Effectively Finding the Optimal Wavelet for Hybrid Wavelet - Large Margin Signal Classification

Julia Neumann Christoph Schnörr
Gabriele Steidl

Dept. of Mathematics and Computer Science
University of Mannheim
D-68131 Mannheim, Germany
{jneumann,schnoerr,steidl}@uni-mannheim.de

March 27, 2003

Abstract

For hybrid wavelet – large margin classifiers, adapting the wavelet may significantly improve the classification performance. We propose to select the wavelet with respect to a large margin classifier and data to improve class separability and minimise the generalisation error.

In this paper, we show that this wavelet adaptation problem can be formulated as an optimisation problem with polynomial objective function and investigate some techniques to solve it. In particular, we propose an adaptive grid search algorithm that efficiently solves the problem compared with standard optimisation techniques.

1 Introduction

Many approaches in signal and image classification rely on filtering for feature extraction [12, 13, 19]. But in most cases, the filter design problem is addressed irrespective of the target data and the subsequent classification stage. However, using 'off–the–shelf' filters like Daubechies' wavelets [4] may result in an unacceptably large classification error. A wavelet adaptation to the problem at hand may significantly improve signal classification as shown in [18]. Given a sample set of labelled patterns, the main idea is to adapt the filter bank based on a criterion measuring class separability and generalisation error to obtain the optimal features for the particular problem under consideration.

The target classifier in our hybrid approach is the Support Vector Machine (SVM) which is well known to belong to the most competitive approaches and has favourable properties from the perspective of optimisation during the learning stage [21]. As the radius – margin generalisation error bound whose direct application to feature selection has been studied in [24] is computationally very demanding and the filter design is a difficult optimisation problem, we have already proposed simple criteria that well approximate the generalisation error if the final classifier is a SVM in [11].

In order to be able to effectively find the optimal wavelet for signal classification, we would like to perform continuous optimisation. So we first study the dependency of the features and these simple criteria from [11] on the filter coefficients or the filter angles resulting of the lattice factorisation of orthonormal filter banks. We show that the filter design problem may be formulated as a polynomial in the filter coefficients. The nature of the objective function points out the structure of the parameter space.

To examine how the optimisation can be efficiently performed, we investigate several standard techniques for constrained optimisation of the filter coefficients and unconstrained optimisation of the lattice angles including Sequential Quadratic Programming, a simplex search method and a restricted step Newton method. As these techniques require too many objective function evaluations, we finally propose a robust grid search algorithm to solve the filter design problem. We derive the optimisation

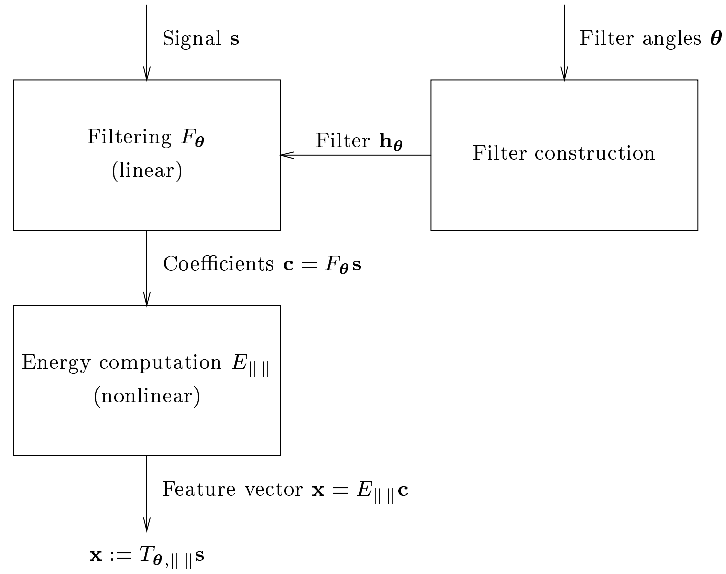

Figure 1: feature extraction: from the signal to the feature vector

problem and show that our algorithm works well for different examples of one–dimensional signals. Furthermore, the results also apply in two dimensions.

Organisation of the paper. We present the feature extraction process, especially the wavelet parametrisation in Sec. 2. Sec. 3 introduces the SVM classifier to be applied. Then we come to the main part. Sec. 4 derives the optimisation problem under consideration and Sec. 5 investigates several methods for the solution of this problem. The results of the paper are summarised in Sec. 6. Background material is given in the appendix. Appendix A gives some mathematical details about filter banks and discrete wavelets.

Notation. Throughout the paper, we denote vectors and matrices by bold face lower and upper case letters, respectively. The matrix $\mathbf{I}$ denotes the identity matrix in appropriate dimensions. The vector $\mathbf{0}$ signifies a vector of zeros in the respective space, and $\mathbf{e}$ a vector of ones. All vectors will be column vectors unless transposed by the superior symbol ^{T}. If $\mathbf{x} \in \mathbb{R}^n$ denotes a vector, we will indicate its components by x_i $(i = 1, \ldots, n)$. We assume vector inequalities coming up in optimisation problems to hold componentwise. Further, ℓ_2 denotes the Hilbert space of real valued quadratic summable sequences $\mathbf{a} = (a_i)_{i\in\mathbb{N}}$ with inner product $\langle \mathbf{a}, \mathbf{b}\rangle_{\ell_2} = \sum_{i\in\mathbb{N}} a_i b_i$ and corresponding norm $\|\mathbf{a}\|_{\ell_2} = \left(\sum_{i\in\mathbb{N}} a_i^2\right)^{1/2}$.

2 Feature Extraction by Discrete Wavelet Transform

This section describes the feature extraction process for our application of signal classification. We briefly summarise the generation of 'feature vectors' used in this paper, and their parametrisation. This provides the basis for the wavelet adaptation and initiates the optimisation problem we will examine here.

Fig. 1 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 l is a power of two and $d \ll l$. The feature extraction process consists of two successive steps, namely filtering and energy computation of the bandpass coefficients. For the filtering we use orthonormal filter banks. As illustrated on the right hand side of the diagram, these filters can be determined by some filter angles $\boldsymbol{\theta}$ which will be the main parameters of our feature extraction process. Therefore the filtering 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 called $E_{\| \ \|}$. In summary, the feature extraction operator is given by $T_{\boldsymbol{\theta}, \| \ \|} := E_{\| \ \|} F_{\boldsymbol{\theta}}$. The single steps will be more closely looked at in the following.

For filtering, we apply the concept of filter banks. For a short introduction see Appendix A.1. Fundamental for the parametrisation of our feature extraction process is the representation of orthonormal filter banks in a lattice structure composed of rotations and delays. According to [20, Theorem 14.3.1], [16, Theorem 4.7], a two–channel FIR filter bank with filter length $2L+2$ is orthogonal (paraunitary) if and only if, up to translation and the sign of the high–pass filter, the corresponding polyphase matrix $\mathbf{H}_{\mathrm{pol}}(z)$ can be decomposed into

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

where $\theta_L \in [0, 2\pi)$ and $\theta_l \in [0, \pi)$ $l = 0, \ldots, L-1$. If the filter bank's high–pass filter has at least one vanishing moment, the filter bank angles add up to $\frac{\pi}{4}$ (see also [16, Theorem 4.6]):

$$\sum_{l=0}^{L} \theta_l = \frac{\pi}{4} \pmod{2\pi} . \tag{2}$$

The resulting parameter space $\{\boldsymbol{\theta} = (\theta_0, \ldots, \theta_{L-1}) : \theta_l \in [0, \pi) \ l = 0, \ldots, L-1\}$ is π–periodic as the angles θ_i can be interpreted as rotation angles: A rotation of $\theta_i + \pi$ implies half a rotation extra because the result just alters sign. As the last angle is computed as $\theta_L = \frac{\pi}{4} - \sum_{i=0}^{L-1} \theta_i$, it is that amount smaller. Hence, the final filter output is just the same as before.

For a given $\boldsymbol{\theta}$ in the parameter space, the synthesis filters are determined by the polyphase matrix. Moreover these filters provide us with an orthonormal basis

$$\{\tilde{\mathbf{v}}^{\mathbf{d}}_{\mathbf{k}} : k = 1, \ldots, \frac{l}{2^d}\} \cup \{\tilde{\mathbf{w}}^{\mathbf{j}}_{\mathbf{k}} : j = 1, \ldots, d; k = 1, \ldots, \frac{l}{2^j}\}$$

of $\mathbb{R}^l$ consisting of the periodic discrete–time scaling sequences $\tilde{\mathbf{v}}^{\mathbf{d}}_{\mathbf{k}}$ and discrete–time wavelets $\tilde{\mathbf{w}}^{\mathbf{j}}_{\mathbf{k}}$ (see Appendix A.2). The representation of the signal $\mathbf{s} \in \mathbb{R}^l$ with respect to this basis is given by

$$\mathbf{s} = \sum_{k=1}^{l/2^d} c^d_k \tilde{\mathbf{v}}^{\mathbf{d}}_{\mathbf{k}} + \sum_{j=1}^{d} \sum_{k=1}^{l/2^j} d^j_k \tilde{\mathbf{w}}^{\mathbf{j}}_{\mathbf{k}}$$

with the wavelet coefficients

$$\begin{aligned} \mathbf{c^d} &:= (c^d_k)_{k=1,\ldots,l/2^d} := (\langle \mathbf{s}, \tilde{\mathbf{v}}^{\mathbf{d}}_{\mathbf{k}} \rangle_{\ell_2})_{k=1,\ldots,l/2^d} , \\ \mathbf{d^j} &:= (d^j_k)_{k=1,\ldots,l/2^j} := (\langle \mathbf{s}, \tilde{\mathbf{w}}^{\mathbf{j}}_{\mathbf{k}} \rangle_{\ell_2})_{k=1,\ldots,l/2^j} , \ j = 1, \ldots, d . \end{aligned}$$

The filtering operator $F_{\boldsymbol{\theta}}$ then performs a wavelet analysis by an octave band filter bank:

$$F_{\boldsymbol{\theta}} : \mathbb{R}^l \to \mathbb{R}^l , \ \mathbf{s} \mapsto \begin{pmatrix} \mathbf{c^d} \\ \mathbf{d^d} \\ \vdots \\ \mathbf{d^1} \end{pmatrix} = \begin{pmatrix} (\langle \mathbf{s}, \tilde{\mathbf{v}}^{\mathbf{d}}_{\mathbf{k}} \rangle_{\ell_2})_{k=1,\ldots,l/2^d} \\ (\langle \mathbf{s}, \tilde{\mathbf{w}}^{\mathbf{d}}_{\mathbf{k}} \rangle_{\ell_2})_{k=1,\ldots,l/2^d} \\ \vdots \\ (\langle \mathbf{s}, \tilde{\mathbf{w}}^{\mathbf{1}}_{\mathbf{k}} \rangle_{\ell_2})_{k=1,\ldots,l/2} \end{pmatrix} . \tag{3}$$

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

$$E_{\| \ \|} : \mathbb{R}^l \to \mathbb{R}^{d+1} , \ \begin{pmatrix} \mathbf{c^d} \\ \mathbf{d^d} \\ \vdots \\ \mathbf{d^1} \end{pmatrix} \mapsto \begin{pmatrix} \|\mathbf{c^d}\| \\ \|\mathbf{d^d}\| \\ \vdots \\ \|\mathbf{d^1}\| \end{pmatrix} , \tag{4}$$

where we restrict our attention to two significant norms $\|\ \|$, namely the ℓ_2–norm and the weighted ℓ_2–norm

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

The weighted ℓ_2–norm seizes the average power or channel variance as proposed by Unser ([19]). 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 sized 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, and the expectation of the low–pass coefficients is a multiple of the average signal value by [11, Lemma 1]. Hence we deal with signals having average value zero, so that the above norm effectively represents the channel variance. For sub–sampled decomposition, this choice of the norm still provides a variance estimate. To set the average signal value is also important to restrict the influence of the low–pass channel. Otherwise, one can as well omit this channel in the energy operator $E_{\|\ \|}$. The powers of other Hölder norms $\|\ \|_{\ell_p}^p$ as proposed in [18] may be used as well in (4).

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

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

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

We have now defined our feature vectors. One could use standard separability measures for feature selection to rate the sets of feature vectors and choose the wavelets. But better criteria result when taking into account the classifier in use. Hence we will next describe the SVM.

3 Support Vector Machine Classification

In this section we give a short introduction to Support Vector Machine classification as it gives rise to the adaptation criteria we want to apply here as proposed in [11].

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$ containing the data 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

$$\mathcal{Z} := \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\} \tag{5}$$

of n associations we are interested in the construction of 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 $i = 1, \ldots, n$. Here

$$\operatorname{sgn}(f(\mathbf{x})) := \begin{cases} 1 & \text{if } f(\mathbf{x}) \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

SVM classification combines the simplicity of a linear learning machine with $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$ ($\mathbf{a} \in \mathbb{R}^d, b \in \mathbb{R}$) with the high generalisation ability only found in nonlinear classifiers. To this end, the so–called *feature map* $\boldsymbol{\phi} : \mathcal{X} \to \ell_2$ non–linearly mapping the input vectors into some generally higher–dimensional space is introduced. We will then search for f as a linear function in the feature vectors.

It is possible to state linear learning machines only in terms of inner products between the input vectors. As only inner products need to be evaluated and 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 a space of functions. A common kernel is the Gaussian

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \quad . \tag{6}$$

The mathematical details with respect to the feature map and reproducing kernel Hilbert spaces can be found in [2, Chapter 3], [23], [11, Appendix B.1].

Let us turn to our classification task. For a given training set (5) we intend to construct a function $f \in \mathcal{H}_K$ which minimises

$$C \sum_{i=1}^{n} (1 - y_i f(\mathbf{x}_i))_+ + \frac{1}{2} ||f||^2_{\mathcal{H}_K} \tag{7}$$

where

$$(\tau)_+ := \begin{cases} \tau & \text{if } \tau \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

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.

By the *Representer Theorem* [7, 23], as also derived in [11, Appendix B.2], the minimiser of (7) can be found by solving the following quadratic problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \; & -\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Y} \mathbf{K} \mathbf{Y} \boldsymbol{\alpha} + \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{e}, \end{aligned} \tag{8}$$

where $\mathbf{Y} := \mathrm{diag}(y_1, \ldots, y_n)$ and the kernel matrix $\mathbf{K} := (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ represents the inner products of the feature vectors.

The *support vectors* (SVs) are those training patterns $\mathbf{x}_i$ for which α_i does not vanish. Let I denote the index set of the support vectors $I := \{i \in \{1, \ldots, n\} : \alpha_i \neq 0\}$ then the function f has the representation

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

which only depends on the SVs.

4 The Wavelet Adaptation Problem

The goal in this paper is to systematically determine the optimal filter for the classification process. We have already described the feature vectors and their dependency on the filter angles in Sec. 2. Assume that for the subsequent classification, an SVM as described in Sec. 3 will be used. In [11], we have shown that simple criteria like the class centre distance and the alignment [3] well measure the discrimination ability of sets of feature vectors, at least if the Gaussian kernel (6) is used in the SVM. We therefore concentrate on maximising these criteria.

The class centre distance is given by

$$D := ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_{-1}|| \tag{9}$$

where $\boldsymbol{\mu}_i$ is the mean of class $i = \pm 1$, i.e., $\boldsymbol{\mu}_i := \frac{1}{|\{j : y_j = i\}|} \sum_{y_j = i} \mathbf{x}_j$. The alignment

$$\hat{A}(\mathbf{K}) := \frac{\langle \mathbf{K}, \mathbf{y}\mathbf{y}^T \rangle_F}{\sqrt{\langle \mathbf{y}\mathbf{y}^T, \mathbf{y}\mathbf{y}^T \rangle_F \langle \mathbf{K}, \mathbf{K} \rangle_F}} = \frac{\mathbf{y}^T \mathbf{K} \mathbf{y}}{n ||\mathbf{K}||_F} \tag{10}$$

measures the conformance of the kernel with the 'optimal kernel' $\mathbf{y}\mathbf{y}^T$.

We now want to set up a simple exemplary optimisation problem directly maximising one of these criteria subject to the filter coefficients or filter angles (as also proposed by [3, 8]). For that purpose, we first make the following assumptions:

- one–dimensional signals $\mathbf{s}_i = (s_{i0}, \ldots, s_{i(l-1)})^T \in \mathbb{R}^l$ $(i = 1, \ldots, n)$ 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 (1)),
- two decomposition steps are performed ($d = 2$),
- we also include the filter bank's low–pass channel $\mathbf{c^2}$ in (4),
- the ℓ_2–norm is used for energy computation in (4), only that we omit taking the square root,
- the objective criterion is the equivalent of the class–centre distance $(\frac{n}{2}D)^2$.

Taking the square of the ℓ_2-features in (4) does not affect the classification if the features all have the same magnitude, but makes the whole transformation differentiable. But the squaring may even invert the rating if one feature is dominant, e.g. the low–pass channel!

The following steps successively define the dependence of the objective $(\frac{n}{2}D)^2$ on the filter angle θ (see also Fig. 1):

1. The filter generation $\theta \mapsto (\mathbf{h_{0\theta}}, \mathbf{h_{1\theta}})$ yields for $L = 1$ in (1) with (2)

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

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 (3) means a convolution with the analysis filters. If $*$ denotes the convolution operator and $(2\downarrow)$ downsampling by 2, the first decomposition step reads

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

with $\mathbf{S^1} := (s^1_{i,j})_{i=0,\ldots,\frac{l}{2}-1; j=0,\ldots,3}, s^1_{i,j} := s_{(2i-j) \bmod l}$.
The second step results in

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

with $\mathbf{S^{2.k}} := (s^{2.k}_{i,j})_{i,j=0,\ldots,3}, s^{2.k}_{i,j} := s^1_{(2k-i) \bmod \frac{l}{2}, j} = s_{(4k-2i-j) \bmod l} \quad k = 0, \ldots, \frac{l}{4} - 1$.
In general, every decomposition step generates an additional power of $\mathbf{h_{0\theta}}$.

3. The energy operator $E_{\|\,\|^2_{\ell_2}}$ from (4) produces a feature vector $\mathbf{x} \in \mathbb{R}^3$ with

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

Note that, as the operand of the sum is no longer linear in the coefficient matrices, in general, there don't exist matrices $\mathbf{\Sigma_1}, \mathbf{\Sigma_2}$ such that, e.g., $x_1 = \mathbf{h_{0\theta}}^T\mathbf{\Sigma_1}\mathbf{h_{0\theta}}\mathbf{h_{0\theta}}^T\mathbf{\Sigma_2}\mathbf{h_{0\theta}}$ holds. The powers of $\mathbf{h_{0\theta}}$ have doubled through the energy computation. When performing d decomposition steps, the feature vectors thus depend on $\mathbf{h_{0\theta}}^{2d}$.

4. The criterion evaluation $\{\mathbf{x_i} : i = 1, \ldots, n\} \mapsto (\frac{n}{2}D)^2$ yields for equally frequent classes

$$\begin{aligned}
(\frac{n}{2}D)^2(\mathbf{h_{0\theta}}) &= \|\sum_{i=1}^{n} y_i\mathbf{x_i}\|^2 \\
&= \left\| \begin{pmatrix} \sum_{i=1}^{n} y_i \sum_{k=0}^{l/4-1} \left(\mathbf{h_{0\theta}}^T\mathbf{S_i^{2.k}}\mathbf{h_{0\theta}}\right)^2 \\ \sum_{i=1}^{n} y_i \sum_{k=0}^{l/4-1} \left(\mathbf{h_{0\theta}}^T\mathbf{D}^T\mathbf{S_i^{2.k}}\mathbf{h_{0\theta}}\right)^2 \\ \mathbf{h_{0\theta}}^T\mathbf{D}^T \left(\sum_{i=1}^{n} y_i \left(\mathbf{S_i^1}\right)^T \mathbf{S_i^1}\right) \mathbf{D}\mathbf{h_{0\theta}} \end{pmatrix} \right\|^2 \\
&= \left\| \begin{pmatrix} M_{\mathbf{X_1}}(\mathbf{h_{0\theta}}) \\ M_{\mathbf{X_2}}(\mathbf{h_{0\theta}}) \\ \mathbf{h_{0\theta}}^T\mathbf{X_3}\mathbf{h_{0\theta}} \end{pmatrix} \right\|^2 \\
&= (M_{\mathbf{X_1}}(\mathbf{h_{0\theta}}))^2 + (M_{\mathbf{X_2}}(\mathbf{h_{0\theta}}))^2 + \left(\mathbf{h_{0\theta}}^T\mathbf{X_3}\mathbf{h_{0\theta}}\right)^2
\end{aligned}$$

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

$$\mathbf{X_3} = \mathbf{D}^T \left(\sum_{i=1}^{n} y_i \left(\mathbf{S_i^1}\right)^T \mathbf{S_i^1}\right) \mathbf{D}$$

and $M_{\mathbf{X_i}}$ $i = 1, 2$ denotes the tensor–vector–multiplication

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

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) for $E_{\| \|}$ simply weights the features, and thus the final summands of the objective function, differently. 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 $E_{\| \|_F^2}$. Alternatively to the class centre distance, if 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 (10) 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 it is more generally applicable.

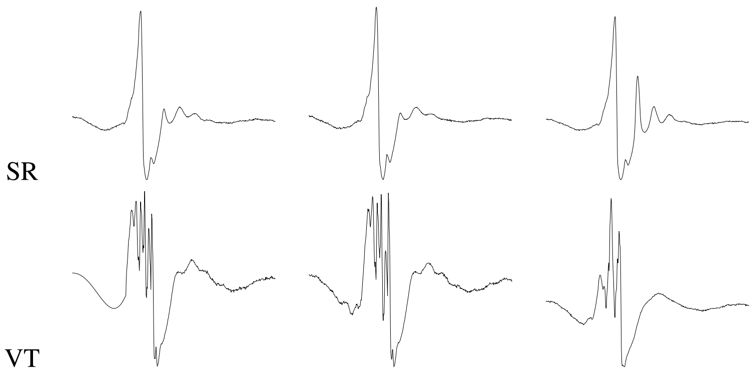

Figure 2: exemplary heart beats for 'heart1': sinus rhythm (SR) and ventricular tachycardia(VT)

The first step uses $\cos$ and $\sin$ 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 idea 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 non-linear, 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. We have either the unconstrained, π–periodic parameter space $[0, \pi)$ or the feasible set is defined by three partially non-linear constrains in $\mathbb{R}^4$, but with a polynomial target function.

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 different data sets: The first classification problem is the detection of ventricular tachycardia from electro–physiological data. 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 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, eight beats from a single episode are used. Some exemplary beats for a sample patient examined here are shown in Fig. 2. We use the data of two patients here and name the resulting problems 'heart1' and 'heart2'.

The second group of data are real–world texture images from the MeasTex collection [14]. We use single rows of the texture images to have one–dimensional data structurally different from the cardiac data. We 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. The corrugated iron images with normalised contrast as well as two exemplary rows are shown in Fig. 3. The task is to classify which of two given textures the rows belong to. Here, the first 32 rows of each texture are used for classifier training.

As argued before, to balance the features corresponding to channel energies, the original sample signals $\mathbf{s}_i$ $(i = 1, \ldots, n)$, 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.

class 1

class -1

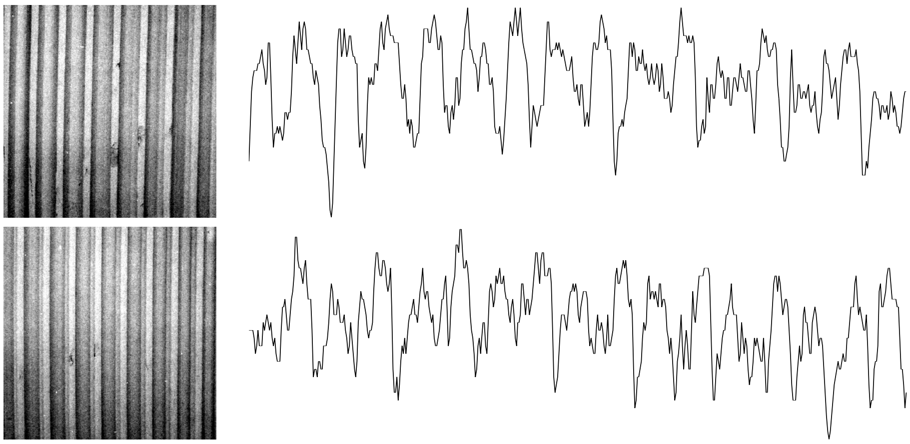

Figure 3: texture sample for 'm2m3': linearly rescaled images and exemplary rows

We indicated two approaches to solve the filter design problem. First, we will examine the profit of a polynomial objective function with its limited number of local maxima. Therefore, Sec. 5.1 tries to solve a constrained polynomial optimisation problem for filter design. We examine the unconstrained angle optimisation problem and indicate and test solution algorithms for it in Sec. 5.2. In any case, with these non–concave problems, only local optima can be found and there still remain suitable start values to be determined. As all suggested methods do not reliably find the problem's solution, we finally propose a heuristic search algorithm in Sec. 5.3.

5.1 Constrained Optimisation

A common state–of–the–art optimisation technique for nonlinearly constrained programs is Sequential Quadratic Programming (SQP) which is well described in [1] as well as in [15], [6, chapter 12.4]. The iterative approach approximates a nonlinear minimisation program locally by a quadratic program 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 a very efficient method in terms of convergence rate.

We used the SQP implementation in MATLAB's optimisation toolbox [10] accessible via the function `fmincon`. We normalised the input signals according to $\|\mathbf{s}_i\| = 1$ ($i = 1, \ldots, n$) so that the values of the objective function have approximately magnitude one and the default options can be applied.

But the SQP method is not recommendable for this problem. The iterates often did not converge. Expectedly, it appears to be practically too difficult to follow the nonlinear constraints. Experimental results with all four problems have shown 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 starting point any longer so no convergence at all is given or the returned solution is eventually even worse than the starting solution.

5.2 Unconstrained Optimisation

As the constrained optimisation for the filter design problem failed, 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. Since functions subject to this parameter space are easily plotted, we will advance our search for methods by first visualising the objective function in order to get an impression of the problem's structure in the following. Subsequently, in Section 5.2.2, we will give results obtained by using standard optimisation techniques for the problem.

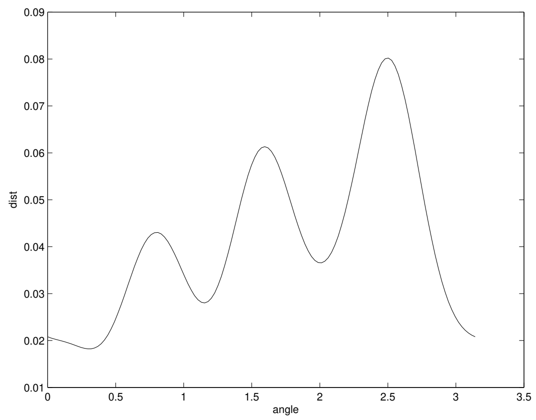

Figure 4: class centre distance for problem ’heart2’ with two decomposition steps

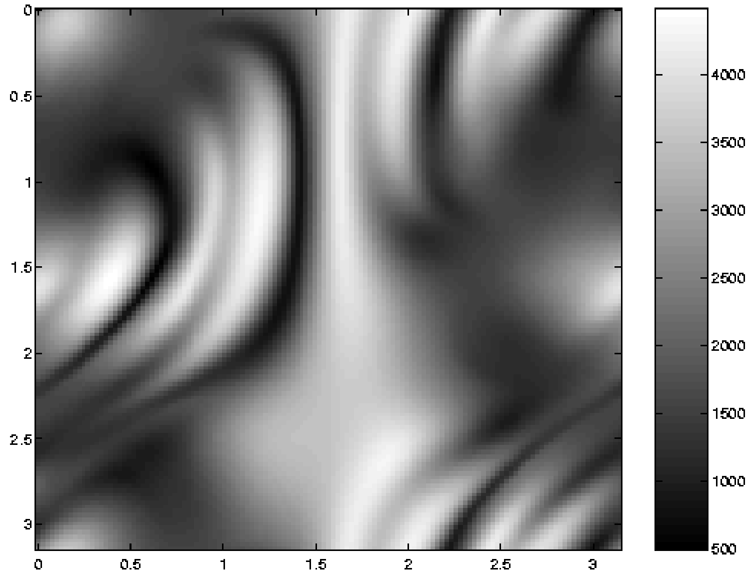

Figure 5: class centre distance for problem ’heart2’ with full decomposition

5.2.1 Problem Illustration

The simplified setting we have studied in Sec. 4 says 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 given in Fig. 4. The function is still smooth and possesses only three local optima due to the reduced setting. 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.

An example for a realistic problem is depicted in Fig. 5: The class centre distance for full decomposition of sample heart beats with $E_{\|\ \|_{\ell_2}}$, but without the low–pass component is shown. The parameter space was discretised with 128 angles per dimension. The resulting values are plotted using a linear grey scale. Light spots represent favourable filter angles. One can see that the class centre distance depending on the filter bank angles is indeed a smooth function, but has several local minima as expected.

5.2.2 Optimisation Results

We now want to try out some standard methods for maximising unconstrained functions as the ones shown in figures 4 and 5. Note that an unconstrained optimisation may also be performed. The bounds $\mathbf{0} \leq \boldsymbol{\theta} < \pi\mathbf{e}$ can be neglected as the parameter space is periodic resulting from the dependency on $\sin$ and $\cos$. But again we are only able to search for local optima.

We will test the methods with the help of MATLAB's optimisation toolbox [10]. The methods we compare here are

- 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 analytic Hessian is provided) and
- the SQP method from Sec. 5.1.

A description of all techniques beside the special gradient method can also be found in [6]. The optimum values given for comparison were 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*, when you are given an angle vector $\boldsymbol{\theta}^{(k)}$ for our problem, the next iterate $\boldsymbol{\theta}^{(k+1)}$ is given 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 $\alpha^{(k)}$ 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 $\alpha^{(k)}$ to a fix value or to define a decreasing sequence $(\alpha^{(k)})_{k\in\mathbb{N}}$ in advance. We implemented a bisection heuristic: Start with an initial guess for $\alpha^{(k)}$. If the objective value of the resulting angle vector is higher than that of $\boldsymbol{\theta}^{(k)}$, double $\alpha^{(k)}$ until the function value does not increase any longer. Otherwise, if the objective value is lower, halve $\alpha^{(k)}$ until the objective value is higher than that of $\boldsymbol{\theta}^{(k)}$. For $\alpha^{(k+1)}$, we use $\alpha^{(k)}$ as an initial guess.

Most of the one–dimensional functions coming from our simple filter design problem have two or three local maxima as, e.g. the one shown in Fig. 4, so we fix four start points for all methods apart from the Golden Section search: the Haar wavelet (with angle $\theta = 0$), the Daubechies wavelet with two vanishing moments ([4], $\theta = \frac{11}{12}\pi$), $\theta = 1$ and $\theta = 2$. For the evaluation, we used 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 considered optimisation problem, we can avail ourselves of the problem formulation in Sec. 4, especially for the simple analytic gradient evaluation. The results for the four discussed classification problems are summarised in Table 1. Confer the objective function plot for 'heart1' in Fig. 4. One can see that all methods find the optimum for all sample problems, but except for the Golden Section search method, the number of function evaluations is not substantially lower than for the complete search that would achieve the same accuracy with approximately 128 function evaluations. And additionally, the cost for calculating the coefficient tensors $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$ from Sec. 4 during the preparation step is dominating the cost for the optimisation process. Furthermore, more decomposition steps d lead to 2^{4d} or $2^{8d} = 4^{4d}$ coefficients for filters of length four or six, respectively, and it quickly becomes impossible even to store them. Consequently, we still know 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 with all signals.

For a more realistic setting with nine decomposition steps which means full decomposition, we apply the gradient descent with finite difference gradients as well as the other available methods listed above. The results are shown in Table 2. One can see that none of the methods was able

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

Table 1: optimisation results for the exemplary problem of Section 4: returned maximal value and number of function evaluations

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

Table 2: optimisation results for one angle and full decomposition: returned maximal value and number of function evaluations

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 considering the overhead for the optimisation methods besides 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 in (4), 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 3. 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 [17]. But still there remains the choice of the number of start values depending on the problem structure.

5.3 A Search Algorithm

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, another possibility is to develop an algorithm built up on grid search. The idea is to start with a grid and then to successively and adaptively refine those sections of the grid where the function behaves different from our expectation or where it exhibits favourable function values. One can then define tolerances no longer depending on the absolute function values and is also independent of possible start values.

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

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

Table 3: optimisation results for two angles and full decomposition: returned maximal value and number of function evaluations

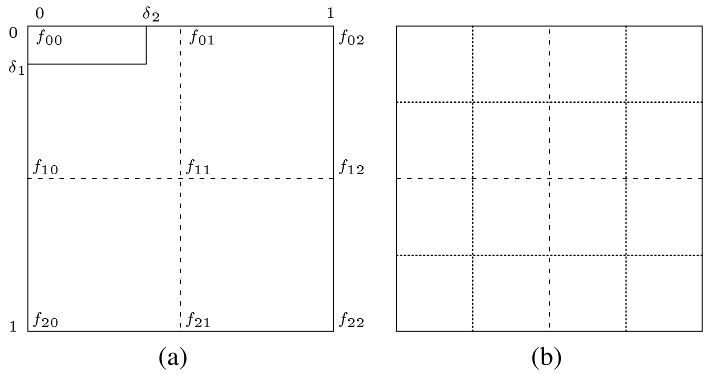

Figure 6: grid section: (a) examined function values on two different refinement levels, (b) further refined grid

5.3.1 Refinement Criterion and Algorithm

We investigated several approaches concerning the grid refinement and the 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. Fig. 6 (a) shows a sample 3×3 grid marked with already calculated function values from two levels defined by the solid and dashed lines. We use the even numbered grid points on the coarsest grid indicated by the solid lines as our interpolation points. The resulting bilinear function $\hat{f}$ for the shown grid section is

$$\hat{f}(\delta_1, \delta_2) = f_{00} + (f_{20} - f_{00})\delta_1 + (f_{02} - f_{00})\delta_2 + (f_{00} - f_{20} - f_{02} + f_{22})\delta_1\delta_2$$

for $\delta_1, \delta_2 \in [0, 1]$. The refinement condition compares the function values of the grid points including odd numbers $f_{01}, f_{10}, f_{11}, f_{12}, f_{21}$ to their estimates by $\hat{f}$. If we denote by $f_{\max}$ the maximal function value found up to now, the refinement criterion reads for f_{11}, e.g.,

$$\begin{aligned} \frac{f_{11} - \hat{f}(\frac{1}{2}, \frac{1}{2})}{f_{\max} - f_{11}} &> tolF \\ \Leftrightarrow f_{11} - (f_{00} + f_{20} + f_{02} + f_{22})/4 &> tolF(f_{\max} - f_{11}) \ . \end{aligned}$$

If this condition is fulfilled for at least one of $f_{01}, f_{10}, f_{11}, f_{12}, f_{21}$, the grid is refined to obtain the grid in Fig. 6 (b) with the dotted lines added. On the next level, the condition is analogously checked for the four resulting smaller sections. Effectively, the improvement $f - \hat{f}$ is a measure for the concavity of the function f - or convexity for minimisation problems -, which is a necessary condition for local maxima of twice differentiable functions. In the beginning of the algorithm's runtime, heavily concave sections with arbitrary function values will also satisfy the condition. In the end only concave sections which at the same time have high function values, i.e. possible maxima, will be refined.

In summary, we propose the algorithm listed in the following:

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

Table 4: grid search results for two angles and full decomposition: returned maximal value and number of function evaluations

Algorithm 5.1: GRIDSEARCH($f, tolF, tolX, maxGrid$)

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

5.3.2 Experiments and Performance

Fig. 7 gives an illustration of the performance of the algorithm. For the first heart patient and the setting used in the end of Sec. 5.2.2, i.e., full decomposition with two lattice angles followed by the weighted ℓ_2-norm in (4), the final grid used by the algorithm is demonstrated. For the parameters of Algorithm 5.1, we used values of $tolF = 4$, again $tolX = 0.01$ and $maxGrid = \frac{\pi}{16}$. 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. One can already see in the figure that the region where f is evaluated rapidly gets smaller with each finer grid. The results for all four problems are summarised in Table 4. Compared with Table 3, the results quite keep up with those 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 the simplex search algorithm. Additionally, it provides a robust algorithm that is not depending on experienced parameter tuning for each optimisation problem and can easily be implemented in an efficient setting with a performance oriented programming language due to its simplicity.

6 Conclusion

In this paper, we set up a precise optimisation problem minimising the generalisation error of a hybrid wavelet – large margin classifier subject to the filter coefficients or the filter's lattice angles. As a result, the filter design problem could be formulated as an optimisation problem with polynomial objective function in the filter coefficients under some assumptions concerning the feature extraction process. In spite of this realisation, there is no simple way to solve the problem. As illustrated, the optimisation problem is apparently too complex.

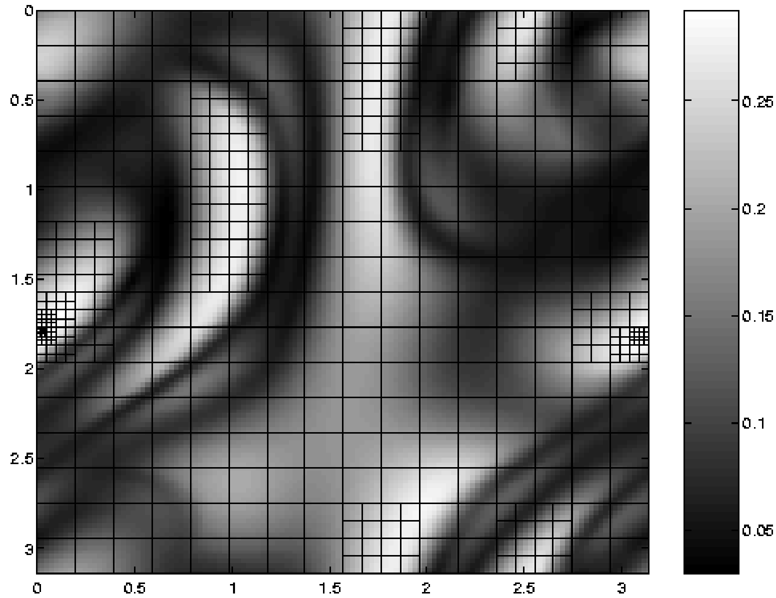

Figure 7: class centre distance for problem 'heart1' with final grid used by Algorithm 5.1

We proposed a simple heuristic grid search algorithm succeeding in solving the sample problems with only few function evaluations. We utilised the algorithm for maximising the class centre distance, but for other criteria it will presumably work alike as we have shown in [11] that many possible criteria have a similar structure on the parameter space.

Beside the specific algorithm suggested, the knowledge that the adaptation criterion is a polynomial of degree four times the maximal decomposition level may give rise to ideas for other adaptation techniques.

A Filter Banks and Discrete Orthonormal Wavelets

We will give some background about filter banks and discrete wavelets here and introduce our notation about them. The concept of 'filter banks' comes from engineering sciences and is widely used in all signal processing areas as pattern recognition. Work on wavelets was pioneered by S. Mallat ([9]) and I. Daubechies ([5]). The usual wavelet is a continuous function. However, we want to apply the term 'wavelet' in the discrete setting. The derivation of the analogies will be given in the following.

A.1 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. The essential information is extracted from the resulting subband signals of an analysis filter bank. Our notion of filter banks is mainly based upon the book [16]. We will only use two–channel filter banks whose analysis filters normally consist of a low–pass and a high–pass filter.

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 the signal decomposition, we are interested in the filter coefficient sequences $(h_0[k])_{k\in\mathbb{Z}}$, $(h_1[k])_{k\in\mathbb{Z}} \in \ell_2$.

A filter bank with analysis filters H_0 and H_1 is called *paraunitary* (also referred to as *orthogonal*) if

$$\begin{pmatrix} H_0(z^{-1}) & H_1(z^{-1}) \\ H_0(-z^{-1}) & H_1(-z^{-1}) \end{pmatrix} \begin{pmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{pmatrix} = \\ \begin{pmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{pmatrix} \begin{pmatrix} H_0(z^{-1}) & H_1(z^{-1}) \\ H_0(-z^{-1}) & H_1(-z^{-1}) \end{pmatrix} = 2\mathbf{I} \ . \tag{11}$$

The corresponding synthesis filters are then given by

$$G_0(z) = H_0(z^{-1}) \, , \; G_1(z) = H_1(z^{-1}) \ . \tag{12}$$

The *polyphase matrix* of a paraunitary 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

$$H_i(z) := H_{i0}(z^2) + z^{-1}H_{i1}(z^2) \quad i = 0, 1 \ . \tag{13}$$

To split the signals into different frequency bands, often high–pass filters with at least one vanishing moment are considered. To this end, the filter bank has to satisfy the low–pass condition

$$H_0(1) = \sqrt{2} \tag{14}$$

or, equivalently,

$$H_1(1) = 0 \ . \tag{15}$$

For our practical purposes, we are interested in finite impulse response (FIR) filters. The analysis filters of order $2L + 2$ then read

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

A.2 Discrete Orthonormal Wavelets

To justify the term 'wavelet decomposition' for our feature extraction process described in Sec. 2, 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 orthonormal filter banks covered by the parameter space here are related to continuous wavelets. To cope with this mismatch, in the style of the books [22, Sec. 3.3.2] and [20, Sec. 11.4], we want to 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 $(g_0[k])_{k\in\mathbb{Z}} \overset{(12)}{=} (h_0[-k])_{k\in\mathbb{Z}}$, $(g_1[k])_{k\in\mathbb{Z}} \overset{(12)}{=} (h_1[-k])_{k\in\mathbb{Z}} \in \ell_2$, we want to analyse the signal with the corresponding filter bank. With $\mathbf{g}_{jk} := (g_j[i-2k])_{i\in\mathbb{Z}} \in \ell_2 (j = 0, 1; k \in \mathbb{Z})$, the orthogonality conditions for the z–transforms (11) and (12) imply the orthogonality conditions

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

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

$$\mathbf{s} = \sum_{k\in\mathbb{Z}} c_k^1 \mathbf{g}_{0k} + \sum_{k\in\mathbb{Z}} d_k^1 \mathbf{g}_{1k} \tag{16}$$

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

$$S(z) = C^1(z^2) G_0(z) + D^1(z^2) G_1(z) \tag{17}$$

with

$$C^1(z^2) = \frac{1}{2}(H_0(z)S(z) + H_0(-z)S(-z))\ , \tag{18}$$

$$D^1(z^2) = \frac{1}{2}(H_1(z)S(z) + H_1(-z)S(-z))\ . \tag{19}$$

If we want to perform several decomposition steps, we refine the signal representations (16) or (17) further and obtain

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

with

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

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

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

with

$$\begin{aligned}C^2(z^4) &= \frac{1}{2}(H_0(z^2)C^1(z^2) + H_0(-z^2)C^1(-z^2)) \ , \\ D^2(z^4) &= \frac{1}{2}(H_1(z^2)C^1(z^2) + H_1(-z^2)C^1(-z^2)) \ .\end{aligned} \tag{20}$$

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

$$V^j(z) := \prod_{m=0}^{j-1} G_0(z^{2^m}) \ , \quad j \in \mathbb{N}_0 \tag{21}$$

$$W^j(z) := G_1(z^{2^{j-1}}) \prod_{m=0}^{j-2} G_0(z^{2^m}) \ , \quad j \in \mathbb{N} \tag{22}$$

as *discrete–time scaling sequences* and *discrete–time wavelets*, respectively. Let $\mathbf{v_k^j} := v_0^j[\cdot - 2^j k]$ and $\mathbf{w_k^j} := w_0^j[\cdot - 2^j k]$ ($k \in \mathbb{Z}$) denote the translates of the sequences by multiples of their sample length 2^j. The following orthogonality relations then hold for all $i, k \in \mathbb{Z}; j, m \in \mathbb{N}$:

$$\langle \mathbf{v_i^j}, \mathbf{v_k^j} \rangle_{\ell_2} = \delta(k-i) \ , \tag{23}$$

$$\langle \mathbf{w_i^j}, \mathbf{w_k^m} \rangle_{\ell_2} = \delta(k-i)\delta(m-j) \ , \tag{24}$$

$$\langle \mathbf{v_i^j}, \mathbf{w_k^j} \rangle_{\ell_2} = 0 \ . \tag{25}$$

As in the case of continuous wavelets, the sequences $\mathbf{v_k^j}$ and $\mathbf{w_k^j}$ ($j \in \mathbb{N}, k \in \mathbb{Z}$) 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 see this, we define the sequence of spaces

$$\begin{aligned}V^j &:= \overline{\mathrm{span}\{\mathbf{v_k^j} : k \in \mathbb{Z}\}} \subset \ell_2 \ , \ j \in \mathbb{N}_0 \ , \\ W^j &:= \overline{\mathrm{span}\{\mathbf{w_k^j} : k \in \mathbb{Z}\}} \subset \ell_2 \ , \ j \in \mathbb{N} \ .\end{aligned}$$

The multiresolution properties

$$V^0 \supset V^1 \supset V^2 \supset \dots \ ,$$

$$\bigcup_{j \in \mathbb{N}_0} V^j = V^0 = \ell_2 \ ,$$

$$\bigcap_{j \in \mathbb{N}_0} V^j = \{\mathbf{0}\}$$

then hold due to the definition of the scaling sequence filters (21). And due to (23), the set $\{\mathbf{v_k^j} : k \in \mathbb{Z}\}$ forms an orthonormal basis of V^j. Further, (22) and (25) imply that the detail spaces W^j form the orthogonal complement to the approximation spaces V^j in the next bigger spaces V^{j-1}

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

As a consequence, the whole space of sequences can be decomposed into $\ell_2 = \bigoplus_{j\in\mathbb{N}} W^j = V^J \oplus \bigoplus_{j=1}^{J} W^j$ 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 of 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 the *wavelet coefficients*

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

Given the finite analysis filters $(h_0[k])_{k=0,\ldots,2L+1}$, $(h_1[k])_{k=0,\ldots,2L+1}$, the decomposition of a finite signal $\mathbf{s}$ with length $l = n2^d$ ($n \in \mathbb{N}$) in $J = d$ 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

$$\begin{aligned} &\{\tilde{\mathbf{v}}_\mathbf{k}^\mathbf{J} = (\sum_{n\in\mathbb{N}} \mathbf{v_k^J}[\mathbf{i} + \mathbf{nl}])_{i=0,\ldots,l-1} : k = 1, \ldots, l/2^J\} \\ \cup\ &\{\tilde{\mathbf{w}}_\mathbf{k}^\mathbf{j} = (\sum_{n\in\mathbb{N}} \mathbf{w_k^j}[\mathbf{i} + \mathbf{nl}])_{i=0,\ldots,l-1} : j = 1, \ldots, J; k = 1, \ldots, l/2^j\} \ . \end{aligned}$$

References

[1] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. In *Acta Numerica*, pages 1–51. 1995.

[2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-base learning methods*. Cambridge University Press, 2000.

[3] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 367–373, Cambridge, MA, 2002. MIT Press.

[4] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, 1988.

[5] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[6] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, second edition, 1987.

[7] G. S. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applications*, 33(1):82–95, 1971.

[8] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of 19th International Conference on Machine Learning*, pages 323–330, 2002.

[9] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, London, 1999.

[10] The MathWorks, Inc. *Optimization Toolbox User's Guide*, July 2002.

[11] J. Neumann, C. Schnörr, and G. Steidl. Feasible adaptation criteria for hybrid wavelet – large margin classifiers. Technical Report TR-02-015, Dept. of Mathematics and Computer Science, University of Mannheim, 2002.

[12] T. Randen and J. H. Husøy. Filtering for texture classification: A comparative study. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.

[13] P. Scheunders, S. Livens, G. Van de Wouwer, P. Vautrot, and D. Van Dyck. Wavelet-based texture analysis. *International Journal on Computer Science and Information Management*, 1(2):22–34, 1998.

[14] G. Smith. MeasTex image texture database and test suite. Available at `http://www.cssip.uq.edu.au/meastex/meastex.html`, May 1997. Version 1.1.

[15] P. Spellucci. *Numerische Verfahren der nichtlinearen Optimierung*, chapter 3.6 : Die Methode der sequentiellen quadratischen Minimierung, pages 455–527. Birkhäuser Verlag, Basel, 1993.

[16] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, 1996.

[17] D. Strauß, T. Sinnwell, A. Rieder, Y. Manoli, and J. Jung. A promising approach to morphological endocardial signal discriminations: Adapted multiresolution signal decompositions. *Applied Signal Processing*, 6:182–193, 1999.

[18] D. Strauß and G. Steidl. Hybrid wavelet-support vector classification of waveforms. *Journal of Computational and Applied Mathematics*, 148:375–400, 2002.

[19] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

[20] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[21] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[22] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1995.

[23] G. Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, chapter 6, pages 69–88. MIT Press, Cambridge, MA, 1999.

[24] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.