) + z H_{i1}(z^2)$$

with

$$H_{i0}(z^2) = \frac{H_i(z) + H_i(-z)}{2} \quad , \quad H_{i1}(z^2) = \frac{H_i(z) - H_i(-z)}{2} z^{-1} \quad . \tag{3.21}$$

Concerning the filter bank analysis depicted in the left part of Fig. 3.2, we have for $i = 0, 1$ that

$$\begin{aligned} S(z)H_i(z) &= [S_0(z^2) + z^{-1}S_1(z^2)][H_{i0}(z^2) + zH_{i1}(z^2)] \ , \\ S(-z)H_i(-z) &= [S_0(z^2) - z^{-1}S_1(z^2)][H_{i0}(z^2) - zH_{i1}(z^2)] \end{aligned}$$

and further by (3.1)

$$\begin{aligned} C^1(z^2) &= S_0(z^2)H_{00}(z^2) + S_1(z^2)H_{01}(z^2) \ , \\ D^1(z^2) &= S_0(z^2)H_{10}(z^2) + S_1(z^2)H_{11}(z^2) \end{aligned}$$

so that

$$\begin{pmatrix} C^1(z^2) \\ D^1(z^2) \end{pmatrix} = \begin{pmatrix} H_{00}(z^2) & H_{01}(z^2) \\ H_{10}(z^2) & H_{11}(z^2) \end{pmatrix} \begin{pmatrix} S_0(z^2) \\ S_1(z^2) \end{pmatrix} \quad , \tag{3.22}$$

where the matrix on the right hand side of the equation is the polyphase matrix $\mathbf{H}_{\mathrm{pol}}(z^2)$ of the analysis filter bank.

Having an explicit decomposition formula in terms of the signal's polyphase components S_0 and S_1, we apply the decomposition on our discrete signal. We therefore calculate its discrete Fourier transform. Let $\widehat{\mathbf{x}}$ denote the Fourier transform of a signal $\mathbf{x}$. With $z := e^{2\pi ik/N} = w_N^{-k}$ this reads

$$\widehat{s}_k = S(e^{2\pi ik/N}) = \sum_{j=0}^{N-1} s_j e^{-2\pi ijk/N} \quad k = 0, \ldots, N-1 \ , \tag{3.23}$$

which requires $\mathcal{O}(N \log N)$ arithmetic operations with a real Fast Fourier Transform (FFT) of length N. For the polyphase components, it holds

$$S_0(z^2) = S_0\left(e^{2\pi ik/(N/2)}\right) = \widehat{s}_{0k} \overset{(3.20)}{=} \frac{1}{2}\left(S\left(e^{2\pi ik/N}\right) + S\left(e^{2\pi i(k+N/2)/N}\right)\right) \ ,$$

analogously for S_1. Thus they may be calculated in the frequency domain by

$$\widehat{s}_{0k} = \frac{1}{2}(\widehat{s}_k + \widehat{s}_{k+N/2}) \ , \quad \widehat{s}_{1k} = \frac{1}{2}e^{2\pi ik/N}(\widehat{s}_k - \widehat{s}_{k+N/2}) \ , \qquad k = 0, \ldots, \frac{N}{2} - 1 \ , \tag{3.24}$$

which requires N complex additions and $N/2$ complex multiplications. Finally, according to (3.22) the first decomposition step reads

$$\begin{pmatrix} C^1(e^{2\pi ik/(N/2)}) \\ D^1(e^{2\pi ik/(N/2)}) \end{pmatrix} = \begin{pmatrix} H_{00}(e^{2\pi ik/(N/2)}) & H_{01}(e^{2\pi ik/(N/2)}) \\ H_{10}(e^{2\pi ik/(N/2)}) & H_{11}(e^{2\pi ik/(N/2)}) \end{pmatrix} \begin{pmatrix} S_0(e^{2\pi ik/(N/2)}) \\ S_1(e^{2\pi ik/(N/2)}) \end{pmatrix} \tag{3.25}$$

for $k = 0, \ldots, N/2 - 1$ which requires $4N/2$ complex multiplications and $2N/2$ complex additions. For reconstructing the coefficients in the time domain, an inverse (real) FFT of length $N/2$ is necessary for both the low–pass and the high–pass coefficients.

Note that the polyphase components of the filters which appear in the polyphase matrix in (3.25) may be computed similar to S_0 and S_1: From (3.21) it follows for $i = 0$ and $k = 0, \ldots, N/2 - 1$, for example,

$$\begin{aligned} H_{00}\left(e^{2\pi ik/(N/2)}\right) &= \frac{1}{2}\left[H_0\left(e^{2\pi ik/N}\right) + H_0\left(e^{2\pi i(k+N/2)/N}\right)\right] \ , \\ H_{01}\left(e^{2\pi ik/(N/2)}\right) &= \frac{1}{2}e^{-2\pi ik/N}\left[H_0\left(e^{2\pi ik/N}\right) - H_0\left(e^{2\pi i(k+N/2)/N}\right)\right] \end{aligned}$$

or equivalently

$$\begin{aligned} \widehat{h}_{00}[k] &= \frac{1}{2}\left(\widehat{h}_0[k] + \widehat{h}_0\left[k + \frac{N}{2}\right]\right) \ , && k = 0, \ldots, \frac{N}{2} - 1 \ , \\ \widehat{h}_{01}[k] &= \frac{1}{2}e^{-2\pi ik/N}\left(\widehat{h}_0[k] - \widehat{h}_0\left[k + \frac{N}{2}\right]\right) \ , && k = 0, \ldots, \frac{N}{2} - 1 \ . \end{aligned}$$

In doing so, one has to take into account the N–periodisation of the shifted filters (3.14), (3.15) originally defined in the frequency range $[-1/2, 1/2)$, which for $j = 2, \ldots, J$ read

$$\begin{aligned} H_{0b}^j(e^{2\pi ik/N}) &= \begin{cases} e^{-\pi ik/N}H_0(e^{2\pi ik/N}) & k = 0, \ldots, \frac{N}{2} - 1 \ , \\ e^{-\pi i(k-N)/N}H_0(e^{2\pi ik/N}) & k = \frac{N}{2}, \ldots, N - 1 \ , \end{cases} \\ H_{1b}^j(e^{2\pi ik/N}) &= e^{\pi i[(k-N)/N+1/2]}H_1(e^{2\pi ik/N}) \\ &= -ie^{\pi ik/N}H_1(e^{2\pi ik/N}) \ , \quad k = 0, \ldots, N - 1 \ . \end{aligned}$$

Further steps $j = 2, \ldots, J$ are applied to the low–pass components C^{j-1} where the signal lengths halve at each step as

$$\begin{pmatrix} C^j(e^{2\pi ik/(N/2^j)}) \\ D^j(e^{2\pi ik/(N/2^j)}) \end{pmatrix} = \begin{pmatrix} H_{00}(e^{2\pi ik/(N/2^j)}) & H_{01}(e^{2\pi ik/(N/2^j)}) \\ H_{10}(e^{2\pi ik/(N/2^j)}) & H_{11}(e^{2\pi ik/(N/2^j)}) \end{pmatrix} \begin{pmatrix} C_0^{j-1}(e^{2\pi ik/(N/2^j)}) \\ C_1^{j-1}(e^{2\pi ik/(N/2^j)}) \end{pmatrix} \tag{3.26}$$

with $k = 0, \ldots, N/2^j - 1$. Note that the polyphase components $H_\cdot(e^{2\pi ik/(N/2^j)}) = \widehat{h}_\cdot[2^{j-1}k]$ are already known from previous steps. One can precompute the polyphase components or matrices. Once a precomputation for length N is done the values can be utilised for any signal of length $N/2^k$ where $k \in \mathbb{N}_0$.

If $N = 2^J$ and we make a full decomposition in the frequency domain, the cost of the algorithm to obtain Fourier transformed wavelet coefficients in terms of number of

complex multiplications reads

$$\begin{array}{ccccccccc} s & \xrightarrow{(3.23)} & S & \xrightarrow{(3.24),(3.25)} & C^1, D^1 & \xrightarrow{(3.24),(3.26)} & C^2, D^2 & \xrightarrow{(3.24),(3.26)} & \dots \xrightarrow{(3.24),(3.26)} C^J, D^J \ , \\ & & \frac{N}{2}\log_2 N & + \frac{N}{2} + 4\frac{N}{2} & & + \frac{N}{4} + 4\frac{N}{4} & & + \quad \cdots \quad + 1 + 4 & \\ & & & \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{5(\frac{N}{2}+\frac{N}{4}+\cdots+1)=5(N-1)} & & & & & \end{array}$$

i.e. $(N/2)\log_2 N + 5(N-1)$ total. As the signal's polyphase components S_0, S_1 are nothing but the sequences of even and odd numbered coefficients, respectively, one can instead also perform two FFTs of length $N/2$ on these two sequences, which reduces the complexity further. If the wavelet coefficients in the time domain are wanted, one additionally needs real inverse FFTs of lengths $N/2, N/4, \dots, 2$ requiring altogether $2^{J-2}\log_2 2^{J-1} + 2^{J-3}\log_2 2^{J-2} + \cdots + \log_2 2 = \sum_{j=1}^{J-1} 2^{j-1} j = 2^{J-1}(J-2) + 1 = (N/2)(\log_2 N - 2) + 1$ multiplications. Note that for our application in Sec. 3.9 in particular, this transform back is not necessary.

Concerning the reconstruction depicted in the right part of Fig. 3.2, we obtain

$$\begin{aligned} S_0(z^2) &\overset{(3.20)}{=} \frac{1}{2}[C^1(z^2)H_0(z^{-1}) + D^1(z^2)H_1(z^{-1}) \\ &\quad + C^1(z^2)H_0(-z^{-1}) + D^1(z^2)H_1(-z^{-1})] \\ &= \frac{1}{2}C^1(z^2)[H_0(z^{-1}) + H_0(-z^{-1})] + \frac{1}{2}D^1(z^2)[H_1(z^{-1}) + H_1(-z^{-1})] \\ &\overset{(3.21)}{=} C^1(z^2)H_{00}(z^{-2}) + D^1(z^2)H_{10}(z^{-2}) \ , \\ S_1(z^2) &\overset{(3.20)}{=} \frac{1}{2}zC^1(z^2)[H_0(z^{-1}) - H_0(-z^{-1})] + \frac{1}{2}zD^1(z^2)[H_1(z^{-1}) - H_1(-z^{-1})] \\ &\overset{(3.21)}{=} C^1(z^2)H_{01}(z^{-2}) + D^1(z^2)H_{11}(z^{-2}) \ . \end{aligned}$$

Consequently, the final reconstruction is simply the multiplication with $\mathbf{H}_{\text{pol}}^{\top}(z^{-2})$

$$\begin{pmatrix} S_0(z^2) \\ S_1(z^2) \end{pmatrix} = \begin{pmatrix} H_{00}(z^{-2}) & H_{10}(z^{-2}) \\ H_{01}(z^{-2}) & H_{11}(z^{-2}) \end{pmatrix} \begin{pmatrix} C^1(z^2) \\ D^1(z^2) \end{pmatrix} \ .$$

Further steps follow the same rule. Several modifications are possible to further reduce the arithmetic complexity of the algorithm.

3.8. Invariance Evaluation

In this section we examine the behaviour of the dual–tree wavelet transform in the frequency domain based on the standard wavelets proposed in Sec. 3.6 with respect to translation and rotational invariance.

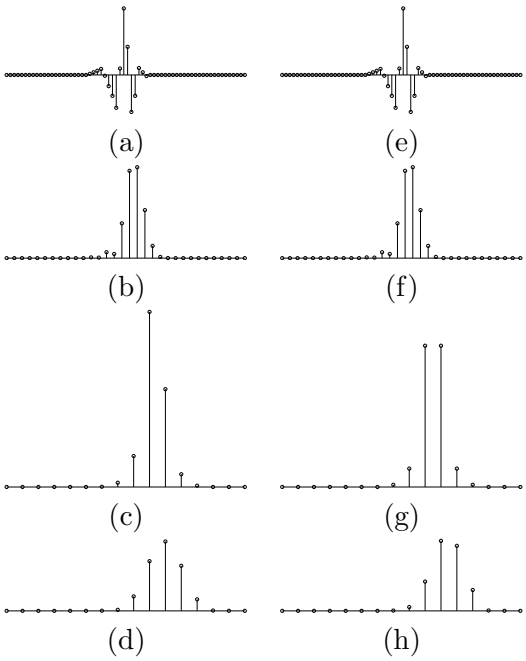


Figure 3.15.: Shift sensitivity of the complex wavelet transform in the frequency domain: (a) original signal equal to Fig. 3.1 (a), (b)–(d) magnitude of wavelet subband coefficients, (e) signal from (a) shifted by one sample, (f)–(h) magnitude of new wavelet subband coefficients

3.8.1. One-dimensional

First we give an illustration of the shift invariance properties. Figure 3.1 illustrates the shift dependence of the common discrete wavelet transform considering the D_3 wavelet as an example. For the same basis filter D_3, the dual–tree complex transform yields the coefficient magnitudes shown in Fig. 3.15. To reconstruct the coefficients in the time domain, we have to apply an inverse FFT as described in Sec. 3.7. As for Kingsbury's transform applied in Fig. 3.8 in Sec. 3.5.1, we are only able to plot the magnitude of the complex coefficients and the distribution of the coefficient energy across the subbands is almost equal for both signals in contrast to the real transform. Even the shape of the coefficient magnitude mainly stays the same; the coefficients provide interpolability as required by [Simoncelli et al., 1992]. As expected, the coefficient magnitudes on the first level come closest to shift invariance.

This illustration already indicates that also the complex wavelet transform in the frequency domain behaves more stable on signal shifts. To assess the influence of signal shifts more methodically, we compute the scatter of the channel energies which is the scatter of the features with ℓ_2-norm energy $\left\|D^j\right\|_2$ with respect to all signal shifts $z^k S(z)$

level / filter	1	2	3	4	5	6	7	8
H real	2.5e-01	2.5e-01	3.8e-01	6.9e-01	1.3e+00	2.7e+00	5.3e+00	5.3e+00
D_3 real	1.2e-02	4.1e-02	9.4e-02	1.8e-01	3.6e-01	6.1e-01	4.1e+00	4.1e+00
Kingsbury (9,7)/14-tap	1.6e-30	6.3e-04	5.3e-04	4.2e-04	1.0e-03	9.2e-04	4.0e-03	4.0e-03
H	1.1e-31	4.1e-02	5.5e-02	9.6e-02	1.8e-01	3.2e-01	3.2e-01	3.2e-01
D_3	2.2e-29	2.7e-03	4.4e-03	7.8e-03	1.5e-02	7.2e-02	6.4e-02	6.4e-02
BW_3	1.6e-29	4.3e-04	7.5e-04	1.4e-03	2.7e-03	8.4e-03	1.4e-02	1.4e-02
BW_{11}	5.1e-30	3.5e-11	7.0e-11	1.1e-10	1.6e-10	7.3e-11	1.1e-08	1.1e-08
BL_1	1.0e-29	1.7e-03	2.9e-03	5.3e-03	1.1e-02	3.1e-02	3.7e-02	3.7e-02
BL_2	3.2e-30	8.8e-05	1.6e-04	3.0e-04	5.9e-04	1.4e-03	4.0e-03	4.0e-03
BL_3	6.6e-30	5.7e-06	1.1e-05	2.0e-05	3.7e-05	6.2e-05	4.4e-04	4.4e-04

Table 3.4.: Energy scatter on shifts of the step signal

for $k = 1, \ldots, N$ for a step signal $\mathbf{s}$ of length $N = 256$. As the coefficients in each tree are real, the energy is $\|D_a^j + iD_b^j\|_2 = (\|D_a^j\|_2^2 + \|D_b^j\|_2^2)^{1/2}$. No FFT back from the frequency domain is necessary since, by the Parseval identity, the ℓ_2-norms of the wavelet coefficients in the time and the frequency domain coincide. Table 3.4 shows the shift variance for real (subsampled) transforms, Kingsbury's time domain dual–tree transform with 14-tap orthogonal filter and (9,7)-tap biorthogonal filters in the first step introduced in Sec. 3.5 and our complex sample filters. The coefficients for the complex and real transforms have the same magnitude as all filters H_{0a}, H_{1a} are normalised. Hence the scatter is comparable. Evidently, the constructed complex filter banks are all less sensitive to signal shifts than the real transforms. Similar to the results in Table 3.1, the scatter even gets close to zero which is the energy variance for the non-subsampled transform. At level one of the complex transform, the variance is zero. The shift invariance improves with the order, and, therewith, the number of vanishing moments of the Butterworth and Battle–Lemarié wavelets as property (3.5) is more closely satisfied. Both wavelet filters of order three are at least comparable to Kingsbury's large filter with respect to shift invariance. Note that the channel energies are used in our signal classification application so that it is important that they do not heavily depend on the signal alignment.

To quantify the effect of the aliasing causing the shift dependence more generally, we determine the aliasing energy ratio as done by [Kingsbury, 2001]. Considering (3.9), we have already observed that the aliasing terms are the summands containing $S(w_{2^j}^k z)$ for $k \neq 0$. Hence we determine the *aliasing energy ratio* for the jth subband as

$$R_{\text{alias}} = \frac{\sum_{k=1}^{2^j-1} \left\| B_a^j(w_{2^j}^k z) B_a^j(z^{-1}) + B_b^j(w_{2^j}^k z) B_b^j(z^{-1}) \right\|_2^2}{\left\| B_a^j(z) B_a^j(z^{-1}) + B_b^j(z) B_b^j(z^{-1}) \right\|_2^2} \ ,$$

where a filter $H(z)$ is regarded as a function $\omega \mapsto H(e^{2\pi i\omega}) \in L_2([-1/2, 1/2))$. Table 3.5

	low–pass					high–pass				
level	1	2	3	4	5	1	2	3	4	5
filter										
H real	- 4.77	- 3.42	- 3.11	- 3.04	- 3.02	- 4.77	1.09	2.51	2.88	2.98
D_3 real	- 7.64	- 7.43	- 7.42	- 7.42	- 7.42	- 7.64	- 1.51	- 1.30	- 1.29	- 1.29
Kingsbury (9,7)/ 14-tap	$-\infty$	-23.19	-29.33	-28.56	-28.57	$-\infty$	-21.81	-18.96	-24.85	-24.15
H	$-\infty$	- 9.80	- 8.71	- 8.47	- 8.41	$-\infty$	- 7.84	- 3.75	- 2.96	- 2.78
D_3	$-\infty$	-18.83	-18.83	-18.82	-18.82	$-\infty$	-17.23	-13.10	-13.09	-13.08
BW_3	$-\infty$	-27.16	-27.09	-27.09	-27.09	$-\infty$	-26.08	-21.63	-21.58	-21.58
BW_{11}	$-\infty$	-92.81	-92.82	-92.87	-92.34	$-\infty$	-92.55	-87.78	-87.78	-87.50
BL_1	$-\infty$	-22.35	-22.11	-22.09	-22.09	$-\infty$	-21.05	-16.75	-16.60	-16.59
BL_3	$-\infty$	-44.16	-44.14	-44.15	-44.15	$-\infty$	-43.55	-38.85	-38.83	-38.80

Table 3.5.: Aliasing energy ratio $10 \log_{10} R_{\text{alias}}$ in dB of wavelet transforms at levels one to five

summarises the aliasing energy ratios up to level five, where the results for Kingsbury's filters are adopted from [Kingsbury, 2001, Table 3]. The ratio is significantly lower for all combined complex transforms shown in the lower part of the table; it is zero at level one in particular. The Haar filter and its dual–tree version exhibit a high aliasing because it is badly localised and does not fulfil condition (3.5). On the other hand, again BW_m and BL_m get of course less shift dependent as their order m increases. We can conclude that all constructed filters with appropriate support property perform well so that we derived a general design method for shift invariant complex filters with perfect reconstruction.

3.8.2. Two-dimensional

To illustrate the behaviour for the two-dimensional transform with respect to rotational invariance, we show the contributions of the levels $S_l^4, S_h^4, \ldots, S_h^1$ for a rotationally symmetric image in Fig. 3.16. The illustration is similar to the one in Fig. 3.9, only that in two dimensions, each high–pass component comprises the information of the appropriate six directional subbands, or two subbands for the final low–pass component. Again we also apply the D_3 filter and the dual–tree complex 6-tap orthogonal filter of original length eight having approximately the same lengths as well as complex filters in the frequency domain.

As a result, the fully decimated real wavelet transform in the second row shows heavy blocky artifacts and aliasing. The complex transforms in the last two rows look a lot better, especially for the same basis filter D_3, but still the resulting reconstruction components are not rotationally invariant; see, e.g., the high–pass components at levels two and three. According to [Kingsbury, 2001], this may result from the two diagonal subbands having higher centre frequencies than the other four. We note that the non-subsampled

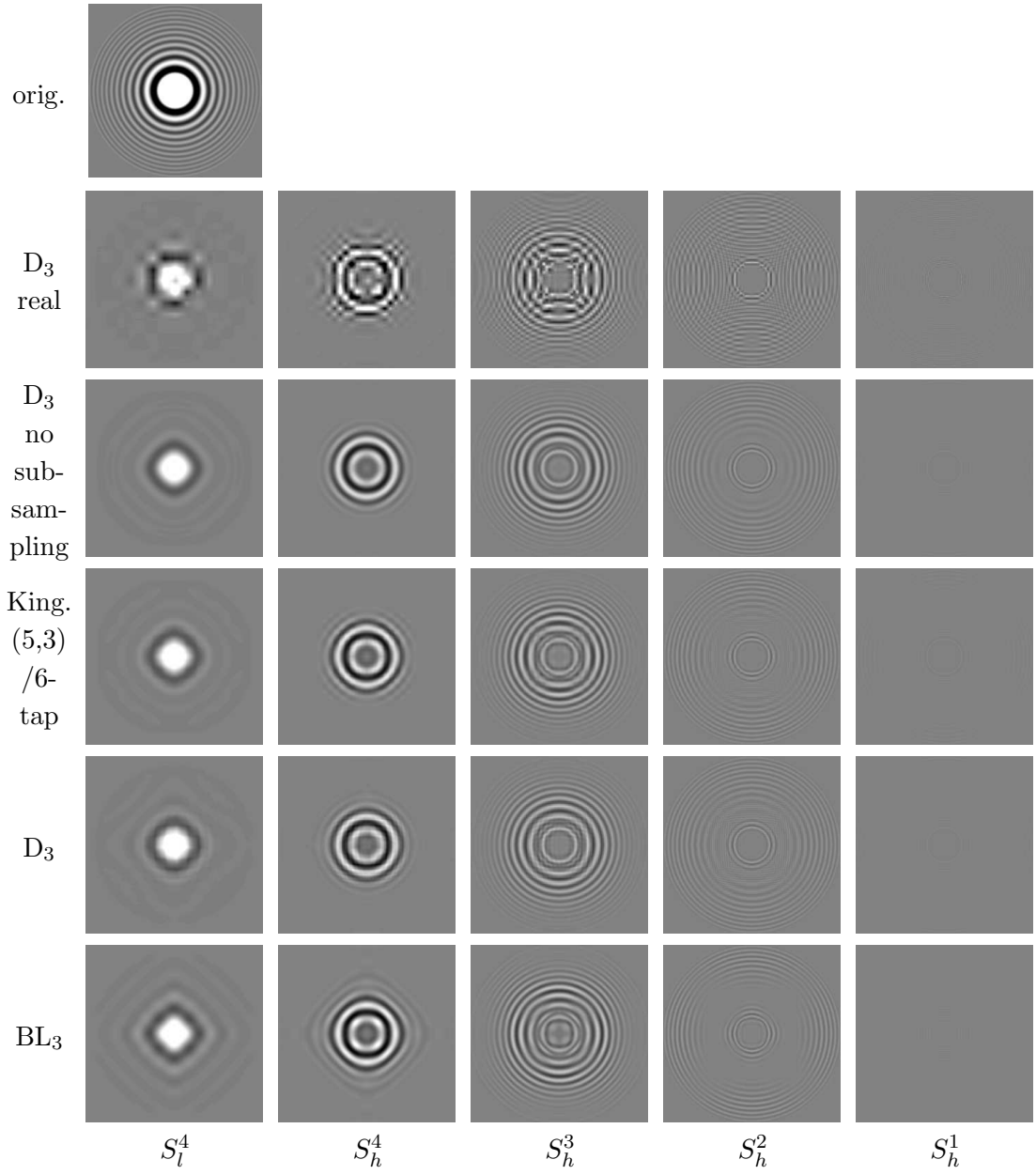


Figure 3.16.: Subband information of real and complex wavelet transforms in 2D

as well as the complex wavelet transform both show no true rotational invariance: The level four wavelet components in the second column rather show a diamond shape than a circle.

3.9. Application to Signal Classification

We investigate the classification performance of a wavelet–SV classifier with these complex wavelets in the frequency domain. We intend to classify two different types of data, namely physiological heart patient data as in [Strauss and Steidl, 2002] and texture image rows as described in [Neumann et al., 2002]. The problems 'heart5', 'heart6' and 'heart7' denote the detection of ventricular tachycardia with real patient data for three different patients. The training data are eight beats of normal heart activity and eight beats with induced ventricular tachycardia, 32 further beats are used for testing. Typical examples of curves from the two classes we want to distinguish are shown in Fig. 1.1 in the introduction.

The problem 'misc2 – misc3' is a texture row classification problem. The first 32 rows of each image are used as training data, the rest is used for evaluating the classification error. In contrast to the heart data, the typical curves depicted in Fig. 1.2 have a bad localisation in the time domain. All signals consist of 512 values, so full decomposition leads to $J = 9$ decomposition levels.

The classification setup is summarised in Chap. 2, where the single steps are described in more detail in Secs. 2.2 and 2.3. For the wavelet transform in the feature extraction step we likewise apply critically sampled, non-subsampled or dual–tree complex transforms here and use the weighted ℓ_2-norm for the energy computation. Note that energy operators based on the ℓ_2-norm, by the Parseval identity, have the advantage that no transform of the wavelet coefficients back from the frequency to the time domain is necessary. For feature extraction, we additionally examine the ideal filter with support property $\operatorname{supp} H_0(e^{2\pi i\omega}) = [-1/4, 1/4]$ which can be implemented by the Fourier transform.

The classification results are given in Table 3.6. In the 'original' problem versions, the data are separated into a training and a test set as described above to evaluate the classification performance. In the 'trn shifted' problem versions, the test data instead consists of all distinct shifts of all training signals to evaluate shift invariance. Finally, in the 'tst shifted' versions, we use again the disjoint training and test sets, but the test set contains all shifts of all test signals too. For 'trn shifted' and the transform without subsampling, no results are given because the feature test vectors to classify are just the same as the training vectors. Consequently, for a hard margin SVM that classifies the training data correctly, the test error has to be zero. Again we include adapted wavelets as for the two-dimensional signal classification in Sec. 3.5.2. [Kingsbury, 2001] claims that for larger filters, smoother wavelets may be obtained and shows that the

filter	problem	misc2 – misc3 'original'	misc2 – misc3 'trn shifted'	misc2 – misc3 'tst shifted'	heart5 'original'	heart5 'trn shifted'	heart5 'tst shifted'	heart6 'original'	heart6 'trn shifted'	heart6 'tst shifted'	heart7 'original'	heart7 'trn shifted'	heart7 'tst shifted'
critically sampled	H	44	45	50	19	38	42	0	40	42	3	42	47
	D_3	49	34	36	9	41	36	6	55	54	13	23	30
	adapted	42	39	46	6	46	41	3	43	45	0	43	46
no subsampling	H	24	-	24	0	-	0	38	-	38	22	-	22
	D_3	24	-	24	0	-	0	6	-	6	16	-	16
	adapted	24	-	24	0	-	0	38	-	38	22	-	22
Kingsbury	(5,3)/6-tap	23	0	25	0	3	1	25	16	26	19	0	18
	(9,7)/14-tap	31	0	32	0	0	0	13	11	19	19	0	19
complex (frequency domain)	H	52	26	40	0	30	20	6	38	44	6	15	30
	D_3	21	3	25	0	11	6	44	28	40	16	0	18
	BW_3	25	0	26	0	0	0	19	15	21	19	0	22
	BW_{11}	16	0	16	0	0	0	13	0	13	25	0	25
	BL_1	24	0	26	0	2	1	34	26	30	16	0	17
	BL_2	24	0	24	0	0	0	9	0	30	19	0	24
	BL_3	22	0	22	0	0	0	13	0	12	25	0	25
	adapted	38	26	34	25	43	35	3	51	51	28	39	53
Fourier (ideal filter)		0	-	0	0	-	0	13	-	13	50	-	50

Table 3.6.: Classification error [%] of a signal classifier for different filters

aliasing can be further reduced by using longer filters. Hence we also include the 14-tap orthogonal filters presented in Sec. 3.5.

As a result, expectedly, the fully decimated wavelet transform is not able to well discriminate between the signal classes, particularly for the translated heart signals. In this case, the wavelet adaptation is also questionable: Obviously, the wavelets are only adapted to the specifically aligned signals and do not generalise well for translated test data, even if they perform well on equally aligned test data for the 'original' problem versions. The frequency domain wavelet features are as discriminatory as those of Kingsbury's wavelets. The resulting classification error is significantly lower than for the critically sampled real transform and also comparable to the non-subsampled real transform. But the most important observation is that the complex transforms achieve approximate shift invariance. The classification error for the 'shifted' problems is definitely lower than that of the critically sampled real transform. Moreover, the classification errors with the complex filters are comparable to those of the computationally expensive totally translation invariant transform without subsampling. It is important

to remark that all directly constructed wavelets perform well so that we derived a general wavelet design method for shift invariant complex wavelets with perfect reconstruction. The relatively high error rate of 11% for the complex D_3 wavelet and 'heart5 trn shifted', e.g., is due to some outliers in the small training set and the fact that this wavelet is more sensible to signal shifts according to Table 3.4. The higher sensitivity is also visible in the principal components plot of the test vectors.

As the data for problem 'misc2 – misc3' show a highly periodic structure, the classification error decreases with the filter order and translation invariance is an important issue. Here the Fourier features perform best. For 'heart5', evidently, classification performance is highly dependent on the shift sensitivity so that the shift insensitive transforms perform well. The two further 'heart' problems are more complicated: Different filters, here, e.g., H and BL, are most successful whereas the Fourier features fail completely for 'heart7'.

In view of these properties, an extension to the classification application is to adapt the complex filters to the classification problem, which means to the data and the classifier at hand. Trying to adapt the basis filters to obtain most discriminative features, we observe that the wavelets producing a large class centre distance in the critically sampled case do not provide the same when combined to complex wavelets and vice versa, so that the criterion plots may look different. The adapted wavelets obtained up to now do not take into account the localisation property necessary to provide near translation invariance. That is probably the reason why the adaptation performs so poorly, which gets apparent by the high error for the 'shifted' problem versions.

3.10. Summary and Conclusions

We have worked out and applied Kingsbury's idea of dual–tree filter banks in the frequency domain where it can be based on standard wavelets. Concerning translation and rotational invariance these complex transforms behave much better than their critically sampled counterparts and show a performance at least as good as Kingsbury's specially designed filters. The advantage of our approach is that it provides a general construction method to obtain various filters. Of course our computation in the frequency domain involves (real) FFTs such that with respect to the arithmetic complexity it can only compete with real filter banks in the time domain involving filters of moderate length. We applied dual–tree filter banks in the feature extraction step of the classification problem. Feature extraction and subsequent classification may benefit from an appropriate adaptation of the wavelet to the problem at hand as also argued in Chap. 4.

Many further applications to complex nearly translation invariant filter banks exist, many of them also mentioned by Kingsbury:

- The phase information of the complex coefficients can be used for motion estimation [Kingsbury and Magarey, 1997].

- Because of the shift invariance of the subband energy, the filters well apply to texture synthesis [Kingsbury, 1998] and retrieval [de Rivaz and Kingsbury, 1999].
- Another application that makes use of the shift invariance and the better directional resolution of the filters is denoising [Kingsbury and Magarey, 1997]. Own experiments also show that complex filters achieve smoother output with fewer blocky artifacts similar to the illustration in Fig. 3.16.

That votes for a whole library of dual–tree complex wavelet filter banks which is available based on known wavelet filters only if we work in the frequency domain as proposed in this chapter.

4. Wavelet Adaptation

4.1. The Adaptation Problem

A persistent problem in signal and image classification concerns filter design for feature extraction and selection addressed by [Randen and Husøy, 1999, Scheunders et al., 1998, Unser, 1995]. As the signal types vary as much as from cardiac signals to texture images, different waveforms are encountered in the classification problem. In most cases, the filter design problem is addressed *irrespective of* the subsequent classification stage, which may result in an unacceptably large classification error. In contrast, we are interested in an approach that takes the target classifier and data into consideration for filter design and the selection of appropriate features. A hybrid architecture was introduced in Chap. 2 consisting of a wavelet feature extraction and an SV classifier applied to the resulting feature vectors.

As already shown by [Strauss and Steidl, 2002] for various applications, the classification error depends on the filters used in the wavelet transform and *jointly* designing both the filter stage and the classifier may considerably outperform standard approaches based on a *separate* design of both stages. [Jones et al., 2001] also claim that a problem specific wavelet choice is promising. Besides, adapting the wavelet has proved advantageous as well in other applications such as audio coding [Sathidevi and Venkataramani, 2002]. In general, it is desirable to adapt the preprocessing or some classifier parameters to the specific classification problem. In contrast to best basis methods [Saito, 1994], the wavelet itself was adapted by [Strauss and Steidl, 2002] while the structure of the basis remained fixed [Strauss et al., 2003]. However, although there exist more sophisticated measures for estimating the classification ability of training sets, only the simple class centre distance was used to adapt the feature extraction step, i.e., the wavelet filter, to the subsequent SV classifier.

This motivates our investigation of suitable adaptation criteria. As summarised in [Neumann et al., 2002, Neumann et al., 2003b, Neumann et al., 2005b], we address the problem of how to choose an orthogonal compactly supported wavelet to optimally preprocess signals for binary classification. For that purpose, several criteria to judge the discrimination ability of a set of feature vectors are presented and closely examined.

Adequate adaptation criteria can obviously be obtained from the classifier's objectives and derived classification error bounds. Besides, criteria used to select the soft margin SVM regularisation parameter C or kernel parameters [Cristianini et al., 2002] may be applied for wavelet adaptation. These are often also error bounds [Chapelle et al., 2002,

Chung et al., 2003, Duan et al., 2001, Schölkopf et al., 1999b]. The most frequently applied error bound for SV classifiers is the radius – margin bound (see [Vapnik, 1995]), where the margin is the objective of the SV classification problem. We show that, for a class of kernels, the radius of the smallest sphere enclosing the feature vectors, the second quantity used in the bound, can be computed by solving another standard SV problem again. This bound is, for example, successfully applied for feature selection by [Weston et al., 2001] with gradient descent methods.

But since we take filter optimisation into account, we have to deal with more complex objective functions. So despite the computational convenience coming from our reduction of the radius computation problem, this method is not applicable here: It still includes repeated QP minimisation, which is computationally expensive because wavelet adaptation criteria typically have many local minima and hence need to be evaluated for many different parameter values. This brings up the problem of finding reliable criteria that are still fast to evaluate to rank a given feature set.

We compare five simple criteria in the example of one-dimensional signal classification by wavelet features. Our experiments show that there exist simple criteria that well approximate the classification error, assessed by the radius – margin error bound. Applied to our wavelet adaptation problem, these criteria establish an easy way to find the wavelet that best discriminates the signal classes.

Our results apply to the two-, and arbitrary-dimensional setting as well by utilising the standard tensor product design of wavelets. The resulting wavelet features may then be used to analyse texture in images, for instance (cf. [Arivazhagan and Ganesan, 2003, Scheunders et al., 1998, Portilla and Simoncelli, 2000]). The proposed criteria can also be used for feature selection discussed in Chap. 5, which aims at discarding features from a predetermined set.

So relying on the wavelet–SVM architecture introduced in Chap. 2, we discuss the range of criteria that approximate the generalisation error in Sec. 4.2. There, we also provide the theorem that simplifies the computation of the radius – margin error bound and that may also be interesting in other contexts. Section 4.3 contains a thorough numerical evaluation of the proposed criteria.

After selecting an appropriate easy to evaluate criterion for the wavelet adaptation, we still have to search in the parameter space $\mathcal{P}_L$ for the angle vector $\boldsymbol{\theta}$ optimising this criterion. Up to now, in previous work [Strauss and Steidl, 2002], the computation of the optimal filters was expensive even with the proposed genetic algorithm [Strauss et al., 2003]. Alternative approaches for efficient criteria optimisation are summarised in [Neumann et al., 2003a] and reviewed next.

First we formulate the wavelet adaptation as a continuous optimisation problem in terms of the filter coefficients in Sec. 4.4. Then we try to solve this problem or a constrained variant with various optimisation techniques, which is summarised in Sec. 4.5.

4.2. Possible Criteria: SVM Class Separability

We consider the task of having to rate sets of labelled feature vectors $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\}$ for $\mathcal{X} \subset \mathbb{R}^d$. For our feature extraction especially, all feature vectors $\mathbf{x}_i$ lie in or even on a sphere centred at the origin. To steer our feature extraction process via the parameters $\boldsymbol{\theta}$ such that the subsequent SVM performance becomes optimal we need a criterion that

- measures the *generalisation error* $\mathrm{err}(f)$ of the SVM, i.e., the probability that $\mathrm{sgn}(f_{\mathbf{w}^*}(\mathbf{x})) \neq y$ for a randomly chosen example $(\mathbf{x}, y) \in \mathcal{X} \times \{-1, 1\}$, and
- can be efficiently evaluated for different sets of $\boldsymbol{\theta}$-dependent feature vectors.

Although there exist many proven bounds for the error risk or its expectation in the literature (see, e.g., [Chapelle et al., 2002, Herbrich, 2002]), in essence, most of them rely either on the number of SVs (see [Vapnik, 1995, Theorem 5.2], [Floyd and Warmuth, 1995] and [Herbrich, 2002, Chap. 5.2.1]) or on the size of the margin ρ separating the classes normalised by a measure of the feature vector variation such as their radius as in [Vapnik, 1998, Theorem 10.6], [Herbrich and Graepel, 2001]. In the following we start with this group of criteria. However, although they match the first requirement they do not fulfil the second one: Minimising the error bound is equivalent to maximising the margin resp. minimising the number of SVs. Unfortunately, both objectives imply solving a QP which, for our purpose, is impracticable. Besides, the resolution of error bounds relying on the number of SVs is too low. Additional quantities used in error bounds are, for example, the eigenvalues of the kernel matrix in [Schölkopf et al., 1999b] or the normalised margin in [Herbrich and Graepel, 2001]. However, they suffer from the same computational drawback as do margin and number of SVs. Moreover, many bounds are not tight, e.g. the stability bound [Bousquet and Elisseeff, 2001, Herbrich, 2002]. At the worst, if the bound's value is above one, one cannot say that a decrease improves the expected classifier performance as there is no conclusion possible at all. This motivates to evaluate the performance of simplified criteria which can be more efficiently evaluated.

In the following, we present the criteria and indicate some properties and relationships between them. In our experiments we investigate five criteria: the radius – margin bound, the margin, the alignment, the class centre distance and the generalised Fisher criterion:

Radius – Margin Let the margin ρ be given by (2.30). Further let R be the radius of the smallest sphere in ℓ_2 enclosing all $\boldsymbol{\phi}(\mathbf{x}_i)$, i.e., the solution of

$$\begin{aligned} &\min_{\mathbf{a} \in \mathcal{F}_K,\, R \in \mathbb{R}} R^2 \\ &\text{subject to} \quad \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 \,, \quad i = 1, \ldots, n \ . \end{aligned} \tag{4.1}$$

Then according to [Vapnik, 1998, Theorem 10.6], the expectation of the quotient

$$\mathcal{C}_1(\boldsymbol{\theta}) := \frac{1}{n}\frac{R^2}{\rho^2} \tag{4.2}$$

forms an upper bound on hard margin SVMs' generalisation error, where expectation is meant over all training sets of equal size n assuming the same underlying distribution. Therefore we consider a minimal value of $\mathcal{C}_1$ as the ultimate criterion for a hard margin SV classifier.

At first glance the computation of ρ and R in $\mathcal{C}_1$ requires the solution of two structurally different optimisation problems (2.29) and (4.1). Fortunately, by the following theorem both ρ and R can be obtained by the same kind of QP (2.34). This is indeed profitable since for standard SV problems (2.34), sophisticated algorithms are available in many implementations as, e.g., SVM*light* [Joachims, 1999].

Theorem 4. *Let K be a kernel with corresponding feature map $\boldsymbol{\phi}$ and $K(\mathbf{x},\mathbf{x}) = \kappa$ for all $\mathbf{x} \in \mathcal{X}$. Then the optimal radius R in (4.1) can be obtained by solving (2.34) with $\mathbf{Y} = \mathbf{I}$ and $C = \infty$. With $\boldsymbol{\alpha}$ being the solution of (2.34) and i an index of a SV, $R^2 = \kappa + \boldsymbol{\beta}^\top \mathbf{K}\boldsymbol{\beta} - 2(\mathbf{K}\boldsymbol{\beta})_i$, where $\boldsymbol{\beta} := (\mathbf{e}^\top\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}$.*

The proof of the theorem, which also reveals an interesting relation to the SV problems used for clustering and novelty detection, is given in Appendix A.

Note that in the soft margin case, there also exists a radius – margin bound. According to [Duan et al., 2001], the expectation of the generalisation error of the SVM is bounded from above by the expectation of the term

$$\frac{1}{n}\left(4R^2\sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\xi_i\right) \ ,$$

where $\boldsymbol{\alpha}$ is again the solution of problem (2.34) and $\xi_i := (1 - y_i f(\mathbf{x}_i))_+$ is the resulting error term in the primal problem (2.27).

Due to property (2.18), if the input signals are normalised, the radius R is bounded. Consequently, the margin or the soft margin minimisation functional by themselves provide an error bound and a justified criterion.

Anyway, the computation of the bound still requires the solution of two QPs of the form (2.34) for each considered parameter vector $\boldsymbol{\theta}$. As the optimal wavelet can only be found by search heuristics due to the complexity of the feature extraction process, i.e., the multi-level wavelet transform and energy computation, and due to the resulting non-convex objective function, the radius – margin is a time-consuming criterion. So we look for simpler criteria. But as the bound is relatively tight to the error (most evaluations at least lead to bounds below the trivial 1/2,

see Sec. 4.3), we selected it for comparison as a representative for all error bounds that are close to the generalisation error but are too computationally intensive for feature adaptation.

Margin Due to (2.18), the radius R is bounded. This motivates to consider only the denominator of (4.2), i.e., to use a maximal

$$\mathcal{C}_2(\boldsymbol{\theta}) := \rho$$

as an objective criterion. Besides, for the hard margin case, the margin ρ itself (obtained by equation (2.37) from the solution of the optimisation problem (2.34) in Sec. 2.3) as the SVM objective criterion may be a first guess for a criterion for the wavelet choice. Indeed, our experiments indicate that if training and test data have the same underlying distribution, the margin behaves much like the classification error. The major disadvantage of taking the margin as adaptation criterion is that every examined wavelet still requires the solution of one QP. Furthermore, the size of the margin depends only on few data points, precisely on the SVs. Thus, the size of the margin is not a 'smooth' function of all input vectors.

The same main drawback, the complexity of the evaluation, holds for the soft margin optimisation criterion $C \sum_{i=1}^{n} \xi_i + \|\mathbf{w}\|^2_{\mathcal{F}_K} /2$, the equivalent of the margin, even though the optimisation functional in the soft margin case is smoother because of the limited influence of single points (cf. the dual constraint $\boldsymbol{\alpha} \leq C\mathbf{e}$ in (2.34)).

Alignment In [Cristianini et al., 2002, Lanckriet et al., 2002] the sample alignment

$$\widehat{A}(\mathbf{K}_1, \mathbf{K}_2) := \frac{\langle \mathbf{K}_1, \mathbf{K}_2 \rangle_{\mathrm{F}}}{\|\mathbf{K}_1\|_{\mathrm{F}} \, \|\mathbf{K}_2\|_{\mathrm{F}}}$$

with Frobenius inner product $\langle \cdot, \cdot \rangle_{\mathrm{F}}$ and corresponding norm $\|\cdot\|_{\mathrm{F}}$ was proposed as a measure of conformance between kernels. It is also used for tuning kernel parameters. Especially, the kernel matrix $\mathbf{y}\mathbf{y}^\top$ is viewed as the optimal kernel matrix for two-class classification. This leads to maximising the criterion

$$\mathcal{C}_3(\boldsymbol{\theta}) := \widehat{A}(\mathbf{K}, \mathbf{y}\mathbf{y}^\top) = \frac{\langle \mathbf{K}, \mathbf{y}\mathbf{y}^\top \rangle_{\mathrm{F}}}{\|\mathbf{K}\|_{\mathrm{F}} \, \|\mathbf{y}\mathbf{y}^\top\|_{\mathrm{F}}} = \frac{\mathbf{y}^\top \mathbf{K} \mathbf{y}}{n \, \|\mathbf{K}\|_{\mathrm{F}}} \tag{4.3}$$

which, by the inequality of Cauchy–Schwarz, only takes values in $[0, 1]$ as $\mathbf{K}$ is always positive definite.

As an extreme case, if σ tends to zero for the Gaussian kernel (2.20), it follows $\mathbf{K} \approx \mathbf{I}$ and hence $\widehat{A}(\mathbf{K}, \mathbf{y}\mathbf{y}^\top) = (\|\mathbf{I}\|_{\mathrm{F}})^{-1} = 1/\sqrt{n}$. Although there is no information available about the feature vectors any more, the alignment is relatively high.

A borderline case of an SVM is the *Parzen window estimator*. Note that by [Cristianini et al., 2002, Theorem 4], the generalisation accuracy of the expected Parzen window estimator is bounded by a function of the alignment:

$$\operatorname{err}(f) \leq 1 - \widehat{A}(\mathbf{K}, \mathbf{y}\mathbf{y}^\top) + \widehat{\epsilon} + \frac{1}{\|\mathbf{K}\|_{\mathrm{F}}}$$

with probability greater than $1-\delta$, where $\widehat{\epsilon}$ is a function of the sample and the level of significance δ. The parameter δ and the term $\widehat{\epsilon}$ are only needed because the sample alignment is used instead of its expected value. When using the true alignment $A(k_1, k_2) := \langle k_1, k_2 \rangle_P (\langle k_1, k_1 \rangle_P \langle k_2, k_2 \rangle_P)^{-1/2}$ where the inner product is defined as $\langle f, g \rangle_P := \int_{\mathcal{X}^2} f(\mathbf{x}, \mathbf{z}) g(\mathbf{x}, \mathbf{z}) dP(\mathbf{x}) dP(\mathbf{z})$, according to [Cristianini et al., 2002] the bound even simplifies to $\operatorname{err}(f) \leq 1 - A(k, t(\cdot)t(\cdot))$ (where t is the target function defined in Sec. 2.3.1). This shows that the alignment is directly related to the expected Parzen window estimator. [Cristianini et al., 2002] claim that, as the empirical Parzen estimator is concentrated, its generalisation is described by the empirical alignment $\widehat{A}$ as well. The Parzen window estimator is equivalent to a soft margin SVM with bias term and minimal outlier penalisation parameter $C = 1/n$. This establishes the choice of the alignment as an adaptation criterion especially for soft margin SVMs. As we show in the next section, the alignment reliably predicts the margin for our hard margin SV problems without outliers as well.

Class Centre Distance According to all our experiments, the denominator $n \|\mathbf{K}\|_{\mathrm{F}}$ in (4.3) doesn't influence the alignment much for the Gaussian kernel with a fixed kernel width. The alignment is governed by its numerator $\langle \mathbf{K}, \mathbf{y}\mathbf{y}^\top \rangle_{\mathrm{F}}$ that varies about 200% whereas its denominator only varies about 20% for different wavelets. This may result from the norm preservation (2.18) which implies that the kernel matrix is more or less normalised. For normalised training vectors $\|\mathbf{x}_i\| = c$ for $i = 1, \ldots, n$ as guaranteed by using the ℓ_2-norm for the energy computation, the Gaussian kernel reads

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)} \\ &= e^{-(\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle)/(2\sigma^2)} \\ &= e^{-c^2/\sigma^2} e^{\langle \mathbf{x}_i, \mathbf{x}_j \rangle/\sigma^2} \quad , \quad i, j = 1, \ldots, n \ , \end{aligned}$$

where the first term is constant and the second one is just the exponential of the linear kernel. First, the exponential e^x is a monotone function of x, and second, it can be approximated for small $x < 1$ by $1 + x$ according to its Taylor series

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + x^2 \sum_{n=0}^{\infty} \frac{x^n}{(n+2)!} \ .$$

Thus, if σ is large, which means that the exponent is small, the linear approximation is close to the exponential. This implies that the alignment with the Gaussian kernel is related to the alignment with the linear kernel

$$K_{\text{linear}}(\mathbf{x}, \mathbf{y}) := \langle \mathbf{x}, \mathbf{y} \rangle \ , \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$$

with feature map $\boldsymbol{\phi}_{\text{linear}} := \text{id}$, or to $\langle \mathbf{D}, \mathbf{y}\mathbf{y}^\top \rangle_{\text{F}}$ where $\mathbf{D} := (\langle \mathbf{x}_i, \mathbf{x}_j \rangle)_{i,j=1}^n$ is the analogue of $\mathbf{K} = (\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{F}_K})_{i,j=1}^n$ with respect to the linear kernel. Motivated by the alignment's relation to linear quantities, we want to look at criteria in the original data space $\mathbb{R}^d$. Denote the class means by $\boldsymbol{\mu}_i := (\sum_{y_j=i} \mathbf{x}_j)/n_i$ with class cardinalities $n_i := |\{j : y_j = i\}|$ for $i = \pm 1$. [Strauss and Steidl, 2002] successfully applied as an adaptation criterion the distance of the two class centres in the Euclidean data space $\mathbb{R}^d$

$$\mathcal{C}_4(\boldsymbol{\theta}) := \left\| \boldsymbol{\mu}_1 - \boldsymbol{\mu}_{-1} \right\| \ .$$

For an equal number of training vectors for both classes $n_1 = n_{-1}$, the criterion is equivalent to

$$\begin{aligned}
\mathcal{C}_4^2 &\propto \left\| \sum_{\{i:y_i=1\}} \mathbf{x}_i - \sum_{\{i:y_i=-1\}} \mathbf{x}_i \right\|^2 \\
&= \left\| \sum_{i=1}^n y_i \mathbf{x}_i \right\|^2 \\
&= \left\langle \sum_{i=1}^n y_i \mathbf{x}_i, \sum_{i=1}^n y_i \mathbf{x}_i \right\rangle \\
&= \sum_{i,j=1}^n y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\
&= \langle \mathbf{D}, \mathbf{y}\mathbf{y}^\top \rangle_{\text{F}} \ .
\end{aligned}$$

The criterion $\mathcal{C}_4$ is thus equivalent to the alignment's numerator in data space. In effect, it approximates the alignment in data space and even for the Gaussian kernel.

As argued in [Strauss and Steidl, 2002], for a normalised isotropic kernel that is monotonically decreasing in the arguments' Euclidean distance, the distance between two points in feature space is maximised if their distance in data space is maximised. For this class of kernels (including, e.g., the Gaussian kernel), this property hints why the criteria in feature space are related to their substitutes in data space. Especially, the true alignment may be close to the class centre distance.

As for the alignment, the class centre distance also has an upper bound. According to (2.18), the class centre distance is bounded by

$$\mathcal{C}_4 \leq 2 \max_{i\in\{1,\ldots,n\}} \|\mathbf{x}_i\| \leq 2 \max_{i\in\{1,\ldots,n\}} \|\mathbf{s}_i\| \quad .$$

Apart from the simple criterion evaluation that comes from the plain form of $\mathcal{C}_4$, the criterion is also easily differentiable. This may be a crucial point from the perspective of optimisation.

While $\mathcal{C}_4$ only takes into account the mean values of the classes we are next looking for classes that are distant from each other and at the same time concentrated around their means.

Scatter Measures A generalisation of $\mathcal{C}_4$ are measures using scatter matrices as described in [Theodoridis and Koutroumbas, 1999, Chap. 5.5.3]:
The *within–class scatter matrix* seizes the average feature variance in the classes and is defined as

$$\mathbf{S}_w := \frac{1}{n} \sum_{i=\pm 1} \sum_{y_j=i} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^\top \quad .$$

The scattering of the whole classes is seized by the *between–class scatter matrix* . Denoting by $\boldsymbol{\mu} := (\sum_{i=\pm 1} n_i \boldsymbol{\mu}_i)/n$ the common mean vector, the matrix is defined as

$$\mathbf{S}_b := \sum_{i=\pm 1} \frac{n_i}{n} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top \quad .$$

Combining both scattering dimensions, the *mixture scatter matrix*

$$\mathbf{S}_m := \frac{1}{n} \sum_{j=1}^{n} (\mathbf{x}_j - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})^\top$$

seizes the common covariance. As a consequence, $\mathbf{S}_m = \mathbf{S}_w + \mathbf{S}_b$ holds.

Using these matrices, separability measures are defined that use the relation of $\mathbf{S}_m$ or $\mathbf{S}_b$ to $\mathbf{S}_w$. This can be done by maximising the quotient of either their traces or their determinants.

The measures $\operatorname{tr}(\mathbf{S}_b)/\operatorname{tr}(\mathbf{S}_w)$ and $|\mathbf{S}_b|/|\mathbf{S}_w|$ for equiprobable classes in one dimension yield the *Fisher discriminant* (see [Fisher, 1936])

$$\frac{(\mu_1 - \mu_{-1})^2}{\sigma_1^2 + \sigma_{-1}^2}$$

with class scatter $\sigma_{\pm 1}^2$.

In the multi-dimensional case, using the determinant poses harder computational requirements than the trace representing only the variances. Moreover, the features are connected by the norm constraint anyway, so it is plausible to ignore their correlation. Hence, for our two-class problem, for example the criterion $\operatorname{tr}(\mathbf{S}_b)/\operatorname{tr}(\mathbf{S}_w)$ leads to the generalised Fisher criterion

$$\begin{aligned}\mathcal{C}_5(\boldsymbol{\theta}) :=& \frac{\operatorname{tr}(\mathbf{S}_b)}{\operatorname{tr}(\mathbf{S}_w)} \\ =& \frac{\frac{n_1}{n}\left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}\right\|^2 + \frac{n_{-1}}{n}\left\|\boldsymbol{\mu}_{-1} - \boldsymbol{\mu}\right\|^2}{\frac{n_1}{n}\sum_{k=1}^{d}\sigma_{1k}^2 + \frac{n_{-1}}{n}\sum_{k=1}^{d}\sigma_{-1k}^2} \ ,\end{aligned}$$

where σ_{ik}^2 denotes the marginal scatter of class i along dimension k. For equiprobable classes, this simplifies to a multiple of

$$\mathcal{C}_5(\boldsymbol{\theta}) \propto \frac{\left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_{-1}\right\|^2}{\sum_{k=1}^{d}(\sigma_{1k}^2 + \sigma_{-1k}^2)} = \frac{\mathcal{C}_4^2(\boldsymbol{\theta})}{\sum_{k=1}^{d}(\sigma_{1k}^2 + \sigma_{-1k}^2)} \ ,$$

which is the class centre distance divided by a variance term. The variances serve to make the measure independent of the scaling as well as to include the classes' scattering. Note that in theory this criterion is unbounded in case of zero variance which, however, rarely happens in practice.

The relation of the matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ is also well known as the basis for a common feature extraction technique. The *Linear Discriminant Analysis* (LDA), also known as Fisher linear discriminant, linearly projects the data to obtain a single feature. Its generalisation to multiple features is called *Multiple Discriminant Analysis* (MDA) and is considered, for example, in [Duda et al., 2000, Chap. 3.8.3], [Devijver and Kittler, 1982]. They maximise the ratio of the determinants $|\mathbf{S}_b|/|\mathbf{S}_w|$ to find the best discriminating linear projections of the data. One is looking for a matrix of $D < d$ projection directions

$$\mathbf{W} := (\mathbf{w}_1 \dots \mathbf{w}_D) \ , \quad \mathbf{w}_1, \dots, \mathbf{w}_D \in \mathbb{R}^d$$

such that the scatter matrices $\widetilde{\mathbf{S}}_w, \widetilde{\mathbf{S}}_b$ for the new feature vectors $\widetilde{\mathbf{x}} := \mathbf{W}^\top \mathbf{x}$ maximise

$$J(\mathbf{W}) = \frac{|\widetilde{\mathbf{S}}_b|}{|\widetilde{\mathbf{S}}_w|} \ .$$

Denote by $i = 1, \dots, c$ the class labels for the multi-class case and by $n_i = |\{j : y_j = i\}|$ the class frequencies. For the classes' feature means, it then holds

$$\widetilde{\boldsymbol{\mu}}_i := \frac{1}{n_i}\sum_{y_j=i}\widetilde{\mathbf{x}}_j = \frac{1}{n_i}\sum_{y_j=i}\mathbf{W}^\top\mathbf{x}_j = \mathbf{W}^\top\boldsymbol{\mu}_i \ , \quad i = 1, \dots, c$$

and, by analogy, $\widetilde{\boldsymbol{\mu}} = \mathbf{W}^\top \boldsymbol{\mu}$ and thereby

$$\begin{aligned}\widetilde{\mathbf{S}}_w &= \frac{1}{n}\sum_{i=1}^{c}\sum_{y_j=i}(\widetilde{\mathbf{x}}_j - \widetilde{\boldsymbol{\mu}}_i)(\widetilde{\mathbf{x}}_j - \widetilde{\boldsymbol{\mu}}_i)^\top \\ &= \frac{1}{n}\sum_{i=1}^{c}\sum_{y_j=i}\mathbf{W}^\top(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^\top\mathbf{W} \\ &= \mathbf{W}^\top\mathbf{S}_w\mathbf{W}\end{aligned}$$

and

$$\begin{aligned}\widetilde{\mathbf{S}}_b &= \sum_{i=1}^{c}\frac{n_i}{n}(\widetilde{\boldsymbol{\mu}}_i - \widetilde{\boldsymbol{\mu}})(\widetilde{\boldsymbol{\mu}}_i - \widetilde{\boldsymbol{\mu}})^\top \\ &= \sum_{i=1}^{c}\frac{n_i}{n}\mathbf{W}^\top(\widetilde{\boldsymbol{\mu}}_i - \widetilde{\boldsymbol{\mu}})(\widetilde{\boldsymbol{\mu}}_i - \widetilde{\boldsymbol{\mu}})^\top\mathbf{W} \\ &= \mathbf{W}^\top\mathbf{S}_b\mathbf{W}\ .\end{aligned}$$

In order to find a stationary point, setting the derivative with respect to the matrix $\mathbf{W}$ to zero and using the differentiation rules given in [Devijver and Kittler, 1982, Appendix B] yields

$$\begin{aligned}&\frac{|\mathbf{W}^\top\mathbf{S}_w\mathbf{W}||\mathbf{W}^\top\mathbf{S}_b\mathbf{W}|\mathbf{S}_b\mathbf{W}(\mathbf{W}^\top\mathbf{S}_w\mathbf{W})^{-1} - |\mathbf{W}^\top\mathbf{S}_b\mathbf{W}||\mathbf{W}^\top\mathbf{S}_w\mathbf{W}|\mathbf{S}_w\mathbf{W}(\mathbf{W}^\top\mathbf{S}_b\mathbf{W})^{-1}}{|\mathbf{W}^\top\mathbf{S}_w\mathbf{W}|^2} \\ &= J(\mathbf{W})(\mathbf{S}_b\mathbf{W}(\mathbf{W}^\top\mathbf{S}_w\mathbf{W})^{-1} - \mathbf{S}_w\mathbf{W}(\mathbf{W}^\top\mathbf{S}_b\mathbf{W})^{-1}) \\ &= 0 \\ \Leftarrow &\mathbf{S}_b\mathbf{W} - \mathbf{S}_w\mathbf{W}(\mathbf{W}^\top\mathbf{S}_w\mathbf{W})^{-1}\mathbf{W}^\top\mathbf{S}_b\mathbf{W} = 0\ .\end{aligned}$$

As a sum of outer vector products, the matrices $\widetilde{\mathbf{S}}_w$ and $\widetilde{\mathbf{S}}_b$ are positive semidefinite. Hence, the inverse $(\mathbf{W}^\top\mathbf{S}_w\mathbf{W})^{-1}$ is positive definite as well. Their matrix product is then diagonalisable with a real eigenvalue matrix $\boldsymbol{\Lambda}$ (cf. [Hackbusch, 1993, Remark 2.10.7]). Setting $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1} := (\mathbf{W}^\top\mathbf{S}_w\mathbf{W})^{-1}\mathbf{W}^\top\mathbf{S}_b\mathbf{W}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix, the maximum has to fulfil

$$\mathbf{S}_b\mathbf{W}\mathbf{U} - \mathbf{S}_w\mathbf{W}\mathbf{U}\boldsymbol{\Lambda} = 0\ ,$$

which means that the solution is a matrix of eigenvectors $\mathbf{V} := \mathbf{W}\mathbf{U}$ of the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ and $\mathbf{W} = \mathbf{V}\mathbf{U}^{-1}$. As the criterion is invariant to invertible transformations

$$J(\mathbf{V}\mathbf{U}^{-1}) = \frac{|(\mathbf{U}^{-1})^\top\mathbf{V}^\top\mathbf{S}_b\mathbf{V}\mathbf{U}^{-1}|}{|(\mathbf{U}^{-1})^\top\mathbf{V}^\top\mathbf{S}_w\mathbf{V}\mathbf{U}^{-1}|} = \frac{|(\mathbf{U}^{-1})^\top||\mathbf{V}^\top\mathbf{S}_b\mathbf{V}||\mathbf{U}^{-1}|}{|(\mathbf{U}^{-1})^\top||\mathbf{V}^\top\mathbf{S}_w\mathbf{V}||\mathbf{U}^{-1}|} = J(\mathbf{V})\ ,$$

the optimal matrix $\mathbf{W}$ consists of eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. Furthermore, the criterion value is just the determinant of the eigenvalue matrix $\mathbf{\Lambda}$, hence the eigenvectors corresponding to the largest eigenvalues are the best choice.

In the case of a single extracted feature for LDA, the determinant criterion is obviously identical to the trace criterion presented above. Besides, the bounded criterion $\operatorname{tr}(\mathbf{S}_m)/\operatorname{tr}(\mathbf{S}_w)$ is equivalent to the unbounded $\operatorname{tr}(\mathbf{S}_b)/\operatorname{tr}(\mathbf{S}_w)$. In general, aside from that, the criteria $\operatorname{tr}(\mathbf{S}_w^{-1}\mathbf{S}_b)$, $\operatorname{tr}(\mathbf{S}_m)/\operatorname{tr}(\mathbf{\Lambda}\mathbf{S}_w)$ and $|\mathbf{S}_m|/|\mathbf{S}_w|$, e.g., are equivalent for feature extraction as argued in [Devijver and Kittler, 1982].

This section proposes different criteria in feature and data space, some of them directly related to generalisation error bounds. The usefulness of the criteria for feature adaptation still remains to be shown. To this end, the next section gives evaluation results of the criteria for several real-world problems.

4.3. Empirical Criteria Comparison

In the previous section we have proposed several criteria for judging the discrimination ability of a set of feature vectors. Some connections between the criteria have already been identified. Now we want to see how the proposed criteria and their theoretical relations behave when analysing real data, especially how close the criteria are together and which ones approximate the true generalisation ability best.

We want to evaluate the proposed criteria for the application described in Chap. 2: One-dimensional signals are to be classified according to the norm of their wavelet coefficients at each level. More precisely, for the feature extraction, we apply orthogonal filter banks with filters of length up to six, which can be parameterised by the two-dimensional space

$$\mathcal{P}_2 = \{\boldsymbol{\theta} = (\theta_1, \theta_0) : \theta_l \in [0, \pi)\,,\ l = 1, 0\}\ ,$$

and make a full wavelet decomposition (i.e., nine decomposition steps for signals of length 512) with subsampling. This generates as many features as possible. We use the ℓ_2-norm as well as the weighted ℓ_2-norm for energy computation and feature extraction. As already described, we thereby omit the low–pass component. By appropriate signal preprocessing, as argued in Sec. 2.2, we can still fix or bound the ℓ_2-norm of the feature vectors. In the classification stage, a hard margin SVM, i.e., $C = \infty$ in (2.34) with Gaussian kernel is used. We apply the hard margin quantities 'margin' and 'radius – margin' to reduce the number of parameters (namely, to fix C to a simple value).

We use three structurally different real data bases to evaluate the criteria: The first are electro-physiological data sets aiming at the detection of ventricular tachycardia as in [Strauss and Steidl, 2002]. The samples used here were obtained by inducing ventricular tachycardia during examinations at the University Hospital of Homburg, Germany. Data segments of 10 sec duration were recorded, equally for periods of normal cardiac

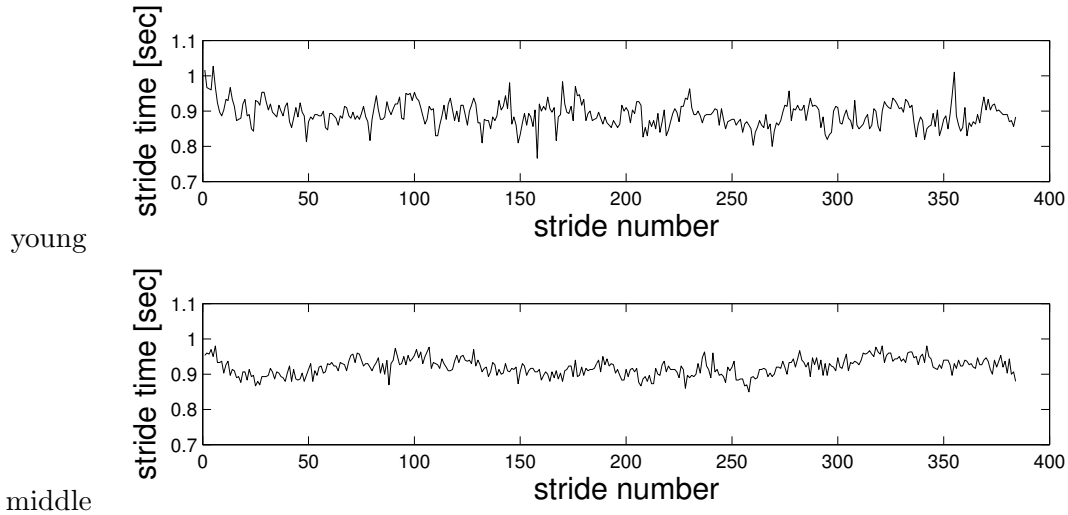
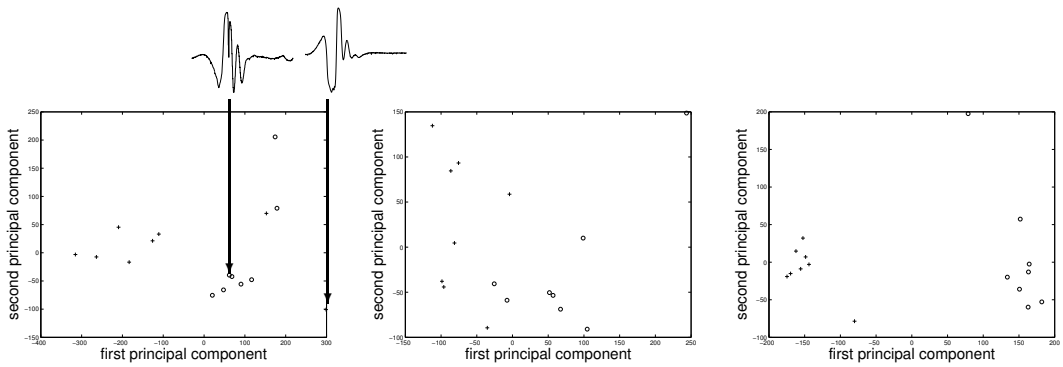


Figure 4.1.: Sample stride time records

activity. The episodes have been filtered and single beats have been cut out within a time frame of 256 ms resulting in waveforms $\mathbf{s} \in \mathbb{R}^{512}$. For each patient and class, eight heartbeats from a single episode are used for classifier training. Some exemplary beats for a sample patient are shown in the introduction in Fig. 1.1. The second data base contains children's stride time records for the examination of gait maturation as in [Hausdorff et al., 1999]. The task is to analyse whether the dynamics of walking still change for healthy children between the ages 3 – 4 (young, $n_1 = 11$) and 6 – 7 (middle, $n_{-1} = 20$). From the data available by [Goldberger et al., 2000], we use the first $l = 384$ strides. Sample time series are depicted in Fig. 4.1. The third group of data are real-world texture images from the MeasTex collection [Smith, 1997]. We use single rows of length $l = 512$ of the corrugated iron images 'Misc.0002' and 'Misc.0003' to have two classes of one-dimensional data. Both images with normalised contrast as well as two exemplary rows are shown in Fig. 1.2 on p. 2. The task is to classify which of the two given textures the rows belong to. Here, the first 32 rows of each texture are used for classifier training.

Accounting for the properties of the feature extraction operator $T_{\boldsymbol{\theta},\|\,\|}$ described in Sec. 2.2, all original sample signals $\mathbf{s}_i$ for $i = 1, \ldots, n$, cardiac data as well as stride time series and texture image rows, have been ℓ_2-normalised according to $\|\mathbf{s}_i\| = 1000$ and their average signal value has been set to zero. We set $\sigma = 100$ for the Gaussian kernel (2.20). (For the rationale of this parameter choice see the end of Sec. 4.3.2.) For the gait maturation data base, it is also possible to classify without prior normalisation as the overall variability may be a useful feature here. In this case, for the appropriate parameter value $\sigma = 1$, the criteria evaluation also yields qualitatively similar results.



(a) (b) (c)

Figure 4.2.: Principal components of training vectors for heartbeat classification with ℓ_2-norm in $E_{\|\,\|}$: (a) for the Haar wavelet, (b) for the Daubechies wavelet with three vanishing moments, (c) for the optimally aligned wavelet ($\mathcal{C}_3$)

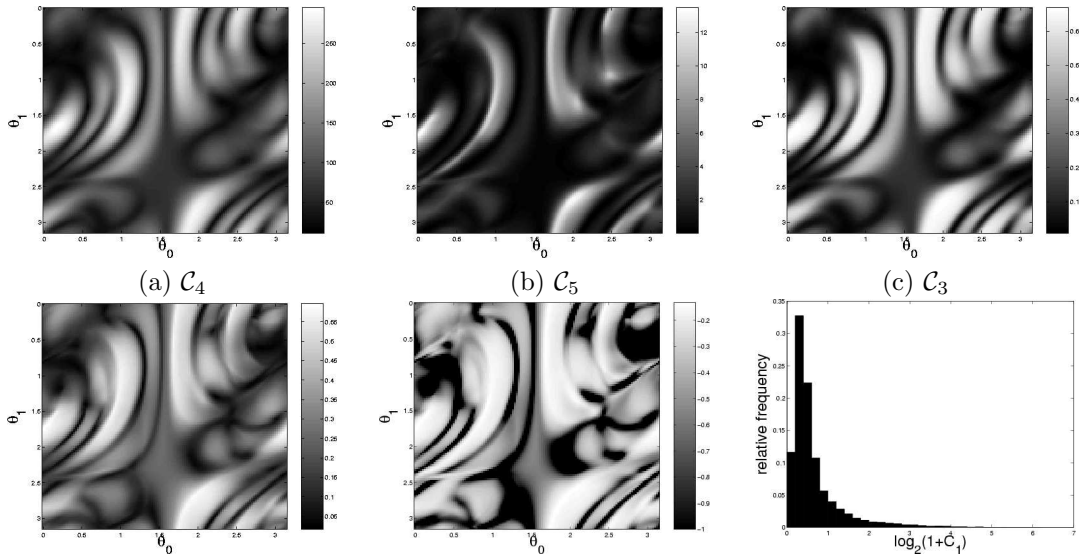
4.3.1. Insight into the Wavelet Adaptation Problem

We start by an example that confirms the tests by [Strauss and Steidl, 2002] and shows that the wavelet choice may heavily influence the classification performance. We illustrate that wavelet feature adaptation may lead to a considerable increase of discriminatory power for real-world signal classification.

For this, we visualise the training data $\mathbf{x}_i \in \mathbb{R}^9$ for the sample heart patient from Fig. 1.1 by extracting its principal two components. The *Principal Components Analysis* (PCA) projects the data down from $\mathbb{R}^9$ so that most of the total variance of the data is retained. The plots for the Haar wavelet ($\boldsymbol{\theta} = (0, 0)$), the Daubechies wavelet with three vanishing moments (see [Daubechies, 1988], ($\boldsymbol{\theta} \approx (0.50, 1.47)$) and the optimal wavelet with respect to $\mathcal{C}_3$ ($\boldsymbol{\theta} \approx (0.56, 2.04)$) for the ℓ_2-norm in $E_{\|\,\|}$ are shown in Fig. 4.2. The variance still contained in the plots is approximately 90%, 75% and 92% of the total variance, respectively.

This single example with few training data already shows that the wavelet choice heavily influences the classification performance: Neither the Haar wavelet nor the Daubechies wavelet appear to make the training data linearly separable. Our optimal wavelet, on the other hand, well separates the data (see Fig. 4.2 (c)). Moreover, the classes are nicely clustered now.

Indeed, for example for this patient with two further test episodes, the classification error for the weighted ℓ_2-norm varies from 0 to 56% for different wavelets. Also, the optimal $\boldsymbol{\theta}$ does not always lie in the same region. Even for different patients (but still the same type of problem), the optimal wavelets differ heavily. As a consequence, utilising standard wavelets such as Haar or Daubechies wavelets does not guarantee well-discriminating features and a small generalisation error.



(a) $\mathcal{C}_4$ (b) $\mathcal{C}_5$ (c) $\mathcal{C}_3$

(d) $\mathcal{C}_2$ (e) $-\log_2(1+\mathcal{C}_1)$ clipped at 1 (f) histogram of (e)

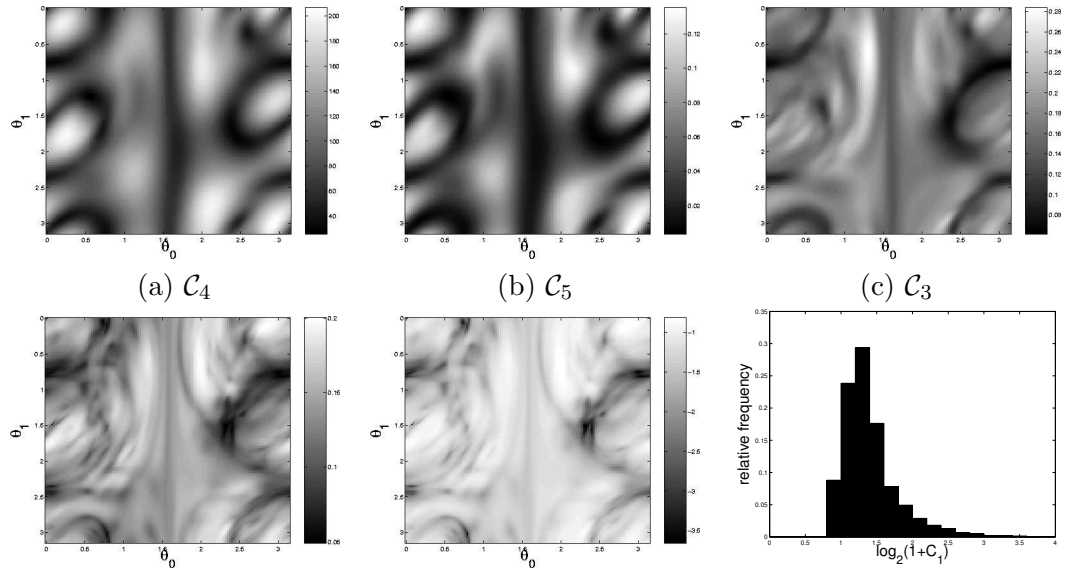
Figure 4.3.: Criteria values for heartbeat classification with weighted ℓ_2-norm in $E_{\|\,\|}$; light spots represent favourable criterion values

4.3.2. Criteria Comparison

Motivated by the results of the previous section, next we evaluate and compare the criteria discussed in Sec. 4.2. For this purpose, we generate plots that show the criterion values subject to the two-dimensional wavelet parameter space $\mathcal{P}_2$. We analyse the distance of the class centres $\mathcal{C}_4$, the generalised Fisher criterion $\mathcal{C}_5 = \operatorname{tr}(\mathbf{S}_b)/\operatorname{tr}(\mathbf{S}_w)$, the alignment $\mathcal{C}_3$ with the kernel matrix $\mathbf{y}\mathbf{y}^\top$, the margin $\mathcal{C}_2$ and the radius – margin classification error bound $\mathcal{C}_1 = R^2/n\rho^2$.

The adaptation criteria are directly visualised over $\mathcal{P}_2$ discretised with 128 angles per dimension. For all parameter combinations, the feature vectors are computed by wavelet decomposition. In Figs. 4.3 to 4.6 the evaluated criteria values are plotted ordered from the computationally most efficient to the most expensive one. The plots use a linear grey scale except for the radius – margin bound $\mathcal{C}_1$ which is plotted on a logarithmic scale due to its large variation. Additionally, its larger values are clipped to the trivial error bound one (except for the gait analysis problem in Fig. 4.4 (e)) to enhance the contrast. To assess the effect of the clipping, the distribution of the logarithm of the bound is indicated by a histogram in Figs. 4.3 to 4.6 (f). Light spots represent favourable criterion values and, hence, beneficial filter operators $F_{\boldsymbol{\theta}}$ in all criteria plots.

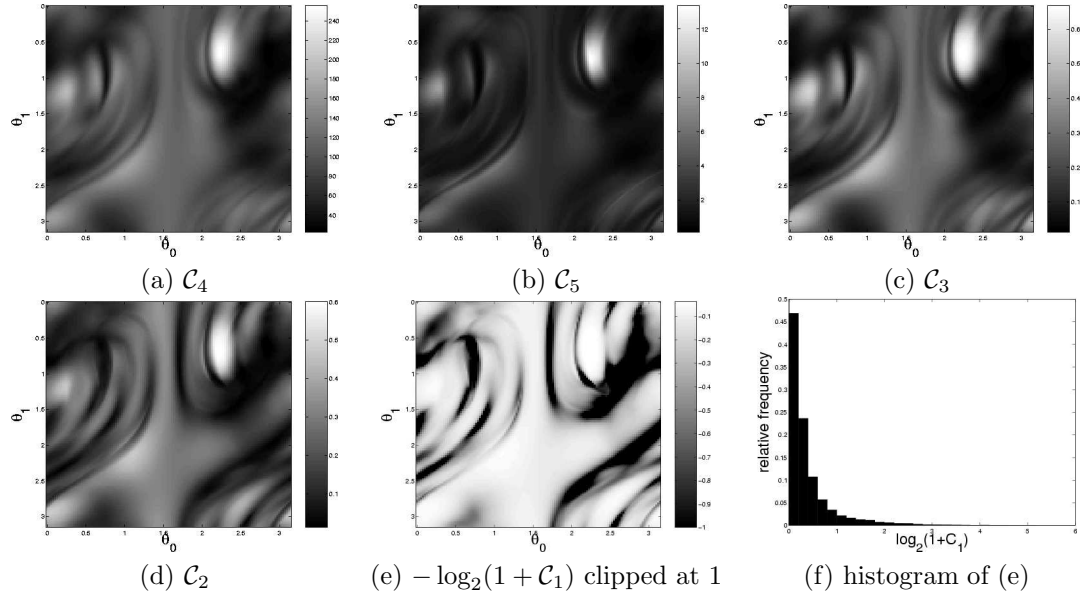
We want to examine the criteria using both norms for energy computation. As for the particular heart patient, e.g., the ℓ_2-norm plots for all criteria much resemble the ones for the weighted norm, further plots are not included.



(a) $\mathcal{C}_4$ (b) $\mathcal{C}_5$ (c) $\mathcal{C}_3$

(d) $\mathcal{C}_2$ (e) $-\log_2(1+\mathcal{C}_1)$ (f) histogram of (e)

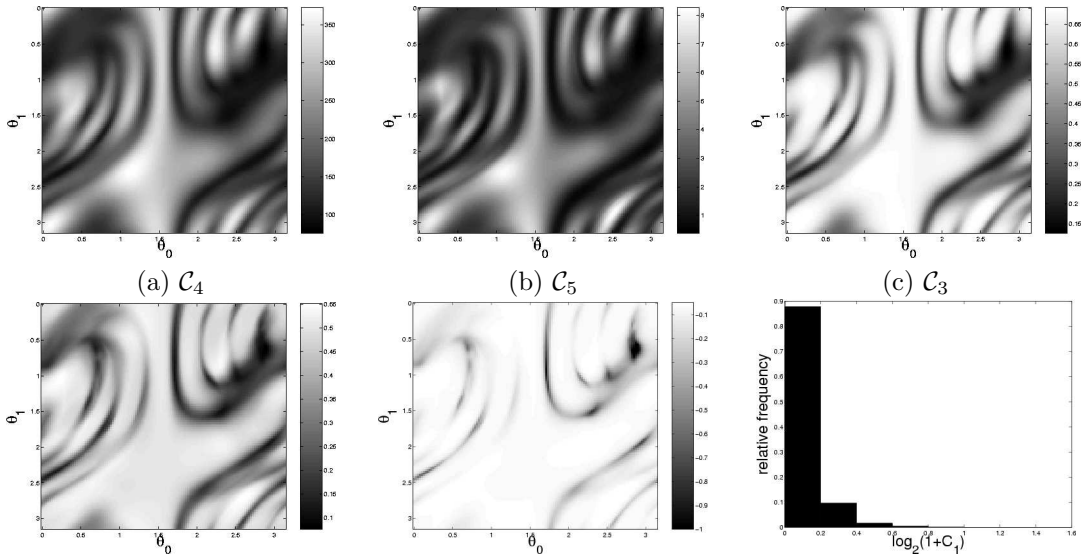
Figure 4.4.: Criteria values for gait dynamics classification with ℓ_2-norm in $E_{\|\,\|}$; light spots represent favourable criterion values



(a) $\mathcal{C}_4$ (b) $\mathcal{C}_5$ (c) $\mathcal{C}_3$

(d) $\mathcal{C}_2$ (e) $-\log_2(1+\mathcal{C}_1)$ clipped at 1 (f) histogram of (e)

Figure 4.5.: Criteria values for texture row classification with weighted ℓ_2-norm in $E_{\|\,\|}$; light spots represent favourable criterion values



(a) $\mathcal{C}_4$ (b) $\mathcal{C}_5$ (c) $\mathcal{C}_3$

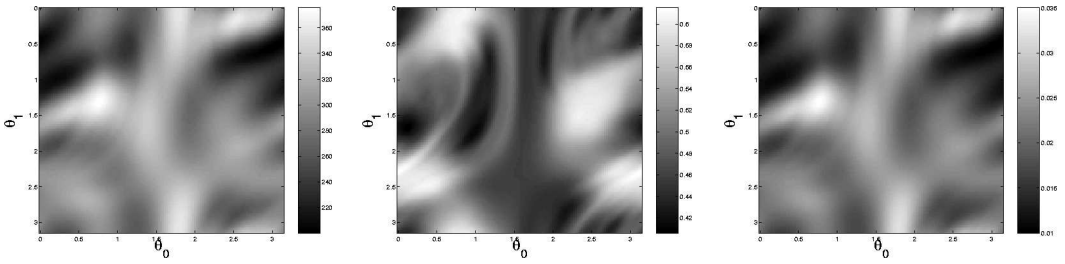
(d) $\mathcal{C}_2$ (e) $-\log_2(1+\mathcal{C}_1)$ clipped at 1 (f) histogram of (e)

Figure 4.6.: Criteria values for texture row classification with ℓ_2-norm in $E_{\|\,\|}$; light spots represent favourable criterion values

Parameter Space Some general properties are visible in the plots: The parameter space is apparently π–periodic in both parameters as argued in Sec. 2.2.1. Additionally, it seems to be structured since some characteristic lines appear in all criteria plots. Another parameter of the feature extraction is the filter length. All orthogonal filters of length four can be generated by a single parameter. Equivalently, all parameter combinations $(0, \cdot) \in \mathcal{P}_2$ correspond to these filters. Regarding the first row of the plots, one can compare the difference between the values achieved there and on the whole parameter space. Only for the texture row classification with weighted ℓ_2-norm plotted in Fig. 4.5, the optimal value on the whole parameter space differs significantly from the optimal value on the first row. For the other classification problems, there is already no systematic gain in augmenting the filter length from four to six.

Criterion Concerning the criteria, for all four problems, the overall impression is that all shown criteria are alike. Moreover, all criteria show a detailed structure for the wavelet parameter space. This indicates that effectively finding the optimal wavelet according to the chosen criterion is not easy even for the simple criteria. We address this problem in Secs. 4.4 and 4.5.

The class centre distance $\mathcal{C}_4$ and particularly the alignment $\mathcal{C}_3$ resemble the margin $\mathcal{C}_2$. That is, the wavelets that generate a high class centre distance or alignment



(a) $\mathcal{C}_4$ (b) $\mathcal{C}_3$ for $\sigma = 100$ (c) $\mathcal{C}_3$ for $\sigma = 1000$

Figure 4.7.: Relationship between class centre distance $\mathcal{C}_4$ and alignment $\mathcal{C}_3$ for the ℓ_2-norm in $E_{\|\,\|}$

also guarantee a large margin. Although the scatter criterion $\mathcal{C}_5$ also takes into account the variances and exhibits more detailed structures, it doesn't seem to be superior to the simplest criterion, the class centre distance $\mathcal{C}_4$.

The radius – margin bound $\mathcal{C}_1$ covers a large range of values: Its maximum goes up to 84 in the examples. Apart from the different distribution of the values, it rates the features mostly like the margin. Confirming the arguments regarding the wavelet adaptation problem, the range of values for the radius – margin bound from 10 resp. 3% to 100% (the maximum meaningful error bound) again indicates the significance of the wavelet choice.

Norm Although the plots for the sample patient did not differ much, for specific signals there may be an important difference between using the ℓ_2-norm and its weighted variant as exhibited by Figs. 4.5 and 4.6, even though the features are only weighted differently. Moreover, for the original ℓ_2-norm as plotted in Fig. 4.6, the class centre distance $\mathcal{C}_4$ differs slightly more from the kernel based criteria, namely alignment $\mathcal{C}_3$ and margin $\mathcal{C}_2$.

Alignment — Class Centre Distance As reasoned in Sec. 4.2, the alignment is linked to the class centre distance. The larger the kernel width σ is, the closer they are to each other. Motivated by this connection, the alignment for different kernel widths σ for a further texture data set where the class centre distance and the alignment differ heavily (images 'Asphalt.0000' and 'Misc.0000' also from the MeasTex collection [Smith, 1997] displayed in Fig. 5.4) is plotted in Fig. 4.7.

Even though the distribution of the alignment for the smaller kernel width $\sigma = 100$ (with exponent $\langle \mathbf{x}_i, \mathbf{x}_j \rangle / \sigma^2 \approx 10^2$) and of the class centre distance are almost inverse, for the larger kernel width $\sigma = 1000$ (with exponent $\langle \mathbf{x}_i, \mathbf{x}_j \rangle / \sigma^2 \approx 1$) they again look similar. Concerning the choice of the kernel parameter, for the heartbeat problem examined in Fig. 4.3, e.g., the highest alignment $\mathcal{C}_3$ for the

optimally aligned wavelet (see Fig. 4.3 (c)) is achieved for $\sigma \approx 200$ for the original ℓ_2-norm and $\sigma \approx 80$ for the weighted norm.

4.3.3. Distances in Feature Space

To confirm the assumption that the distances in feature space resemble the original distances, we visualise the feature vectors. To visualise the original feature vectors $\mathbf{x}_i$, we again use PCA as in Sec. 4.3.1. To retain most of the total variance of the data, PCA projects the points on the eigendirections of the sample covariance matrix or mixture scatter matrix $\mathbf{S}_m = (\sum_{j=1}^{n}(\mathbf{x}_j - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})^\top)/n$ corresponding to its largest eigenvalues. For the points $\boldsymbol{\phi}(\mathbf{x}_j)$ in feature space, we only know the matrix $\mathbf{K}$ of inner products as the mapping $\boldsymbol{\phi}$ isn't given explicitly and the feature space may even be infinite-dimensional. The PCA in feature space introduced by [Schölkopf et al., 1998] is called *kernel PCA* and is carried out by projecting the potentially infinite-dimensional feature vectors onto the eigendirections of the centred kernel matrix

$$\widetilde{\mathbf{K}} := \mathbf{CKC}$$

with centring matrix

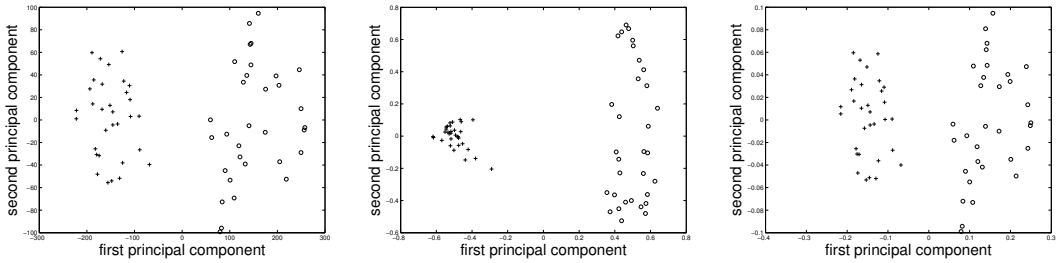
$$\mathbf{C} := \mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^\top \ .$$

As $\widetilde{\mathbf{K}}$ is symmetric and positive definite, it is unitary diagonalisable with positive eigenvalues, which reads

$$\begin{aligned}\widetilde{\mathbf{K}} &= \mathbf{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{1/2}\mathbf{V}^\top \\ &=: \widehat{\mathbf{X}}\widehat{\mathbf{X}}^\top \ .\end{aligned}$$

The rows of $\widehat{\mathbf{X}}$ provide vectors in $\mathbb{R}^n$ that have the same distances as the vectors $\boldsymbol{\phi}(\mathbf{x}_i)$ in feature space. We get our approximations in $\mathbb{R}^2$ by only taking into account the first two components of these vectors (corresponding to the largest eigenvalues).

The resulting feature vectors for one wavelet in the example of Fig. 4.7 (corresponding to a single point in each of the plots in Fig. 4.7) are given in Fig. 4.8. For the visualisation, we chose the optimal wavelet according to the alignment with the smaller Gaussian kernel with $\sigma = 100$. Its filter bank angles are $\boldsymbol{\theta} \approx (3.1, 2.3)$ marking the lightest spot in Fig. 4.7 (b). One has to be careful with the interpretation of the resulting vector plots. If two scatter plots look different, there may be two effects influencing this: Naturally, if the points are differently distributed, their projection in $\mathbb{R}^2$ is likely to be different. But if this is the case, also other principal components may be chosen. The best variance preserving linear projection isn't unique anyway, but we restrict the projection directions to the eigendirections and take care with the signs and scales as well. Nevertheless, the quality of the projection has to be examined. In the example, approximately 88%,



(a) (b) (c)

Figure 4.8.: Principal components of feature vectors for optimally aligned wavelet in example of Fig. 4.7: (a) in data space $\mathbb{R}^d$, (b) in feature space for Gaussian kernel with $\sigma = 100$, (c) in feature space for Gaussian kernel with $\sigma = 1000$

		Support Vector Machine						
data set	energy	Haar	Daub. 3	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_3$	$\mathcal{C}_2$	$\mathcal{C}_1$
heart	weighted ℓ_2	19	9	6	16	16	16	16
texture	weighted ℓ_2	8	4	0	2	1	0	0
texture	ℓ_2	1	2	0	0	0	0	0

Table 4.1.: Classification error [%] for the SVM; different wavelets (Haar, Daub. 3) and adaptation criteria ($\mathcal{C}_i$)

46% and 87% of the total variance is retained for (a), (b) and (c), respectively. These numbers seem sufficient to draw conclusions, and the scattering of the feature vectors for the larger kernel closely matches the scattering of the original feature vectors, as predicted in Sec. 4.2. For our feature vectors $\mathbf{x}$ with $\|\mathbf{x}\| = 1000$, we observe that for a kernel width $\sigma \geq 500$–750, the relative point positions resemble the original ones so that one can easily identify single points in feature space with the input points $\mathbf{x}_i$.

Besides, Fig. 4.8 (b) shows how the optimal wavelet combined with the nonlinear feature map succeed in making the points easily separable.

4.3.4. Classification experiments

To demonstrate the impact of wavelet adaptation on classification, we compare adapted wavelets to both the Haar and the Daubechies wavelet with three vanishing moments. In addition to the results for the SVM listed in Table 4.1, we include in Table 4.2 also results for the Gaussian Bayes classifier with piecewise quadratic decision boundary (see, e.g., [Duda et al., 2000]).

These results show that wavelet adaptation may significantly improve classification performance. We note that the results for the heart data should be taken with care, due to the small sample size. Sixteen additional heartbeats were available as test data for

		Gaussian Bayes Classifier						
data set	energy	Haar	Daub. 3	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_3$	$\mathcal{C}_2$	$\mathcal{C}_1$
heart	weighted ℓ_2	19	28	25	25	13	9	16
texture	weighted ℓ_2	3	3	0	1	0	0	0
texture	ℓ_2	3	3	0	0	0	0	0

Table 4.2.: Classification error [%] for the Gaussian Bayes Classifier; different wavelets (Haar, Daub. 3) and adaptation criteria ($\mathcal{C}_i$)

each class, yet 480 for the texture data. Surprisingly, the Gaussian Bayes classifier shows comparable performance, at least for the texture data. Moreover, wavelet adaptation proves to be favourable here as well.

For further experimental results, we refer to [Strauss and Steidl, 2002].

4.4. An Optimisation Problem for Filter Adaptation

After selecting an adaptation criterion $f \in \{\mathcal{C}_1, \ldots, \mathcal{C}_5\}$, our goal is to systematically determine the optimal filter for the classification process. We have described the feature vectors and their dependency on the filter angles in Sec. 2.2. Unfortunately, there is no hope to exactly solve the filter adaptation problem. In the general case with several filter parameters, in two dimensions, with several filtering steps performed etc., the problem soon becomes quite complicated. Additionally, the problem isn't convex, i.e., there may exist many local minima (cf. Theorem 11 in Appendix B). Some examples for real-world problems are depicted in Figs. 4.3 to 4.6: The margin depending on the filter bank angles is a smooth, i.e. continuous function, but has several minima.

In the previous sections, we have shown that simple criteria like the class centre distance and the alignment well measure the discrimination ability of a set of feature vectors, at least if the Gaussian kernel (2.20) is used in the SVM. We therefore concentrate on maximising these criteria.

We want to set up a simple exemplary optimisation problem directly maximising one of these criteria subject to the filter coefficients or filter angles. As a by-product, the nature of the objective reveals insight into the structure of the parameter space.

For that purpose, we first make the following assumptions:

- one-dimensional signals $\mathbf{s}_i = (s_{i0}, \ldots, s_{i(l-1)})^\top \in \mathbb{R}^l$ for $i = 1, \ldots, n$ are to be classified,
- an equal number of training samples for both classes is given,
- one filter angle θ is used corresponding to filters of length four with $L = 1$ in (2.4),
- two decomposition steps are performed ($d = 2$),

- we also include the norm of the filter bank's low–pass channel $\left\|\mathbf{c}^2\right\|$ in (2.17),
- the ℓ_2-norm is used for energy computation in (2.17), only that we omit taking the square root,
- the objective criterion is the equivalent of the class centre distance $((n/2)\mathcal{C}_4)^2$.

Taking the square of the ℓ_2-features in (2.17) does not affect the classification if the features all have the same magnitude, but makes the whole transform differentiable. But if one feature is dominant, e.g. the low–pass channel, the squaring may even invert the rating.

The following steps successively define the dependence of the objective $((n/2)\mathcal{C}_4)^2$ on the filter angle θ (see also Fig. 2.2):

1. The filter generation $\theta \mapsto (\mathbf{h}_{0\theta}, \mathbf{h}_{1\theta})$ yields for $L = 1$ in (2.4) with (2.5)

$$\mathbf{h}_{0\theta} = \begin{pmatrix} \cos\theta\cos(\frac{\pi}{4}-\theta) \\ \cos\theta\sin(\frac{\pi}{4}-\theta) \\ -\sin\theta\sin(\frac{\pi}{4}-\theta) \\ \sin\theta\cos(\frac{\pi}{4}-\theta) \end{pmatrix} ,$$

$$\mathbf{h}_{1\theta} = \begin{pmatrix} -\sin\theta\cos(\frac{\pi}{4}-\theta) \\ -\sin\theta\sin(\frac{\pi}{4}-\theta) \\ -\cos\theta\sin(\frac{\pi}{4}-\theta) \\ \cos\theta\cos(\frac{\pi}{4}-\theta) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mathbf{h}_{0\theta} =: \mathbf{D}\mathbf{h}_{0\theta} \ .$$

2. The filtering $F_\theta : (\mathbf{s}, \mathbf{h}_{0\theta}, \mathbf{h}_{1\theta}) \mapsto (\mathbf{c}^2, \mathbf{d}^2, \mathbf{d}^1)$ with the synthesis filters $H_{0\theta}(z^{-1})$, $H_{1\theta}(z^{-1})$ as indicated by (2.15) represents a convolution with the analysis filters. With $*$ denoting the convolution operator and $(2\downarrow)$ downsampling by 2, the first decomposition step reads

$$\begin{aligned} \mathbf{c}^1(\mathbf{s}, \mathbf{h}_{0\theta}) &= (\mathbf{s} * \mathbf{h}_{0\theta})(2\downarrow) = \mathbf{S}^1\mathbf{h}_{0\theta} \ , \\ \mathbf{d}^1(\mathbf{s}, \mathbf{h}_{1\theta}) &= (\mathbf{s} * \mathbf{h}_{1\theta})(2\downarrow) = \mathbf{S}^1\mathbf{D}\mathbf{h}_{0\theta} \ , \end{aligned}$$

   where $\mathbf{S}^1 := (s^1_{i,j})_{i=0,\ldots,l/2-1,\, j=0,\ldots,3}$, $s^1_{i,j} := s_{(2i-j) \bmod l}$.

   The second step results in

$$\begin{aligned} \mathbf{c}^2(\mathbf{s}, \mathbf{h}_{0\theta}) &= (\mathbf{c}^1 * \mathbf{h}_{0\theta})(2\downarrow) \ , \\ c^2_k &= \mathbf{h}_{0\theta}^\top \mathbf{S}^{2.k} \mathbf{h}_{0\theta} \ , \qquad k = 0, \ldots, \frac{l}{4} - 1 \ , \\ \mathbf{d}^2(\mathbf{s}, \mathbf{h}_{0\theta}) &= (\mathbf{c}^1 * \mathbf{h}_{1\theta})(2\downarrow) \ , \\ d^2_k &= \mathbf{h}_{0\theta}^\top \mathbf{D}^\top \mathbf{S}^{2.k} \mathbf{h}_{0\theta} \ , \quad k = 0, \ldots, \frac{l}{4} - 1 \ , \end{aligned}$$

where $\mathbf{S}^{2.k} := (s_{i,j}^{2.k})_{i,j=0,\ldots,3}$, $s_{i,j}^{2.k} := s^1_{(2k-i) \bmod 1/2,j} = s_{(4k-2i-j) \bmod l}$ for $k = 0,\ldots,l/4-1$.

In general, every decomposition step generates an additional power of $\mathbf{h}_{0\theta}$.

3. The energy operator $E_{\|\,\|_{\ell_2}^2}$ (2.17) produces a feature vector $\mathbf{x} \in \mathbb{R}^3$ with

$$\begin{aligned} x_1 &= \left(\mathbf{c}^2\right)^\top \mathbf{c}^2 = \sum_{k=0}^{l/4-1} \left(\mathbf{h}_{0\theta}^\top \mathbf{S}^{2.k} \mathbf{h}_{0\theta}\right)^2 \ , \\ x_2 &= \left(\mathbf{d}^2\right)^\top \mathbf{d}^2 = \sum_{k=0}^{l/4-1} \left(\mathbf{h}_{0\theta}^\top \mathbf{D}^\top \mathbf{S}^{2.k} \mathbf{h}_{0\theta}\right)^2 \ , \\ x_3 &= \left(\mathbf{d}^1\right)^\top \mathbf{d}^1 = \mathbf{h}_{0\theta}^\top \mathbf{D}^\top \left(\mathbf{S}^1\right)^\top \mathbf{S}^1 D \mathbf{h}_{0\theta} \ . \end{aligned}$$

Note that, as the summands are no longer linear in the signal matrices, in general, there do not exist matrices $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2$ such that, e.g., $x_1 = \mathbf{h}_{0\theta}^\top \mathbf{\Sigma}_1 \mathbf{h}_{0\theta} \mathbf{h}_{0\theta}^\top \mathbf{\Sigma}_2 \mathbf{h}_{0\theta}$ holds. The powers of $\mathbf{h}_{0\theta}$ have doubled by the energy computation. When performing d decomposition steps, the feature vectors thus depend on $\mathbf{h}_{0\theta}^{2d}$.

4. The criterion evaluation $\{(\mathbf{x}_i, y_i) : i = 1,\ldots,n\} \mapsto ((n/2)\mathcal{C}_4)^2$ yields for equally frequent classes

$$\begin{aligned} (\frac{n}{2}\mathcal{C}_4)^2(\mathbf{h}_{0\theta}) &= \left\| \sum_{i=1}^n y_i \mathbf{x}_i \right\|^2 \\ &= \left\| \begin{pmatrix} \sum_{i=1}^n y_i \sum_{k=0}^{l/4-1} \left(\mathbf{h}_{0\theta}^\top \mathbf{S}_i^{2.k} \mathbf{h}_{0\theta}\right)^2 \\ \sum_{i=1}^n y_i \sum_{k=0}^{l/4-1} \left(\mathbf{h}_{0\theta}^\top \mathbf{D}^\top \mathbf{S}_i^{2.k} \mathbf{h}_{0\theta}\right)^2 \\ \mathbf{h}_{0\theta}^\top \mathbf{D}^\top \left(\sum_{i=1}^n y_i \left(\mathbf{S}_i^1\right)^\top \mathbf{S}_i^1\right) \mathbf{D} \mathbf{h}_{0\theta} \end{pmatrix} \right\|^2 \\ &= \left\| \begin{pmatrix} M_{\mathbf{X}_1}(\mathbf{h}_{0\theta}) \\ M_{\mathbf{X}_2}(\mathbf{h}_{0\theta}) \\ \mathbf{h}_{0\theta}^\top \mathbf{X}_3 \mathbf{h}_{0\theta} \end{pmatrix} \right\|^2 \\ &= (M_{\mathbf{X}_1}(\mathbf{h}_{0\theta}))^2 + (M_{\mathbf{X}_2}(\mathbf{h}_{0\theta}))^2 + (\mathbf{h}_{0\theta}^\top \mathbf{X}_3 \mathbf{h}_{0\theta})^2 \ , \end{aligned}$$

where $\mathbf{X}_1, \mathbf{X}_2$ are the four-dimensional tensors

$$\begin{aligned} \mathbf{X}_1 &= \left(\sum_{i=1}^n y_i \sum_{k=0}^{l/4-1} s_{i,j,m}^{2.k} s_{i,p,q}^{2.k} \right)_{j,m,p,q=0,\ldots,3} \ , \\ \mathbf{X}_2 &= \left(\sum_{i=1}^n y_i \sum_{k=0}^{l/4-1} (-1)^{j+p} s_{i,3-j,m}^{2.k} s_{i,3-p,q}^{2.k} \right)_{j,m,p,q=0,\ldots,3} \end{aligned}$$

and

$$\mathbf{X}_3 = \mathbf{D}^\top \left(\sum_{i=1}^{n} y_i \left(\mathbf{S}_i^1 \right)^\top \mathbf{S}_i^1 \right) \mathbf{D}$$

and $M_{\mathbf{X}}$ denotes the tensor–vector multiplication

$$M_{\mathbf{X}}(\mathbf{h}_{0\theta}) := \sum_{j,m,p,q=0}^{3} x_{j,m,p,q} h_{0\theta}[j] h_{0\theta}[m] h_{0\theta}[p] h_{0\theta}[q] \ .$$

This describes the complete derivation of the objective criterion from the single angle θ for the selected setting. Now the objective function is a polynomial of degree eight in the filter coefficients. In general, for d decomposition steps, it is a polynomial of degree $4d$. In a different setting, if we include the square root to obtain, e.g., the energy operator $E_{\|\,\|_{\ell_2}}$, the objective function is still continuous, but no longer differentiable if the argument of the square root becomes zero. In contrast, a change to the weighted ℓ_2-norm (or its square) in $E_{\|\,\|}$ simply reweights the features, and thus the final summands of the objective function. Furthermore, if more angles are to be determined corresponding to longer filters, the problem structurally remains the same. A generalisation to two-dimensional signals implies a filter tensor product in the filtering step and thus results in polynomials of degree $8d$ for d decomposition steps when using the energy operator $E_{\|\,\|_{\mathrm{F}}^2}$. Alternatively to the class centre distance, when using the alignment as an objective criterion, the last step requires a kernel evaluation $\{\mathbf{x}_i : i = 1, \ldots, n\} \mapsto (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ and the computation of the alignment (4.3) itself. Mainly because of the kernel function, this is more complicated than the formula for the class centre distance, but it is still continuous and perhaps more generally applicable.

The first step uses cosine and sine to generate the filter coefficients, so the objective function is not a polynomial in θ, but still infinitely differentiable. But the function is more complicated to handle than a polynomial. Hence one possible approach is to directly adjust the filter coefficients. Instead of an unconstrained optimisation problem, to guarantee the perfect reconstruction property and one vanishing moment, in the example for filters of length four, we then face the constraints

$$\begin{aligned} h_0[0]h_0[2] + h_0[1]h_0[3] &= 0 \ , \\ h_0[0]^2 + h_0[1]^2 + h_0[2]^2 + h_0[3]^2 &= 1 \ , \\ h_0[0] + h_0[1] + h_0[2] + h_0[3] &= \sqrt{2} \ . \end{aligned}$$

As these are partly nonlinear, the feasible set reduces to a spherical curve in $\mathbb{R}^4$.

Let us resume the properties of our optimisation problem. We have defined a smooth objective function that is easily differentiable, but non-concave. Either we have to deal with the unconstrained, π–periodic parameter space $[0, \pi)$ or the feasible set is defined by three partly nonlinear, convex constraints in $\mathbb{R}^4$, but with a polynomial objective.

4.5. The Optimisation Process

In the previous section we have defined an optimisation problem for designing a filter with regard to the discrimination ability of the resulting feature vectors. We now want to consider the solution of the problem for real-world data.

We use the electro-physiological and texture data specified in Sec. 4.3. For the heartbeat classification, we use data of two patients and name the problems 'heart1' and 'heart2'. From the texture data, we again use the two images of corrugated iron 'Misc.0002' and 'Misc.0003' (problem 'm2m3') and two images of ground texture 'Asphalt.0000' and 'Misc.0000' (problem 'a0m0') here. Similar as before, the original signals, cardiac data as well as texture image rows, have been ℓ_2-normalised to one and their average signal value has been set to zero.

We restrict our numerical experiments to the maximisation of the simplest criterion, the class centre distance $f = \mathcal{C}_4$ and to the parameter spaces $\mathcal{P}_1$ and $\mathcal{P}_2$. Most methods work for the other criteria and higher-dimensional parameter spaces as well.

We have indicated two approaches to solve the filter design problem. First, we examine the profit of a polynomial objective with its limited number of local maxima: We try to solve a constrained polynomial optimisation problem for filter design in Sec. 4.5.1. We examine the unconstrained angle optimisation problem and indicate and test solution algorithms for it in Sec. 4.5.2. In any case, with these non-concave maximisation problems, only local optima can be found and there still remain suitable start values to be determined. As all examined methods do not reliably find the problem's solution, we finally propose a robust grid search heuristic in Sec. 4.5.3 for efficiently finding the global optimum over the resulting parameter space that succeeds in solving our problems in acceptable time.

4.5.1. Constrained Optimisation

A common state of the art optimisation technique for nonlinearly constrained programs is *Sequential Quadratic Programming (SQP)* well described by [Boggs and Tolle, 1995] as well as by [Spellucci, 1993], [Fletcher, 1987, Chap. 12.4]. The iterative approach approximates a nonlinear minimisation program locally by a QP to generate a descent direction. As our problem is continuously differentiable and we only face quadratic constraints for whom linearisation should hopefully not be too inaccurate, the problem approximation seems feasible. Moreover, we already know many good filters offering as initial points, and SQP is an efficient method in terms of convergence rate.

We use the SQP implementation accessible via the function `fmincon` in MATLAB's optimisation toolbox [MathWorks, 2002]. Due to the signal normalisation, the values of the objective function have approximately magnitude one so that the default options can be applied.

But the SQP method is not recommendable for this problem. The iterates often do not

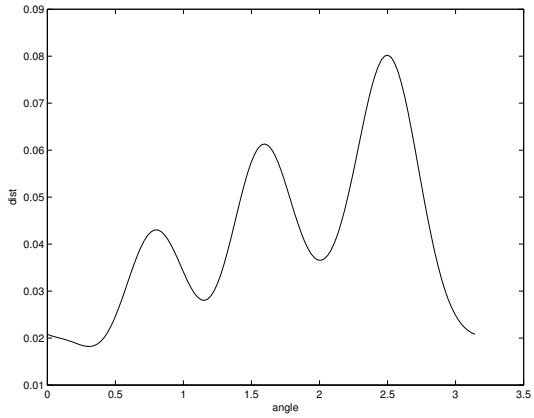


Figure 4.9.: Class centre distance for problem 'heart1' with two decomposition steps

converge. Expectedly, it appears to be practically too difficult to follow the nonlinear constraints. Experiments with all four problems show that the iterates often quickly depart from the feasible region, hence no global convergence can be expected. As a result, the solution found by the algorithm has no relation to the start value any longer and no convergence at all is given or the returned solution is eventually even worse than the start solution.

4.5.2. Unconstrained Optimisation

As the constrained optimisation for the filter design problem fails, we are now looking for methods that find the optimum angle(s) in the parameter space $[0, \pi)$ or $[0, \pi)^2$ for filters of length four or six, respectively. Note that our parameter space is π–periodic, but many local optima exist. Since functions subject to this parameter space are easily plotted, we advance our search for methods by first visualising the objective function to get an impression of the problem's structure in the following. Subsequently, we give optimisation results for the problem obtained by using standard techniques.

Problem Illustration

In the simplified setting studied in Sec. 4.4 we only perform two decomposition steps with filters of length four and then use the energy operator $E_{\|\,\|_{\ell_2}^2}$. The resulting unconstrained objective function for the problem 'heart1' is plotted in Fig. 4.9. The function is still smooth and possesses only three local optima due to the restricted setting. But the levels that provide the highest discrimination potential according to their energy are chiefly levels five to eight for the heartbeats and four to six for the texture rows.

Examples for realistic problems are depicted in Figs. 4.3 to 4.6 (a) in Sec. 4.3.2: The

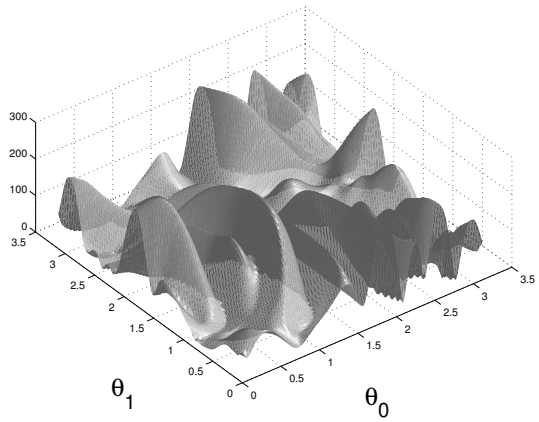


Figure 4.10.: Objective criterion for wavelet adaptation: example from Fig. 4.3 (a)

class centre distance for full decomposition of sample heartbeats with $E_{\|\,\|_{\ell_2}}$, but without the low–pass component is shown. One can see that the class centre distance depending on the filter bank angles is indeed a smooth function, but exhibits several local minima as expected. One example is demonstratively illustrated in Fig. 4.10.

Optimisation Results

We now want to try out standard methods for maximising unconstrained functions as illustrated in Figs. 4.9 and 4.10. Note that an unconstrained optimisation may be performed: The bounds $\mathbf{0} \leq \boldsymbol{\theta} < \pi\mathbf{e}$ may be neglected as the parameter space is periodic resulting from the construction with sine and cosine. But again we are only able to search for local optima.

With the help of MATLAB's optimisation toolbox [MathWorks, 2002], we compare

- *gradient descent* with bisecting line search (see below),
- *Golden Section search* and parabolic interpolation by the function `fminbnd` (only in one dimension),
- the *Nelder–Mead simplex search* method by the function `fminsearch`,
- a *restricted step Newton method* by the function `fminunc` (only if the analytic Hessian is provided), and
- the SQP method applied in Sec. 4.5.1.

A description of all techniques beside the special gradient descent method can also be found in [Fletcher, 1987]. The optimum values given for comparison are computed by

discretising the angles $\theta \in [0, \pi)$ to 128 equally spaced values and picking the maximum on this grid.

With the gradient descent or *steepest descent method*, given an angle vector $\boldsymbol{\theta}^k$ for our problem, the next iterate $\boldsymbol{\theta}^{k+1}$ is determined by $\boldsymbol{\theta}^k$ plus a multiple $\alpha^k \geq 0$ of the gradient of the objective function at $\boldsymbol{\theta}^k$. We then have to solve the *line search subproblem* by choosing α^k to be a minimiser of the objective on the resulting line. As we only deal with optimisation problems in low dimensions, actually solving the line search subproblem isn't advisable as this is almost as complex as solving the whole problem. Simple approaches are to set α^k to a fixed value or to define a decreasing sequence $(\alpha^k)_{k\in\mathbb{N}}$ in advance. We implement a bisection heuristic: Start with an initial guess for α^k. If the objective value of the resulting angle vector is higher than that of $\boldsymbol{\theta}^k$, double α^k until the function value does not increase any longer. Otherwise, if the objective value is lower, halve α^k until the objective value is higher than that of $\boldsymbol{\theta}^k$. For α^{k+1}, we use α^k as an initial guess.

Most of the univariate functions coming from our simple filter design problem have two or three local maxima as, e.g., the one shown in Fig. 4.9, so we fix four start values for all methods apart from Golden Section search: the Haar wavelet (with angle $\theta = 0$), the Daubechies wavelet with two vanishing moments (see [Daubechies, 1988], $\theta = 11\pi/12$), $\theta = 1$ and $\theta = 2$. For the evaluation, we use a tolerance for the solution angle of 0.01 only as this precision suffices for our practical filtering purposes and to be able to compare it with simpler methods.

For the simplest optimisation problem considered, we can avail ourselves of the problem formulation in Sec. 4.4, especially for the analytic gradient evaluation. The optimisation results for the four discussed classification problems are summarised in Table 4.3. Confer the objective function plot for 'heart1' in Fig. 4.9. One can see that all methods find the optimum for all sample problems, but except for the Golden Section search, the number of function evaluations is not substantially lower than for the complete search that would achieve the same accuracy with approximately 128 evaluations. And additionally, the cost for calculating the signal tensors $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ from Sec. 4.4 during the preparation step is dominating the cost for the optimisation. Furthermore, more decomposition steps d lead to $4^{2d} = 2^{4d}$ or 6^{2d} tensor coefficients for filters of length four or six, respectively, and it quickly becomes impossible even to store them. Consequently, we are still aware that the objective function is differentiable, but we are searching for algorithms for whom we needn't supply the gradients so that we can compute the objective values ad hoc by just performing a wavelet decomposition for all signals.

For a more realistic setting with nine decomposition steps, which means full decomposition, we apply gradient descent with finite difference gradients as well as the other available methods listed above. The optimisation results are shown in Table 4.4. One can see that none of the methods is able to always find the optimum from the given start values, especially Golden Section search doesn't work well any longer. Augmenting the number of start values gets us close to the performance of total sampling, especially when

problem	heart1		heart2		a0m0		m2m3	
method	value	evals	value	evals	value	evals	value	evals
optimum	0.0802		0.0188		111.98		3.0923	
gradient descent	0.0802	38	0.0188	38	111.98	41	3.0919	32
Golden Section	0.0802	12	0.0188	11	111.98	11	3.0922	12
simplex search	0.0802	46	0.0188	48	111.98	100	3.0922	88
Newton	0.0802	22	0.0188	21	111.98	20	3.0923	27
SQP	0.0802	46	0.0179	20	111.98	37	3.0923	68

Table 4.3.: Optimisation results for the exemplary problem of Sec. 4.4: maximal value found and number of function evaluations

problem	heart1		heart2		a0m0		m2m3	
method	value	evals	value	evals	value	evals	value	evals
optimum	0.7194		0.5092		0.2503		0.3023	
gradient descent	0.7194	60	0.5074	55	0.2503	55	0.3022	61
Golden Section	0.5393	12	0.4817	11	0.2377	13	0.2537	13
simplex search	0.7194	66	0.5074	66	0.2503	40	0.3023	70
SQP	0.7194	43	0.5074	49	0.2503	33	0.3023	55

Table 4.4.: Optimisation results for one angle and full decomposition: maximal value found and number of function evaluations

considering the overhead for the optimisation methods beside the function evaluations.

In two dimensions corresponding to filters of length six, the optimisation becomes more complicated. In a realistic setting with full decomposition and the use of the weighted ℓ_2-norm for the energy operator (2.17), we perform an optimisation with 4×4 equally spaced start values in $[0, \pi)^2$. As we still assume differentiability, we again evaluate all available methods listed above. The results are shown in Table 4.5. All three methods work well for the examples, but all pose the question of the choice of start values. To obtain maximally independent start values, one can use the notion of orthogonality of the discrete–time wavelets similar to the approach in [Strauss et al., 1999]. But still there remains the number of start values to be chosen depending on the problem's nature.

4.5.3. A Search Algorithm

In the previous sections we have considered various constrained and unconstrained optimisation strategies to find $\boldsymbol{\theta}$, e.g. SQP, a simplex search method and a restricted step Newton method. As the number of function evaluations for the solution of the two-dimensional problem with standard optimisation techniques is close to the number of points on a medium spaced grid of about 32×32 points, we have developed the following adaptive grid search algorithm which seems to be the most efficient method: The idea

problem	heart1		heart2		a0m0		m2m3	
method	value	evals	value	evals	value	evals	value	evals
optimum	0.2923		0.2601		0.1670		0.2564	
gradient descent	0.2921	850	0.2601	803	0.1670	935	0.2564	913
simplex search	0.2923	655	0.2601	721	0.1670	774	0.2564	611
SQP	0.2923	378	0.2598	454	0.1669	406	0.2559	520

Table 4.5.: Optimisation results for two angles and full decomposition: maximal value found and number of function evaluations

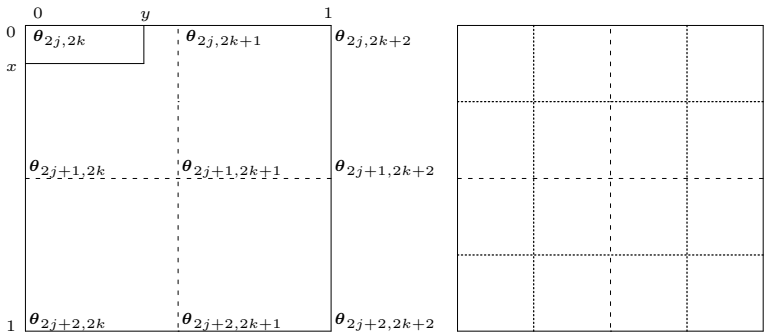


Figure 4.11.: (a) segment $I_{j,k}$ of a coarse grid, examined function values on two different refinement levels, (b) refined segment

is to start with an equispaced coarse grid

$$\mathcal{G}_0 := \left\{ \boldsymbol{\theta}_{j,k} := \left(\frac{\pi j}{N}, \frac{\pi k}{N} \right) : j,k = 0,\ldots,N-1 \right\} \ .$$

On $\mathcal{G}_0$ we compute the function values $f_{j,k} := f(\boldsymbol{\theta}_{j,k})$ and $f_{\max} := \max_{j,k} f_{j,k}$. Now we consider the neighbourhood $I_{j,k}$ of each point $\boldsymbol{\theta}_{2j+1,2k+1}$ depicted in Fig. 4.11 (a) and adaptively refine those sections of the grid where the function behaves differently from our expectation or where it exhibits favourable values. One can then define tolerances no longer depending on the absolute function values and is also independent of possible start values.

We first motivate our proposed refinement criterion and formulate the algorithm and then give some experimental results.

Refinement Criterion and Algorithm

We investigated several approaches concerning the grid refinement and criteria for adaptive local refinement. After adding the finishing touches, the criterion for a further local

grid refinement is the ratio of the improvement towards a bilinear interpolation to the rating compared with the optimum. The quotient balances these two important aspects against each other. Let us illustrate the bilinear interpolation. Figure 4.11 (a) shows a 3×3 grid section marked with already calculated function values on two levels defined by the solid and dashed lines. We use the even indexed grid points

$$(\boldsymbol{\theta}_{2j,2k}, f_{2j,2k}), (\boldsymbol{\theta}_{2j,2k+2}, f_{2j,2k+2}), (\boldsymbol{\theta}_{2j+2,2k}, f_{2j+2,2k}), (\boldsymbol{\theta}_{2j+2,2k+2}, f_{2j+2,2k+2})$$

on the coarser grid indicated by the solid lines as our interpolation points. The bilinear interpolation polynomial $\widehat{f}$ on $I_{j,k}$ is then given by

$$\begin{aligned} \widehat{f}(\boldsymbol{\theta}_{2j,2k} + h \cdot (x,y)) &= f_{2j,2k} + x\,(f_{2j+2,2k} - f_{2j,2k}) + y\,(f_{2j,2k+2} - f_{2j,2k}) \\ &\quad + x\,y\,(f_{2j,2k} - f_{2j+2,2k} - f_{2j,2k+2} + f_{2j+2,2k+2}) \end{aligned}$$

for $0 \leq x, y \leq 1$, where $h := (\boldsymbol{\theta}_{2j+2,\cdot} - \boldsymbol{\theta}_{2j,\cdot})_1$ is the coarse grid width. Then $\widehat{f}$ is a continuous function on the whole parameter space.

f is concave on $I_{j,k}$ if and only if

$$f(\boldsymbol{\theta}_{2j,2k} + h(\alpha(x,y) + (1-\alpha)(\widetilde{x},\widetilde{y}))) \geq \alpha f(\boldsymbol{\theta}_{2j,2k} + h \cdot (x,y)) + (1-\alpha) f(\boldsymbol{\theta}_{2j,2k} + h \cdot (\widetilde{x},\widetilde{y}))$$

for all $0 \leq \alpha, x, y, \widetilde{x}, \widetilde{y} \leq 1$. Hence the condition of improvement on f at the four boundary points including odd indices is just the concavity condition with respect to their neighbour points on the coarse grid and $\alpha = 1/2$. In general, if f is concave on $I_{j,k}$ we have that

$$\begin{aligned} & f(\boldsymbol{\theta}_{2j+\cdot,2k} + h \cdot (0,y)) \geq f_{2j+\cdot,2k} + y\,(f_{2j+\cdot,2k+2} - f_{2j+\cdot,2k}) \\ \Rightarrow \quad & f(\boldsymbol{\theta}_{2j,2k} + h \cdot (x,y)) \geq x f(\boldsymbol{\theta}_{2j+2,2k} + h \cdot (0,y)) + (1-x) f(\boldsymbol{\theta}_{2j,2k} + h \cdot (0,y)) \\ & \qquad \geq x\,(f_{2j+2,2k} + y\,(f_{2j+2,2k+2} - f_{2j+2,2k})) \\ & \qquad\quad + (1-x)\,(f_{2j,2k} + y\,(f_{2j,2k+2} - f_{2j,2k})) \\ & \qquad = f_{2j,2k} + x\,(f_{2j+2,2k} - f_{2j,2k}) + y\,(f_{2j,2k+2} - f_{2j,2k}) \\ & \qquad\quad + x\,y\,(f_{2j,2k} - f_{2j+2,2k} - f_{2j,2k+2} + f_{2j+2,2k+2}) \end{aligned}$$

for all $x, y \in [0,1]$, which implies for $\boldsymbol{\theta} \in I_{j,k}$ that

$$f(\boldsymbol{\theta}) \geq \widehat{f}(\boldsymbol{\theta}) \ .$$

If f is even strictly concave, the improvement is positive for all non-corner points. Further, if f is heavily concave, the improvement is still greater, so that the improvement towards bilinear interpolation $f - \widehat{f}$ may be considered as local measure for concavity of the function f — or convexity for minimisation problems approximating the degree of convexity ρ defined in Def. 5. As local concavity is a necessary condition for a local maximum of a twice differentiable function, the refinement condition compares the function values

of the grid points including odd indices $f_{2j,2k+1}, f_{2j+1,2k}, f_{2j+1,2k+1}, f_{2j+1,2k+2}, f_{2j+2,2k+1}$ with their estimates by $\widehat{f}$. Denoting by $f_{\max}$ the maximal function value found up to now, we use the following refinement strategy: If

$$\frac{f(\boldsymbol{\theta}) - \widehat{f}(\boldsymbol{\theta})}{f_{\max} - f(\boldsymbol{\theta})} > \text{tolF} \tag{4.4}$$

for at least one $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_{2j,2k+1}, \boldsymbol{\theta}_{2j+1,2k}, \boldsymbol{\theta}_{2j+1,2k+1}, \boldsymbol{\theta}_{2j+1,2k+2}, \boldsymbol{\theta}_{2j+2,2k+1}\}$ then we further refine the segment $I_{j,k}$ to obtain the grid segment in Fig. 4.11 (b) with the dotted lines added, otherwise we leave the segment as it is. The refinement criterion reads for $\boldsymbol{\theta}_{2j+1,2k+1}$, e.g.,

$$\frac{f_{2j+1,2k+1} - \widehat{f}\left(\boldsymbol{\theta}_{2j,2k} + h \cdot (\frac{1}{2}, \frac{1}{2})\right)}{f_{\max} - f_{2j+1,2k+1}} > \text{tolF}$$

$$\Leftrightarrow f_{2j+1,2k+1} - (f_{2j,2k} + f_{2j+2,2k} + f_{2j,2k+2} + f_{2j+2,2k+2})/4 > \text{tolF}(f_{\max} - f_{2j+1,2k+1}) \ .$$

The quotient (4.4) balances the improvement towards the bilinear interpolation with the rating compared with the optimum. We apply our refinement strategy to all segments of $\mathcal{G}_0$ and end up with a new adaptively refined grid $\mathcal{G}_1$. On the next level, we apply the procedure again on the refined grid $\mathcal{G}_1$ also containing the four resulting smaller sections for each refined segment and so on until the finest segments have grid width $h \leq \text{tolX}$. Note that function evaluations are only necessary on the new grid points.

In the beginning of the algorithm's runtime, heavily concave sections with arbitrary function values satisfy condition (4.4), in the end only concave sections that at the same time have high function values, i.e. possible maxima, are refined.

In summary, we propose the following algorithm:

Algorithm 4.5.1: GRIDSEARCH$(f, \text{tolF}, \text{tolX}, N)$

local $h, \text{index}, \text{indexnew}$
calculate f on $\left\{0, \frac{\pi}{N}, \ldots, \pi - \frac{\pi}{N}\right\}^2$
$h \leftarrow \frac{\pi}{N}$
$\text{index} \leftarrow \left\{0, \ldots, \frac{N}{2} - 1\right\}^2$
while $(\text{index} \neq \varnothing) \wedge (h > \text{tolX})$
do
- $\text{indexnew} \leftarrow \varnothing$
- **for each** $(i, j) \in \text{index}$
  - **do**
    - **if** improvement towards bilinear interpolation of f on intermediate grid points in $([2i, 2i+2] \times [2j, 2j+2]) * h$ /(current maximum − function value) $> \text{tolF}$
    - **then**
      - refine f on $([2i, 2i+2] \times [2j, 2j+2]) * h$
      - $\text{indexnew} \leftarrow \text{indexnew} \cup \{2i, 2i+1\} \times \{2j, 2j+1\}$
- $h \leftarrow h/2$
- $\text{index} \leftarrow \text{indexnew}$

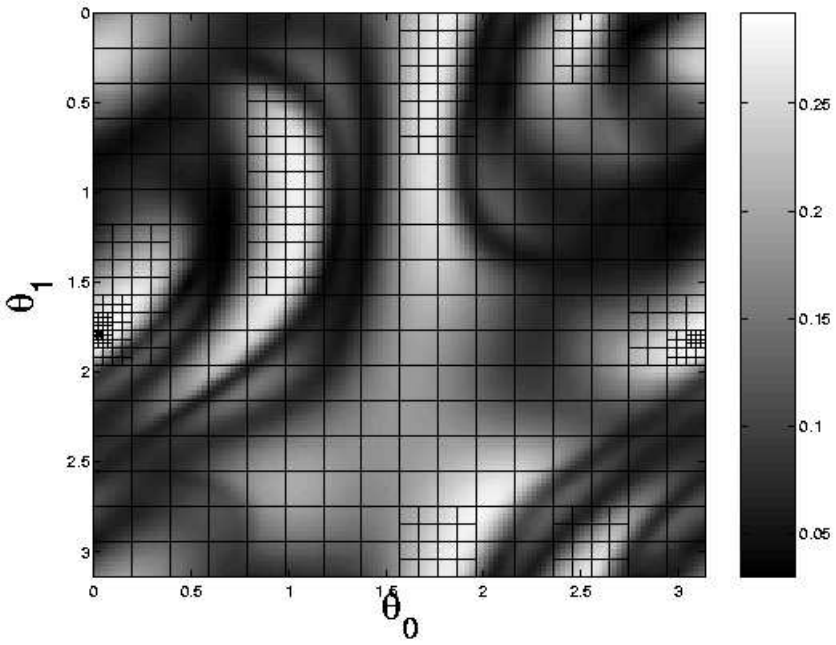


Figure 4.12.: Class centre distance for problem 'heart1' with final grid used by Algorithm 4.5.1

Experiments and Performance

In our numerical examples, we stick to the setting used in the end of Sec. 4.5.2, i.e., full decomposition with two lattice angles (θ_1, θ_0) followed by the weighted ℓ_2-norm for $E_{\|\,\|}$ in (2.17). For the parameters of Algorithm 4.5.1, we use values of tolF $= 4$, again tolX $= 0.01 \approx \pi/512$ and $N = 16$. Note that the number of function evaluations and the optimal value found by the algorithm are sensible to the parameter tolF, but its value can be used for all problems as we apply an absolute criterion.

Figure 4.12 demonstrates the final grid generated by the algorithm for the first heart patient. One can already see that the region where f is evaluated rapidly gets smaller with each finer grid.

Table 4.6 presents the algorithm's results for all four sample problems. Thereby, the optimum values given for comparison were computed by picking the maximum class centre distance $f = \mathcal{C}_4$ on the equispaced grid with $h = \pi/128$ leading to $128^2 = 16384$ function evaluations. So the heuristic finds the optimum or a close value with only 2–3% of the criterion evaluations. Compared with Table 4.5, the algorithm performance keeps up with that of the optimisation. Moreover, not only due to the few function evaluations the grid search is faster than all other evaluated methods, especially much faster than

problem	heart1		heart2		a0m0		m2m3	
method	value	evals	value	evals	value	evals	value	evals
optimum	0.2923		0.2601		0.1670		0.2564	
grid search	0.2923	538	0.2601	392	0.1669	350	0.2564	364

Table 4.6.: Grid search results for two angles and full decomposition: maximal value found and number of function evaluations

the simplex search algorithm. So it establishes a robust and fast optimisation method for our problems that is easy to tune.

4.6. Summary and Outlook

We have addressed the problem how to efficiently adapt the feature extraction process by orthogonal filter banks and norm computation of the subband coefficients to the subsequent classification by an SVM. We have proposed several criteria for judging the discrimination ability of a set of feature vectors and have highlighted some connections between these criteria. A theorem was provided that simplifies the computation of the radius – margin error bound. We have numerically shown that simple adaptation criteria like the class centre distance and the alignment suffice to promisingly design filters for our hybrid wavelet–large margin classifiers with Gaussian kernels.

We have also presented an adaptive grid search algorithm that effectively finds the optimal orthogonal filter bank for our applications. This grid search can easily be implemented due to its simplicity and provides a robust algorithm that does not depend on experienced parameter tuning for each optimisation problem

Multi-class SVMs are reviewed in Sec. 2.3.4. Using binary classifiers to solve multi-class problems, the wavelet adaptation can be applied with a different wavelet for every classifier. As an alternative approach, e.g. the generalised Fisher criterion $\mathcal{C}_5$ naturally generalises to multiple classes.

The classification of images and higher-dimensional signals works analogously to one-dimensional signals according to the construction of multivariate wavelets by tensor products. Hence, in principle our results are relevant for higher dimensions as well. Just that in this case, the additional question whether to choose the same wavelet for all directions or separately adapt wavelets comes up. In practice, however, features extracted by tensor wavelets — no matter whether adapted or not — suffer from a lack of rotational invariance so that one should consider applying a dual–tree complex wavelet transform as introduced in Chap. 3 instead.

5. Adaptation and Embedded Feature Selection

5.1. Feature Selection

As it was the aim of our *feature extraction* process defined in Sec. 2.2, *feature selection* intends to reduce the number of features d in the context of supervised pattern classification. But unlike feature extraction that may be defined by an arbitrary operator $T : \mathbb{R}^l \to \mathbb{R}^d$, we are now just picking out a subset of all features $\{1, \ldots, d\}$. So feature selection is a combinatorial optimisation problem. The goal is to retain only those features that ensure a high accuracy of the classifier. Different notions of *feature relevance* have been defined by [Kohavi and John, 1997]. In particular, feature selection is another approach to adapt wavelets to the classification problem.

At first glance, feature selection seems reasonable in order to save resources. But as resumed by [Guyon and Elisseeff, 2003, Weston et al., 2001], the motivations for doing feature selection are manifold:

- performance issues:
  - facilitating data collection,
  - reducing storage space,
  - reducing classification time,
- understanding the classification problem: semantics analysis,
- improving prediction performance (by preventing the *curse of dimensionality*).

Beside the number of features increasing the computation time of, e.g., kernel functions, a subtle effect of the dimensionality may also be observed concerning the geometry of the data. If, e.g., the kernel matrix $\mathbf{K}$ in the SV problem (2.34) is dense, the solution is costly. In [Hegland, 2003], it was argued that for the Gaussian kernel (2.20), the probability for small elements in $\mathbf{K}$ is bounded according to

$$P\left(K(\mathbf{x}_i, \mathbf{x}_j) \leq te^{-M^2/(2\sigma^2)}\right) \leq P\left(\|\mathbf{x}_i - \mathbf{x}_j\| - M \geq \frac{2\sigma^2 |\ln t|}{M + D}\right) \quad ,$$

where M is the median and D the maximum of $\{\|\mathbf{x}_i - \mathbf{x}_j\| : i, j = 1, \ldots, n\}$. If we further assume for the concentration function $P(\|\mathbf{x}_i - \mathbf{x}_j\| - M \geq s) \approx e^{-cds^2}$ with $c > 0$, we obtain for $t \in (0, 1]$ the estimate

$$P\left(K(\mathbf{x}_i, \mathbf{x}_j) \leq te^{-M^2/(2\sigma^2)}\right) \leq t^{\frac{4cd\sigma^4|\ln t|}{(M+D)^2}} \quad .$$

As a consequence, the probability that an element of $\mathbf{K}$ falls below a threshold drops exponentially with the dimension d. In view of the sparsity of $\mathbf{K}$, this also suggests to apply feature selection.

According to [Guyon and Elisseeff, 2003, John et al., 1994, Kohavi and John, 1997, Bradley, 1998], feature selection approaches essentially divide into

- *filters* which act as a preprocessing step and select features *a priori* independently of the final classifier built [Hermes and Buhmann, 2000, Steel and Hechter, 2004, Shashua and Wolf, 2004, Duda et al., 2000, Heiler et al., 2001],
- *wrappers* which take the classifier into account as a black box ([John et al., 1994, Kohavi and John, 1997, Weston et al., 2001]), and
- *embedded approaches* which try to determine the optimal feature set and classifier simultaneously during the training process.

The three approaches are listed with increasing complexity and accuracy of the feature selection. Feature dependencies and the feature relevance for the classification accuracy are taken into account more and more [Guyon and Elisseeff, 2003, Weston et al., 2001].

As illustrated in Fig. 5.1, most known approaches are filters (left); known embedded methods in [Bradley and Mangasarian, 1998] are based on a linear approach (middle) although the SVM provides better generalisation ability by its ℓ_2 regulariser.

A wrapper method for nonlinear SVMs is given by [Weston et al., 2001], where instead of minimising the classification error, the features are selected to minimise a generalisation error bound. The embedded methods in [Bradley and Mangasarian, 1998] are based on a linear classifier. Similar to the wrappers, there exist only few embedded methods addressing feature selection in connection with nonlinear classifiers up to now. An embedded approach for the quadratic ℓ_1–SVM was suggested by [Zhu et al., 2004]. The authors penalise the features by the ℓ_1-norm and apply the nonlinear mapping explicitly. This makes the approach feasible only for low-dimensional feature maps such as the quadratic one. In particular, original features are not suppressed so that no performance improvements or semantics analysis are possible.

We focus on embedded approaches for feature selection. We investigate different *direct objective minimising feature selection* approaches formulating the task as an optimisation problem. The starting point for our investigation is the FSV approach by [Bradley and Mangasarian, 1998], which along with other linear embedded approaches

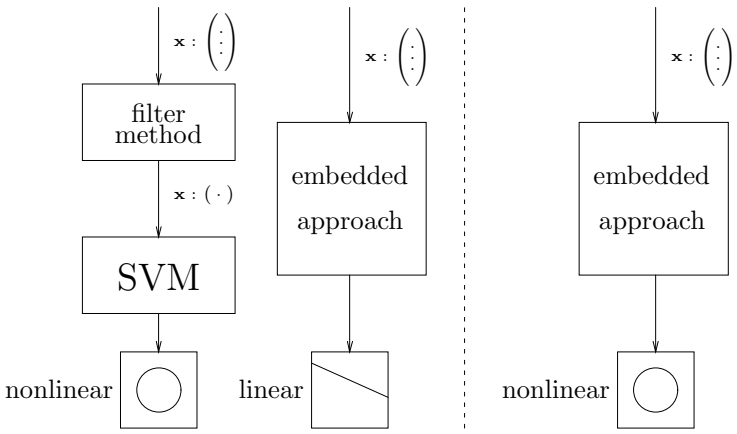


Figure 5.1.: Feature selection and classification: known approaches (**left**) and our new approaches (**right**)

is reviewed in Sec. 5.2. It minimises the training errors of a linear classifier while penalising the number of features by a concave penalty approximating the ℓ_0-"norm". In this way, the linear classifier is constructed while implicitly discarding features.

After studying the application of this particular feature selection approach to wavelet adaptation in Sec. 5.3, we introduce own enhanced approaches both for linear and nonlinear classification in Sec. 5.4.

Some of our new approaches require the solution of non-convex optimisation problems. To solve these problems, we apply a general difference of convex functions (d.c.) optimisation algorithm in an appropriate way. The d.c. optimisation approach and its application to our feature selection problems is described in Sec. 5.5. Moreover, we show that the *Successive Linearisation Algorithm* (SLA) proposed by [Bradley and Mangasarian, 1998] for concave minimisation is in effect a special case of our general optimisation approach.

Numerical results illustrating and evaluating various approaches are given in Sec. 5.6. To illustrate that feature selection is especially profitable for high-dimensional problems, we investigate as part of our in-depth method evaluation the problem of selecting a suitable subset from 650 image features in order to segment organs in computed tomography (CT) scans.

After considering an extension to multi-class problems in Sec. 5.7, we summarise in Sec. 5.8.

5.2. Known Feature Penalties and Feature Selection Methods

As in the classification setting in Sec. 2.3, we assume we are given a training set $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\}$ with $\mathcal{X} \subset \mathbb{R}^d$. Our goal is both to find a classifier $F : \mathcal{X} \to$

$\{-1, 1\}$ and to select features. We introduce the linear classification approach on which the presented embedded feature selection approaches are based in Sec. 5.2.1, and then add penalties for feature suppression to obtain common feature selection methods in Sec. 5.2.2.

5.2.1. Robust Linear Programming

The baseline approach is similar to a linear soft margin SVM only that it omits the regularisation by omitting the margin maximisation: Hence, we construct two parallel bounding hyperplanes in $\mathbb{R}^d$ such that the differently labelled sets are maximally located in the two opposite half spaces determined by these hyperplanes by solving

$$f_{\mathrm{RLP}}(\mathbf{w}, b) := \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))_+ \longrightarrow \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} \quad . \tag{5.1}$$

With $(\mathbf{w}, b)$ being the solution of (5.1), the classifier is $F(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top \mathbf{x} + b)$. The linear method (5.1) was proposed by [Bennett and Mangasarian, 1992] as *Robust Linear Programming* (RLP). Note that these authors weight the training errors and instead solve

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}^+\in\mathbb{R}^n,\, \boldsymbol{\xi}^-\in\mathbb{R}^n} \quad & \frac{\mathbf{e}^\top \boldsymbol{\xi}^+}{n_{+1}} + \frac{\mathbf{e}^\top \boldsymbol{\xi}^-}{n_{-1}} \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{x}_i + b \geq 1 - \xi_i^+ \ , \qquad y_i = 1 \ , \\ & \mathbf{w}^\top \mathbf{x}_i + b \leq -1 + \xi_i^- \ , \qquad y_i = -1 \ , \\ & \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \mathbf{0} \ , \end{aligned} \tag{5.2}$$

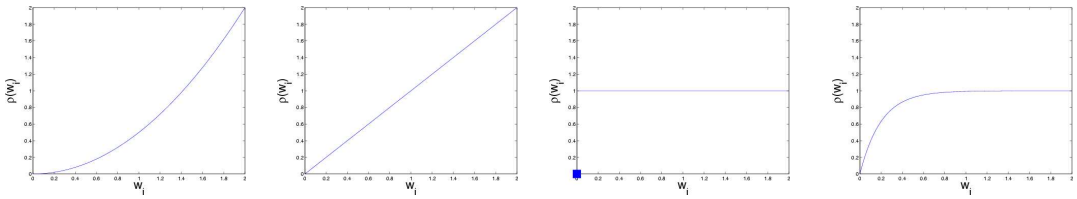
where again $n_{\pm 1} = |\{i : y_i = \pm 1\}|$. For equiprobable classes, both versions are equivalent.

5.2.2. Feature Penalties

In general, optimisation approaches to statistical classification include an additional penalty term ρ beside a "goodness of fit" term as f_{RLP} in (5.1) whose competition is controlled by a weight parameter $\lambda \in [0, 1)$:

$$\min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} (1 - \lambda) f_{\mathrm{RLP}}(\mathbf{w}, b) + \lambda \rho(\mathbf{w}) \ . \tag{5.3}$$

For example for the SVM, to maximise the margin between the two parallel hyperplanes, $\rho(\mathbf{w}) = 1/2 \, \|\mathbf{w}\|_2^2$ is used. In order to concurrently suppress features, we consider different feature penalties in the following.



$\rho(\mathbf{w}) = 1/2\,\|\mathbf{w}\|_2^2$ $\rho(\mathbf{w}) = \|\mathbf{w}\|_1$ $\rho(\mathbf{w}) = \|\mathbf{w}\|_0^0$ $\rho(\mathbf{w}) = \mathbf{e}^\top \left(\mathbf{e} - e^{-\alpha|\mathbf{w}|}\right) \approx \|\mathbf{w}\|_0^0$

Figure 5.2.: Feature selection penalties

ℓ_1–SVM

Feature selection is achieved by suppressing components of the normal vector $\mathbf{w}$ to the separating hyperplane since a component of the weight vector $\mathbf{w}$ is non-zero if and only if the feature is used. Proposed feature penalty terms are $\rho(\mathbf{w}) = \|\mathbf{w}\|_p^p$ for $0 \leq p < 2$ (see, e.g., [Bradley and Mangasarian, 1998, Daubechies et al., 2004]) as illustrated in Fig. 5.2. In [Bradley and Mangasarian, 1998], the ℓ_1-norm (lasso penalty) $\rho(\mathbf{w}) = \|\mathbf{w}\|_1$ leads to good feature selection and classification results. Accordingly, (5.3) reads

$$\min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} (1-\lambda) \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))_+ + \lambda \mathbf{e}^\top |\mathbf{w}| \ ,$$

which can be solved by a linear program. This penalty term was originally introduced in the statistical context of linear regression in the 'lasso' ('Least Absolute Shrinkage and Selection Operator') by [Tibshirani, 1996], and also applied by [Zhu et al., 2004].

Feature Selection Concave (FSV)

As can be guessed from Fig. 5.2, the most apparent penalty term is the dimension of the feature space, the so-called ℓ_0-"norm" $\rho(\mathbf{w}) = \|\mathbf{w}\|_0^0 = \lim_{p\to 0} \|\mathbf{w}\|_p^p = |\{i : w_i \neq 0\}|$ [Bradley and Mangasarian, 1998, Weston et al., 2003]. Note that $\|\cdot\|_0$ is no norm because the canonical definition for ℓ_p-"norms" for $p < 1$ does not fulfil the triangle inequality any longer. Since the ℓ_0-"norm" is non-smooth, it was approximated by [Bradley and Mangasarian, 1998] by the continuous concave functional

$$\rho(\mathbf{w}) = \mathbf{e}^\top (\mathbf{e} - e^{-\alpha|\mathbf{w}|}) \approx \|\mathbf{w}\|_0^0 \tag{5.4}$$

with approximation parameter $\alpha \in \mathbb{R}_+$ illustrated in Fig. 5.2 right. The larger α is, the better is the approximation, but the authors fix its value to 5 in their experiments as large values may lead to numerical instability of the solution algorithm. A logarithmic approximation of the ℓ_0-"norm" is used by [Weston et al., 2003]. This penalty tends to $-\infty$ if w_i tends to zero and therefore suggests their fast iterative method for the solution.

Problem (5.3) with penalty term (5.4) yields with suitable constraints the mathematical program:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & (1-\lambda)\mathbf{e}^\top\boldsymbol{\xi} + \lambda\mathbf{e}^\top(\mathbf{e}-e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1-\xi_i \quad , \quad i=1,\ldots,n \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ , \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \ . \end{aligned} \tag{5.5}$$

For the error weighting as in (5.2), we obtain

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}^+\in\mathbb{R}^n,\, \boldsymbol{\xi}^-\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & (1-\lambda)\left(\frac{\mathbf{e}^\top\boldsymbol{\xi}^+}{n_{+1}} + \frac{\mathbf{e}^\top\boldsymbol{\xi}^-}{n_{-1}}\right) + \lambda\mathbf{e}^\top(\mathbf{e}-e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & \mathbf{w}^\top\mathbf{x}_i + b \geq 1-\xi_i^+ \quad , \quad y_i = 1 \ , \\ & \mathbf{w}^\top\mathbf{x}_i + b \leq -1+\xi_i^- \quad , \quad y_i = -1 \ , \\ & \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \mathbf{0} \ , \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \end{aligned} \tag{5.6}$$

which is known as *Feature Selection concaVe* (FSV) by [Bradley and Mangasarian, 1998].

Note that the above problems are non-convex, but concave minimisation problems, which are not easy to solve. For this kind of problem, namely

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) \ , \tag{5.7}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is concave, but not necessarily differentiable, and $X \subset \mathbb{R}^d$ is a polyhedral set, [Mangasarian, 1997] shows that the minimum value is always attained at a vertex of the polyhedral feasible set X, so that 'arg min' may be written as 'arg vertex min'. Let the symbol ∂f denote the *super*differential of f, which, for f concave, is the analogue of the subdifferential for (not necessarily differentiable) convex functions introduced in Appendix B. It is a generalisation of $\{\nabla f\}$ to non-differentiable concave functions. For such concave minimisation problems, and especially for problem FSV (5.6), the following iterative algorithm was proposed by [Bradley and Mangasarian, 1998]:

Algorithm 5.2.1: Successive Linearisation Algorithm (SLA)(f, X)

choose $\mathbf{x}^0 \in \mathbb{R}^d$ arbitrarily
for $k \in \mathbb{N}_0$
do $\begin{cases} \text{select } \mathbf{z} \in \partial f(\mathbf{x}^k) \text{ arbitrarily} \\ \text{select } \mathbf{x}^{k+1} \in \arg\text{vertex}\min_{\mathbf{x}\in X} \mathbf{z}^\top(\mathbf{x}-\mathbf{x}^k) \text{ arbitrarily} \\ \textbf{if } \mathbf{z}^\top(\mathbf{x}^{k+1}-\mathbf{x}^k) = 0 \\ \quad \textbf{then return } (\mathbf{x}^k) \end{cases}$

The algorithm produces high quality solutions by a sequence of linear programs and terminates after a finite number of iterations [Mangasarian, 1997].

Applied to FSV (5.6), the SLA gives

Algorithm 5.2.2: SLA FOR FSV($\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}, \lambda$)

choose $\mathbf{v^0} \in \mathbb{R}^d$

for $k \in \mathbb{N}_0$

do

$$\begin{cases} \text{select } (\mathbf{w}^{k+1}, b^{k+1}, \boldsymbol{\xi}^{+k+1}, \boldsymbol{\xi}^{-k+1}, \mathbf{v}^{k+1}) \in \arg\text{vertex} \\ \quad \min\limits_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}^+\in\mathbb{R}^n, \boldsymbol{\xi}^-\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} (1-\lambda)\left(\dfrac{\mathbf{e}^\top\boldsymbol{\xi}^+}{n_{+1}} + \dfrac{\mathbf{e}^\top\boldsymbol{\xi}^-}{n_{-1}}\right) + \lambda\alpha\left(e^{-\alpha\mathbf{v}^k}\right)^\top\mathbf{v} \\ \qquad\qquad \text{subject to} \quad \mathbf{w}^\top\mathbf{x}_i + b \geq 1 - \xi_i^+ \quad , \quad y_i = 1 \ , \\ \qquad\qquad\qquad\qquad \mathbf{w}^\top\mathbf{x}_i + b \leq -1 + \xi_i^- \quad , \quad y_i = -1 \ , \\ \qquad\qquad\qquad\qquad \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \mathbf{0} \ , \\ \qquad\qquad\qquad\qquad -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \\ \textbf{if } (1-\lambda)\left(\frac{\mathbf{e}^\top(\boldsymbol{\xi}^{+k+1} - \boldsymbol{\xi}^{+k})}{n_{+1}} + \frac{\mathbf{e}^\top(\boldsymbol{\xi}^{-k+1} - \boldsymbol{\xi}^{-k})}{n_{-1}}\right) + \lambda\alpha\left(e^{-\alpha\mathbf{v}^k}\right)^\top\left(\mathbf{v}^{k+1} - \mathbf{v}^k\right) = 0 \\ \quad \textbf{then return } (\mathbf{w}^k, b^k) \end{cases}$$

5.2.3. FSV Evaluation

FSV solved by the SLA was extensively evaluated and tested for problems of the UCI repository [Blake and Merz, 1998] and of [Weston et al., 2003] in the diploma thesis "Feature Selection with Concave Minimization" [Jakubik, 2003]. The material in this and the following section is adopted from the thesis.

The algorithm was implemented in MATLAB using the simplex algorithm by CPLEX [Ilog, Inc., 2001] to solve the linear programs. The results are:

- The algorithm was able to perform feature selection fast and with a stable behaviour.
- Agreeing with [Bradley and Mangasarian, 1998], the value of the ℓ_0-"norm" approximation parameter may be fixed to $\alpha = 5$ unless numerical problems occur (e.g., for badly scaled data).
- But the solution is depending on the initial value $\mathbf{v}^0$: Choosing $\mathbf{v}^0 = \mathbf{0}$ suppresses much more features than $\mathbf{v}^0 = \mathbf{1}$ or $\mathbf{v}^0 = |\mathbf{w}|$ where $\mathbf{w}$ is the RLP solution.
- As designed, the weight parameter λ steers the feature selection. Figure 5.3 shows how the classification accuracy and the problem dimension change with the value of λ for the nine-dimensional 'Breast Cancer Wisconsin' data set with SLA start value computed by RLP.

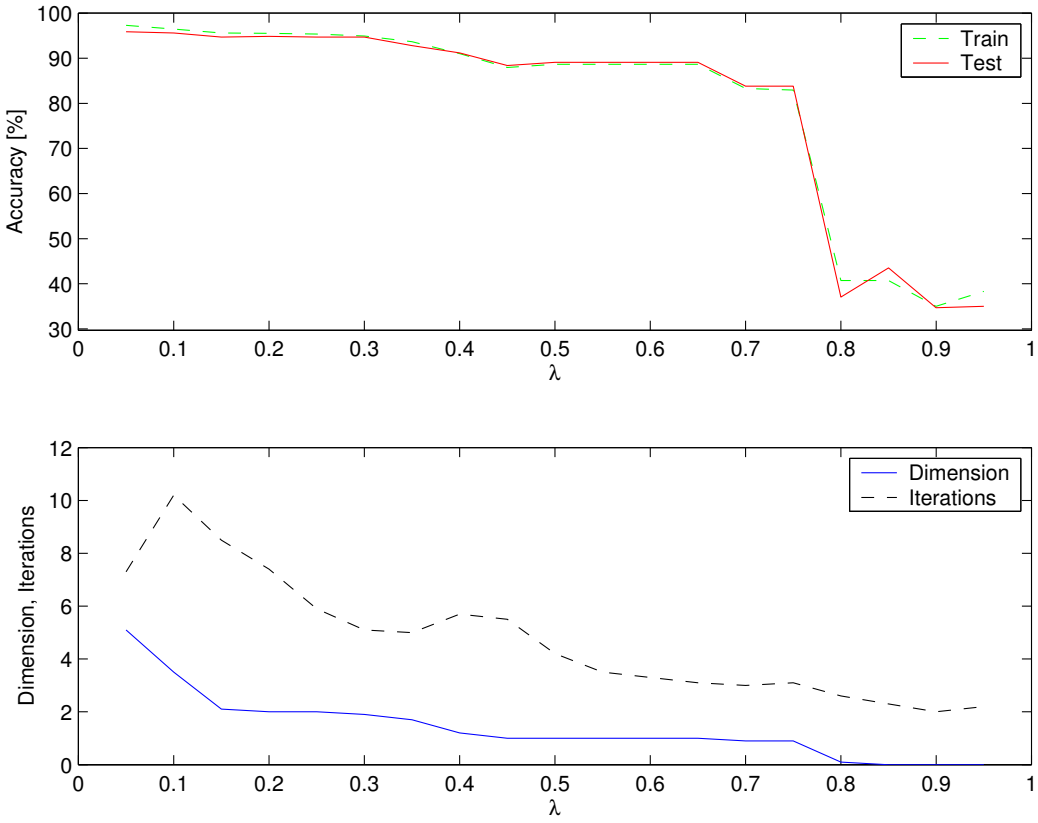


Figure 5.3.: FSV accuracy and problem dimension depending on the parameter λ

- In detailed tests with further problems, most of them also introduced in Sec. 5.6.2, the FSV performance with λ chosen by cross-validation to minimise the test classification error is compared with RLP and soft margin SVMs with linear and Gaussian kernels, where the SVM weight parameter C is also chosen by cross-validation. As argued in Sec. 5.1 to motivate feature selection, the resulting test error may even be smaller than for the appropriate classifier (RLP) applied to the original problem whereas the problem dimension is often reduced as can be seen in Table 5.1.

- But concerning the classification error, we observe that even a linear SVM achieves a much better classification than RLP, similarly on the reduced feature sets (SVM (reduced) versus FSV). The Gaussian SVM is still more accurate.

5.3. Wavelet Feature Selection by FSV

As the selection of a wavelet that is best suited for the feature extraction for a given classification problem is also a feature selection problem, a further aim of the diploma thesis [Jakubik, 2003] was to apply FSV to this problem. We consider texture classification with features as defined in Sec. 2.2.

To apply FSV, we have to provide the features for all considered wavelets at once. Again, we therefore discretise the parameter space resulting of the lattice factorisation (2.4) as also done in Sec. 4.3.2. We again use filters of length six with $L = 2$ in (2.4). If we discretise each parameter with resolution p, we generate feature vectors of dimension $d = 3qp^2$ for q levels of the 2D non-standard wavelet transform by concatenating the feature vectors for all p^2 wavelets.

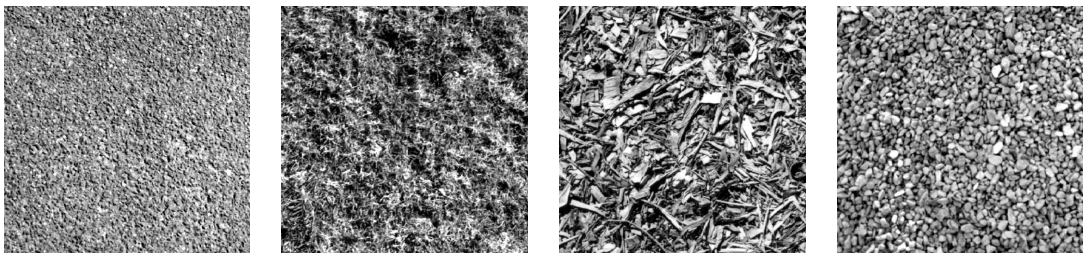
FSV does not take into account the correspondence of a wavelet to its features. As a consequence, FSV may select features from many different wavelets as an optimal feature set. Effectively, the filter operators $F_{\boldsymbol{\theta}}$ for all selected wavelets then still have to be applied for the classification of new samples. Therefore, we modify FSV to use this background knowledge.

To force FSV to retain or to discard all features corresponding to a wavelet, instead of penalising the number of features, we penalise the number of wavelets. For our new wavelet indicator variables $\mathbf{v}$ of dimension $p^2 < d$, the constraints couple all associated wavelet features to the single wavelet indicator variable. To formalise this, we introduce the feature mapping

$$\begin{aligned} \text{group} : \{1, \ldots, d\} &\rightarrow \{1, \ldots, p^2\} \ , \\ i &\mapsto \left\lceil \frac{i}{3q} \right\rceil \ . \end{aligned}$$

	FSV			RLP	SVM		SVM (reduced)	
	$\mathbf{v}^0$: RLP	$\mathbf{v}^0 = \mathbf{e}$	$\mathbf{v}^0 = \mathbf{0}$		linear	Gauss.	linear	Gauss.
	train.	train.	train.	train.	train.	train.	train.	train
	test	test	test	test	test	test	test	test
	λ^*	λ^*	λ^*		$\ln C^*$	$\ln C^*$	$\ln C^*$	$\ln C^*$
data set	dim	dim	dim	dim	dim	dim	dim	dim
bcw	3	3	3	3	3	3	3	2
	4	4	4	3	3	3	3	3
	0.05	0.05	0.05		0	-2	0	-1
	5.1	5.1	3.9	9	9	9	6	6
liver	32	32	32	32	29	15	29	15
	36	36	37	34	33	30	33	30
	0.05	0.05	0.05		4	6	4	6
	5.9	5.9	5.9	6	6	6	6	6
pima	23	23	25	23	22	20	22	20
	25	25	25	25	23	23	23	23
	0.05	0.05	0.05		18	0	18	0
	7.8	7.8	6.6	8	8	8	8	8
tic tac toe	2	2	35	2	2	0	2	0
	2	2	35	2	2	0	2	0
	0.05	0.05	0.2		16	9	16	9
	9	9	0	9	9	9	9	9
wdbc	0	2	4	0	2	1	2	0
	5	4	6	6	2	2	2	4
	0.75	0.05	0.05		0	1	0	9
	23	6	5	30	30	30	27	27
wpbc24	20	29	25	15	13	14	11	15
	33	30	29	33	19	17	18	18
	0.15	0.95	0.10		2	-1	5	-1
	21.5	1.0	5.0	32	32	32	23	23
wpbc60	22	29	29	13	23	18	25	14
	39	29	29	41	31	30	32	30
	0.65	0.50	0.50		0	-2	1	1
	18.1	1.4	1.4	32	32	32	16	16
ionosphere	14	10	17	5	7	1	11	11
	14	14	19	16	12	9	11	11
	0.90	0.20	0.05		0	0	18	1
	2.4	7.5	2.3	34	34	34	2	2
microarray	18	1	12	0	0	0	20	19
	25	22	13	42	13	17	20	17
	0.65	0.1	0.15		7	9	9	3
	3.3	3.9	1.8	2000	2000	2000	1	1

Table 5.1.: Tenfold cross-validation average performance (error [%], number of features) with parameter values (for λ / C) chosen via smallest test error



'Asphalt.0000' 'Grass.0000' 'Misc.0000' 'Rock.0023'

Figure 5.4.: Texture images from the MeasTex collection [Smith, 1997]

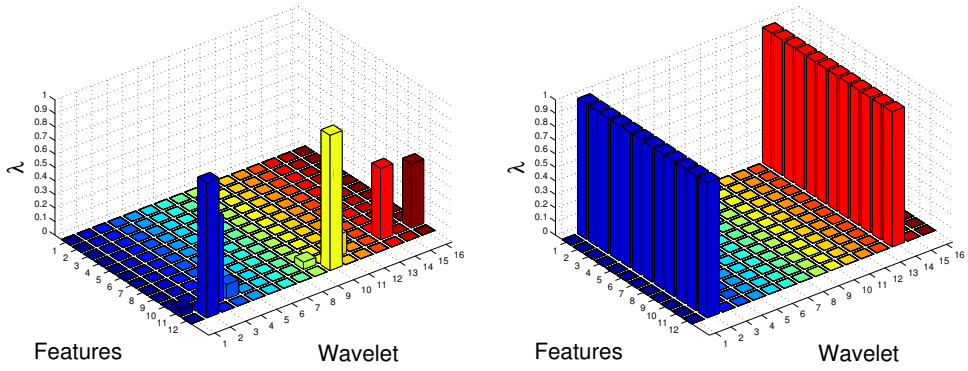
Now FSV with grouped features (gFSV) reads

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}^+\in\mathbb{R}^n,\, \boldsymbol{\xi}^-\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^{p^2}} \quad & (1-\lambda)\left(\frac{\mathbf{e}^\top\boldsymbol{\xi}^+}{n_{+1}} + \frac{\mathbf{e}^\top\boldsymbol{\xi}^-}{n_{-1}}\right) + \lambda\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & \mathbf{w}^\top\mathbf{x}_i + b \geq 1 - \xi_i^+ \quad , \quad y_i = 1 \ , \\ & \mathbf{w}^\top\mathbf{x}_i + b \leq -1 + \xi_i^- \quad , \quad y_i = -1 \ , \\ & \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \mathbf{0} \ , \\ & -v_{\text{group}(i)} \leq w_i \leq v_{\text{group}(i)} \quad , \quad i = 1, \ldots, d \ . \end{aligned}$$

We now present the results of using both feature selection approaches on real-world textures. Figure 5.4 shows the four different grey value textures (512×512 pixels) that are used for the classification; the images will be denoted by the first part of their name. The textures are normalised to span the full range of grey values. As done in Sec. 3.5, the images are split into fragments of 64×64 pixels. The first 16 samples are used for training and the remaining 48 are used for testing the classifier.

In our experiments with the 4-level non-standard decomposition, the discretisation resolution p is set to 4 or 8. Figure 5.5 shows the relevant wavelet features subject to the parameter λ for the binary classification task 'asphalt - grass'. The RLP solution is used as initial value for the SLA. In Fig. 5.5, one can see that the features selected by FSV are robust subject to λ and the number of features is already heavily reduced for small values of λ. As λ becomes larger, even fewer wavelet features are selected. It is also instructive that only features corresponding to subbands eleven and twelve on level one are selected by FSV. Due to the effect of the feature grouping, gFSV finds another solution for the same classification task, which is in this particular case independent of the chosen value of λ. Some features of the two selected wavelets are also chosen by FSV. All other wavelets are suppressed.

Figure 5.6 illustrates the selected filter angles for the classification task 'grass - rock'. The colour marks the relevance of the wavelet: Black means that it is never chosen for any value of λ, white means it is chosen for all values of λ, grey marks something in



FSV gFSV

Figure 5.5.: Selected wavelet features for the classification task 'asphalt - grass'

data set	asphalt	grass	misc	rock
asphalt	-	1	0	2
grass	-	-	6	17
misc	-	-	-	14
rock	-	-	-	-

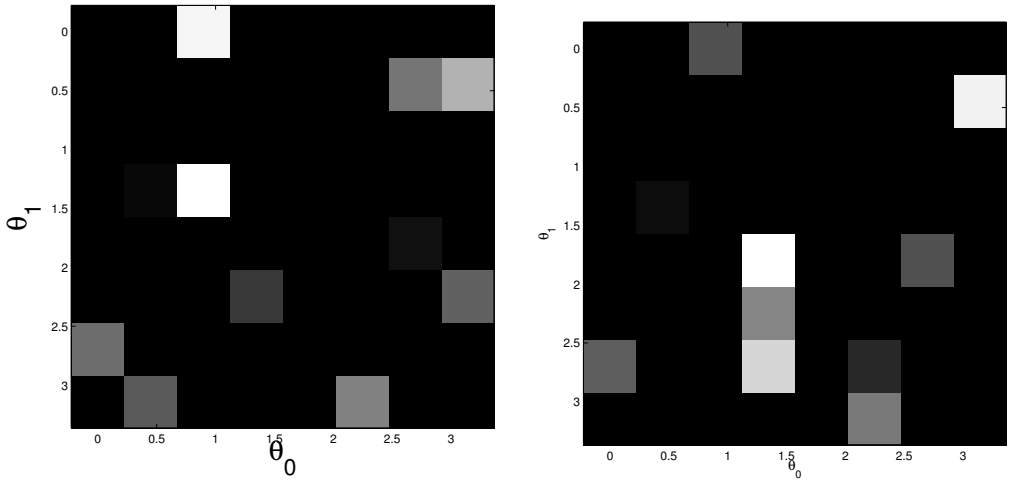
Table 5.2.: FSV test error [%] for wavelet texture classification with λ chosen by cross-validation

between. Again, we find that some wavelets are chosen for both approaches - they seem to be strongly relevant for the classification - and others are irrelevant. The number of relevant wavelets is low.

Now we have a look at the resulting classifiers' performance. Again we determine the value of λ by cross-validation. Tables 5.2 and 5.3 summarise the test error of FSV and gFSV for all binary combinations of the four textures. Table 5.4 shows the respective number of features and wavelets used. On the one hand, the number of features for gFSV is higher than for FSV but on the other hand the number of used wavelets is lower. Thus, as the test error is nearly the same, it is better to use our proposed gFSV which requires fewer wavelet decompositions for classification.

Table 5.5 summarises the test error for RLP on all considered wavelet features for all texture combinations. The accuracy of RLP is also good but for the cases 'asphalt - grass' and 'asphalt - misc' FSV and gFSV find better solutions yet with fewer features and wavelets.

To get an idea how well FSV and gFSV work we also compare their test error with that for a standard wavelet. Table 5.6 summarises the test error of RLP with wavelet features calculated with the Daubechies wavelet with three vanishing moments for all texture combinations. In most cases, FSV and gFSV find better solutions than RLP



FSV gFSV

Figure 5.6.: Selected wavelets averaged over λ for the classification task 'grass - rock'

data set	asphalt	grass	misc	rock
asphalt	-	1	0	1
grass	-	-	13	8
misc	-	-	-	10
rock	-	-	-	-

Table 5.3.: gFSV test error [%] for wavelet texture classification with λ chosen by cross-validation

data sets	FSV features wavelets	gFSV features wavelets
asphalt - grass	4	12
	4	1
asphalt - misc	4	24
	4	2
asphalt - rock	3	24
	3	2
grass - misc	7	48
	7	4
grass - rock	7	60
	5	5
misc - rock	9	96
	7	8

Table 5.4.: Number of features/wavelets determined by FSV and gFSV with λ chosen by cross-validation

data set	asphalt	grass	misc	rock
asphalt	-	5	6	0
grass	-	-	7	0
misc	-	-	-	6
rock	-	-	-	-

Table 5.5.: RLP test error [%] for wavelet texture classification on the whole feature set

data set	asphalt	grass	misc	rock
asphalt	-	5	0	2
grass	-	-	16	0
misc	-	-	-	4
rock	-	-	-	-

Table 5.6.: RLP test error [%] for wavelet texture classification with the Daubechies 3 wavelet

with this special wavelet.

In summary, we showed how FSV can be used to select wavelet features for texture classification. Therefore, we presented an extension - gFSV - to Bradley and Mangasarian's original FSV algorithm, which takes into account the whole wavelet and not only single wavelet features. Both approaches achieve good classification results with only a few features. The difference between the FSV and gFSV solutions is that FSV selects few subbands of many wavelets while gFSV selects all subbands of few wavelets.

5.4. New Feature Selection Approaches

SVMs with linear and especially with Gaussian kernels perform better than a linear approach (cf. results in Sec. 5.2.3). So we try to perform well-generalising embedded feature selection for linear and nonlinear classifiers.

Our first objective is to extend the FSV approach with the aim to improve the generalisation performance of the linear classifier. Taking into account that the SVM provides good generalisation ability by its ℓ_2 regulariser $\|\mathbf{w}\|_2^2$, we propose new methods by introducing additional regularisation terms. Of course these approaches are only of interest if the corresponding non-convex problems may be solved. We tackle that later by formulating the tasks as d.c. problems and applying the appropriate d.c. algorithm in Sec. 5.5.

As a second goal, we construct *direct objective minimising feature selection* methods for nonlinear SV classifiers. Due to the non-quadratic feature penalties, it is not possible to apply the "kernel trick" to nonlinear feature maps in the same manner as for SVMs. First, we generalise the approach for the quadratic SVM of [Zhu et al., 2004] in two directions: We apply the approximate ℓ_0 penalty considered superior to the ℓ_1-norm by [Bradley and Mangasarian, 1998] and we focus on feature selection in the *original* feature space to further improve the performance and enable semantics analysis. Second, we incorporate "kernel – target alignment" [Cristianini et al., 2002] within this framework which performs appropriate feature selection if, e.g., the Gaussian kernel SVM is used as classifier.

We present the approaches in detail in the following subsections. A summary of our algorithms has also appeared in [Neumann et al., 2004], a detailed description is

[Neumann et al., 2005a].

Further approaches that may be interesting to examine include a variable ranking (filter) approach according to the mixture scatter components, which are the diagonal elements of $\mathbf{S}_m$. Besides, gradient descent to the error bound (4.2) is applied by [Weston et al., 2001, Chapelle et al., 2002] to suppress features through expressions of the kernel K. This can be achieved, e.g., by scaling componentwise with $\boldsymbol{\sigma} \in \mathbb{R}^d$ in the Gaussian kernel (2.20) instead of $\sigma \in \mathbb{R}$. But the approach requires evaluation of the derivatives of the SVM solution variables $\boldsymbol{\alpha}$ depending on the kernel. Then for every gradient evaluation an SVM has to be determined. (Kernel derivatives are also used by [Hermes and Buhmann, 2000, Heiler et al., 2001]). Another idea is to penalise a least squares SV classifier [Suykens et al., 2002] by the ℓ_1 penalty.

5.4.1. Combined ℓ_p Penalties

FSV performs well for feature selection. However, its classification accuracy can be improved by applying a standard SVM on the selected features only, as shown by [Jakubik, 2003] (see also Sec. 5.2.3) and also indicated by [Weston et al., 2003]. Therefore, since the ℓ_2 penalty term is responsible for the good SVM classification results while the ℓ_1 and ℓ_0 penalty terms focus on feature selection, we suggest combinations of these terms. Consequently, we need two weight parameters $C, D \in \mathbb{R}_+$.

ℓ_1–ℓ_2–SVM

Denote again by $\mathbf{X} \in \mathbb{R}^{n \times d}$ the matrix of transposed pattern vectors ($\mathbf{X}_{i\cdot} = \mathbf{x}_i^\top$ for $i = 1, \ldots, n$) and by $\mathbf{Y}$ the diagonal label matrix. For the ℓ_1–ℓ_2–SVM, we are interested in solving the constrained convex QP

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & \frac{C}{n}\mathbf{e}^\top\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^\top\mathbf{w} + D\mathbf{e}^\top\mathbf{v} \\ \text{subject to} \quad & \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi} \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ , \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \ . \end{aligned} \tag{5.8}$$

This is just the SVM problem (2.28) for a linear kernel with bias term and with the last term added to the objective function. Here, it is advantageous to replace the first weighting factor C in the objective function by C/n. Then the optimal value for C is independent of the training set size, especially for large problems.

Alternatively to our implementation of the weight parameters C, D, a convex combination of the objective terms

$$(1 - \lambda - \mu)\frac{1}{n}\mathbf{e}^\top\boldsymbol{\xi} + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} + \mu\mathbf{e}^\top\mathbf{v}$$

for $\lambda, \mu \in \mathbb{R}_+$, $\lambda + \mu < 1$ may be easier to discretise; or the parameter C may weight the second term as regularisation weight parameter also.

ℓ_0–ℓ_2–SVM

For the ℓ_0–ℓ_2–*SVM* with approximate ℓ_0-"norm", we examine

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & \frac{C}{n}\mathbf{e}^\top\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^\top\mathbf{w} + D\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi} \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ , \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \ . \end{aligned} \tag{5.9}$$

An appropriate approach to optimise (5.9) is developed in Sec. 5.5.

5.4.2. Nonlinear Classification

When trying to penalise features in a kernel classification problem, several difficulties occur:

- The Representer Theorem does not hold for a not purely quadratic functional as, e.g., for the ℓ_1–SVM or FSV. As we have no explicit kernel expression of our hyperplane, the decision boundary has to be determined in feature space.
- Upon extension of the SVM's functional (2.28a) by a feature penalty, as, e.g., for the ℓ_1–ℓ_2–SVM, the dual problem retains variables related to the primal space, which is impractical for many feature maps.
- Besides, a feature penalty $\|\mathbf{w}\|$ for $\mathbf{w} \in \mathcal{F}_K \approx \boldsymbol{\phi}(\mathbb{R}^d)$ is questionable in terms of the original features in R^d.

So a dual approach with feature penalty is not practicable. Hence, we consider two popular feature maps $\boldsymbol{\phi} : \mathbb{R}^d \to \mathcal{F}_K$ as introduced in Sec. 2.3.2 in connection with different feature selection approaches:

Quadratic FSV

We examine the simplest common generalisation to linear decision surfaces as, e.g., also done by [Zhu et al., 2004]: the quadratic feature map

$$\begin{aligned} \boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^{d'} \ , \quad \mathbf{x} \ &\mapsto \ (\mathbf{x}^{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}_0^d \ , \, 0 < \|\boldsymbol{\alpha}\|_1 \leq 2) \\ &= (x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} : \boldsymbol{\alpha} \in \mathbb{N}_0^d \ , \, 0 < \|\boldsymbol{\alpha}\|_1 \leq 2) \\ &= (x_i^k x_j : i, j = 1, \ldots, d, \, k = 0, 1, \, i - 1 \leq k(j-1)) \\ &= (\widetilde{x}_i \widetilde{x}_j : \widetilde{\mathbf{x}}^\top = (1, \mathbf{x}^\top), \, 1 \leq i \leq j \leq d+1, \, ij \neq 1) \ , \end{aligned}$$

where $d' = d(d+3)/2$. As $d' = \dim \mathcal{F}_K < \infty$, instead of applying the "kernel trick" and solving a dual problem we generate nonlinear decision surfaces by explicitly carrying out $\boldsymbol{\phi}$ and suppress features in $\mathcal{F}_K$.

Straightforward application of uniform FSV (5.5) with approximate ℓ_0 penalty in $\mathbb{R}^{d'}$ leads to the minimisation problem

$$(1-\lambda)\sum_{i=1}^{n}(1-y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i)+b))_+ + \lambda\mathbf{e}^\top(\mathbf{e}-e^{-\alpha\mathbf{v}}) + \sum_{i=1}^{d'}\chi_{[-v_i,v_i]}(w_i)$$

for $\mathbf{w}\in\mathbb{R}^{d'}$, $b\in\mathbb{R}$ and $\mathbf{v}\in\mathbb{R}^{d'}$. This approach, as well as a similar one for the ℓ_1 penalty in [Zhu et al., 2004], achieve feature selection only in the *transformed* feature space $\mathbb{R}^{d'}$. Our goal, however, is to select features in the *original* space $\mathbb{R}^d$ in order to get insight into our original problem, too, and to reduce the number of primary features. To this end, instead of penalising v_i for $\mathbf{v}\in\mathbb{R}^{d'}$, we examine for each w_i $(i=1,\ldots,d')$ which original features are included in computing ϕ_i. So we replace the constraints $\chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w})$ by '$\chi_{[-\boldsymbol{\phi}(\mathbf{v}),\boldsymbol{\phi}(\mathbf{v})]}(\mathbf{w})$' for $\mathbf{v}\in\mathbb{R}^d$. With $\mathbf{e}_j\in\mathbb{R}^d$ denoting the jth unit vector, taking the maximal bound for each v_i leads to

$$\begin{aligned} f(\mathbf{w},b,\mathbf{v}) :=& (1-\lambda)\sum_{i=1}^{n}(1-y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i)+b))_+ + \lambda\mathbf{e}^\top(\mathbf{e}-e^{-\alpha\mathbf{v}}) \\ &+ \sum_{i=1}^{d'}\sum_{\phi_i(\mathbf{e}_j)\neq 0}\chi_{[-v_j,v_j]}(w_i) \qquad \longrightarrow \min_{\mathbf{w}\in\mathbb{R}^{d'},\, b\in\mathbb{R},\, \mathbf{v}\in\mathbb{R}^d} . \end{aligned} \tag{5.10}$$

In total, this gives $2*d*(d+1)$ linear coupling constraints, but the constraint matrix is sparse, which can be taken into account during optimisation. In the following, we refer to (5.10) as *quadratic FSV*. In principle, the approach can be extended to other explicit feature maps $\boldsymbol{\phi}$, especially by choosing other polynomial degrees.

Due to the higher computational requirements, we first try this direct approach with FSV. In the same manner as done for FSV here, it is possible to generalise the ℓ_p–ℓ_2–SVMs for $p=0,1$ by explicitly applying, e.g., the quadratic feature map.

Kernel – Target Alignment Approach

Compared with linear SVMs, further improvements of classification accuracy in our context may be achieved by using Gaussian kernel SVMs, as has been confirmed by experiments in [Jakubik, 2003] (see also Sec. 5.2.3). Therefore, we also consider SVMs with the feature map $\boldsymbol{\phi}:\mathcal{X}\to\ell_2$ induced by $K(\mathbf{x},\mathbf{z})=\langle\boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{z})\rangle$ for the Gaussian kernel

$$K(\mathbf{x},\mathbf{z}) = K_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) = e^{-\|\mathbf{x}-\mathbf{z}\|^2_{2,\boldsymbol{\theta}}/(2\sigma^2)} \tag{5.11}$$

with componentwise weighted ℓ_2-norm $\|\mathbf{x}\|_{2,\boldsymbol{\theta}}^2 = \sum_{k=1}^d \theta_k |x_k|^2$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$. As the feature space has infinite dimension, feature selection as done for the quadratic feature map is no longer applicable. We apply the common SV classifier without bias term b. We obtain the commonly used kernel and classifier for $\boldsymbol{\theta} = \mathbf{e}$. Direct feature selection, i.e., the setting of as many θ_k to zero as possible while retaining or improving the classification ability, is a difficult problem. One possible approach is to use a wrapper as in [Weston et al., 2001]. Instead, we aim at directly maximising the alignment (4.3) also used for feature selection by [Steel and Hechter, 2004]. To simplify this optimisation task, we drop the denominator, which is justified in view of the boundedness of the kernel elements (5.11). To cope with unequal sample partitioning as, e.g., in Fig. 5.7 left on p. 132, we replace $\mathbf{y}$ by $\mathbf{y}_n = (y_i/n_{y_i})_{i=1}^n$. This leads to

$$\begin{aligned} \mathbf{y}_n^\top \mathbf{K} \mathbf{y}_n &= \sum_{i,j=1}^n \frac{y_i}{n_{y_i}} \frac{y_j}{n_{y_j}} \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \left\langle \sum_{i=1}^n \frac{y_i}{n_{y_i}} \boldsymbol{\phi}(\mathbf{x}_i), \sum_{j=1}^n \frac{y_j}{n_{y_j}} \boldsymbol{\phi}(\mathbf{x}_j) \right\rangle \\ &= \left\| \frac{1}{n_{+1}} \sum_{\{i:y_i=1\}} \boldsymbol{\phi}(\mathbf{x}_i) - \frac{1}{n_{-1}} \sum_{\{i:y_i=-1\}} \boldsymbol{\phi}(\mathbf{x}_i) \right\|^2 \geq 0 \ , \end{aligned} \tag{5.12}$$

which is the class centre distance in feature space, and further makes the magnitude of our criterion independent of the number of training samples. A different view on the alignment criterion is obtained by considering the linear classifier F in feature space with $\mathbf{w} = \sum_{i=1}^n y_{ni} \boldsymbol{\phi}(\mathbf{x}_i)$, $b = 0$. Then maximising the correct class responses $\sum_{i=1}^n y_{ni} F(\mathbf{x}_i)$ also leads to the expression above. Adding penalty (5.4) and bounds for $\boldsymbol{\theta}$, we define as our *kernel – target alignment approach* to feature selection

$$f(\boldsymbol{\theta}) := -(1-\lambda) \frac{1}{2} \mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n + \lambda \frac{1}{d} \mathbf{e}^\top (\mathbf{e} - e^{-\alpha \boldsymbol{\theta}}) + \chi_{[\mathbf{0},\mathbf{e}]}(\boldsymbol{\theta}) \longrightarrow \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \tag{5.13}$$

for $\lambda \in [0, 1)$ again. Some remarks to the objective criterion:

- The alignment term (5.12) is also bounded from above:

$$\begin{aligned} \mathbf{y}_n^\top \mathbf{K} \mathbf{y}_n &= \frac{1}{n_{+1}^2} \sum_{\substack{i,j=1 \\ y_i=y_j=1}}^n K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n_{-1}^2} \sum_{\substack{i,j=1 \\ y_i=y_j=-1}}^n K(\mathbf{x}_i, \mathbf{x}_j) \\ &\quad - \frac{2}{n_{+1} n_{-1}} \sum_{\substack{i,j=1 \\ y_i=1, y_j=-1}}^n K(\mathbf{x}_i, \mathbf{x}_j) \\ &\leq 1 + 1 - 0 = 2 \ . \end{aligned}$$

  Consequently, the scaling factors $1/2$, $1/d$ ensure that both objective terms take values in $[0, 1]$ so that their convex combination with weight λ is bounded in $[0, 1]$ also.

- Considering the boundary values, it follows for $\boldsymbol{\theta} = \mathbf{0}$ that $\mathbf{K}_{\boldsymbol{\theta}} = (1)_{n\times n}$ and $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n = 0$. (Using $\mathbf{y}$ instead of $\mathbf{y}_n$ leads to $(n_{+1} - n_{-1})^2$.)

- For $\boldsymbol{\theta} \to \infty$, we have $\mathbf{K}_{\boldsymbol{\theta}} \to \mathbf{I}$ and $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n \to \frac{1}{n_{+1}} + \frac{1}{n_{-1}}$. (Using $\mathbf{y}$ leads to n.)

- This suggests that (5.12) has a maximum. From realistic experiments, $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n$ is mostly, but not always, a unimodal function on $\mathbb{R}_+^d$, and in general increasing for $\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{e}$ for a reasonable value of σ (unless $n_{+1} n_{-1} = 0$).

- It is essential here that the features are normalised as their variances influence the objective with initially equal weights. In experiments, it shows that otherwise features with large variances are preferred.

- As the entries of the kernel matrix $K_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) > 0$ are monotonically decreasing in $\boldsymbol{\theta}$ we have the same for $\|\mathbf{K}_{\boldsymbol{\theta}}\|_{\mathrm{F}} = (\sum_{i,j} K_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)^2)^{1/2}$ and, hence, $\|\mathbf{K}_{\mathbf{0}}\|_{\mathrm{F}} \geq \|\mathbf{K}_{\boldsymbol{\theta}}\|_{\mathrm{F}} \geq \|\mathbf{K}_{\mathbf{e}}\|_{\mathrm{F}}$ for $\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{e}$, i.e., $n \geq \|\mathbf{K}_{\boldsymbol{\theta}}\|_{\mathrm{F}} \geq \|\mathbf{K}\|_{\mathrm{F}} \geq \sqrt{n}$ by the two former points. To get the alignment's denominator $\|\mathbf{K}_{\boldsymbol{\theta}}\|_{\mathrm{F}}$ small, $\boldsymbol{\theta}$ has to be large in conflict with the objective term $\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\boldsymbol{\theta}})$. But when the features are normalised, the term $\|\mathbf{K}_{\boldsymbol{\theta}}\|_{\mathrm{F}}$ does not much influence the decision which features to suppress.

- We intend to find $\boldsymbol{\theta} \in \{0,1\}^d$, which is in most cases implicitly satisfied due to the nature of the objective.

The minimisation problem (5.13) is subjected to bound constraints only, but the variable $\boldsymbol{\theta}$ is included in the exponential norm expressions in the first term as well as in the concave second term. As a result, the problem is likely to have local minima and is difficult to solve. This is treated in the next section.

A further interesting point is whether it is possible to simultaneously adapt the kernel width σ for the kernel (5.11) either by allowing $\boldsymbol{\theta} > \mathbf{e}$ or by a large initial choice of σ. But from our experiments so far, the optimisation can be expected to be slower then.

5.5. D.C. Decomposition and Optimisation

Whereas RLP (5.1), SVM (2.32) and ℓ_1–ℓ_2–SVM (5.8) are still convex QPs, adding the concave penalty term (5.4) makes problems FSV (5.6), the ℓ_0–ℓ_2–SVM (5.9), quadratic FSV (5.10) and, particularly, the kernel – target alignment approach (5.13) difficult to solve due to possible local minima.

A robust algorithm for minimising non-convex problems is the *Difference of Convex functions Algorithm* (DCA) proposed by [Pham Dinh and Hoai An, 1998] in a different context. Based on the theory of convex analysis summarised in Appendix B, it can be

used to minimise a *non-convex* function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ that reads

$$f(\mathbf{x}) = g(\mathbf{x}) - h(\mathbf{x}) \longrightarrow \min_{\mathbf{x} \in \mathbb{R}^d} , \tag{5.14}$$

where $g, h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ are lower semi-continuous, proper convex functions. A property of this approach, particularly convenient for applications, is that f may be non-smooth. For example, constraint sets $C \ni x$ may be taken into account by adding a corresponding indicator function χ_C to the objective f. Besides, the set of d.c. functions is closed under multiplication and contains, e.g., all functions whose gradient is locally Lipschitzian (cf. [Pham Dinh and Hoai An, 1998]). But an example of a function that cannot be modelled as a d.c. function is the discontinuous ℓ_0-"norm". Due to the convexity, by Prop. 14, discontinuities of g, h may only occur at steps with function value ∞. But as depicted in Fig. 5.2, $\|\cdot\|_0^0$ has a discontinuity at zero not involving infinite value and hence cannot be modelled.

In the next subsections, we first sketch the DCA and then apply it to our non-convex feature selection problems, where the precise algorithm is determined by the appropriate d.c. decomposition of f in each case.

5.5.1. D.C. Programming

Our algorithm presentation is based on the standard notation and results of convex analysis summarised in Appendix B. In the remainder of this section, we apply the following general algorithm, the simplified DCA according to [Pham Dinh and Hoai An, 1998]:

Algorithm 5.5.1: D.C. MINIMISATION ALGORITHM (DCA)(g, h, tol)

choose $\mathbf{x}^0 \in \operatorname{dom} g$ arbitrarily
for $k \in \mathbb{N}_0$
do
- select $\widetilde{\mathbf{x}}^k \in \partial h(\mathbf{x}^k)$ arbitrarily
- select $\mathbf{x}^{k+1} \in \partial g^*(\widetilde{\mathbf{x}}^k)$ arbitrarily
- **if** $\min\left(\left|x_i^{k+1} - x_i^k\right|, \left|\frac{x_i^{k+1} - x_i^k}{x_i^k}\right|\right) \leq \text{tol} \quad \forall\, i = 1, \ldots, d$
  - **then return** $(\mathbf{x}^{k+1})$

We can compute the subgradients by the relations in Prop. 20. The following theorem was proven in [Pham Dinh and Hoai An, 1998, Lemma 3.6, Theorem 3.7]:

Theorem 5 (DCA convergence). *If $g, h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ are lower semi-continuous, proper convex functions so that $\operatorname{dom} g \subset \operatorname{dom} h$ and $\operatorname{dom} h^* \subset \operatorname{dom} g^*$, then it holds for the DCA Algorithm 5.5.1:*

(i) The sequences $\left(\mathbf{x}^k\right)_{k \in \mathbb{N}_0}$, $\left(\widetilde{\mathbf{x}}^k\right)_{k \in \mathbb{N}_0}$ are well defined.

(ii) $\left(f(\mathbf{x}^k) = g(\mathbf{x}^k) - h(\mathbf{x}^k)\right)_{k \in \mathbb{N}_0}$ is monotonously decreasing.

(iii) Every limit point of $\left(\mathbf{x}^k\right)_{k\in\mathbb{N}_0}$ *is a critical point of* $f = g - h$*. In particular, if* $f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k)$*, then* $\mathbf{x}^k$ *is a critical point of* f *in* (5.14)*.*

Notice that the convergence of the DCA is guaranteed without any restrictions concerning parameter choices. For any bounded d.c. function we have convergence to a critical point independently of any parameters.

In the following, we study the application of the DCA to our non-convex feature selection problems. Another example of its application is given by [Schüle et al., 2003].

5.5.2. Application to Direct Objective Minimising Feature Selection

The crucial point in applying the DCA is to define a suitable d.c. decomposition (5.14) of the objective function. The aim of this section is to propose such decompositions for the different approaches under consideration.

FSV

Let us consider the general non-convex problems (5.7) in the d.c. optimisation framework. It turns out that our new feature selection approaches not only generalise the FSV approach, but also that the DCA generalises the SLA: We show that the DCA applied to a *particular* d.c. decomposition (5.14) of (5.7) coincides with the SLA.

Proposition 6 (SLA equivalence). *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *be concave and* $X \subset \mathbb{R}^d$ *be a polyhedral set. Then for solving the concave minimisation problem* (5.7) *the SLA with* $\mathbf{x}^0 \in X$ *and DCA with* $tol = 0$ *are equivalent.*

Proof. Modelling problem (5.7) as a d.c. problem reads

$$\min_{\mathbf{x}\in\mathbb{R}^d} \chi_X(\mathbf{x}) - (-f(\mathbf{x})) \ ,$$

where the first term is defined as function g in (5.14), and the second one as h. Then we have in the DCA Algorithm 5.5.1

- $\mathbf{x}^0 \in \operatorname{dom} g \Leftrightarrow \mathbf{x}^0 \in X$, and for $k \in \mathbb{N}_0$:
- $\widetilde{\mathbf{x}}^k \in \partial h(\mathbf{x}^k) \Leftrightarrow \widetilde{\mathbf{x}}^k \in -\partial f(\mathbf{x}^k)$,
- $\mathbf{x}^{k+1} \in \partial g^*(\widetilde{\mathbf{x}}^k) \overset{\text{Prop. } 20}{\Longleftrightarrow} \mathbf{x}^{k+1} \in \arg\min_{x\in X} -(\widetilde{\mathbf{x}}^k)^\top(\mathbf{x} - \mathbf{x}^k)$.

The problem given in the theorem has exactly the form for which the SLA Algorithm 5.2.1 is defined. Algorithm 5.2.1 and the above DCA are almost identical with $\mathbf{z} = -\widetilde{\mathbf{x}}^k$. If we use tol = 0 in the DCA, choose our start value $\mathbf{x}^0 \in X$ in the SLA and apply, e.g., the simplex algorithm to obtain only vertex solutions, the algorithms are identical. □

Note that again f does not have to be differentiable.

Corollary 7 (FSV algorithms). *For the FSV problems* (5.5) *and* (5.6)*, the algorithms DCA and SLA are equivalent.*

Proof. Proposition 6 □

Note that, e.g. for (5.5), the alternative d.c. decomposition

$$\begin{aligned} g(\mathbf{w}, b, \mathbf{v}) &= (1-\lambda)\mathbf{e}^\top(\mathbf{e} - \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}))_+ + \chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w}) \ , \\ h(\mathbf{v}) &= -\lambda\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) \ . \end{aligned}$$

leads to the same DCA because all linear terms may be assigned to the components g, h arbitrarily.

ℓ_1–ℓ_2–SVM

The ℓ_1–ℓ_2–SVM is a convex problem so that the DCA just amounts to solving it in a single step. Similar to the SVM, it is advisable here to solve the dual problem. The dual to the convex quadratic problem (5.8) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^n, \boldsymbol{\gamma}\in\mathbb{R}^d} \ & \frac{1}{2}(\boldsymbol{\alpha}^\top\mathbf{Y}\mathbf{X}\mathbf{X}^\top\mathbf{Y}\boldsymbol{\alpha} + 4\boldsymbol{\gamma}^\top\mathbf{X}^\top\mathbf{Y}\boldsymbol{\alpha} + 4\boldsymbol{\gamma}^\top\boldsymbol{\gamma}) - (\mathbf{D}\mathbf{e}^\top\mathbf{X}^\top\mathbf{Y} + \mathbf{e}^\top)\boldsymbol{\alpha} - 2\mathbf{D}\mathbf{e}^\top\boldsymbol{\gamma} \\ \text{subject to} \quad & \mathbf{0} \le \boldsymbol{\alpha} \le \frac{C}{n}\mathbf{e} \ , \\ & \mathbf{0} \le \boldsymbol{\gamma} \le D\mathbf{e} \ , \\ & \boldsymbol{\alpha}^\top\mathbf{y} = 0 \ , \end{aligned}$$

which is again a convex quadratic problem with almost the same number of constraints, but this problem has dimension $n+d < n+2d+1$ and the constraints are mostly variable bounds. The hyperplane's normal vector can be obtained by $\mathbf{w} = \mathbf{X}^\top\mathbf{Y}\boldsymbol{\alpha} + 2\gamma - D\mathbf{e}$ and the bias term for $0 < \alpha_i < C/n$ by $b = y_i - \mathbf{w}^\top\mathbf{x}_i$. The Hessian for this quadratic problem is

$$\mathbf{H} = \begin{pmatrix} \mathbf{Y}\mathbf{X}\mathbf{X}^\top\mathbf{Y} & 2\mathbf{Y}\mathbf{X} \\ 2\mathbf{X}^\top\mathbf{Y} & 4\mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{Y}\mathbf{X} \\ 2\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{X}^\top\mathbf{Y} & 2\mathbf{I} \end{pmatrix} =: \mathbf{A}^\top\mathbf{A}$$

(as $\mathbf{Y}$ is a diagonal matrix), which is certainly positive semidefinite. Due to numerical problems during either the matrix creation or factorisation, the solvers - especially CPLEX - complain about $\mathbf{H}$ being indefinite. To resolve this problem, we use different tactics according to the solver: A proper solution is to define new variables $\boldsymbol{\beta} = \mathbf{A}\begin{pmatrix}\boldsymbol{\alpha}^\top & \boldsymbol{\gamma}^\top\end{pmatrix}^\top$ and to include them into the problem by the constraints $\mathbf{A}\begin{pmatrix}\boldsymbol{\alpha}^\top & \boldsymbol{\gamma}^\top\end{pmatrix}^\top - \boldsymbol{\beta} = \mathbf{0}$. Now the quadratic objective term can be rewritten as $\begin{pmatrix}\boldsymbol{\alpha}^\top & \boldsymbol{\gamma}^\top\end{pmatrix}\mathbf{H}\begin{pmatrix}\boldsymbol{\alpha}^\top & \boldsymbol{\gamma}^\top\end{pmatrix}^\top = \boldsymbol{\beta}^\top\boldsymbol{\beta}$ corresponding to a diagonal Hessian with diagonal entries zero and one only. Even

though this approach works well for CPLEX' barrier optimiser, it leads to convergence problems for MATLAB's active set method. If we instead replace $\mathbf{H}$ by $\mathbf{H} + \epsilon\mathbf{I}$, where $\epsilon = (-\lambda_{\min})_+$ and $\lambda_{\min}$ is the smallest eigenvalue of $\mathbf{H}$ computed by MATLAB (For our problems, $\lambda_{\min}$ was about -10^{-15} times the largest entry of $\mathbf{H}$.), the runtime is reduced by a factor of 3, but the components of the resulting normal vector $\mathbf{w}$ aren't as small as before ($> 10^{-8}$). Convergence is always achieved for $\epsilon \geq 10^{-9}$, but this regularisation is difficult to compensate for in the primal concave part of the d.c. function as a slightly convex term in the dual corresponds to a heavily convex term in the primal problem. Notably, the same approach leads to an increase of runtime for the CPLEX optimiser. The solutions obtained for both regularisation approaches are mostly identical; only for exceptional parameter combinations in CPLEX or where `quadprog` has convergence problems, the resulting classification errors differ. The normal vectors $\mathbf{w}$ also vary slightly ($\pm 10^{-5}$).

From some small experiments, it follows that feature selection is indeed carried out, but in most cases depending on the ratio D/C rather than on the single parameter D. But it remains to be shown in Sec. 5.6 that at the same time feature selection and better generalisation ability than for the FSV classifier can be obtained.

ℓ_0–ℓ_2–SVM

As already argued above, even the ℓ_0–ℓ_2–SVM using the concave exponential approximation of the non-d.c. ℓ_0-"norm" in combination with the ℓ_2-norm $\|\mathbf{w}\|_2^2$ is neither concave nor convex. So the duality theory of convex analysis is not applicable, but the primal problem may be solved with the DCA.

A viable d.c. decomposition (5.14) for (5.9) with

$$f(\mathbf{w}, b, \mathbf{v}) = \frac{C}{n}\mathbf{e}^\top(\mathbf{e} - \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}))_+ + \frac{1}{2}\mathbf{w}^\top\mathbf{w} + D\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) + \chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w})$$

is given by

$$\begin{aligned} g(\mathbf{w}, b, \mathbf{v}) &= \frac{C}{n}\mathbf{e}^\top(\mathbf{e} - \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}))_+ + \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w}) \ , \\ h(\mathbf{v}) &= -D\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) \ . \end{aligned}$$

Concerning the convergence conditions for the DCA, g and h are lower semi-continuous, proper convex functions and $\operatorname{dom} h = \mathbb{R}^d$. As for FSV, $\operatorname{dom} h^* = \{(\mathbf{0}, 0, \mathbf{v}) : \mathbf{v} \leq \mathbf{0}\} \subset \operatorname{dom} g^*$. Here and for the following problems, h is differentiable, so in the first step of DCA iteration $k \in \mathbb{N}_0$ we have $\widetilde{\mathbf{x}}^k = \nabla h(\mathbf{x}^k)$, which yields $\widetilde{\mathbf{v}}^k = -D\alpha e^{-\alpha\mathbf{v}^k}$ here. Combining the two DCA steps for each iteration k by Prop. 20 leads to $\mathbf{x}^{k+1} \in \partial g^*(\nabla h(\mathbf{x}^k)) = \arg\max_{\mathbf{x}}\{\nabla h(\mathbf{x}^k)^\top\mathbf{x} - g(\mathbf{x})\}$ so that we arrive at the constrained convex

QP

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & \frac{C}{n}\mathbf{e}^\top\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^\top\mathbf{w} + D\alpha\left(e^{-\alpha\mathbf{v}^k}\right)^\top\mathbf{v} \\ \text{subject to} \quad & \mathbf{Y}(\mathbf{X}\mathbf{w} + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi} \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ , \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \ , \end{aligned}$$

which is similar to the ℓ_1–ℓ_2–SVM. Hence, analogously, we solve the dual problem which is the same as for the ℓ_1–ℓ_2–SVM except that the term $D\mathbf{e}$ is replaced by $D\alpha e^{-\alpha\mathbf{v}^\mathbf{k}}$ in iteration k. So for $\mathbf{v}^\mathbf{k} = \mathbf{0}$, the problem solved in the next step is exactly the ℓ_1–ℓ_2–SVM. Note that the sequence of solutions to these QPs converges, due to Theorem 5, as f is bounded from below.

For the MATLAB solver, without regularisation of the Hessian, the algorithm didn't converge. Consequently, we applied the same regularisation techniques for the Hessian as for the ℓ_1–ℓ_2–SVM in this case. We further use the iterate solutions as restart values for the optimisers and state convergence of the DCA if the relative or absolute change of our primal variables $\mathbf{v}$ is lower than a tolerance of 10^{-5} as the algorithm doesn't terminate for tol $\leq 10^{-7}$.

Like the ℓ_1–ℓ_2–SVM, the ℓ_0–ℓ_2–SVM performs feature selection well on small experimental data sets. Fewer different solutions subject to the parameters come at the cost of higher computation time. The solution is also mainly depending on the ratio C/D.

Quadratic FSV

To solve (5.10), we use the d.c. decomposition

$$\begin{aligned} g(\mathbf{w}, b, \mathbf{v}) &= (1-\lambda)\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b))_+ \\ &\quad + \sum_{i=1}^{d'}\sum_{\phi_i(\mathbf{e}_j)\neq 0}\chi_{[-v_j, v_j]}(w_i) \ , \\ h(\mathbf{v}) &= -\lambda\mathbf{e}^\top(\mathbf{e} - e^{-\alpha\mathbf{v}}) \ , \end{aligned}$$

which, analogously to the previous approach, in each DCA iteration $k \in \mathbb{N}_0$ leads to a linear problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^{d'},\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n,\, \mathbf{v}\in\mathbb{R}^d} \quad & (1-\lambda)\mathbf{e}^\top\boldsymbol{\xi} + \lambda\alpha\left(e^{-\alpha\mathbf{v}^k}\right)^\top\mathbf{v} \\ \text{subject to} \quad & y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i \ , \quad i = 1, \ldots, n \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \ , \\ & -v_j \leq w_i \leq v_j \ , \qquad i = 1, \ldots, d';\ \phi_i(\mathbf{e}_j) \neq 0 \ . \end{aligned}$$

For linear target functions the algorithm's results are similar as for the linear FSV approach, but quadratic FSV is also able to detect the quadratic decision functions correctly.

Without rescaling of the attributes, single convergence problems occurred, e.g. for the 'wpbc60' and 'cleveland' problems, with the message "Optimal solution found, unscaled infeasibilities.". So we normalise the variables x_i to also assure equal convergence for all components.

In experiments with real-world data, we observed that unequal class distribution as, e.g., for 'wpbc24', results in $\mathbf{w} = \mathbf{0}$. A possible remedy is to introduce variables $\boldsymbol{\xi}^+, \boldsymbol{\xi}^-$ and weight the training errors like Bradley and Mangasarian (cf. Sec. 5.2.2).

Kernel – Target Alignment Approach

To apply the DCA, we have to split $f = g - h$ for f defined in (5.13). The term $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n$ is neither concave nor convex in $\boldsymbol{\theta}$. But the exponential kernel expression (5.11) is always convex as the second derivative of $e^{\mathbf{a}^\top \boldsymbol{\theta}}$ for $\mathbf{a} \in \mathbb{R}^d$ is $\mathbf{a}\mathbf{a}^\top e^{\mathbf{a}^\top \boldsymbol{\theta}}$, which is positive semidefinite regardless of $\mathbf{a}$ (by Prop. 13). Only positive linear combinations of these terms are convex with regard to $\boldsymbol{\theta}$, so we split the quadratic form $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n$: All summands with positive sign, i.e., $y_i \neq y_j$, are assigned to g, the rest to h. Assigning the indicator function to g and the concave penalty term to h again, we end up with the convex functions

$$
\begin{aligned}
g(\boldsymbol{\theta}) &= \frac{1-\lambda}{2n_{+1}n_{-1}} \sum_{\substack{i,j=1 \\ y_i \neq y_j}}^{n} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/(2\sigma^2)} + \chi_{[\mathbf{0},\mathbf{e}]}(\boldsymbol{\theta}) \ , \\
h(\boldsymbol{\theta}) &= \frac{1-\lambda}{2} \sum_{\substack{i,j=1 \\ y_i = y_j}}^{n} \frac{1}{n_{y_i}^2} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/(2\sigma^2)} - \frac{\lambda}{d} \mathbf{e}^\top (\mathbf{e} - e^{-\alpha \boldsymbol{\theta}}) \ .
\end{aligned}
$$

Again h is differentiable, so by applying the DCA we find the dual solution in the first step of iteration k as

$$
\widetilde{\boldsymbol{\theta}}^k = \nabla h(\boldsymbol{\theta}^k) = -\frac{1-\lambda}{4\sigma^2} \sum_{\substack{i,j=1 \\ y_i = y_j}}^{n} \frac{1}{n_{y_i}^2} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}^k}/(2\sigma^2)} \left((x_{il} - x_{jl})^2\right)_{l=1}^{d} - \frac{\lambda}{d} \alpha e^{-\alpha \boldsymbol{\theta}^k} \ .
$$

In the second step, looking for $\boldsymbol{\theta}^{k+1} \in \partial g^*(\widetilde{\boldsymbol{\theta}}^k) \overset{\text{Prop. 20}}{=} \arg\max_{\boldsymbol{\theta}} \{\boldsymbol{\theta}^\top \widetilde{\boldsymbol{\theta}}^k - g(\boldsymbol{\theta})\}$ leads to solving the *convex non-quadratic* problem

$$
\begin{aligned}
&\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1-\lambda}{2n_{+1}n_{-1}} \sum_{\substack{i,j=1 \\ y_i \neq y_j}}^{n} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/(2\sigma^2)} - \boldsymbol{\theta}^\top \widetilde{\boldsymbol{\theta}}^k \\
&\text{subject to } \mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{e}
\end{aligned}
\tag{5.15}
$$

in each DCA iteration with a valid initial point $\mathbf{0} \leq \boldsymbol{\theta}^0 \leq \mathbf{e}$.

In our experiments, we set $\boldsymbol{\theta}^0 = \mathbf{e}/2$. We stop the DCA with $\text{tol} = 10^{-3}$ and determine the relevant features as $\{j = 1, \ldots, d : \theta_j^{k+1} > 10^{-2}\}$.

For the convex optimisation problems (5.15), the objective as well as the constraints are differentiable and the Slater condition is fulfilled. Therefore, by Theorem 12 a point $\boldsymbol{\theta} \in \mathbb{R}^d$ is a solution to the problem if and only if $\boldsymbol{\theta}$ is a Kuhn–Tucker point. Solution methods are alternatively:

Penalty/Barrier Multiplier Method. To find the solution, one may use a technique basing upon minimising the (augmented) Lagrangian. Penalty/barrier multiplier methods according to [Ben-Tal and Zibulevsky, 1997] are iterative methods where a Kuhn–Tucker point is found by solving an unconstrained minimisation problem in each step. The authors recommend the logarithmic-quadratic penalty function, which is logarithmic in the interior of the valid region and quadratic in the penalty branch. The algorithm converges to a solution if the Slater condition holds and the solution set for the convex problem is non-empty and compact as proven in [Ben-Tal and Zibulevsky, 1997, Theorem 1]. We choose the penalty updating function $\pi^k(t) = \pi_0(\mu)^k$. If penalty parameter p increases, an actual constraint violation is penalised less! For our constraints a penalty branch is always active if $p \geq 1$! In our implementation, we omit the "safeguard rule" $\mu \leq u_i^k / u_i^{k-1} \leq 1/\mu$ mentioned by [Ben-Tal and Zibulevsky, 1997] to prevent the Lagrange multipliers to change too much in one step. It is not necessary in our experiments, but only leads to an increase of computation time, iterations, and further convergence problems induced by this. We solve the unconstrained optimisation problem occurring in each step of the method by MATLAB's optimisation toolbox function `fminunc`. This causes trouble for high-dimensional data sets, e.g., 'microarray', as the function terminates depending on $\|\nabla f\|_\infty$ for objective f, which may be small in high dimensions. So one may scale the tolerance with factor $1/d$.

In our experiments, we set the initial penalty parameter to 1, the initial Lagrange multipliers to 0.01 and terminate in the penalty/barrier multiplier method if all Kuhn–Tucker complementarity conditions are satisfied up to 10^{-6} and the constraints are satisfied within a tolerance of 10^{-7}. If the difference between both stages' tolerances is not that high, the DCA may get stuck in an infinite loop rarely. According to toy problem experiments, these parameter values seem to be optimal and the method is robust only that it is slower for nonlinear constraints. Although the penalty/barrier multiplier method often exhibits large variable changes during the optimisation process for our problems, it did always converge within the tolerance and solve the problems reliably.

Trust Region Algorithm. Faster than the penalty/barrier multiplier method is MATLAB's constrained optimisation toolbox function `fmincon` [MathWorks, 2002], which of course yields essentially the same solutions. (In the cross-validation tests, only for 'wpbc60' and 'pima' (and 'bcw') the solutions differed noticeably in six out of seventy runs with validation due to the DCA.) For our problems, it applies a trust region algorithm based on a Newton method. It decreases the runtime at least by a factor of

stage	parameter	value	objective calls	time [sec]	features
	current setting		1285	32	67
DCA	$\boldsymbol{\theta}^0$	$\mathbf{0}$	1269		17
		$\mathbf{e}$	5010		97
	tol	10^{-2}	657		68
		10^{-4}	1397		68
`fmincon`	Hessian	none	42405	339	
	tolF	tol/10	509		71
		tol/100	1069		69
		tol/10000	1182		69
	tolX	tol/10	1285		
		tol	1285		
		tol*10	1305		
	tolPCG	0.01	1165		67
		1	1153		71
	PrecondBandWidth	1	1323	33	
		∞	1284	33	

Table 5.7.: Optimisation parameter evaluation for the kernel – target alignment approach on problem 'wpbc60'

two and the number of objective function calls to roughly one fifth in our real-world experiments compared with the penalty/barrier multiplier method. Still, the number of objective calls is dominating the runtime. For calling `fmincon`, we use the last iterate for $\boldsymbol{\theta}$ as a start value and also provide the gradient and Hessian for the objective. The further parameters are a function value tolerance tolF — the relative change or the norm of the gradient — of tol/1000, a variable tolerance tolX of 10^{-6} and the default PCG tolerance tolPCG of 0.1 and PCG preconditioner bandwidth zero. An overview of an optimisation parameter evaluation is given in Table 5.7, where we apply the method on the real-world problem 'wpbc60' (see Sec. 5.6.2). We rescaled the features linearly to zero mean and unit variance, set $\sigma = \sqrt{d}/2$, $\alpha = 5$ and iterate for $\lambda = 0, 0.1, \ldots, 0.9$ with $n = 334$ training samples followed by a final training with $n = 668$ training samples with the value of λ minimising the error. The execution time of the MATLAB programs was measured on a Pentium 4 with 3 GHz and 2GB memory running under Linux. The DCA's, `fmincon`'s and the PCG tolerance and especially the start value $\boldsymbol{\theta}^0$ affect the features selected. Especially for $\boldsymbol{\theta}^0 = \mathbf{0}$, often all features are discarded. Taking into account that the convergence is critical if the tolerances at the lower stages are increased, the current parameter settings are reasonable.

In experiments with the kernel – target alignment approach, we observe the following:

- The solution is highly sensible to the DCA start value $\boldsymbol{\theta}^0$.

- We look for feature indicators $\boldsymbol{\theta} \in \{0,1\}^d$. Indeed, the solution mostly satisfies this.
- Implicit feature selection is carried out for $\lambda = 0$ as detailed in Sec. 5.6.1.

For other kernels than the Gaussian, the approach may be applied as well provided that there exists a feasible d.c. decomposition for the kernel function.

5.6. Evaluation

To study the performance of our new methods in detail, we first present computer generated ground truth experiments in Sec. 5.6.1 illustrating the general behaviour and robustness of the nonlinear classification methods. To evaluate the performance of the suggested approaches at large, we study various real-world problems in Sec. 5.6.2 and finally examine the high-dimensional research problem of organ segmentation in CT scans in Sec. 5.6.3.

5.6.1. Ground Truth Experiments

In this section, we consider artificial training sets in $\mathbb{R}^2$ and $\mathbb{R}^4$ where y is a function of the first two features x_1 and x_2. We examine specially designed points $(x_1, x_2) \in \mathbb{R}^2$ on the left of the figures and $n = 64$ normally distributed points $(x_1, x_2, x_3, x_4) \in \mathbb{R}^4$ on the right.

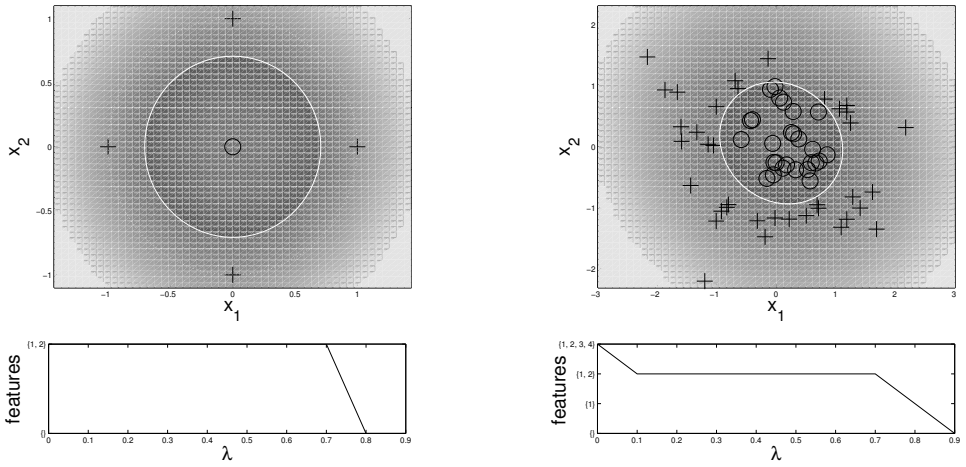
We first conduct experiments with quadratic rule similar to those by [Zhu et al., 2004] except that in our case $P_1(x)P_{-1}(x) = 0$ for all x. The examples in Fig. 5.7 show that our quadratic FSV approach indeed performs feature selection and finds classification rules for quadratic, not linearly separable problems. Ranking methods for feature selection as well as linear classification approaches do not appreciate the feature relevance for these problems.

For the 'XOR' classification problems in Fig. 5.8, the kernel – target alignment approach and again quadratic FSV perform well in contrast to the approaches with linear classification rule.

For the non-quadratic chess board classification problems in Fig. 5.9, our kernel – target alignment approach performs well, in contrast to all other feature selection approaches presented. Again, the features by themselves do not contain relevant information and all linear methods are doomed to fail.

In all test examples, only relevant feature sets are selected by our methods as can be seen in the bottom plots. Particularly the correct feature set $\{1, 2\}$ is selected for most values of λ. This clearly shows the favourable properties of *embedded* feature selection also in connection with nonlinear classification.

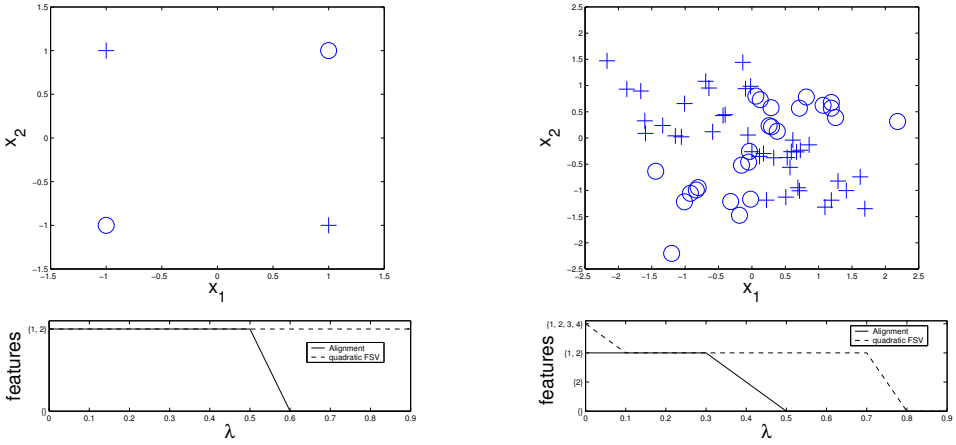
Figures 5.8 and 5.9 show on the right a remarkable property: The kernel – target alignment approach discards the two noise features x_3, x_4 even for $\lambda = 0$, which indicates



deterministic in $\mathbb{R}^2$

random in $\mathbb{R}^4$ (four normally distributed variables $x_1, \ldots, x_4$ with variances 1, 1, 1 and 2)

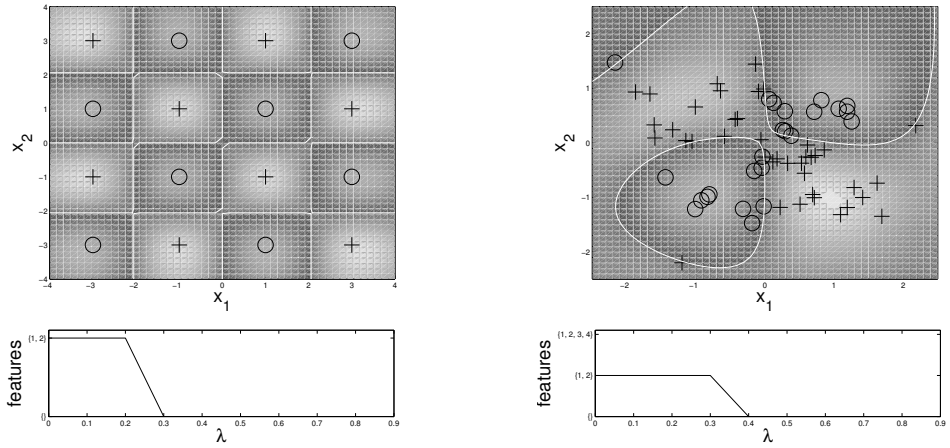
Figure 5.7.: Quadratic classification problems with unit circle rule $y = \operatorname{sgn}(x_1^2 + x_2^2 - 1)$. *Top:* Training points and decision boundaries (*white lines*) computed by quadratic FSV for $\lambda = 0.1$, *left:* in $\mathbb{R}^2$, *right:* projection of $\mathbb{R}^4$ onto selected features. *Bottom:* Features determined by quadratic FSV



deterministic ($\mathbf{x}_i \in \{-1, 1\}^2$)

four random features (same as Fig. 5.7 right)

Figure 5.8.: 'XOR' classification problems with rule $y = -\operatorname{sgn}(x_1 x_2)$. *Top:* Training points, *left:* in $\mathbb{R}^2$, *right:* projection of $\mathbb{R}^4$ onto first two features. *Bottom:* Features determined by nonlinear classification approaches (all linear approaches always discard all features)



deterministic ($\mathbf{x}_i \in \{-3,-1,1,3\}^2$) four random features (same as Fig. 5.7 right)

Figure 5.9.: Chess board classification problems with $(y+1)/2 = (\lfloor x_1/2 \rfloor \bmod 2) \oplus (\lfloor x_2/2 \rfloor \bmod 2)$. *Top:* Training points and Gaussian SVM decision boundaries (*white lines*) for $\sigma = 1$, $\lambda = 0.1$, *left:* in $\mathbb{R}^2$, *right:* zoomed projection of $\mathbb{R}^4$ onto selected features. *Bottom:* Features determined by kernel – target alignment approach

that the alignment functional (5.12) incorporates implicit feature selection. This is due to the isotropic properties of the Gaussian kernel where the feature space distances are bounded by $\|\boldsymbol{\phi}(\mathbf{x})\|^2 = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}) \rangle = K(\mathbf{x},\mathbf{x}) = 1$. As argued in Sec. 5.4.2, maximising the alignment term $\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n$ amounts to maximising the class centre distance of the feature vectors which lie on the unit sphere in ℓ_2. Adding random features disturbs the original distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ and so distributes the feature vectors $\boldsymbol{\phi}(\mathbf{x}_i)$ more uniformly on the sphere potentially moving the class means closer to each other. More precisely, adding features

$$\mathbf{x} \mapsto \begin{pmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{pmatrix}$$

leads for $\boldsymbol{\theta} = \mathbf{e}$ to kernel matrix elements

$$e^{(-\|\mathbf{x}-\mathbf{z}\|_2^2 - \|\widetilde{\mathbf{x}}-\widetilde{\mathbf{z}}\|_2^2)/(2\sigma^2)} = K(\mathbf{x},\mathbf{z}) \cdot e^{-\|\widetilde{\mathbf{x}}-\widetilde{\mathbf{z}}\|_2^2/(2\sigma^2)}$$

for $\mathbf{x}, \mathbf{z} \in \mathcal{X}$. If the new features are random, roughly all off-diagonal elements are damped by the same factor α. Splitting the diagonal from the off-diagonal terms, the original alignment $\mathbf{y}_n^\top \mathbf{K} \mathbf{y}_n =: (1/n_{+1} + 1/n_{-1}) + c$ is reduced if $c > 0$ or $\mathbf{y}_n^\top \mathbf{K} \mathbf{y}_n > 1/n_{+1} + 1/n_{-1}$. For large n_i, the value of the alignment term is reduced to $(1/n_{+1} + 1/n_{-1}) + \alpha c$ by almost the factor α too. As an example, consider

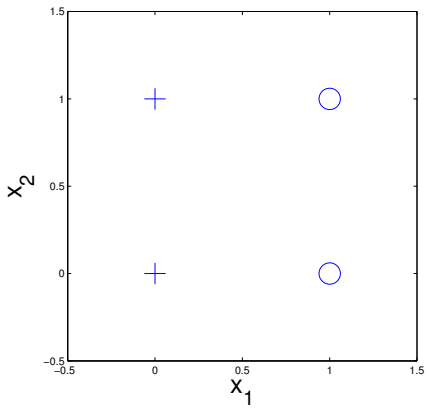


Figure 5.10.: Classification problem example with an irrelevant feature

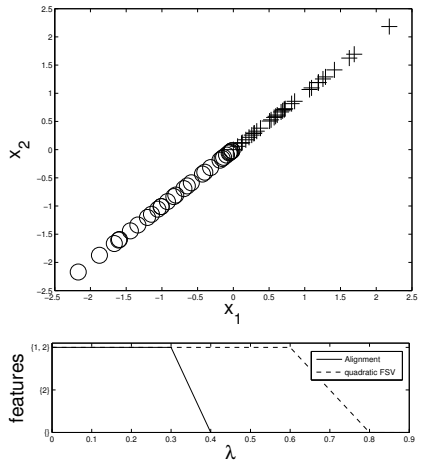
$$\mathbf{X} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad , \quad \mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

illustrated in Fig. 5.10 and with $\sigma = \sqrt{d}/2 = 1/\sqrt{2}$ leading to

$$\mathbf{y}_n^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n = 1 + e^{-\theta_2} - e^{-\theta_1} - e^{-\theta_1-\theta_2} = (1 + e^{-\theta_2})(1 - e^{-\theta_1}) \longrightarrow \max_{\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{e}} .$$

The solution obviously implies $\theta_1 = 1$, $\theta_2 = 0$ which is a strict maximum. The implicit feature selection of the alignment functional does not apply for arbitrary kernels: The linear kernel, e.g., leads to a (nonnegative) alignment summand for each feature. If one considers the common alignment (4.3), the arguments above do no longer hold as, when damping all kernel elements by the same factor, also the denominator $\|\mathbf{K}\|_{\mathrm{F}}$ is scaled roughly by this factor so that the alignment stays nearly the same. Only if the scaling factor is very small, the alignment decreases. Nevertheless, a weak implicit feature selection still applies as can be verified for the example above. But the exact conditions and reasons for the implicit feature selection of the alignment term is still an open problem.

Finally, we examine how the approaches tackle redundant features. Figure 5.11 gives the nonlinear methods' results for a problem with two completely redundant features. Similar feature sets are also selected by the linear methods. The (quadratic) FSV requires sophisticated parameter tuning for removing one redundant feature. The linear SVM and kernel – target alignment approaches did not remove a redundant feature, presumably as the alignment or the margin, respectively, is increased by the redundancy. Besides, many optimisation steps are necessary, which indicates a sensible minimum.



random in $\mathbb{R}^2$ (standard normal variables)

Figure 5.11.: classification problem in $\mathbb{R}^2$ with two identical features and rule $y = \mathrm{sgn}(x_1) = \mathrm{sgn}(x_2)$. *Top:* Training points, *Bottom:* Features determined by nonlinear classification approaches

5.6.2. Real-World Data

We compare our approaches with RLP, standard linear and Gaussian kernel SVMs and FSV which is favoured by [Bradley and Mangasarian, 1998]. To compare with RLP and FSV, we equally penalise the outliers of both classes minimising the total error, i.e., we apply problems (5.1) and (5.5) instead of (5.2) and (5.6). An appropriate class-dependent weight may be chosen individually for each classification problem.

We first introduce the data sets and our experimental setup and then present the results for all methods. We particularly examine the kernel – target alignment approach in detail at the end of the section.

Data Sets and Preprocessing

To systematically test the different feature selection methods on real-world data, we use common pattern recognition data sets from the UCI repository [Blake and Merz, 1998] — among these all those used by [Bradley and Mangasarian, 1998] — as well as the high-dimensional Colon Cancer data set from [Weston et al., 2003] — originally taken from [Alon et al., 1999] — which will be denoted as follows:

- Wisconsin prognostic breast cancer, recurrence before 24 resp. 60 months ('wpbc24' resp. 'wpbc60')
- BUPA liver disorders ('liver')

data set	no. of features d	no. of samples n	class distribution n_{+1}/n_{-1}
wpbc60	32	110	41/ 69
wpbc24	32	155	28/127
liver	6	345	145/200
cleveland	13	297	160/137
ionosphere	34	351	225/126
pima	8	768	500/268
bcw	9	683	444/239
microarray	2000	62	22/ 40

Table 5.8.: Statistics for data sets used

- Cleveland heart disease ('cleveland') collected by Robert Detrano at the Cleveland Clinic Foundation distinguishing presence from absence of heart disease (in contrast to the labelling by [Bradley and Mangasarian, 1998]) based on various features with samples with missing values omitted; we use only the thirteen commonly used features
- Johns Hopkins University ionosphere ('ionosphere')
- Pima Indians diabetes ('pima')
- breast cancer Wisconsin ('bcw')
- colon cancer microarray ('microarray')

The problems mostly treat medical diagnoses based on genuine patient data. (See also [Bradley and Mangasarian, 1998] for a brief review of most of the data sets used.) Some properties of the data sets are summarised in Table 5.8. It is essential that the features are normalised, especially for the kernel – target alignment approach as their variances influence its sensible objective with initially equal weights. In experiments, it shows that otherwise features with large variances are preferred. So we rescale the features linearly to zero mean and unit variance. By the normalisation, at large, the values $|w_i|$ increase so that the penalty $\|\mathbf{w}\|_2^2$ implicitly gets a larger weight in the objective function. We also tried linearly rescaling the features to the range $[-1, 1]$ as a normalisation which leads to similar results.

Choice of Parameters

We fix the ℓ_0-"norm" approximation parameter to $\alpha = 5$ in penalty (5.4) as in Sec. 5.2.3, where more - in particular mostly more than zero - features are selected for a larger value of α, but the accuracy does not necessarily increase, especially for the ℓ_0–ℓ_2–SVM. The

influence of the parameter α has been studied by [Jakubik, 2003]. We set $\sigma = \sqrt{d}/2$ in the Gaussian kernel (5.11), which maximises the alignment of the problems. We start the DCA with $\mathbf{v}^0 = \mathbf{e}$ for the ℓ_0–ℓ_2–SVM, FSV and quadratic FSV and with $\boldsymbol{\theta}^0 = \mathbf{e}/2$ for the kernel – target alignment approach and stop on $\mathbf{v}$ with tol $= 10^{-5}$ resp. tol $= 10^{-3}$ for $\boldsymbol{\theta}$.

To determine the weight parameters, we discretise their range of values and perform a parameter selection step minimising the error on an independent validation set before actually applying the feature selection algorithm. The validation set is chosen arbitrarily as one half of each run's (cross-validation) training set to select $\ln C \in \{0, \ldots, 10\}$, $\ln D \in \{-5, \ldots, 5\}$, $\lambda \in \{0.05, 0.1, 0.2, \ldots, 0.9, 0.95\}$ for (quadratic) FSV and $\lambda \in \{0, 0.1, \ldots, 0.9\}$ for the kernel – target alignment approach. On ambiguity, in case of equal validation error, we choose the larger values for (D, C) resp. λ. In the same manner, the SVM weight parameter C is chosen according to the smallest in $\{e^{-5}, e^{-4}, \ldots, e^{5}\}$ independently of the selected features. The final classifier is then built from the training and validation sets. To solve the elementary optimisation problems, we use the CPLEX solver library [Ilog, Inc., 2001] with the barrier optimiser for the quadratic problems called via [Musicant, 2000] with a reduced convergence tolerance of 10^{-10}. It is also possible to use MATLAB's active set method in `quadprog`. If the data are not normalised, both solvers have convergence problems sometimes invariantly of changes of the approximation parameter α. They occur mostly for large values of C, only for bad parameter combinations or the algorithm recurs thereafter for the ℓ_0–ℓ_2–SVM. The proposed normalisation leads to stable convergence in all tests. We further use MATLAB's constrained optimisation method `fmincon` documented in [MathWorks, 2002] for the kernel – target alignment approach.

Results

We first partition the data equally into a training, a validation and a test set. The validation and test performance and validated parameters for the linear classifiers are summarised in Table 5.9. For comparison, the test results on non-normalised and normalised range data are given in Tables 5.10 and 5.11, respectively. (For the high-dimensional 'microarray' data, the evaluation of quadratic FSV requires too much memory.) While trying different normalisations, we also observed that quadratic FSV is most sensible to changes in the data. While the other approaches are stable, small changes may cause quadratic FSV solutions to differ heavily, which may also be fortified by the additional validation step. As the optimisation process is stopped numerically, we determine the number of features as $|\{j = 1, \ldots, d : |w_j| > 10^{-8}\}|$ resp. $|\{j = 1, \ldots, d : \theta_j > 10^{-2}\}|$.

As a result of the validation, the optimal combination for (C, D) mostly falls within the range of discretised values (Mind the resolution of ambiguities preferring larger values.). Further, from the error plots subject to the two parameters the classifier performance is mostly depending on the ratio C/D and not on the absolute values. Our linear methods

	RLP (5.1)		linear SVM			FSV uniform (5.5)				ℓ_1–ℓ_2–SVM (5.8)				ℓ_0–ℓ_2–SVM (5.9)			
data set	dim	tst err	dim	val err	tst err	dim	val err	tst err	λ^*	dim	val err	tst err	($\ln C^*$, $\ln D^*$)	dim	val err	tst err	($\ln C^*$, $\ln D^*$)
wpbc60	32	44	32	27	31	0	38	31	0.95	27	24	33	(1,-4)	27	24	33	(1,-5)
wpbc24	32	25	32	25	22	0	25	22	0.95	19	25	20	(10, 5)	13	25	22	(8, 5)
liver	6	28	6	32	30	2	34	33	0.3	6	31	30	(9, 5)	6	31	31	(5, 1)
cleveland	13	17	13	14	16	4	18	23	0.05	9	13	17	(8, 5)	7	14	17	(2,-2)
ionosphere	33	12	34	15	11	2	11	14	0.2	19	13	11	(9, 5)	3	10	15	(6, 3)
pima	8	26	8	20	27	1	22	29	0.05	7	19	27	(6,-1)	8	19	27	(5,-3)
bcw	9	4	9	2	4	1	5	9	0.2	9	2	4	(3,-2)	8	2	4	(5,-3)
microarray	41	**40**	2000	14	**10**	1	24	**15**	0.3	21	14	**5**	(1, 0)	18	14	**5**	(0,-3)

Table 5.9.: Feature selection and linear classification performance (number of features, validation error [%], test error [%]) and weight parameters that minimise classification error on validation set

	RLP (5.1)		FSV uniform (5.5)		ℓ_1–ℓ_2–SVM (5.8)		ℓ_0–ℓ_2–SVM (5.9)		quadratic FSV (5.10)	
data set	dim	err	dim	err	dim	err	dim	err	dim	err
wpbc60	32	44	1	31	21	47	19	47	12	36
wpbc24	32	25	0	22	18	22	14	22	13	20
liver	6	28	0	41	6	28	6	28	6	25
cleveland	13	17	8	19	10	17	10	16	13	19
ionosphere	33	12	3	13	19	14	3	15	5	14
pima	8	26	8	26	8	27	8	27	8	28
bcw	9	4	9	4	9	4	9	4	9	5
microarray	41	40	1	15	21	5	18	5	-	-

Table 5.10.: Feature selection and classification performance (number of features, test error [%]) on non-normalised features with weight parameters chosen to minimise classification error on validation set

	RLP (5.1)		FSV uniform (5.5)		ℓ_1–ℓ_2–SVM (5.8)		ℓ_0–ℓ_2–SVM (5.9)		quadratic FSV (5.10)		k.-t. align. (5.13)	
data set	dim	err	dim	err	dim	err	dim	err	dim	err	dim	err
wpbc60	32	44	1	33	30	31	25	33	0	31	4	33
wpbc24	32	25	0	22	3	22	0	22	0	22	1	22
liver	6	28	4	31	6	31	6	31	4	30	0	41
cleveland	13	17	6	19	12	15	10	16	1	27	1	18
ionosphere	33	12	2	14	22	11	5	11	3	10	3	15
pima	8	26	3	29	7	27	2	29	1	31	4	29
bcw	9	4	1	9	9	4	9	4	8	4	2	4

Table 5.11.: Feature selection and classification performance (number of features, test error [%]) on normalised range features with weight parameters chosen to minimise classification error on validation set

achieve feature selection and are often able to improve the classification performance compared with the baseline RLP classifier. Especially for the very high-dimensional 'microarray' data, both our linear feature selection methods ℓ_1–ℓ_2–SVM and ℓ_0–ℓ_2–SVM are more accurate than even the linear SVM. In the validation phase, the ℓ_1–ℓ_2– and ℓ_0–ℓ_2–SVMs attain an even higher accuracy compared with the other methods. This indicates that parameter selection is difficult, but that the methods are powerful if the right parameters are chosen.

In order to make the results less variant on the sample partitioning, we also conducted experiments with cross-validation by dividing the data set into ten equally sized test runs. The aggregate results are displayed in Table 5.12 for linear and in Table 5.13 for nonlinear classifiers. Clearly, all proposed approaches perform feature selection: FSV discards most features followed by the kernel – target alignment approach and then the ℓ_0–ℓ_2–SVM, then the ℓ_1–ℓ_2–SVM. At the same time, all our approaches mostly achieve a higher classification accuracy than RLP. The quadratic FSV performs well mainly for special problems (e.g., 'liver' and 'ionosphere'), where presumably a nonlinear relation is given. For true linear classification problems, it may be more difficult to find the best classification rule in this more general setting. But the classification is good in general for all other approaches. Both double norm classifiers achieve feature selection and a low error rate at least comparable to FSV. The ℓ_0–ℓ_2–SVM suppresses more features, but its computation time is several times higher than for the ℓ_1–ℓ_2–SVM. Even more features would be suppressed for higher values of D with maybe only a small decrease of accuracy, but the parameters were chosen here so as to minimise the classification error. For the kernel – target alignment approach, apart from the apparent feature reduction, also the number of SVs is generally reduced, which can be seen in Table 5.13. This allows again faster classification and also indicates a higher generalisation ability. The average number of DC iterations given in Table 5.13 for a run with ten validation calls and the final evaluation is still moderate. The number of iterations is for each problem also approximately proportional to the total optimisation time.

Kernel – Target Alignment Approach

The kernel – target alignment approach is two-stage with separate feature selection and classification steps. Here we also try to apply parameter selection to the feature indicator $\boldsymbol{\theta}$ instead of the weight parameter λ. This means that we do not only determine λ in the validation step, but directly adopt the best feature indicator $\boldsymbol{\theta}$ for the final classifier. This leads to a general increase of the number of features without significant gain in accuracy as can be verified in Table 5.14 in comparison with Table 5.13, which indicates that our parameter validation procedure is sensible.

We have already pointed out in Secs. 5.5.2 and 5.6.1 that the kernel – target alignment approach performs feature selection implicitly, which means without feature penalty ($\lambda = 0$). To illustrate this, the respective results are given in Table 5.15. Of course the

	RLP (5.1)		linear SVM		FSV uniform (5.5)		ℓ_1–ℓ_2–SVM (5.8)		ℓ_0–ℓ_2–SVM (5.9)	
data set	dim	err	dim	err	dim	err	dim	err	dim	err
wpbc60	32.0	40.9	32.0	33.6	0.4	36.4	12.4	**35.5**	13.4	37.3
wpbc24	32.0	27.7	32.0	18.1	0.0	18.1	12.6	**17.4**	2.9	18.1
liver	6.0	31.9	6.0	32.5	2.1	36.2	6.0	35.1	5.0	34.2
cleveland	13.0	16.2	13.0	15.8	1.8	23.2	9.9	**16.5**	8.2	**16.5**
ionosphere	33.0	13.4	34.0	13.4	2.3	21.7	24.8	13.4	14.0	15.7
pima	8.0	22.5	8.0	23.2	0.7	28.9	6.6	25.1	6.1	**24.7**
bcw	9.0	3.4	9.0	2.9	2.4	4.8	8.7	**3.2**	7.9	**3.1**

Table 5.12.: Feature selection and linear classification tenfold cross-validation average performance (number of features, test error [%]), bold numbers indicate lowest errors of feature selection methods including Table 5.13

	Gaussian SVM			quadratic FSV (5.10)		kernel – target alignment (5.13)			
data set	dim	err	SVs	dim	err	dim	err	DCA iter	SVs
wpbc60	32.0	32.7	94.3	3.2	37.3	4.4	**35.5**	248.1	92.0
wpbc24	32.0	16.8	123.8	0.0	18.1	1.9	18.1	215.2	131.5
liver	6.0	33.3	233.1	3.2	**32.5**	2.5	35.4	242.6	262.3
cleveland	13.0	15.8	241.0	9.2	32.3	3.2	23.6	139.6	224.4
ionosphere	34.0	7.1	159.7	32.9	10.5	6.6	**7.7**	192.2	109.6
pima	8.0	23.4	481.1	4.7	29.9	1.4	27.0	202.2	444.2
bcw	9.0	2.9	229.0	5.9	9.4	2.8	4.2	74.9	160.5

Table 5.13.: Feature selection and nonlinear classification tenfold cross-validation average performance (number of features, test error [%], number of DCA iterations, number of SVs), bold numbers indicate lowest errors of feature selection methods including Table 5.12

	kernel – target alignment (5.13)	
data set	dim	err
wpbc60	5.2	35.5
wpbc24	2.8	18.7
liver	2.8	37.7
cleveland	3.3	21.2
ionosphere	7.7	7.4
pima	1.8	25.4
bcw	2.8	4.2

Table 5.14.: kernel – target alignment approach tenfold cross-validation average performance (number of features, test error [%]) with features that minimise classification error on validation set

	kernel – target alignment (5.13)	
data set	dim	err
wpbc60	9.0	38.2
wpbc24	6.5	17.4
liver	4.0	29.6
cleveland	4.2	19.9
ionosphere	8.9	7.1
pima	2.0	25.9
bcw	3.0	4.0

Table 5.15.: kernel – target alignment approach tenfold cross-validation average performance (number of features, test error [%]) for $\lambda = 0$

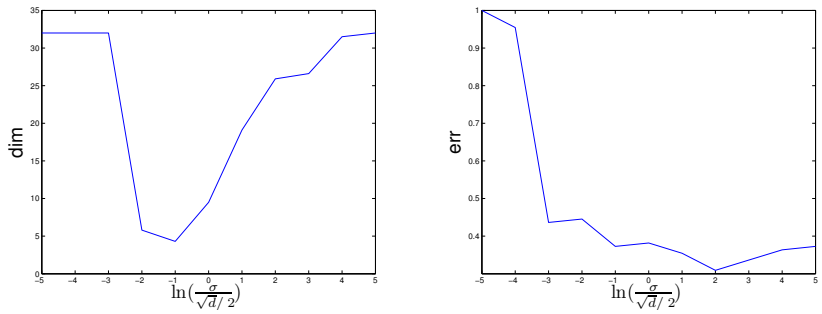


Figure 5.12.: Performance of kernel – target alignment approach (5.13) for problem 'wpbc60' with $\lambda = 0$ subject to kernel parameter σ

number of selected features is larger than with feature penalty as in Table 5.13, but many features are discarded inherently along with a sound classification performance. Note that this gives a reliable feature selection approach without any necessity for parameter selection.

As the alignment approach is implicitly also subject to the parameter σ of the Gaussian kernel (5.11), we examine its influence in Fig. 5.12. Results are given for the problem 'wpbc60' with $\lambda = 0$ to simplify the analysis and to be able to compare the results for different values of σ. The plots show that the feature selection only works within a range of several orders of magnitude for σ, and that the value selected in the other experiments (corresponding to 0 on the abscissa) is expedient.

5.6.3. Organ Segmentation in CT Scans

The classification results on the 'microarray' data set in the previous section already indicate that feature selection methods are more important in higher dimensions. The evaluation of medical data is a prominent area where this occurs. Due to the unknown

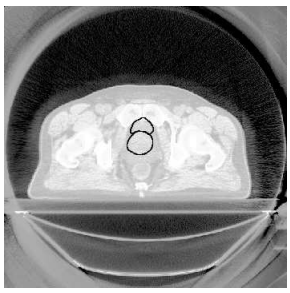


Figure 5.13.: Sample CT slice from data set 'organs22' with contours of organs bladder and prostate

relevant factors and problem nature, at first often large feature sets are collected.

Here, we study the segmentation of specific organs in CT scans where no satisfactory algorithms exist up to now. However this automatic detection is essential for the treatment of, e.g., cancer patients. The data originates from three-dimensional CT scans of the masculine hip region. An exemplary two-dimensional image slice is depicted in Fig. 5.13. To label the images, the adjacent organs bladder and prostate have been masked manually by experts. The contours of both organs are also shown in Fig. 5.13 where the organs are difficult to distinguish visually.

As described by [Schmidt, 2004], the images are filtered by a three-dimensional steerable pyramid filter bank with 16 angular orientations and four decomposition levels. Then local histograms are built for the filter responses with ten bins per channel. Including the original grey values, this results in 650 features per voxel. The task is to label each voxel with the correct organ. Here, the high dimension of the feature space is induced by the filtering which requires many directions due to the three primary input dimensions. In total, for example for problem 'organs22', the data for the region where bladder or prostate are contained amount to $117 \times 80 \times 31$ feature vectors $\in \mathbb{R}^{650}$.

In our experiments, we consider three different patients or data sets. For each of those, we select 500 feature vectors from each class. From those, we use 334 arbitrary samples for training and test, respectively, during the parameter validation and then train our final classifier on all 1000 training vectors. Note that, by choosing an equal number of training samples from both classes different from the entire test set where $n_{+1}/n_{-1} \in [1/12, 1/4]$, we put more weight on the errors of the smaller class 'prostate'.

As done in [Schmidt, 2004], we also apply an SV classifier with χ^2 kernel

$$K_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = e^{-\rho \sum_{k=1}^{d} \theta_k \frac{(x_k - z_k)^2}{x_k + z_k}}$$

for $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$ with $\rho = 2^{-11}$ on unmodified features. This kernel achieves a performance significantly superior to the Gaussian kernel for histogram features in experiments by [Chapelle et al., 1999]. According to [Haasdonk and Bahlmann, 2004, Prop. 1], the ker-

	RLP		lin. SVM		FSV		ℓ_1–ℓ_2–SVM		ℓ_0–ℓ_2–SVM	
data set	dim	err	dim	err	dim	err	dim	err	dim	err
organs4	225	13.2	650	1.1	4	2.3	61	0.9	18	0.7
organs20	242	15.2	650	1.4	6	3.6	79	1.5	43	2.7
organs22	231	11.7	650	1.3	3	11.4	106	2.2	66	2.2

Table 5.16.: Feature selection and linear classification performance for CT data (number of features, test error [%]) with weight parameters chosen to minimise classification error on validation set

	Gaussian SVM		χ^2 SVM		χ^2 SVM ranking		k.-t. align.	
data set	dim	err	dim	err	dim	err	dim	err
organs4	650	1.5	650	0.8	25	1.2	16	1.6
organs20	650	2.3	650	1.1	32	1.8	29	1.9
organs22	650	2.2	650	1.9	22	2.7	35	3.9

Table 5.17.: Feature selection and nonlinear classification performance for CT data (number of features, test error [%]) with weight parameters chosen to minimise classification error on validation set

nel is positive definite if and only if the distance in the exponent is isometric to an ℓ_2-norm, which is the case for the χ^2 distance. Nevertheless, we include a bias term b as in the linear case. To apply the kernel – target alignment approach for feature selection, one has to replace the Gaussian kernel by the new kernel, which is still convex in $\boldsymbol{\theta}$, in Sec. 5.5.2.

In our experiments, we also include a fast SVM-based filter method for feature selection [Heiler et al., 2001] ranking the features according to the χ^2 SVM decision function, where we select C by validation again and successively include features until the validation error drops five times by no more than 0.1%. For the other approaches, we use the same parameter settings as in the previous section. The results for the three patients are given in Table 5.16 for linear and in Table 5.17 for nonlinear classification methods.

The data sets seem to be well linearly separable, which also results in much lower classification and training times. Even more, the Gaussian SVM yields astonishingly high errors compared with its linear and χ^2 variants although reasonable values for the weight λ are selected and our chosen kernel width σ produces an alignment of around 12% on the training set, which is maximised for a near kernel width $\in [\sigma/2, \sigma]$. This slight overestimation of σ is due to the sparsity of the histogram features. The error of the Gaussian SVM always increases compared with its validation error of 0.3 – 2.1% whereas it decreases for the other SVMs. But the scant superiority of Gaussian SVMs over linear ones is also consistent with [Chapelle et al., 1999].

Both our linear feature selection methods perform well: They sometimes reduce the

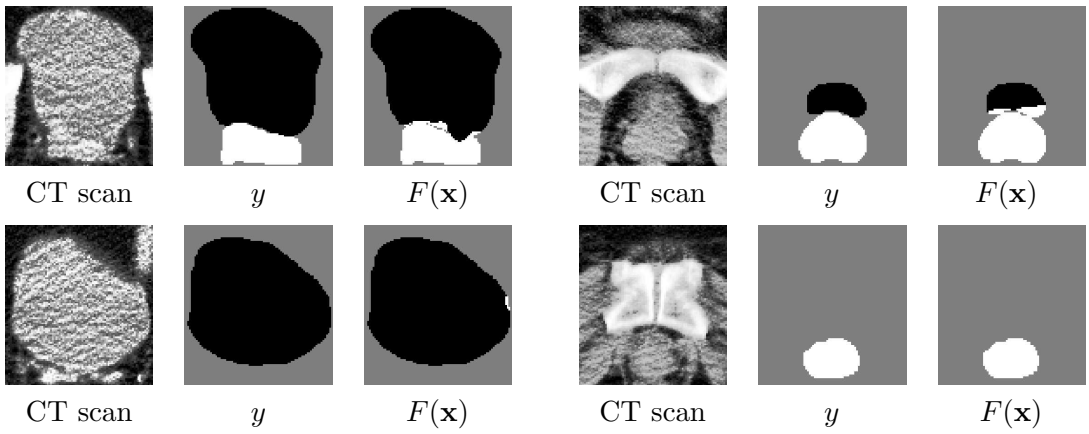


Figure 5.14.: Sample results for ℓ_0–ℓ_2–SVM on segmentation problem 'organs4'; classes are marked black and white

classification error compared with RLP and the linear SVM using the whole feature set and reliably reduce the number of features. The kernel – target alignment approach and the filter method select very few features only, in particular only few features corresponding to each filter subband. So the alignment approach well copes with the redundancy of the histogram features. The classification results for the ℓ_0–ℓ_2–SVM on the data set 'organs4' may be visually compared with the mask considered as ground truth in Fig. 5.14. The organs are classified with a high accuracy although the classes are again difficult to distinguish visually. The dimension reduction also leads to a reduced classification time for all feature selection approaches which is essential in real-time medical applications.

In the kernel – target alignment approach we found by chance that the (validation) errors decrease roughly by a factor of two if one uses different weights in the χ^2 kernel during the computation of $\widetilde{\boldsymbol{\theta}}^k$: If the other class's weights are used, the denominators of the non-zero elements are generally smaller so that the features are penalised by a smaller $\widetilde{\theta}_j^k$, especially if the feature x_j has different values for the two classes.

5.7. Possible Extensions to Multi-Class Problems

As they are, the proposed embedded feature selection approaches only work for binary classification problems. A common way to treat with multi-class problems is to reduce them to a sequence of binary classifiers as mentioned in Sec. 2.3.4. This makes it possible to generalise the feature selection to multiple classes. One can apply feature selection to every binary classifier in order to obtain improved accuracy and prediction time and analyse the problems. As an alternative approach, it is desirable to use the same features for all binary classifiers in order to also facilitate data collection, reduce storage space,

and eventually compute the kernel values only once for all classifiers. For (quadratic) FSV, ℓ_1–ℓ_2–SVM and ℓ_0–ℓ_2–SVM, this may easily be achieved by applying one of the embedded approaches for all classifiers simultaneously. This amounts to optimising all binary classifiers while penalising a common feature indicator similar to the idea in quadratic FSV. The only (computational) drawback is that all binary problems have to be solved at once:
Consider, e.g., linear classifiers. When we have k binary classification problems

$$\begin{aligned} \min_{\mathbf{w}^i \in \mathbb{R}^d,\, \boldsymbol{\beta}^i \in \mathbb{R}^{n_\beta}} \quad & f(\mathbf{w}^i, \boldsymbol{\beta}^i) \\ \text{subject to} \quad & C(\mathbf{w}^i, \boldsymbol{\beta}^i) \end{aligned}$$

with hyperplane normals $\mathbf{w}^i$, objective function f, constraint set C and additional variables $\boldsymbol{\beta}^i$ for $i = 1, \ldots, k$, the proposed algorithm with feature penalty ρ reads

$$\begin{aligned} \min_{\mathbf{w}^i \in \mathbb{R}^d,\, \boldsymbol{\beta}^i \in \mathbb{R}^{n_\beta}, i=1,\ldots,k,\, \mathbf{v} \in \mathbb{R}^d} \quad & (1-\lambda) \sum_{i=1}^{k} f(\mathbf{w}^i, \boldsymbol{\beta}^i) + \rho(\mathbf{v}) \\ \text{subject to} \quad & C(\mathbf{w}^i, \boldsymbol{\beta}^i) \ , \qquad i = 1, \ldots, k \ , \\ & -\mathbf{v} \leq \mathbf{w}^i \leq \mathbf{v} \ , \quad i = 1, \ldots, k \ . \end{aligned}$$

An evaluation is left for future research.

5.8. Summary and Conclusions

Wavelet adaptation is a special case of feature selection, which is a prominent problem in pattern recognition. We studied known efficient feature selection methods and examined special issues for applying them to wavelet adaptation.

Motivated by the results on real-world data, we proposed several novel methods that extend existing linear embedded feature selection approaches towards better generalisation ability by improved regularisation, and constructed feature selection methods in connection with nonlinear classifiers. To solve the corresponding optimisation problems with the DCA, we found appropriate d.c. splittings of our non-convex objective functions.

Our results show that embedded nonlinear methods, which have been rarely examined up to now, are indispensable for feature selection. In the experiments with real data, effective feature selection was always carried out by our methods in conjunction with a small classification error. In particular, the proposed feature selection methods were able to improve the classification of organs in CT scans which is still a research problem in medical imaging. So direct objective minimising feature selection is profitable and viable for different types of classifiers. In higher dimensions, the curse of dimensionality

affects the classification error even more such that our methods are also more important here.

We sketched possible application of the approaches to multi-class classification problems. The approaches may also be extended to incorporate other feature maps in the same manner as quadratic FSV. For the kernel – target alignment approach, an application to kernels other than the Gaussian is possible as we have shown in the experiments with histogram features.

6. Conclusions

This thesis investigates jointly designing both stages of an adaptive wavelet–Support Vector classifier. In particular, the wavelet feature extraction stage is adapted to the subsequent classifier and the problem at hand.

After introducing the two-stage classifier architecture in Chap. 2, we focus on three aspects of optimally adapting the wavelet features to the classifier.

The classifier performance is strongly affected by the choice of the wavelet used for feature extraction, as becomes clear from our argumentation and many different experiments. Chapter 4 shows how to effectively adapt the wavelet to the problem and classifier. We suggest possible adaptation criteria. Simple criteria well approximate the expected classification error in experiments with different classification problems. The adaptive grid search algorithm we devise proves to robustly optimise the selected criterion faster than standard optimisation procedures.

The central assumption during the wavelet adaptation is that the extracted features and the resulting classifier performance depend on the wavelet shape. This is to decide for each class of signals individually. Further, for other classifiers than the SVM it is not clear which adaptation criteria should be used. The adaptation fundamentally relies on the lattice factorisation of orthogonal filter banks. The criteria comparison and especially the grid search algorithm are based on the resulting cuboid parameter space. Other wavelet parameterisations, e.g. for symmetric wavelets, as discussed in the wavelet literature could be examined as well.

We illustrate that the common discrete wavelet transform splits into subbands severely depending on the signal alignment. In Chap. 3, we propose enhanced wavelet transforms to cope with shifts of the input signals. We derive how Kingsbury's dual–tree transform achieves shift invariance by combining two appropriately supported real filters to a complex filter with only positive frequency response. Our extension to filter banks in the frequency domain proves to achieve the same favourable shift invariance properties, also when applied to signal classification, while providing a vast library of filters. The main drawback of the resulting transform is its calculation in the frequency domain, which requires an initial Fourier transform of the signal. It remains to decide for the specific application whether the flexibility and shift invariance gained are worth that overhead.

The property allowing for shift invariance is the filter's single peak in the frequency domain which can only be achieved with complex wavelets. It would be interesting to extend the theory to complex multiwavelets. Further, there also exist other transforms aiming at shift invariance. They should be compared and similarities ought to

be analysed. Beside shift invariance, additional properties such as the availability of complex phase information or better directional resolution in multiple dimensions make the complex transform well suited for other applications as sketched in Sec. 3.10. An obvious example is denoising by wavelet shrinkage, where rotational invariance is a desired property. Own experiments similar to those in [Mrázek and Weickert, 2003] are promising.

Chapter 5 is devoted to general feature selection methods. Having examined wavelet adaptation by means of appropriate criteria in Chap. 4, we discuss how the task can be solved with standard feature selection approaches also. Based on state of the art embedded approaches, we devise novel feature selection approaches. Two approaches aim at improved generalisation by including an additional regularisation term. As nonlinear classifiers are able to solve more complicated classification problems, two further approaches for the first time focus on embedded original feature selection for nonlinear classifiers. Experiments with many real-world data bases show that all proposed approaches reliably perform feature selection. At the same time, they often improve over the classification performance of well-established classifiers and feature selection approaches. The approaches are also able to improve the classification of organs in up to date computed tomography scans. The enhancement of the feature selection functionals is only possible due to the potent difference of convex functions programming. We review the necessary theory in Appendix B, present the optimisation algorithm and discuss how it can be applied. This general framework for non-convex non-differentiable optimisation invariably solves all four different problems. It may be interesting for many other applications not only in pattern recognition or signal processing.

Apart from the four different feature selection approaches we elaborate on in Chap. 5, other approaches are listed in Sec. 5.4 which would be interesting to examine and to compare. Further, our experiments only investigate three different kernels, or nonlinear feature maps. This leaves room to extend the feature selection to others.

Supplementing the discussion of wavelet adaptation criteria, Appendix A deduces equivalence of different Support Vector problems. As a result, the radius of a set of feature vectors can be computed by a standard SVM.

As a general focus of our work, we try to adapt the features to the classification. The paradigm of a conjoint design may be applied to areas other than wavelet selection. Only some aspects in designing the classifier with respect to the problem are the choice of appropriate kernels for the SVM, an automated choice of the regularisation parameter or the kernel parameters as suggested in Sec. 2.3.1. Furthermore, it may be sensible to consider other classifiers than the SVM. In this case, the methods and results presented are to be transferred and verified.

Further research with respect to our signal classification architecture should consider two-dimensional signals more thoroughly. An additional issue in two dimensions is rotational invariance that can still not be guaranteed using complex wavelet transforms. So

the first task is to find an appropriate signal representation. In preliminary experiments, a Fourier representation corresponding to an ideal filter shows promise, for example. On account of the energy operator, there is ongoing work in image analysis. It seems worth investigating more sophisticated norms or features here. Other norms, also in the one-dimensional case, are also considered in the wavelet literature to characterise function spaces.

Parallel to the generalisation to multiple dimensions is the generalisation to multiple classes. Multi-class SVMs are discussed in Sec. 2.3.4, and possible extensions are already sketched in Secs. 4.6 and 5.7. But much still remains to be done to design an efficient signal classifier for more than two classes.

A. An SVM Formulation for Radius Computation

A.1. SV Clustering Problem

This section derives a QP equivalence that may be used to efficiently compute the radius of the smallest sphere enclosing a set of points as stated earlier in Theorem 4. The radius is involved in the radius – margin error bound (4.2) and its efficient computation was exploited to evaluate the wavelet adaptation criterion visualised in Figs. 4.3 to 4.6 (e).

The QP (4.1) to determine the radius R for the points $\boldsymbol{\phi}(\mathbf{x}_i)$ for $i = 1, \ldots, n$ in feature space is a special case of the problem

$$\begin{aligned} \min_{\mathbf{a} \in \mathcal{F}_K,\, R \in \mathbb{R},\, \boldsymbol{\xi} \in \mathbb{R}^n} \quad & R^2 + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \;, \quad i = 1, \ldots, n \;, \\ & \xi_i \geq 0 \;, \quad i = 1, \ldots, n \end{aligned} \tag{A.1}$$

considered by [Ben-Hur et al., 2001] for clustering. Therefore we refer to (A.1) as *SV clustering problem*. We show that (A.1) can be solved by a single-class SVM, i.e., an SV classification problem with all points belonging to the same class. Then the matrix $\mathbf{Y}$ in (2.34) is the identity matrix so that (2.34) simplifies to

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad & \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} - \mathbf{e}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \widehat{C} \mathbf{e} \;. \end{aligned} \tag{A.2}$$

Although this is still a QP it is profitable to use this connection, since, for standard SVMs, sophisticated algorithms are included into many software implementations. Note that according to [Cristianini and Shawe-Taylor, 2000, p.104], the radius can also be used to determine the weight parameter C for soft margin SVMs as $C = 1/R^2$ whereupon the soft margin radius – margin error bound is directly minimised by the SVM. Hence it is profitable to have a simple way of computing the radius.

We will prove the following theorem, which generalises Theorem 4 also including the soft margin case $C < \infty$:

Theorem 8. *Let K be a kernel with corresponding feature map $\boldsymbol{\phi}$ and with the property that $K(\mathbf{x},\mathbf{x}) = \kappa$ for all $\mathbf{x} \in \mathcal{X}$. Then there exists $\widehat{C} > 0$ such that the optimal radius R in* (A.1) *can be obtained by solving the dual problem* (A.2) *of a single-class SVM. More precisely, $\boldsymbol{\alpha}$ being the solution of* (A.2)*, it holds*

$$R^2 = \kappa + \boldsymbol{\beta}^\top \mathbf{K}\boldsymbol{\beta} - 2(\mathbf{K}\boldsymbol{\beta})_i \ , \tag{A.3}$$

where $\boldsymbol{\beta} := (\mathbf{e}^\top\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}$ and $i \in \{1,\dots,n\}$ denotes some index with $0 < \beta_i < C$.

Note that $C = \widehat{C} = \infty$ for our original problem (4.1).

Our proof proceeds in two steps: first we show that the SV clustering problem (A.1) is equivalent to a single-class SVM with additional bias term also included in the objective function. This SVM is used for novelty detection by [Schölkopf et al., 2000] and is therefore called *SV novelty detection problem* in the following. Then we prove that the SV novelty detection problem is equivalent to the ordinary single-class SVM (A.2) without bias term.

A.2. Equivalence of the SV Clustering Problem and the SV Novelty Detection Problem

The equivalence is best shown considering the dual problems. For solving (A.1), we introduce the Lagrangian

$$\mathcal{L}(\mathbf{a}, R, \boldsymbol{\xi}, \boldsymbol{\beta}, \boldsymbol{\mu}) := R^2 + C\mathbf{e}^\top\boldsymbol{\xi} - \sum_{i=1}^{n} \beta_i(R^2 + \xi_i - \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2) - \boldsymbol{\mu}^\top\boldsymbol{\xi}$$

with Lagrange multipliers $\boldsymbol{\beta}, \boldsymbol{\mu} \geq \mathbf{0}$. Setting the derivative of $\mathcal{L}$ with respect to R, $\mathbf{a}$ and $\boldsymbol{\xi}$ to zero, it follows

$$\mathbf{e}^\top\boldsymbol{\beta} = 1 \ ,$$

$$\mathbf{a} = \sum_{i=1}^{n} \beta_i \boldsymbol{\phi}(\mathbf{x}_i) \ , \tag{A.4}$$

$$\boldsymbol{\beta} = C\mathbf{e} - \boldsymbol{\mu} \ . \tag{A.5}$$

Using these equations, the Lagrangian yields the dual problem

$$\begin{aligned} \max_{\boldsymbol{\beta}\in\mathbb{R}^n} \quad & \left(W(\boldsymbol{\beta}) := \sum_{i=1}^{n} \beta_i \|\boldsymbol{\phi}(\mathbf{x}_i)\|^2 - \sum_{i,j=1}^{n} \beta_i\beta_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle \right) \\ \text{subject to} \quad & \mathbf{e}^\top\boldsymbol{\beta} = 1 \ , \\ & \mathbf{0} \leq \boldsymbol{\beta} \leq C\mathbf{e} \ . \end{aligned}$$

By (2.23), the function $W(\boldsymbol{\beta})$ can be rewritten as

$$W(\boldsymbol{\beta}) = \sum_{i=1}^{n} \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{n} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \ .$$

In our applications we are mainly interested in isotropic kernels $K(\mathbf{x}, \mathbf{y}) = k(\|\mathbf{x} - \mathbf{y}\|)$, e.g. in the Gaussian kernel (2.20). These kernels have $K(\mathbf{x}, \mathbf{x}) = \kappa$ for some $\kappa > 0$ and all $\mathbf{x} \in \mathcal{X}$. Then $W(\boldsymbol{\beta})$ can be further simplified to

$$W(\boldsymbol{\beta}) = \kappa - \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta}$$

so that we finally have to solve the dual optimisation problem

$$\begin{aligned} &\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \ \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta} \\ \text{subject to} \quad &\mathbf{e}^\top \boldsymbol{\beta} = 1 \ , \\ &\mathbf{0} \leq \boldsymbol{\beta} \leq C\mathbf{e} \ . \end{aligned} \tag{A.6}$$

Note that this problem coincides with our optimisation problem (A.2) except for the first constraint $\mathbf{e}^\top \boldsymbol{\beta} = 1$. The Kuhn–Tucker complementarity conditions for problem (A.1) are

$$\beta_i (R^2 + \xi_i - \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2) = 0 \ , \quad i = 1, \ldots, n \ , \tag{A.7}$$

$$\mu_i \xi_i = 0 \ , \quad i = 1, \ldots, n \ . \tag{A.8}$$

For $0 < \beta_i < C$, equations (A.5) and (A.8) imply that $\mu_i > 0$ and thereby $\xi_i = 0$. Now it follows from (A.7) that

$$\begin{aligned} R^2 &= \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2 \\ &\overset{(A.4)}{=} K(\mathbf{x}_i, \mathbf{x}_i) + \sum_{j,k=1}^{n} \beta_j \beta_k K(\mathbf{x}_j, \mathbf{x}_k) - 2 \sum_{j=1}^{n} \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \kappa + \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta} - 2(\mathbf{K}\boldsymbol{\beta})_i \ . \end{aligned}$$

Let us turn to the SV novelty detection problem investigated by [Schölkopf et al., 2000]. We are looking for a decision function

$$f(\mathbf{x}) = a(\mathbf{x}) + b := \sum_{j=1}^{n} \alpha_j K(\mathbf{x}, \mathbf{x}_j) + b \tag{A.9}$$

with bias term b that solves the modified single-class SV problem

$$\begin{aligned} &\min_{\boldsymbol{\alpha} \in \mathbb{R}^n, \, b \in \mathbb{R}, \, \boldsymbol{\xi} \in \mathbb{R}^n} \ \left(C\mathbf{e}^\top \boldsymbol{\xi} + \frac{1}{2} \|a\|_{\mathcal{H}_K}^2 + b = C\mathbf{e}^\top \boldsymbol{\xi} + \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} + b \right) \\ \text{subject to} \quad &a(\mathbf{x}_i) + b \geq 1 - \xi_i \ , \quad i = 1, \ldots, n \ , \\ &\boldsymbol{\xi} \geq \mathbf{0} \ . \end{aligned} \tag{A.10}$$

Analogous to the SV clustering problem, we build the Lagrangian

$$\mathcal{L}(\boldsymbol{\alpha}, b, \boldsymbol{\xi}, \boldsymbol{\beta}, \boldsymbol{\mu}) := C\mathbf{e}^\top \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} + b - \sum_{i=1}^{n} \beta_i (\sum_{j=1}^{n} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b - 1 + \xi_i) - \boldsymbol{\mu}^\top \boldsymbol{\xi}$$

with Lagrange multipliers $\boldsymbol{\beta}, \boldsymbol{\mu} \geq \mathbf{0}$. Setting the derivative of $\mathcal{L}$ with respect to b, $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ to zero, it follows

$$\begin{aligned} \mathbf{e}^\top \boldsymbol{\beta} &= 1 \ , \\ \boldsymbol{\alpha} &= \boldsymbol{\beta} \ , \\ \boldsymbol{\beta} &= C\mathbf{e} - \boldsymbol{\mu} \ . \end{aligned}$$

Using these equations, the Lagrangian yields the dual problem

$$\begin{aligned} \max_{\boldsymbol{\beta} \in \mathbb{R}^n} \ & 1 - \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{K}\boldsymbol{\beta} \\ \text{subject to} \quad & \mathbf{e}^\top \boldsymbol{\beta} = 1 \ , \\ & \mathbf{0} \leq \boldsymbol{\beta} \leq C\mathbf{e} \ . \end{aligned} \tag{A.11}$$

This problem is obviously equivalent to the dual SV clustering problem (A.6). We summarise:

Lemma 9. *Let K be a kernel with corresponding feature map $\boldsymbol{\phi}$ and with the property that $K(\mathbf{x}, \mathbf{x}) = \kappa$ for all $\mathbf{x} \in \mathcal{X}$. Then the optimisation problems* (A.1) *and* (A.10) *are equivalent in that they lead to equivalent dual problems.*

From the dual solution $\boldsymbol{\beta}$ of (A.6)*, the primal solution $\mathbf{a}, R, \boldsymbol{\xi}$ of* (A.1) *may be obtained by* (A.4) *and* (A.3) *and the Kuhn–Tucker conditions* (A.7) *and* (A.8)*. The optimal values $b, \boldsymbol{\xi}$ for problem* (A.10) *may be obtained by the Kuhn–Tucker complementarity conditions as well.*

This lemma was also proven by [Schölkopf et al., 2000]. Further, Vapnik already showed in [Vapnik, 1998, Chap. 10.7] that R^2 can be computed as described by (A.3) with problem (A.6) for hard margin ($C = \infty$).

At first sight, it is astonishing that although the QPs for SV clustering and SV novelty detection are deviated from different initial problems (A.1) and (A.10), they are equivalent. The report [Schölkopf et al., 1999a] provides a nice geometric interpretation for that: The above condition on the kernel implies that all feature vectors lie on a sphere centred at the origin. The hyperplane that separates the data from the origin with maximal margin is then spanned by the smallest enclosing sphere's centre, that is,

$$a(\mathbf{x}) = f_{\mathbf{a}}(\mathbf{x}) = \langle \mathbf{a}, \boldsymbol{\phi}(\mathbf{x}) \rangle \ .$$

In the hard margin case, the distance of the sphere's centre from the origin is $\sqrt{1-b}$. The relation between R and b in general reads

$$R^2 + (1-b) = \kappa - C\mathbf{e}^\top \boldsymbol{\xi} \ ,$$

where $\boldsymbol{\xi}$ are the residuals with respect to the SV novelty detection problem. (As a matter of fact, the residuals only differ by a factor of two because of the quadratic constraint terms in the clustering problem.) For the hard margin case ($C = \infty$) and the Gaussian kernel, this implies that

$$R^2 = b \ .$$

A.3. Equivalence of the SV Novelty Detection Problem and the Single-Class SVM without Bias Term

The previous subsection shows the equivalence of the SV clustering problem, which can be used for radius computation, to a modified SVM (A.10) with bias term. We now show that this special problem is equivalent to a single-class SVM without bias term. With $a(\mathbf{x})$ defined as in (A.9), the common single-class SVM is described by the problem

$$\begin{aligned} \min_{\boldsymbol{\beta}\in\mathbb{R}^n,\, \boldsymbol{\xi}\in\mathbb{R}^n} \ & \widehat{C}\mathbf{e}^\top\boldsymbol{\xi} + \frac{1}{2}\left\|a\right\|^2_{\mathcal{H}_K} = \widehat{C}\mathbf{e}^\top\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} \\ \text{subject to} \ & a(\mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1,\ldots,n \ , \\ & \boldsymbol{\xi} \geq \mathbf{0} \end{aligned} \tag{A.12}$$

similar to (2.27). Setting up the Lagrangian as above leads to the dual QP (A.2). To prove Theorem 8, it remains to show

Lemma 10. *There exists $\widehat{C} > 0$ such that the SV novelty detection problem* (A.10) *with parameter C is equivalent to the standard SV problem* (A.12) *with parameter $\widehat{C}$ in that the solutions are derivable from one another. The dual solutions $\boldsymbol{\alpha}$ of* (A.2) *and $\boldsymbol{\beta}$ of* (A.11) *are related by*

$$\boldsymbol{\beta} = \frac{\boldsymbol{\alpha}}{\mathbf{e}^\top\boldsymbol{\alpha}}$$

or conversely by $\boldsymbol{\alpha} = \boldsymbol{\beta}/(1-b)$ with the primal variable b from (A.10).

Proof. The proof consists of two parts. Firstly, the dual solution of the biased SVM (A.11) is derived from the dual solution of the SVM without bias (A.2). Secondly, the primal solution of the unbiased SVM (A.12) is derived from the primal solution of the biased SVM (A.10). Due to the duality of convex QPs, this establishes the proof.

1. Suppose problem (A.2) is solved by $\boldsymbol{\alpha}$. With $a := \mathbf{e}^\top\boldsymbol{\alpha} > 0$, set $\boldsymbol{\beta} := \boldsymbol{\alpha}/a$. Then $\boldsymbol{\beta}$ is valid in problem (A.11) if $C = \widehat{C}/a$. Suppose that $\boldsymbol{\beta}$ is not the optimal solution of

problem (A.11), then there exists some $\widetilde{\boldsymbol{\beta}}$ satisfying $\mathbf{e}^\top\widetilde{\boldsymbol{\beta}} = 1$, $\mathbf{0} \leq \widetilde{\boldsymbol{\beta}} \leq C\mathbf{e}$ so that

$$\begin{aligned} \frac{1}{2}\widetilde{\boldsymbol{\beta}}^\top\mathbf{K}\widetilde{\boldsymbol{\beta}} &< \frac{1}{2}\boldsymbol{\beta}^\top\mathbf{K}\boldsymbol{\beta} \\ \Rightarrow \frac{1}{2}(a\widetilde{\boldsymbol{\beta}})^\top\mathbf{K}(a\widetilde{\boldsymbol{\beta}}) &< \frac{1}{2}(a\boldsymbol{\beta})^\top\mathbf{K}(a\boldsymbol{\beta}) \\ \Rightarrow \frac{1}{2}\widetilde{\boldsymbol{\alpha}}^\top\mathbf{K}\widetilde{\boldsymbol{\alpha}} - a &< \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} - a \ , \end{aligned}$$

where $\widetilde{\boldsymbol{\alpha}} := a\widetilde{\boldsymbol{\beta}}$. Since $\widetilde{\boldsymbol{\alpha}}$ fulfils $\mathbf{0} \leq \widetilde{\boldsymbol{\alpha}} \leq \widehat{C}\mathbf{e}$ and $a = \mathbf{e}^\top\widetilde{\boldsymbol{\alpha}}$ holds, this is a contradiction to the assumption that $\boldsymbol{\alpha}$ is the optimal solution of (A.2).

2. On the other hand, let $(\boldsymbol{\beta},\, b,\, \boldsymbol{\xi}^{\boldsymbol{\beta}})$ be the optimal solution of the primal problem (A.10). Then $(\boldsymbol{\alpha} := \boldsymbol{\beta}/(1-b),\, \boldsymbol{\xi}^{\boldsymbol{\alpha}} := \boldsymbol{\xi}^{\boldsymbol{\beta}}/(1-b))$ is a valid solution for (A.12). Note that $b < 1$ due to the dual constraints and the Kuhn–Tucker conditions. Assume that $(\widetilde{\boldsymbol{\alpha}},\, \boldsymbol{\xi}^{\widetilde{\boldsymbol{\alpha}}})$ is valid for (A.12) as well, then $(\widetilde{\boldsymbol{\beta}},\, b,\, \boldsymbol{\xi}^{\widetilde{\boldsymbol{\beta}}}) := ((1-b)\widetilde{\boldsymbol{\alpha}}, b, (1-b)\boldsymbol{\xi}^{\widetilde{\boldsymbol{\alpha}}})$ is valid for problem (A.10). Now we obtain for $\widehat{C} = C/(1-b)$ that

$$\begin{aligned} \frac{1}{2}\widetilde{\boldsymbol{\alpha}}^\top\mathbf{K}\widetilde{\boldsymbol{\alpha}} + \widehat{C}\mathbf{e}^\top\boldsymbol{\xi}^{\widetilde{\boldsymbol{\alpha}}} &< \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} + \widehat{C}\mathbf{e}^\top\boldsymbol{\xi}^{\boldsymbol{\alpha}} \\ \Leftrightarrow \frac{1}{2}((1-b)\widetilde{\boldsymbol{\alpha}})^\top\mathbf{K}((1-b)\widetilde{\boldsymbol{\alpha}}) + (1-b)\widehat{C}\mathbf{e}^\top\left((1-b)\boldsymbol{\xi}^{\widetilde{\boldsymbol{\alpha}}}\right) &< \\ \frac{1}{2}((1-b)\boldsymbol{\alpha})^\top\mathbf{K}((1-b)\boldsymbol{\alpha}) + (1-b)\widehat{C}\mathbf{e}^\top\left((1-b)\boldsymbol{\xi}^{\boldsymbol{\alpha}}\right) & \\ \Leftrightarrow \frac{1}{2}\widetilde{\boldsymbol{\beta}}^\top\mathbf{K}\widetilde{\boldsymbol{\beta}} + C\mathbf{e}^\top\boldsymbol{\xi}^{\widetilde{\boldsymbol{\beta}}} + b &< \frac{1}{2}\boldsymbol{\beta}^\top\mathbf{K}\boldsymbol{\beta} + C\mathbf{e}^\top\boldsymbol{\xi}^{\boldsymbol{\beta}} + b \ . \end{aligned}$$

Consequently, since $(\boldsymbol{\beta}, b, \boldsymbol{\xi}^{\boldsymbol{\beta}})$ is the optimal solution for problem (A.10), $\boldsymbol{\alpha}$ is the optimal solution of (A.12). □

We have shown that for special values of C depending on the data, the biased and unbiased single-class SVMs are equivalent. Anyway, as C is a tuning parameter that cannot be determined analytically, this condition does not restrain the equivalence. Especially for $C = \infty$, the hard margin case, no condition with respect to the weight factor C has to be taken into account.

B. Convexity

In the context of pattern recognition and especially in this work, many tasks are solved by means of optimisation problems. Due to the favourable properties of convex minimisation problems, the theory of convexity is therefore especially important in this setting. Particularly, if the objective function f is convex, one may also handle the case where f is non-differentiable.

Based on the manuscripts [Butzmann, 0203] and [Burger, 2003, Chap. 3] itself based on [Ekeland and Temam, 1976] and the book [Rockafellar, 1970], we first review the basic definitions and some results of convexity in Sec. B.1. Then we introduce subdifferential theory for convex non-differentiable functions in Sec. B.2, conjugate functions in Sec. B.3 and finally establish the duality between optimisation problems in Sec. B.4.

B.1. Basic Concepts

Definition 1. A set $C \subset \mathbb{R}^n$ is called *convex* if

$$\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in C \quad \forall\, \mathbf{x}, \mathbf{y} \in C,\, 0 \leq \alpha \leq 1 \ .$$

Definition 2. Let $C \subset \mathbb{R}^n$ be convex. A function $f : C \to \mathbb{R}$ is called *convex* if

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) \quad \forall\, \mathbf{x}, \mathbf{y} \in C,\, 0 \leq \alpha \leq 1 \ .$$

An equivalent definition requires that the *epigraph* of f, namely $\{(\mathbf{x}, \mu) : \mathbf{x} \in C,\, \mu \geq f(\mathbf{x})\}$, is convex. The definitions naturally extend to functions $f : C \to \mathbb{R}^p$ by applying them to all components.

Definition 3. An optimisation problem

$$\min_{\mathbf{x} \in C} f(\mathbf{x})$$

with $C \subset \mathbb{R}^n$ and $f : C \to \mathbb{R}$ is called *convex* if C and f are convex.

A fundamental property of convex optimisation problems is given by

Theorem 11. *Let $\bar{\mathbf{x}}$ be a local minimiser of the convex optimisation problem*

$$\min_{\mathbf{x} \in C} f(\mathbf{x}) \ .$$

Then $\bar{\mathbf{x}}$ is a global minimiser.

Proof. As $\bar{\mathbf{x}}$ is a local minimiser, for any $\mathbf{x} \in C$ there exists $\epsilon > 0$ so that

$$f(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}} + \alpha(\mathbf{x} - \bar{\mathbf{x}})) \leq f(\bar{\mathbf{x}}) + \alpha(f(\mathbf{x}) - f(\bar{\mathbf{x}})) \quad \forall\, 0 \leq \alpha \leq \epsilon \ ,$$

which leads to

$$\alpha(f(\mathbf{x}) - f(\bar{\mathbf{x}})) \geq 0 \quad \forall\, 0 \leq \alpha \leq \epsilon$$

and hence $f(\mathbf{x}) \geq f(\bar{\mathbf{x}})$. □

For convex problems, the convex feasible set is often characterised by (in)equalities which reads $C = \{\mathbf{x} \in D : g(\mathbf{x}) \leq \mathbf{0}, h(\mathbf{x}) = \mathbf{0}\}$ with $D \subset \mathbb{R}^n$ where $g : D \to \mathbb{R}^p$, $h : D \to \mathbb{R}^q$ are convex. Then optimality conditions are given in terms of Kuhn–Tucker points:

Definition 4. Let $D \subset \mathbb{R}^n$ be open and $f : D \to \mathbb{R}$, $g : D \to \mathbb{R}^p$ and $h : D \to \mathbb{R}^q$ be continuously differentiable. A feasible point $\mathbf{x} \in D$ is called *Kuhn–Tucker point* of the problem

$$\min_{\substack{g(\mathbf{x}) \leq \mathbf{0} \\ h(\mathbf{x}) = \mathbf{0}}} f(\mathbf{x})$$

if there exist $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, $\boldsymbol{\mu} \in \mathbb{R}^q$ so that

(i) $\nabla f(\mathbf{x}) + \sum_{i=1}^p \lambda_i \nabla g_i(\mathbf{x}) + \sum_{j=1}^q \mu_j \nabla h_j(\mathbf{x}) = 0$,

(ii) $\lambda_i g_i(\mathbf{x}) = 0 \quad \forall\, i = 1, \ldots, p$.

The conditions in Def. 4 are called *Kuhn–Tucker conditions*, conditions (ii) are called *Kuhn–Tucker complementarity conditions* in particular.

We are now able to state necessary and sufficient conditions for optimality:

Theorem 12. *Let $C \subset \mathbb{R}^n$ be open and convex, $f : C \to \mathbb{R}$ and $g : C \to \mathbb{R}^p$ be convex and differentiable and $h : C \to \mathbb{R}^q$ affine. For the problem*

$$\min_{\substack{g(\mathbf{x}) \leq \mathbf{0} \\ h(\mathbf{x}) = \mathbf{0}}} f(\mathbf{x})$$

then hold:

(i) Every Kuhn–Tucker point is a solution.

(ii) If there exists $\mathbf{x} \in C$ with $g(\mathbf{x}) \leq \mathbf{0}$, $h(\mathbf{x}) = \mathbf{0}$ and $g_i(\mathbf{x}) < 0$ for all g_i that are not affine, then every solution is a Kuhn–Tucker point.

Proof. [Butzmann, 0203, Theorem 6.21] □

The condition for the necessity of the Kuhn–Tucker conditions in (ii) is called *Slater condition*.

We continue with some characterisations of convex functions:

Proposition 13. *Let $D \subset \mathbb{R}^n$ be open, $C \subset D$ be convex and $f : D \to \mathbb{R}$ be twice continuously differentiable. Then f is convex on C if and only if its second derivative $Hf(\mathbf{x})$ is positive semidefinite for every $\mathbf{x} \in C$.*

Proof. [Burger, 2003, Prop. 3.3], [Rockafellar, 1970, Theorem 4.5] □

Strict convexity is generally given by excluding equality in the inequality conditions on convexity as in Def. 2. To measure the degree of convexity, according to [Pham Dinh and Hoai An, 1998, equation (5)] we introduce

Definition 5. Let $C \subset \mathbb{R}^n$ and $f : C \to \mathbb{R}$ be convex. We denote the *modulus of strict convexity* $\rho(f, C)$ by

$$\rho(f, C) := \sup\{\rho \geq 0 : f - \frac{\rho}{2} \|\cdot\|^2 \text{ is convex on } C\} \ .$$

Again f is *strictly convex* if $\rho(f, C) > 0$. If f is twice continuously differentiable, by Prop. 13, we have $\rho(f, C) = \sup\{\rho \geq 0 : Hf(\mathbf{x}) - \rho\mathbf{I} \text{ is positive semidefinite } \forall\, \mathbf{x} \in C\}$.

Proposition 14. *Let $C \subset \mathbb{R}^n$ be open and convex and $f : C \to \mathbb{R}^p$ convex. Then f is continuous.*

Proof. [Butzmann, 0203, Prop. 5.19] □

B.2. Subgradients

Many results and methods in optimisation rely on the gradient. For non-differentiable functions, it is possible to consider subgradients instead. As we have the application of convexity theory to the solution of real-valued problems in mind, we introduce the theory of subgradients for functions on $\mathbb{R}^n$ instead of general Banach spaces. This also simplifies the considerations insofar as the dual space to $\mathbb{R}^n$ may be characterised by $\mathbb{R}^n$ itself.

So, if $f : \mathbb{R}^n \to \mathbb{R}$ is convex, even if it is non-differentiable, it is possible to define a generalised gradient at all points $\mathbf{x} \in \mathbb{R}^n$. To motivate this, first assume that f is convex and twice continuously differentiable. By Prop. 13 its second derivative is positive semidefinite, so the Taylor formula at $\mathbf{x} \in \mathbb{R}^n$ yields by

$$\begin{aligned} f(\mathbf{y}) &= f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top Hf(\alpha\mathbf{x} + (1-\alpha)\mathbf{y})(\mathbf{y} - \mathbf{x}) \quad (0 < \alpha < 1) \\ &\geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \end{aligned}$$

an affine minorisation of f for all $\mathbf{y} \in \mathbb{R}^n$, or, more precisely, a supporting hyperplane of the epigraph of f at $\mathbf{x}$. This inspires

Definition 6. Let $f : C \to \mathbb{R}$ with $C \subset \mathbb{R}^n$ be convex. The *subdifferential* of f at $\mathbf{x} \in C$ is defined by

$$\partial f(\mathbf{x}) := \{\mathbf{x}^* \in \mathbb{R}^n : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{x}^*, \mathbf{y} - \mathbf{x} \rangle \quad \forall\, \mathbf{y} \in C\} \ .$$

Each element of this set is called *subgradient* of f at $\mathbf{x}$.

The subdifferential $\partial f(\mathbf{x})$ is a closed convex set for all $\mathbf{x} \in C$. To see the difference to the usual gradient, have a look at

Example 1. Consider the absolute value function $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto |x|$ which is of course convex, but non-differentiable at $x = 0$. For $x > 0$, select $0 < x_1 < x < x_2$. Then $x^* \in \partial f(x)$ implies

$$(x_1 - x)(1 - x^*) \geq 0 \ , \quad (x_2 - x)(1 - x^*) \geq 0 \ ,$$

which is equivalent to $x^* = 1$. Indeed, it holds

$$|y| \geq |x| + (y - x) = y$$

for all $y \in \mathbb{R}$, and hence $\partial f(x) = \{1\}$. Similarly, for $x < 0$ we have $\partial f(x) = \{-1\}$. For $x = 0$, the subgradient condition is

$$|y| \geq x^* y \quad \forall\, y \in \mathbb{R} \ ,$$

which is satisfied if and only if $x^* \in [-1, 1]$. Thus we have

$$\partial f(x) = \begin{cases} \{-1\} & x < 0 \ , \\ [-1, 1] & x = 0 \ , \\ \{1\} & x > 0 \ , \end{cases}$$

so the subdifferential at all differentiable points is just the set containing the gradient, and an interval — that is, a convex set — at $x = 0$.

As observed in the example, the subgradient is a true generalisation of the gradient:

Proposition 15. *Let $f : C \to \mathbb{R}$ be convex and differentiable at $\mathbf{x} \in C$. Then*

$$\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\} \ .$$

Proof. As f is convex and differentiable at $\mathbf{x}$, we have for any $\mathbf{y} \in C$ again by the Taylor formula

$$f(\mathbf{x}) + \alpha(f(\mathbf{y}) - f(\mathbf{x})) \geq f(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x})) = f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + R_{\mathbf{x}}(\alpha(\mathbf{y} - \mathbf{x}))$$

for $0 < \alpha \leq 1$, which leads to

$$f(\mathbf{y}) - f(\mathbf{x}) \geq \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{1}{\alpha} R_{\mathbf{x}}(\alpha(\mathbf{y} - \mathbf{x})) \ ,$$

so that by

$$\frac{1}{\alpha} R_{\mathbf{x}}(\alpha(\mathbf{y} - \mathbf{x})) = \|\mathbf{y} - \mathbf{x}\| \frac{1}{\|\alpha(\mathbf{y} - \mathbf{x})\|} R_{\mathbf{x}}(\alpha(\mathbf{y} - \mathbf{x})) \xrightarrow{\alpha \to 0} 0$$

for $\mathbf{y} \neq \mathbf{x}$ follows $\nabla f(\mathbf{x}) \in \partial f(\mathbf{x})$.

On the other hand, select $\mathbf{x}^* \in \partial f(\mathbf{x})$. If f is differentiable at $\mathbf{x}$, then $\mathbf{x}$ has to be an interior point of C. Then for each $\mathbf{y} \in \mathbb{R}^n$ there exists $\alpha > 0$ so that $\mathbf{x} + \alpha\mathbf{y} \in C$ and the subgradient inequality yields

$$\begin{aligned} \frac{f(\mathbf{x} + \alpha\mathbf{y}) - f(\mathbf{x})}{\alpha} &\geq \langle \mathbf{x}^*, \mathbf{y} \rangle \ , \\ \frac{f(\mathbf{x} - \alpha\mathbf{y}) - f(\mathbf{x})}{\alpha} &\geq -\langle \mathbf{x}^*, \mathbf{y} \rangle \ , \end{aligned}$$

which implies for $\alpha \to 0$ in the limit $\nabla f(\mathbf{x})^\top \mathbf{y} = \langle \mathbf{x}^*, \mathbf{y} \rangle$. As this holds for all $\mathbf{y} \in \mathbb{R}^n$, it follows $\mathbf{x}^* = \nabla f(\mathbf{x})$ which completes the proof. □

Expectedly, a local optimality condition for convex problems also translates from gradients to subgradients:

Proposition 16. *Let $f : C \to \mathbb{R}$ with $C \subset \mathbb{R}^n$ be convex. Then $\mathbf{x} \in C$ is a minimiser of f if and only if*

$$\mathbf{0} \in \partial f(\mathbf{x}) \ .$$

Proof. If $\mathbf{0} \in \partial f(\mathbf{x})$, then

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{0}, \mathbf{y} - \mathbf{x} \rangle = f(\mathbf{x}) \quad \forall\, \mathbf{y} \in C$$

and hence $\mathbf{x}$ is a global minimiser. If, on the other hand, $\mathbf{0} \notin \partial f(\mathbf{x})$, there exists $\mathbf{y} \in C$ so that

$$f(\mathbf{y}) < f(\mathbf{x}) + \langle \mathbf{0}, \mathbf{y} - \mathbf{x} \rangle = f(\mathbf{x})$$

and $\mathbf{x}$ cannot be a global minimiser. □

Recall that, by Theorem 11, each local minimiser is also a global one.

B.3. Conjugate Functions

The concept of duality between optimisation problems attains particular interest for convex problems as properties of these problems and their solutions may often be characterised by their duals. When constraints are incorporated into the objective, it is necessary to generalise to functions mapping to $\overline{\mathbb{R}}$ instead of $\mathbb{R}$. Hence we first introduce

Definition 7. Let $f : C \to \overline{\mathbb{R}}$ be convex. We define the *domain* of f by

$$\operatorname{dom} f = \{\mathbf{x} \in C : f(\mathbf{x}) < \infty\} \ .$$

Although for problems with non-differentiable functions, other notions than the Lagrangian duality have to be considered, it is also possible to generalise the Lagrangian duality concept. In any case, the duality is based on conjugate functions:

Definition 8. Let $f : D \to \overline{\mathbb{R}}$ with $D \subset \mathbb{R}^n$. Then its *conjugate function* $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ is defined by

$$f^*(\mathbf{x}^*) = \sup_{\mathbf{x} \in D} \{\langle \mathbf{x}^*, \mathbf{x} \rangle - f(\mathbf{x})\} \ .$$

Note that f does not have to be convex. The passing from a function to its conjugate is also called *Legendre transform*.

Proposition 17. *Let $f : D \to \overline{\mathbb{R}}$. Then f^* is convex.*

Proof. For $\mathbf{x}^*, \mathbf{y}^* \in \mathbb{R}^n$, $0 \leq \alpha \leq 1$ we have

$$\begin{aligned} f^*(\alpha \mathbf{x}^* + (1-\alpha)\mathbf{y}^*) &= \sup_{\mathbf{x} \in D} \{\alpha \langle \mathbf{x}^*, \mathbf{x} \rangle + (1-\alpha)\langle \mathbf{y}^*, \mathbf{x} \rangle - f(\mathbf{x})\} \\ &\leq \sup_{\mathbf{x}, \mathbf{y} \in D} \{\alpha(\langle \mathbf{x}^*, \mathbf{x} \rangle - f(\mathbf{x})) + (1-\alpha)(\langle \mathbf{y}^*, \mathbf{y} \rangle - f(\mathbf{y}))\} \\ &= \alpha \sup_{\mathbf{x} \in D} \{\langle \mathbf{x}^*, \mathbf{x} \rangle - f(\mathbf{x})\} + (1-\alpha) \sup_{\mathbf{y} \in D} \{\langle \mathbf{y}^*, \mathbf{y} \rangle - f(\mathbf{y})\} \\ &= \alpha f^*(\mathbf{x}^*) + (1-\alpha) f^*(\mathbf{y}^*) \ . \end{aligned}$$

□

As we are now dealing with functions with image $\overline{\mathbb{R}}$ instead of $\mathbb{R}$, further characterisation of the functions' nature is necessary:

Definition 9. Let $f : C \to \overline{\mathbb{R}}$ be convex. The function f is said to be *proper* if $f(\mathbf{x}) < +\infty$ for at least one $\mathbf{x} \in C$ and $f(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in C$.

Proper concave functions may be defined by sign inversion.

As, first of all, f^* may be regarded as a pointwise supremum of affine functions $\langle \cdot, \mathbf{x} \rangle - \mu$ such that $(\mathbf{x}, \mu)$ belongs to the epigraph of f and, further, the only lower semi-continuous improper convex functions are the constant functions $+\infty$ and $-\infty$ which are conjugate to each other, we state

Proposition 18 ([Rockafellar, 1970, Theorem 12.2]). *Let f be convex. Then its conjugate f^* is lower semi-continuous and f^* is proper if and only if f is proper.*

Example 2. For $C \subset \mathbb{R}^n$ consider the indicator function

$$\chi_C : \mathbb{R}^n \to \mathbb{R} \ , \quad \mathbf{x} \mapsto \begin{cases} 0 & \mathbf{x} \in C \ , \\ \infty & \text{otherwise} \ . \end{cases}$$

If C is convex, so is χ_C. The indicator function may be used to incorporate constraints in the objective function and, hence, transform a constrained into an unconstrained problem with objective in $\overline{\mathbb{R}}$. Its conjugate function is

$$\chi_C^*(\mathbf{x}^*) = \sup_{\mathbf{x} \in \mathbb{R}^n} \left\{ \langle \mathbf{x}^*, \mathbf{x} \rangle - \begin{cases} 0 & \mathbf{x} \in C \ , \\ \infty & \text{otherwise} \end{cases} \right\} = \sup_{x \in C} \{ \langle \mathbf{x}^*, \mathbf{x} \rangle \} \ .$$

Another prominent example is

Example 3. A class of lower semi-continuous proper convex functions are the norm functions

$$f_p : \mathbb{R}^n \to \mathbb{R} \ , \quad \mathbf{x} \mapsto \frac{1}{p} \|\mathbf{x}\|_p^p$$

for $p \geq 1$. For $p > 1$ and $\mathbf{x}^* \in \mathbb{R}^n$, the supremum over $\langle \mathbf{x}^*, \mathbf{x} \rangle - f_p(\mathbf{x})$, which is concave, is attained at $\mathbf{x} = \operatorname{sgn}(\mathbf{x}^*)|\mathbf{x}^*|^{1/(p-1)}$ by Prop. 16 so that

$$\begin{aligned} f_p^*(\mathbf{x}^*) &= \left\langle \mathbf{x}^*, \operatorname{sgn}(\mathbf{x}^*)|\mathbf{x}^*|^{1/(p-1)} \right\rangle - \frac{1}{p} \left\| \operatorname{sgn}(\mathbf{x}^*)|\mathbf{x}^*|^{1/(p-1)} \right\|_p^p \\ &= \|\mathbf{x}^*\|_{p/(p-1)}^{p/(p-1)} - \frac{1}{p} \|\mathbf{x}^*\|_{p/(p-1)}^{p/(p-1)} \\ &= \frac{p-1}{p} \|\mathbf{x}^*\|_{p/(p-1)}^{p/(p-1)} \\ &= f_{p/(p-1)}(\mathbf{x}^*) \ . \end{aligned}$$

So in general, for $q = p/(p-1)$ or $1/p + 1/q = 1$, it holds $f_p^* = f_q$. For $p = 2$, we have $f_2^* = f_2$, so the squared Euclidean norm is self-conjugate. For $p = 1$ and $q = \infty$, instead $f_1^* = \chi_{[-\mathbf{e}, \mathbf{e}]}$ and $(\|\cdot\|_\infty)^* = \chi_{\|\cdot\|_1 \leq 1}$.

As may already be conjectured from the example, conjugacy often defines a symmetric relationship:

Proposition 19. *Let $f : D \to \overline{\mathbb{R}}$ with $D \subset \mathbb{R}^n$. Then $f^{**}(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in D$ and $g(\mathbf{x}) \leq f^{**}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$ if $g(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in D$ and $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is convex. In particular, $f^{**} = f$ if and only if f is convex.*

Proof. For $\mathbf{x} \in \mathbb{R}^n$ we compute f^{**} as

$$\begin{aligned} f^{**}(\mathbf{x}) &= \sup_{\mathbf{x}^* \in \mathbb{R}^n} \{\langle \mathbf{x}, \mathbf{x}^* \rangle - f^*(\mathbf{x}^*)\} \\ &= \sup_{\mathbf{x}^* \in \mathbb{R}^n} \{\langle \mathbf{x}^*, \mathbf{x} \rangle - \sup_{\mathbf{y} \in D} \{\langle \mathbf{x}^*, \mathbf{y} \rangle - f(\mathbf{y})\}\} \\ &= \sup_{\mathbf{x}^* \in \mathbb{R}^n} \inf_{\mathbf{y} \in D} \{\langle \mathbf{x}^*, \mathbf{x} - \mathbf{y} \rangle + f(\mathbf{y})\} \ . \end{aligned}$$

As for any $\mathbf{x} \in D$, $\mathbf{x}^* \in \mathbb{R}^n$ we have

$$\inf_{\mathbf{y} \in D} \{\langle \mathbf{x}^*, \mathbf{x} - \mathbf{y} \rangle + f(\mathbf{y})\} \leq \langle \mathbf{x}^*, \mathbf{x} - \mathbf{x} \rangle + f(\mathbf{x}) = f(\mathbf{x}) \ ,$$

this proves that $f^{**} \leq f$.

Now assume that g is convex. Then either $g \equiv \pm\infty$, which is no contradiction to the proposition, or g is proper. In the latter case, by [Rockafellar, 1970, Theorem 23.4], there exist $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y}^* \in \partial g(\mathbf{x})$. Then $g(\mathbf{y}) \geq g(\mathbf{x}) + \langle \mathbf{y}^*, \mathbf{y} - \mathbf{x} \rangle$ for all $\mathbf{y} \in \mathbb{R}^n$ and it follows from the calculation above

$$\begin{aligned} g^{**}(\mathbf{x}) &= \sup_{\mathbf{x}^* \in \mathbb{R}^n} \inf_{\mathbf{y} \in \mathbb{R}^n} \{\langle \mathbf{x}^*, \mathbf{x} - \mathbf{y} \rangle + g(\mathbf{y})\}\} \\ &\geq \sup_{\mathbf{x}^* \in \mathbb{R}^n} \inf_{\mathbf{y} \in \mathbb{R}^n} \{\langle \mathbf{x}^* - \mathbf{y}^*, \mathbf{x} - \mathbf{y} \rangle + g(\mathbf{x})\}\} \\ &\geq \inf_{\mathbf{y} \in \mathbb{R}^n} \{\langle \mathbf{y}^* - \mathbf{y}^*, \mathbf{x} - \mathbf{y} \rangle + g(\mathbf{x})\}\} = g(\mathbf{x}) \end{aligned}$$

and consequently $g^{**} = g$. If $g(\mathbf{y}) \leq f(\mathbf{y})$ for all $\mathbf{y} \in D$ then again by the calculation above

$$\begin{aligned} g(\mathbf{x}) = g^{**}(\mathbf{x}) &= \sup_{\mathbf{x}^* \in \mathbb{R}^n} \inf_{\mathbf{y} \in \mathbb{R}^n} \{\langle \mathbf{x}^*, \mathbf{x} - \mathbf{y} \rangle + g(\mathbf{y})\}\} \\ &\leq \sup_{\mathbf{x}^* \in \mathbb{R}^n} \inf_{\mathbf{y} \in D} \{\langle \mathbf{x}^*, \mathbf{x} - \mathbf{y} \rangle + f(\mathbf{y})\}\} = f^{**}(\mathbf{x}) \ , \end{aligned}$$

which proves the second statement. □

The proposition states that f^{**} is the maximal convex function below f.

We now state other useful relations characterising the subgradient with the help of the conjugate function:

Proposition 20. *Let $f : D \to \overline{\mathbb{R}}$ with $D \subset \mathbb{R}^n$ be lower semi-continuous and proper convex. Then for any $\mathbf{x} \in D$, $\mathbf{x}^* \in \mathbb{R}^n$ hold*

$$\begin{aligned} \partial f(\mathbf{x}) &= \arg\max_{\mathbf{y} \in \mathbb{R}^d} \{\mathbf{x}^\top \mathbf{y} - f^*(\mathbf{y})\} \ , \\ \partial f^*(\mathbf{x}^*) &= \arg\max_{\mathbf{x} \in \mathbb{R}^d} \{(\mathbf{x}^*)^\top \mathbf{x} - f(\mathbf{x})\} \ . \end{aligned}$$

Proof. The equivalence of (a) and (b*) in [Rockafellar, 1970, Theorem 23.5] establishes the first equality, of (a*) and (b) the second one. □

B.4. Optimisation: Duality

To introduce dual problems, according to [Burger, 2003, Chap. 3.2] we consider bivariate functions $\phi : \mathbb{R}^n \times \mathbb{R}^p \to \overline{\mathbb{R}}$. We assume that our original objective f is obtained by $\phi(\mathbf{x}, \mathbf{0}) = f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$ so that the primal problem reads

$$\min_{\mathbf{x} \in \mathbb{R}^n} \phi(\mathbf{x}, \mathbf{0}) \ . \tag{P}$$

Now the conjugate function $\phi^* : \mathbb{R}^n \times \mathbb{R}^p \to \overline{\mathbb{R}}$ is given by

$$\phi^*(\mathbf{x}^*, \mathbf{y}^*) = \sup_{\mathbf{x} \in \mathbb{R}^n, \, \mathbf{y} \in \mathbb{R}^p} \{\langle \mathbf{x}^*, \mathbf{x} \rangle + \langle \mathbf{y}^*, \mathbf{y} \rangle - \phi(\mathbf{x}, \mathbf{y})\}$$

and the dual is defined as

$$. \max_{\mathbf{y}^* \in \mathbb{R}^p} \{-\phi^*(\mathbf{0}, \mathbf{y}^*)\} \ . \tag{P^*}$$

The variables $\mathbf{y}^* \in \mathbb{R}^p$ are the *Lagrange multiplier*s of (P) with respect to ϕ. For all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y}^* \in \mathbb{R}^p$ we have

$$\begin{aligned} -\phi^*(\mathbf{0}, \mathbf{y}^*) &= \inf_{\mathbf{x} \in \mathbb{R}^n, \, \mathbf{y} \in \mathbb{R}^p} \{\phi(\mathbf{x}, \mathbf{y}) - \langle \mathbf{y}^*, \mathbf{y} \rangle\} \\ &\leq \phi(\mathbf{x}, \mathbf{0}) - \langle \mathbf{y}^*, \mathbf{0} \rangle \qquad\qquad = \phi(\mathbf{x}, \mathbf{0}) \end{aligned}$$

and hence we obtain the classical duality gap relation

$$\sup (P^*) \leq \inf (P) \ . \tag{B.1}$$

In the convex case, we even have a tighter relationship between the dual programs:

Theorem 21. *Let $\phi : \mathbb{R}^n \times \mathbb{R}^p \to \overline{\mathbb{R}}$ be convex. Then the following statements are equivalent for $\bar{\mathbf{x}} \in \mathbb{R}^n$ and $\bar{\mathbf{y}}^* \in \mathbb{R}^p$:*

(i) Problems (P) and (P^*) have solutions $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}^*$, and $\inf P = \sup P^*$.

(ii) $\phi(\bar{\mathbf{x}}, \mathbf{0}) + \phi^*(\mathbf{0}, \bar{\mathbf{y}}^*) = 0$.

(iii) $(\mathbf{0}, \bar{\mathbf{y}}^*) \in \partial\phi(\bar{\mathbf{x}}, \mathbf{0})$.

Proof. If (i) holds, then

$$\phi(\bar{\mathbf{x}}, \mathbf{0}) = \inf P = \sup P^* = -\phi^*(\mathbf{0}, \bar{\mathbf{y}}^*)$$

which implies (ii). Vice versa, if (ii) holds, then by (B.1) we have

$$\sup P^* \leq \phi(\bar{\mathbf{x}}, \mathbf{0}) = -\phi^*(\mathbf{0}, \bar{\mathbf{y}}^*) \leq \inf P$$

which implies (i) by the crosswise inequalities. Finally, (iii) reads

$$\begin{aligned} & (\mathbf{0}, \bar{\mathbf{y}}^*) \in \partial\phi(\bar{\mathbf{x}}, \mathbf{0}) \\ \Leftrightarrow \quad & \phi(\mathbf{x}, \mathbf{y}) \geq \phi(\bar{\mathbf{x}}, \mathbf{0}) + \langle \bar{\mathbf{y}}^*, \mathbf{y} \rangle \quad \forall\, \mathbf{x} \in \mathbb{R}^n,\, \mathbf{y} \in \mathbb{R}^p \\ \Leftrightarrow \quad & \phi(\bar{\mathbf{x}}, \mathbf{0}) \leq \inf_{\mathbf{x} \in \mathbb{R}^n,\, \mathbf{y} \in \mathbb{R}^p} \{\phi(\mathbf{x}, \mathbf{y}) - \langle \bar{\mathbf{y}}^*, \mathbf{y} \rangle\} = -\phi^*(\mathbf{0}, \bar{\mathbf{y}}^*) \ , \end{aligned}$$

which is equivalent to (i) by (B.1). □

The dual of (P^*), the bidual problem, is given by

$$\min_{\mathbf{x} \in \mathbb{R}^n} \phi^{**}(\mathbf{x}, \mathbf{0}) \ ,$$

which again yields (P) if ϕ is convex by Prop. 19.

At last, we now apply our duality concept to linear programs:

Example 4. For $\mathbf{b} \in \mathbb{R}^p$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{p \times n}$ consider the LP in canonical form

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \ , \\ & \mathbf{x} \geq 0 \ . \end{aligned}$$

With $X_{\mathbf{y}} := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} - \mathbf{b} \leq \mathbf{y}\}$ we set $\phi(\mathbf{x}, \mathbf{y}) := \mathbf{c}^\top \mathbf{x} + \chi_{X_{\mathbf{y}}}(\mathbf{x}) + \chi_{\mathbb{R}^n_{0+}}(\mathbf{x})$. Then

$$\begin{aligned} -\phi^*(0, \mathbf{y}^*) \quad &= \quad \inf_{\mathbf{x} \in \mathbb{R}^n,\, \mathbf{y} \in \mathbb{R}^p} (\phi(\mathbf{x}, \mathbf{y}) - \langle \mathbf{y}, \mathbf{y}^* \rangle) \\ &= \quad \inf_{\mathbf{x} \in \mathbb{R}^n,\, \mathbf{y} \in \mathbb{R}^p} \left(\mathbf{c}^\top \mathbf{x} + \chi_{X_{\mathbf{y}}}(\mathbf{x}) + \chi_{\mathbb{R}^n_{0+}}(\mathbf{x}) - \langle \mathbf{y}, \mathbf{y}^* \rangle \right) \\ &= \quad \inf_{\mathbf{A}\mathbf{x} - \mathbf{b} \leq \mathbf{y},\, \mathbf{x} \in \mathbb{R}^n_{0+}} \left(\mathbf{c}^\top \mathbf{x} - \langle \mathbf{y}, \mathbf{y}^* \rangle \right) \\ &\overset{\mathbf{y} = \mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{v}}{=} \quad \inf_{\mathbf{x} \in \mathbb{R}^n_{0+},\, \mathbf{v} \in \mathbb{R}^p_{0+}} \left(\mathbf{c}^\top \mathbf{x} - (\mathbf{y}^*)^\top (\mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{v}) \right) \\ &= \quad \inf_{\mathbf{x} \in \mathbb{R}^n_{0+},\, \mathbf{v} \in \mathbb{R}^p_{0+}} \left((\mathbf{c} - \mathbf{A}^\top \mathbf{y}^*)^\top \mathbf{x} - (\mathbf{y}^*)^\top \mathbf{v} + \mathbf{b}^\top \mathbf{y}^* \right) \\ &= \quad \inf_{\mathbf{x} \in \mathbb{R}^n_{0+}} \left((\mathbf{c} - \mathbf{A}^\top \mathbf{y}^*)^\top \mathbf{x} \right) + \inf_{\mathbf{v} \in \mathbb{R}^p_{0+}} \left((-\mathbf{y}^*)^\top \mathbf{v} \right) + \mathbf{b}^\top \mathbf{y}^* \\ &= \quad \begin{cases} \mathbf{b}^\top \mathbf{y}^* & \mathbf{A}^\top \mathbf{y}^* \leq \mathbf{c},\, \mathbf{y}^* \leq \mathbf{0} \ , \\ -\infty & \text{otherwise} \end{cases} \\ &\longrightarrow \quad \max_{\mathbf{y}^* \in \mathbb{R}^p} \ , \end{aligned}$$

which leads to the usual dual

$$\begin{aligned} \max_{\mathbf{y}\in\mathbb{R}^p} \quad & \mathbf{b}^\top \mathbf{y} \\ \text{subject to} \quad & \mathbf{A}^\top \mathbf{y} \leq \mathbf{c} \ , \\ & \mathbf{y} \leq 0 \ . \end{aligned}$$

In Sec. 5.4 we introduced d.c. (difference of convex functions) optimisation problems. These problems are, in general, not convex. In the d.c. optimisation context, a different notion of duality as introduced by [Toland, 1979] is used. With the above concept, it is not possible to deduce the commonly used dual d.c. program and vice versa for the common LP dual.

We now introduce an alternative concept of duality between optimisation problems according to [Toland, 1979] and then reflect some properties of the new concept and compare it with the one just introduced. Again we restrict ourselves to working in the particular spaces $\mathbb{R}^n$. Starting from the problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{$\widetilde{P}$}$$

with $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ supposed to be non-convex, again we introduce a bivariate function $\widetilde{\phi} : \mathbb{R}^n \times \mathbb{R}^p \to \overline{\mathbb{R}}$ so that $\widetilde{\phi}(\mathbf{x}, \mathbf{0}) = -f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$ and also $\widetilde{\phi}_{\mathbf{x}} = \widetilde{\phi}(\mathbf{x}, \cdot)$ is convex and lower semi-continuous for all $\mathbf{x} \in \mathbb{R}^n$ or just $\widetilde{\phi}^{**}_{\mathbf{x}}(\mathbf{0}) = \widetilde{\phi}_{\mathbf{x}}(\mathbf{0})$ for all $\mathbf{x} \in \mathbb{R}^n$. The second condition is a direct consequence of the first one by Prop. 19. With the Lagrangian function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^p \to \overline{\mathbb{R}}$ given by

$$-\mathcal{L}(\mathbf{x}, \mathbf{y}^*) := \sup_{\mathbf{y}\in\mathbb{R}^p} \{\langle \mathbf{y}^*, \mathbf{y}\rangle - \widetilde{\phi}(\mathbf{x}, \mathbf{y})\} \ ,$$

which is the negative of $\widetilde{\phi}^*_{\mathbf{x}}(\mathbf{y}^*)$, we define

$$-L(\mathbf{y}^*) := \sup_{\mathbf{x}\in\mathbb{R}^n} \mathcal{L}(\mathbf{x}, \mathbf{y}^*) \ .$$

Then the dual optimisation problem is

$$\min_{\mathbf{y}^*\in\mathbb{R}^p} L(\mathbf{y}^*) \ . \tag{$\widetilde{P}^*$}$$

Like the classical duality of differentiable functions, this duality is also defined via the Lagrangian function. But even if f is not bounded from above and $\mathcal{L}$ does not have a saddle point, the dual problem is defined. If $\widetilde{\phi}$ satisfies the conditions above, we have similar relationships as for the formerly introduced duality concept: The primal and dual optimum values are equal and dual solutions may be obtained from primal ones via

subgradient relations as in Theorem 21. But the definitions differ substantially: If we put the definitions constituting the dual problem $(\widetilde{P}^*)$ together we have to minimise

$$L(\mathbf{y}^*) = \inf_{\mathbf{x}\in\mathbb{R}^n} \sup_{\mathbf{y}\in\mathbb{R}^p} \{\langle \mathbf{y}^*, \mathbf{y}\rangle - \widetilde{\phi}(\mathbf{x}, \mathbf{y})\}$$

for $\mathbf{y}^* \in \mathbb{R}^p$ whereas the former notion yields

$$\phi^*(\mathbf{0}, \mathbf{y}^*) = \sup_{\mathbf{x}\in\mathbb{R}^n} \sup_{\mathbf{y}\in\mathbb{R}^p} \{\langle \mathbf{y}^*, \mathbf{y}\rangle - \phi(\mathbf{x}, \mathbf{y})\} \longrightarrow \min_{\mathbf{y}^*\in\mathbb{R}^p} \ ,$$

where both problems are different even for $\phi = -\widetilde{\phi}$.

Example 5. With our new concept of duality, for the d.c. problem (5.14) with two lower semi-continuous, proper convex functions $g, h : \mathbb{R}^n \to \overline{\mathbb{R}}$ we define $\widetilde{\phi}(\mathbf{x}, \mathbf{y}) := -g(\mathbf{x}) + h(\mathbf{x} + \mathbf{y})$. Then $\widetilde{\phi}(\mathbf{x}, \cdot)$ is convex and lower semi-continuous for all $\mathbf{x} \in \mathbb{R}^n$ and the the dual objective to minimise reads

$$\begin{aligned}
& \inf_{\mathbf{x}\in\mathbb{R}^n} \sup_{\mathbf{y}\in\mathbb{R}^n} \{\langle \mathbf{y}^*, \mathbf{y}\rangle - h(\mathbf{x}+\mathbf{y}) + g(\mathbf{x})\} \\
\overset{\mathbf{z}=\mathbf{x}+\mathbf{y}}{=} & \inf_{\mathbf{x}\in\mathbb{R}^n} \sup_{\mathbf{z}\in\mathbb{R}^n} \{\langle \mathbf{y}^*, \mathbf{z}\rangle - h(\mathbf{z}) - \langle \mathbf{y}^*, \mathbf{x}\rangle + g(\mathbf{x})\} \\
= & \inf_{\mathbf{x}\in\mathbb{R}^n} \{h^*(\mathbf{y}^*) - \langle \mathbf{y}^*, \mathbf{x}\rangle + g(\mathbf{x})\} \\
= & \ h^*(\mathbf{y}^*) - \sup_{\mathbf{x}\in\mathbb{R}^n} \{\langle \mathbf{y}^*, \mathbf{x}\rangle - g(\mathbf{x})\} \\
= & \ h^*(\mathbf{y}^*) - g^*(\mathbf{y}^*) \ .
\end{aligned}$$

Following [Pham Dinh and Hoai An, 1998, Sec. 3], due to $h^{**} = h$ by Prop. 19 we can also deduce directly that

$$\begin{aligned}
\inf_{\mathbf{x}\in\mathbb{R}^n} \{g(\mathbf{x}) - h(\mathbf{x})\} &= \inf_{\mathbf{x}\in\mathbb{R}^n} \left\{ g(\mathbf{x}) - \sup_{\mathbf{y}\in\mathbb{R}^n} \{\langle \mathbf{x}, \mathbf{y}\rangle - h^*(\mathbf{y})\} \right\} \\
&= \inf_{\mathbf{x},\mathbf{y}\in\mathbb{R}^n} \{g(\mathbf{x}) - \langle \mathbf{x}, \mathbf{y}\rangle + h^*(\mathbf{y})\} \\
&= \inf_{\mathbf{y}\in\mathbb{R}^n} \left\{ h^*(\mathbf{y}) - \sup_{\mathbf{x}\in\mathbb{R}^n} \{\langle \mathbf{x}, \mathbf{y}\rangle - g(\mathbf{x})\} \right\} \\
&= \inf_{\mathbf{y}\in\mathbb{R}^n} \{h^*(\mathbf{y}) - g^*(\mathbf{y})\} \ ,
\end{aligned}$$

which also establishes the relationship between the primal and dual problems.

Bibliography

[Alon et al., 1999] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.*, 96(12):6745–6750. 5.6.2

[Arivazhagan and Ganesan, 2003] Arivazhagan, S. and Ganesan, L. (2003). Texture segmentation using wavelet transform. *Pattern Recognition Letters*, 24(16):3197–3203. 2.1, 4.1

[Azencott et al., 1997] Azencott, R., Wang, J.-P., and Younes, L. (1997). Texture classification using windowed Fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):148–153. 2.1

[Ben-Hur et al., 2001] Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2001). A support vector method for clustering. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 367–373. MIT Press, Cambridge, MA, USA. A.1

[Ben-Tal and Zibulevsky, 1997] Ben-Tal, A. and Zibulevsky, M. (1997). Penalty/barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization*, 7(2):347–366. 5.5.2

[Bennett and Mangasarian, 1992] Bennett, K. P. and Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34. 5.2.1

[Bernard, 1999] Bernard, C. P. (1999). Discrete wavelet analysis for fast optic flow computation. Technical Report RI415, Centre de Mathématiques Appliquées, École Polytechnique. 3.3

[Blake and Merz, 1998] Blake, C. L. and Merz, C. J. (1998). UCI repository of machine learning databases. 5.2.3, 5.6.2

[Blatter, 2003] Blatter, C. (2003). *Wavelets — Eine Einführung*. Vieweg, second edition. In german. 3.6

[Boggs and Tolle, 1995] Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. In Iserles, A., editor, *Acta Numerica*, pages 1–51. Cambridge University Press, Cambridge, MA, USA. 4.5.1

[Bousquet and Elisseeff, 2001] Bousquet, O. and Elisseeff, A. (2001). Algorithmic stability and generalization performance. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 196–202. MIT Press, Cambridge, MA, USA. 4.2

[Bradley, 1998] Bradley, P. S. (1998). *Mathematical Programming Approaches to Machine Learning and Data Mining*. PhD thesis, Computer Sciences Dept., University of Wisconsin. TR-98-11. 5.1

[Bradley and Mangasarian, 1998] Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In Shavlik, J., editor, *Proc. of the 15th International Conference on Machine Learning*, pages 82–90, San Francisco, CA, USA. Morgan Kaufmann. 1, 5.1, 5.1, 5.2.2, 5.2.2, 5.2.2, 5.2.2, 5.2.3, 5.4, 5.6.2, 5.6.2

[Brodatz, 1966] Brodatz, P. (1966). *Textures: A photographic album for artists and designers*. Dover Publications, New York, NY, USA. 3.5.2

[Burger, 2003] Burger, M. (fall 2003). Infinite-dimensional optimization and optimal design. UCLA lecture notes 285J. B, B.1, B.4

[Butzmann, 0203] Butzmann, H.-P. (winter term 2002/03). Nichtlineare Optimierung. University of Mannheim, lecture notes. In german. B, B.1, B.1

[Chapelle et al., 1999] Chapelle, O., Haffner, P., and Vapnik, V. (1999). SVMs for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064. 5.6.3, 5.6.3

[Chapelle et al., 2002] Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159. 4.1, 4.2, 5.4

[Chung et al., 2003] Chung, K.-M., Kao, W.-C., Sun, C. L., Wang, L.-L., and Lin, C.-J. (2003). Radius margin bounds for support vector machines with the RBF kernel. *Neural Computation*, 15(11):2643–2681. 4.1

[Coifman and Donoho, 1995] Coifman, R. R. and Donoho, D. L. (1995). Translation–invariant de–noising. In Antoniadis, A. and Oppenheim, G., editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer, New York, NY, USA. 3.1

[Coifman and Wickerhauser, 1992] Coifman, R. R. and Wickerhauser, M. V. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 32:712–718. 1

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297. 2.1

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, MA, USA. 2.3.1, A.1

[Cristianini et al., 2002] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. (2002). On kernel-target alignment. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, Cambridge, MA, USA. 4.1, 4.2, 4.2, 5.4

[Cvetković and Vetterli, 1998] Cvetković, Z. and Vetterli, M. (1998). Oversampled filter banks. *IEEE Transactions on Signal Processing*, 46(5):1245–1255. 3.1

[Daubechies, 1988] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996. 4.3.1, 4.5.2

[Daubechies, 1992] Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. 2.2.1, 3.20

[Daubechies et al., 2004] Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457. 2.2.4, 5.2.2

[Daubechies and Sweldens, 1998] Daubechies, I. and Sweldens, W. (1998). Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):245–267. 2.2.1

[de Rivaz and Kingsbury, 1999] de Rivaz, P. F. C. and Kingsbury, N. G. (1999). Complex wavelet features for fast texture image retrieval. In *Proc. of the 1999 International Conference on Image Processing*, volume 1, pages 109–113. 3.1, 3.10

[Devijver and Kittler, 1982] Devijver, P. A. and Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Prentice Hall, Englewood Cliffs, NJ, USA. 4.2

[Duan et al., 2001] Duan, K., Keerthi, S. S., and Poo, A. N. (2001). Evaluation of simple performance measures for tuning SVM hyperparameters. Technical Report CD-01-11, Dept. of Mechanical Engineering, National University of Singapore. 4.1, 4.2

[Duda et al., 2000] Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*. John Wiley & Sons, New York, NY, USA, second edition. 4.2, 4.3.4, 5.1

[Dunn et al., 1994] Dunn, D., Higgins, W. E., and Wakeley, J. (1994). Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149. 2.1

[Ekeland and Temam, 1976] Ekeland, I. and Temam, R. (1976). *Convex Analysis and Variational Problems*. North-Holland Elsevier, Amsterdam, NL. B

[Fernandes et al., 2003] Fernandes, F. C. A., Selesnick, I. W., van Spaendonck, R. L. C., and Burrus, C. S. (2003). Complex wavelet transforms with allpass filters. *Signal Processing*, 83(8):1689–1706. 3.2

[Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188. 4.2

[Fletcher, 1987] Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, New York, NY, USA, second edition. 4.5.1, 4.5.2

[Floyd and Warmuth, 1995] Floyd, S. and Warmuth, M. K. (1995). Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304. 4.2

[Forster et al., 2003] Forster, B., Blu, T., and Unser, M. (2003). A new family of complex rotation-covariant multiresolution bases in 2D. In Aldroubi, A., Laine, A. F., and Unser, M., editors, *Wavelet Applications in Signal and Image Processing X*, Proc. of the SPIE Conference on Mathematical Imaging. 3.7

[Frigo and Johnson, 1998] Frigo, M. and Johnson, S. G. (1998). FFTW: An adaptive software architecture for the FFT. In *Proc. of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1381–1384. IEEE Computer Society. http://fftw.org. 3.7

[Girosi, 1998] Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480. 2.3.3

[Goldberger et al., 2000] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. 4.3

[Gottscheber and Steidl, 1999] Gottscheber, A. and Steidl, G. (1999). On a family of orthogonal wavelets on the quincunx grid. In Haußmann, W., Jetter, K., and Reimer,

M., editors, *Advances in Multivariate Approximation*, pages 175–184. Wiley-VCH, Berlin, D. 3.6

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182. 5.1

[Haasdonk and Bahlmann, 2004] Haasdonk, B. and Bahlmann, C. (2004). Learning with distance substitution kernels. In Rasmussen, C. E., Bülthoff, H. H., Giese, M. A., and Schölkopf, B., editors, *Pattern Recognition, Proc. of 26th DAGM Symposium*, volume 3175 of *LNCS*, pages 220–227. Springer. 2.3.2, 5.6.3

[Hackbusch, 1993] Hackbusch, W. (1993). *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, Stuttgart, D, second edition. In german. 4.2

[Hausdorff et al., 1999] Hausdorff, J. M., Zemany, L., Peng, C.-K., and Goldberger, A. L. (1999). Maturation of gait dynamics: stride-to-stride variability and its temporal organization in children. *Journal of Applied Physiology*, 86(3):1040–1047. 4.3

[Hegland, 2003] Hegland, M. (2003). Sparse grids and the analysis of high dimensional large scale data. Available at `http://datamining.anu.edu.au/talks.html`. Talk at International Conference on Imaging Science and Information Processing, Singapore. 5.1

[Heiler, 2001] Heiler, M. (2001). Optimization criteria and learning algorithms for large margin classifiers. Master's thesis, Dept. of Mathematics and Computer Science, University of Mannheim. 2.3.4

[Heiler et al., 2001] Heiler, M., Cremers, D., and Schnörr, C. (2001). Efficient feature subset selection for support vector machines. Technical Report TR-01-021, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim. 5.1, 5.4, 5.6.3

[Herbrich, 2002] Herbrich, R. (2002). *Learning kernel classifiers: theory and algorithms*. MIT Press, Cambridge, MA, USA. 4.2

[Herbrich and Graepel, 2001] Herbrich, R. and Graepel, T. (2001). A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information System Processing 13*, pages 224–230. MIT Press, Cambridge, MA, USA. 4.2

[Hermes and Buhmann, 2000] Hermes, L. and Buhmann, J. M. (2000). Feature selection for support vector machines. In *Proc. of the 15th International Conference on Pattern Recognition (ICPR'00)*, volume 2, pages 716–719. 5.1, 5.4

[Hsu and Lin, 2002] Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425. 2.3.4

[Ilog, Inc., 2001] Ilog, Inc. (2001). Ilog cplex 7.5. 5.2.3, 5.6.2

[Jakubik, 2003] Jakubik, O. J. (2003). Feature selection with concave minimization. Master's thesis, Dept. of Mathematics and Computer Science, University of Mannheim. 1, 5.2.3, 5.3, 5.4.1, 5.4.2, 5.6.2

[Joachims, 1999] Joachims, T. (1999). Making large–scale SVM learning practical. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods — Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, USA. 2.3.3, 4.2

[John et al., 1994] John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In Cohen, W. W. and Hirsh, H., editors, *Proc. of the 11th International Conference on Machine Learning*, pages 121–129, San Francisco, CA, USA. Morgan Kaufmann. 5.1

[Jones et al., 2001] Jones, E., Runkle, P., Dasgupta, N., Couchman, L., and Carin, L. (2001). Genetic algorithm wavelet design for signal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):890–895. 2.1, 2.2.1, 4.1

[Kimeldorf and Wahba, 1971] Kimeldorf, G. S. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95. 2.3.3

[Kingsbury, 1998] Kingsbury, N. G. (1998). The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters. In *Proc. of the IEEE Digital Signal Processing Workshop*. 3.1, 3.2, 3.3, 3.4, 3.10

[Kingsbury, 1999] Kingsbury, N. G. (1999). Image processing with complex wavelets. *Phil. Transactions Royal Society London A*, 357:2543–2560. 3.2, 3.4

[Kingsbury, 2001] Kingsbury, N. G. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3):234–253. 1, 3.1, 3.1, 3.2, 3.2, 3.3, 3.4, 3.5, 3.8.1, 3.8.2, 3.9

[Kingsbury and Magarey, 1997] Kingsbury, N. G. and Magarey, J. F. A. (1997). Wavelet transforms in image processing. In Procházka, A., Uhlir, J., and Sovka, P., editors, *Proc. First European Conference on Signal Analysis and Prediction*, pages 23–34. ICT Press. 3.1, 3.4, 3.10

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324. 5.1

[Lanckriet et al., 2002] Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L., and Jordan, M. I. (2002). Learning the kernel matrix with semi-definite programming. In Sammut, C. and Hoffmann, A. G., editors, *Proc. of the 19th International Conference on Machine Learning*, pages 323–330, San Francisco, CA, USA. Morgan Kaufmann. 4.2

[Li et al., 2003] Li, S., Kwok, J. T., Zhu, H., and Wang, Y. (2003). Texture classification using the support vector machines. *Pattern Recognition*, 36(12):2883–2893. 2.1

[Mallat, 1999] Mallat, S. (1999). *A Wavelet Tour of Signal Processing*. Academic Press, London, UK. 1, 2.2.4, 3.1, 3.6

[Mangasarian, 1997] Mangasarian, O. L. (1997). Minimum-support solutions of polyhedral concave programs. Technical Report TR-1997-05, Mathematical Programming, University of Wisconsin. 5.2.2, 5.2.2

[MathWorks, 2002] MathWorks (2002). *Optimization Toolbox User's Guide*. The MathWorks, Inc. 4.5.1, 4.5.2, 5.5.2, 5.6.2

[Moulin and Mihçak, 1998] Moulin, P. and Mihçak, M. K. (1998). Theory and design of signal–adapted FIR paraunitary filter banks. *IEEE Transactions on Signal Processing*, 46(4):920–929. 2.2.1

[Mrázek and Weickert, 2003] Mrázek, P. and Weickert, J. (2003). Rotationally invariant wavelet shrinkage. In Michaelis, B. and Krell, G., editors, *Pattern Recognition, Proc. of 25th DAGM Symposium*, volume 2781 of *LNCS*, pages 156–163. Springer. 6

[Musicant, 2000] Musicant, D. R. (2000). MATLAB/CPLEX MEX-files. Available at `http://www.mathcs.carleton.edu/faculty/dmusican/cplex/`. 5.6.2

[Neumann et al., 2002] Neumann, J., Schnörr, C., and Steidl, G. (2002). Feasible adaptation criteria for hybrid wavelet – large margin classifiers. Technical Report TR-02-015, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim. 3.9, 4.1

[Neumann et al., 2003a] Neumann, J., Schnörr, C., and Steidl, G. (2003a). Effectively finding the optimal wavelet for hybrid wavelet – large margin signal classification. Technical Report TR-03-005, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim. 4.1

[Neumann et al., 2003b] Neumann, J., Schnörr, C., and Steidl, G. (2003b). Feasible adaptation criteria for hybrid wavelet – large margin classifiers. In Petkov, N. and

Westenberg, M. A., editors, *Computer Analysis of Images and Patterns*, volume 2756 of *LNCS*, pages 588–595. Springer. 4.1

[Neumann et al., 2004] Neumann, J., Schnörr, C., and Steidl, G. (2004). SVM-based feature selection by direct objective minimisation. In Rasmussen, C. E., Bülthoff, H. H., Giese, M. A., and Schölkopf, B., editors, *Pattern Recognition, Proc. of 26th DAGM Symposium*, volume 3175 of *LNCS*, pages 212–219. Springer. 5.4

[Neumann et al., 2005a] Neumann, J., Schnörr, C., and Steidl, G. (2005a). Combined SVM-based feature selection and classification. *Machine Learning*. Accepted. 5.4

[Neumann et al., 2005b] Neumann, J., Schnörr, C., and Steidl, G. (2005b). Efficient wavelet adaptation for hybrid wavelet–large margin classifiers. *Pattern Recognition*. Accepted. 4.1

[Neumann and Steidl, 2003] Neumann, J. and Steidl, G. (2003). Dual–tree complex wavelet transform in the frequency domain and an application to signal classification. Technical Report TR-03-013, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim. 3.1

[Neumann and Steidl, 2005] Neumann, J. and Steidl, G. (2005). Dual–tree complex wavelet transform in the frequency domain and an application to signal classification. *International Journal of Wavelets, Multiresolution and Information Processing*. Accepted. 3.1

[Ojala et al., 2002] Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987. 2.1

[Oppenheim and Schafer, 1989] Oppenheim, A. V. and Schafer, R. W. (1989). *Discrete-Time Signal Processing*. Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, USA. 3.6

[Osher et al., 2003] Osher, S., Solé, A., and Vese, L. (2003). Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Modeling and Simulation*, 1(3):349–370. UCLA TR 02-57. 2.2.4

[Pham Dinh and Hoai An, 1998] Pham Dinh, T. and Hoai An, L. T. (1998). A d.c. optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505. 1, 5.5, 5.5, 5.5.1, 5.5.1, B.1, 5

[Plonka and Tasche, 1995] Plonka, G. and Tasche, M. (1995). On the computation of periodic spline wavelets. *Journal of Applied and Computational Harmonic Analysis*, 2(1):1–14. 3.7

[Portilla and Simoncelli, 2000] Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70. 2.1, 3.5.2, 4.1

[Randen and Husøy, 1999] Randen, T. and Husøy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310. 2.1, 3.5.2, 4.1

[Reed and du Buf, 1993] Reed, T. R. and du Buf, J. H. (1993). A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 57:359–372. 2.1

[Rockafellar, 1970] Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University press, Princeton, NJ, USA. B, B.1, 18, B.3, B.3

[Roussos, 1995] Roussos, G. (1995). Calculation and evaluation of radial basis function interpolants. Master's thesis, University of Manchester Institute of Science and Technology. 2.3.1

[Saito, 1994] Saito, N. (1994). *Local feature extraction and its application using a library of bases*. PhD thesis, Dept. of Mathematics, Yale University. 4.1

[Sathidevi and Venkataramani, 2002] Sathidevi, P. S. and Venkataramani, Y. (2002). Perceptual audio coding using sinusoidal/optimum wavelet representation. *Circuits, Systems and Signal Processing*, 21(5):508–521. 4.1

[Schaback, 1995] Schaback, R. (1995). Creating surfaces from scattered data using radial basis functions. In Daehlen, M., Lyche, T., and Schumaker, L. L., editors, *Mathematical Methods for Curves and Surfaces*, pages 477–496. Vanderbilt University Press, Nashville, TN, USA. 2.3.2

[Scheunders et al., 1998] Scheunders, P., Livens, S., Van de Wouwer, G., Vautrot, P., and Van Dyck, D. (1998). Wavelet-based texture analysis. *International Journal on Computer Science and Information Management*, 1(2):22–34. 2.1, 4.1

[Schmidt, 2004] Schmidt, S. (2004). Context-sensitive image labeling based on logistic regression. Master's thesis, Dept. of Mathematics and Computer Science, University of Mannheim. 1, 5.6.3

[Schölkopf, 1997] Schölkopf, B. (1997). *Support Vector Learning*. PhD thesis, Technische Universität Berlin. 2.1

[Schölkopf et al., 1999a] Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (1999a). Estimating the support of a high-dimensional distribution. Technical Report 99-87, Microsoft Research. Short version appeared in *Neural Computation*, 2001. A.2

[Schölkopf et al., 1999b] Schölkopf, B., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (1999b). Kernel-dependent support vector error bounds. In Willshaw, D. and Murray, A., editors, *Proc. of the Ninth International Conference on Artificial Neural Networks*, volume 470 of *Conference Publications*, pages 103–108, London, UK. IEE. 4.1, 4.2

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA, USA. 2.3.3, 2.3.3

[Schölkopf et al., 1998] Schölkopf, B., Smola, A. J., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319. 4.3.3

[Schölkopf et al., 1997] Schölkopf, B., Sung, K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765. 2.3.1

[Schölkopf et al., 2000] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. (2000). Support vector method for novelty detection. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pages 582–588. MIT Press, Cambridge, MA, USA. A.1, A.2, A.2

[Schüle et al., 2003] Schüle, T., Schnörr, C., Weber, S., and Hornegger, J. (2003). Discrete tomography by convex-concave regularization and d.c. programming. Technical Report TR-03-015, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim. Submitted to *Discrete Applied Mathematics*. 5.5.1

[Selesnick, 2001] Selesnick, I. W. (2001). Hilbert transform pairs of wavelet bases. *IEEE Signal Processing Letters*, 8(6):170–173. 1, 3.2, 3.3

[Shashua and Wolf, 2004] Shashua, A. and Wolf, L. (2004). Kernel feature selection with side data using a spectral approach. In Pajdla, T. and Matas, J., editors, *Proc. of the European Conference on Computer Vision (ECCV), Part III*, volume 3023 of *LNCS*, pages 39–53. Springer. 2.3.3, 5.1

[Simoncelli et al., 1992] Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. (1992). Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607. MIT Media Laboratory Vision and Modeling Technical Report 161. 3.1, 3.8.1

[Smith, 1997] Smith, G. (1997). MeasTex image texture database and test suite. Available at `http://www.cssip.uq.edu.au/meastex/meastex.html`. Version 1.1. 3.5.1, 4.3, 4.3.2, 5.4

[Spellucci, 1993] Spellucci, P. (1993). *Numerische Verfahren der nichtlinearen Optimierung*, chapter 3.6: Die Methode der sequentiellen quadratischen Minimierung, pages 455–527. Birkhäuser Verlag, Basel, CH. In german. 4.5.1

[Steel and Hechter, 2004] Steel, S. J. and Hechter, G. K. (2004). Application of support vector machines in a life assurance environment. In *Classification: the ubiquitous challenge, Proc. 28th Annual GfKl Conference*. Springer. To appear. 5.1, 5.4.2

[Steidl et al., 2005] Steidl, G., Didas, S., and Neumann, J. (2005). Relations between higher order TV regularization and support vector regression. In *Scale-Space*. To appear. 2.3.3

[Steinwart, 2001] Steinwart, I. (2001). On the influence of the kernel on the generalization ability of support vector machines. Technical Report 01-01, FSU Jena. 2.3.3

[Strang and Nguyen, 1996] Strang, G. and Nguyen, T. (1996). *Wavelets and Filter Banks*. Wellesley–Cambridge Press, Wellesley, MA, USA. 1, 2.2.1, 2.2.1, 2.2.1, 2.2.1, 3.7

[Strauss et al., 1999] Strauss, D. J., Sinnwell, T., Rieder, A., Manoli, Y., and Jung, J. (1999). A promising approach to morphological endocardial signal discriminations: Adapted multiresolution signal decompositions. *Applied Signal Processing*, 6:182–193. 4.5.2

[Strauss and Steidl, 2002] Strauss, D. J. and Steidl, G. (2002). Hybrid wavelet-support vector classification of waveforms. *Journal of Computational and Applied Mathematics*, 148:375–400. 1, 1, 2.2.4, 3.9, 4.1, 4.2, 4.3, 4.3.1, 4.3.4

[Strauss et al., 2003] Strauss, D. J., Steidl, G., and Delb, W. (2003). Feature extraction by shape-adapted local discriminant bases. *Signal Processing*, 83(2):359–376. 1, 4.1

[Suykens et al., 2002] Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). *Least Squares Support Vector Machines*. World Scientific Pub. Co., Singapore, SGP. 2.3.3, 5.4

[Sweldens, 1998] Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546. 2.2.1

[Theodoridis and Koutroumbas, 1999] Theodoridis, S. and Koutroumbas, K. (1999). *Pattern Recognition*. Academic Press, London, UK. 4.2

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288. 5.2.2

[Toland, 1979] Toland, J. F. (1979). On subdifferential calculus and duality in nonconvex optimization. *Bull. Soc. math. France*, Mémoire 60:177–183. B.4

[Unser, 1995] Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560. 2.1, 2.2.4, 4.1

[Vaidyanathan, 1993] Vaidyanathan, P. P. (1993). *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, USA. 2.2.1, 2.2.1, 2.2.2

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA. 2.1, 2.3.1, 4.1, 4.2

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons, New York, NY, USA. 1, 2.1, 2.3.1, 2.3.3, 4.2, 4.2, A.2

[Vetterli and Kovačević, 1995] Vetterli, M. and Kovačević, J. (1995). *Wavelets and Subband Coding*. Signal Processing. Prentice Hall, Englewood Cliffs, NJ, USA. 2.2.1, 2.2.2

[Wahba, 1999] Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods — Support Vector Learning*, chapter 6, pages 69–88. MIT Press, Cambridge, MA, USA. 2.3.3

[Weston et al., 2003] Weston, J., Elisseeff, A., Schölkopf, B., and Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461. 5.2.2, 5.2.2, 5.2.3, 5.4.1, 5.6.2

[Weston et al., 2001] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001). Feature selection for SVMs. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, Cambridge, MA, USA. 4.1, 5.1, 5.1, 5.4, 5.4.2

[Weston and Watkins, 1998] Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Dept. of Computer Science, Royal Holloway, University of London. 2.3.4

[Weston and Watkins, 1999] Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In Verleysen, M., editor, *Proc. of the Seventh European Symposium On Artificial Neural Networks*, pages 219–224, Brussels, B. D-Facto. 2.3.4

[Zhu et al., 2004] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2004). 1-norm support vector machines. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA. 5.1, 5.2.2, 5.4, 5.4.2, 5.6.1

Index