
Unmixing Hyperspectral Data

Lucas Parra, Clay Spence, Paul Sajda
Sarnoff Corporation, CN-5300, Princeton, NJ 08543, USA
{lparra,cspence,psajda}@sarnoff.com

Andreas Ziehe, Klaus-Robert Müller
GMD FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
{ziehe,klaus}@first.gmd.de

Abstract

In hyperspectral imagery one pixel typically consists of a mixture of the reflectance spectra of several materials, where the mixture coefficients correspond to the abundances of the constituting materials. We assume linear combinations of reflectance spectra with some additive normal sensor noise and derive a probabilistic MAP framework for analyzing hyperspectral data. As the material reflectance characteristics are not known a priori, we face the problem of unsupervised linear unmixing. The incorporation of different prior information (e.g. positivity and normalization of the abundances) naturally leads to a family of interesting algorithms, for example in the noise-free case yielding an algorithm that can be understood as constrained independent component analysis (ICA). Simulations underline the usefulness of our theory.

1 Introduction

Current hyperspectral remote sensing technology can form images of ground surface reflectance at a few hundred wavelengths simultaneously, with wavelengths ranging from 0.4 to 2.5 μm and spatial resolutions of 10-30m. The applications of this technology include environmental monitoring and mineral exploration and mining. The benefit of hyperspectral imagery is that many different objects and terrain types can be characterized by their spectral signature.

The first step in most hyperspectral image analysis systems is to perform a spectral unmixing to determine the original spectral signals of some set of prime materials. The basic difficulty is that for a given image pixel the spectral reflectance patterns of the surface materials is in general not known a priori. However there are general physical and statistical priors which can be exploited to potentially improve spectral unmixing. In this paper we address the problem of unmixing hyperspectral imagery through incorporation of physical and statistical priors within an unsupervised Bayesian framework.

We begin by first presenting the linear superposition model for the reflectances measured. We then discuss the advantages of unsupervised over supervised systems.

We derive a general maximum a posteriori (MAP) framework to find the material spectra and infer the abundances. Interestingly, depending on how the priors are incorporated, the zero noise case yields (i) a simplex approach or (ii) a constrained ICA algorithm. Assuming non-zero noise our MAP estimate utilizes a constrained least squares algorithm. The two latter approaches are new algorithms whereas the simplex algorithm has been previously suggested for the analysis of hyperspectral data.

Linear Modeling To a first approximation the intensities \mathbf{X} ($x_{i\lambda}$) measured in each spectral band $\lambda = 1, \dots, L$ for a given pixel $i = 1, \dots, N$ are linear combinations of the reflectance characteristics \mathbf{S} ($s_{m\lambda}$) of the materials $m = 1, \dots, M$ present in that area. Possible errors of this approximation and sensor noise are taken into account by adding a noise term \mathbf{N} ($n_{i\lambda}$). In matrix form this can be summarized as

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}, \text{ subject to: } \mathbf{A}\mathbf{1}_M = \mathbf{1}_L, \quad \mathbf{A} \geq \mathbf{0}, \quad (1)$$

where matrix \mathbf{A} (a_{im}) represents the abundance of material m in the area corresponding to pixel i , with positivity and normalization constraints. Note that ground inclination or a changing viewing angle may cause an overall scale factor for all bands that varies with the pixels. This can be incorporated in the model by simply replacing the constraint $\mathbf{A}\mathbf{1}_M = \mathbf{1}_L$ with $\mathbf{A}\mathbf{1}_M \leq \mathbf{1}_L$ which does not affect the discussion in the remainder of the paper. This is clearly a simplified model of the physical phenomena. For example, with spatially fine grained mixtures, called *intimate mixtures*, multiple reflectance may cause departures from this first order model. Additionally there are a number of inherent spatial variations in real data, such as inhomogeneous vapor and dust particles in the atmosphere, that will cause a departure from the linear model in equation (1). Nevertheless, in practical applications a linear model has produced reasonable results for *areal mixtures*.

Supervised vs. Unsupervised techniques *Supervised* spectral unmixing relies on the prior knowledge about the reflectance patterns \mathbf{S} of candidate surface materials, sometimes called *endmembers*, or expert knowledge and a series of semi-automatic steps to find the constituting materials in a particular scene. Once the user identifies a pixel i containing a single material, i.e. $a_{im} = 1$ for a given m and i , the corresponding spectral characteristics of that material can be taken directly from the observations, i.e., $s_{m\lambda} = x_{i\lambda}$ [4]. Given knowledge about the endmembers one can simply find the abundances by solving a constrained least squares problem. The problem with such supervised techniques is that finding the correct \mathbf{S} may require substantial user interaction and the result may be error prone, as a pixel that actually contains a mixture can be misinterpreted as a pure endmember. Another approach obtains endmembers directly from a database. This is also problematic because the actual surface material on the ground may not match the database entries, due to atmospheric absorption or other noise sources. Finding close matches is an ambiguous process as some endmembers have very similar reflectance characteristics and may match several entries in the database.

Unsupervised unmixing, in contrast, tries to identify the endmembers and mixtures directly from the observed data \mathbf{X} without any user interaction. There are a variety of such approaches. In one approach a simplex is fit to the data distribution [7, 6, 2]. The resulting vertex points of the simplex represent the desired endmembers, but this technique is very sensitive to noise as a few boundary points can potentially change the location of the simplex vertex points considerably. Another approach by Szu [9] tries to find abundances that have the highest entropy subject to constraints that the amount of materials is as evenly distributed as possible – an assumption

which is clearly not valid in many actual surface material distributions. A relatively new approach considers modeling the statistical information across wavelength as statistically independent AR processes [1]. This leads directly to the contextual linear ICA algorithm [5]. However, the approach in [1] does not take into account constraints on the abundances, noise, or prior information. Most importantly, the method [1] can only integrate information from a small number of pixels at a time (same as the number of endmembers). Typically however we will have only a few endmembers but many thousand pixels.

2 The Maximum A Posterior Framework

2.1 A probabilistic model of unsupervised spectral unmixing

Our model has observations or data \mathbf{X} and hidden variables \mathbf{A} , \mathbf{S} , and \mathbf{N} that are explained by the noisy linear model (1). We estimate the values of the hidden variables by using MAP

$$p(\mathbf{A}, \mathbf{S} | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{A}, \mathbf{S}) p(\mathbf{A}, \mathbf{S})}{p(\mathbf{X})} = \frac{p_n(\mathbf{X} | \mathbf{A}, \mathbf{S}) p_a(\mathbf{A}) p_s(\mathbf{S})}{p(\mathbf{X})} \quad (2)$$

with $p_a(\mathbf{A})$, $p_s(\mathbf{S})$, $p_n(\mathbf{N})$ as the a priori assumptions of the distributions. With MAP we estimate the most probable values for given priors after observing the data,

$$\mathbf{A}_{\text{MAP}}, \mathbf{S}_{\text{MAP}} = \arg \max_{\mathbf{A}, \mathbf{S}} p(\mathbf{A}, \mathbf{S} | \mathbf{X}) \quad (3)$$

Note that for maximization the constant factor $p(\mathbf{X})$ can be ignored. Our first assumption, which is indicated in equation (2) is that the abundances are independent of the reflectance spectra as their origins are completely unrelated: (A0) \mathbf{A} and \mathbf{S} are independent.

The MAP algorithm is entirely defined by the choices of priors that are guided by the problem of hyperspectral unmixing: (A1) \mathbf{A} represent probabilities for each pixel i . (A2) \mathbf{S} are independent for different material m . (A3) \mathbf{N} are normal i.i.d. for all i, λ . In summary, our MAP framework includes the assumptions A0-A3.

2.2 Including Priors

Priors on the abundances Positivity and normalization of the abundances can be represented as,

$$p_a(\mathbf{A}) = \delta(\mathbf{A} \mathbf{1}_M - \mathbf{1}_N) \Theta(\mathbf{A}), \quad (4)$$

where $\delta()$ represent the Kronecker delta function and $\Theta()$ the step function. With this choice a point not satisfying the constraint will have zero a posteriori probability. This prior introduces no particular bias of the solutions other than abundance constraints. It does however assume the abundances of different pixels to be independent.

Prior on spectra Usually we find systematic trends in the spectra that cause significant correlation. However such an overall trend can be subtracted and/or filtered from the data leaving only independent signals that encode the variation from that overall trend. For example one can capture the conditional dependency structure with a linear auto-regressive (AR) model and analyze the resulting “innovations” or prediction errors [3]. In our model we assume that the spectra represent independent instances of an AR process having a white innovation process $e_{m\lambda}$ distributed according to $p_e(e)$. With a Toeplitz matrix \mathbf{T} of the AR coefficients we

can write, $\mathbf{e}_m = \mathbf{s}_m \mathbf{T}$. The AR coefficients can be found in a preprocessing step on the observations \mathbf{X} . If \mathbf{S} now represents the innovation process itself, our prior can be represented as,

$$p_e(\mathbf{S}) \propto p_e(\mathbf{S}\mathbf{T}) = \prod_{m=1}^M \prod_{\lambda=1}^L p_e\left(\sum_{\lambda'=1}^L s_{m\lambda'} t_{\lambda\lambda'}\right), \quad (5)$$

Additionally $p_e(e)$ is parameterized by a mean and scale parameter and potentially parameters determining the higher moments of the distributions. For brevity we ignore the details of the parameterization in this paper.

Prior on the noise As outlined in the introduction there are a number of problems that can cause the linear model $\mathbf{X} = \mathbf{A}\mathbf{S}$ to be inaccurate (e.g. multiple reflections, inhomogeneous atmospheric absorption, and detector noise.) As it is hard to treat all these phenomena explicitly, we suggest to pool them into one noise variable that we assume for simplicity to be normal distributed with a wavelength dependent noise variance σ_λ ,

$$p(\mathbf{X}|\mathbf{A}, \mathbf{S}) = p_n(\mathbf{N}) = \mathcal{N}(\mathbf{X} - \mathbf{A}\mathbf{S}, \Sigma) = \prod_{\lambda=1}^L \mathcal{N}(\mathbf{x}_\lambda - \mathbf{A}\mathbf{s}_\lambda, \sigma_\lambda \mathbf{I}), \quad (6)$$

where $\mathcal{N}(\cdot, \cdot)$ represents a zero mean Gaussian distribution, and \mathbf{I} the identity matrix indicating the independent noise at each pixel.

2.3 MAP Solution for Zero Noise Case

Let us consider the noise-free case. Although this simplification may be inaccurate it will allow us to greatly reduce the number of free hidden variables - from $NM + ML$ to M^2 . In the noise-free case the variables \mathbf{A}, \mathbf{S} are then deterministically dependent on each other through a NL -dimensional δ -distribution, $p_n(\mathbf{X}|\mathbf{A}\mathbf{S}) = \delta(\mathbf{X} - \mathbf{A}\mathbf{S})$. We can remove one of these variables from our discussion by integrating (2). It is instructive to first consider removing \mathbf{A}

$$p(\mathbf{S}|\mathbf{X}) \propto \int d\mathbf{A} \delta(\mathbf{X} - \mathbf{A}\mathbf{S}) p_a(\mathbf{A}) p_s(\mathbf{S}) = |\mathbf{S}^{-1}| p_a(\mathbf{X}\mathbf{S}^{-1}) p_s(\mathbf{S}). \quad (7)$$

We omit tedious details and assume $L = M$ and invertible \mathbf{S} so that we can perform the variable substitution that introduces the Jacobian determinant $|\mathbf{S}^{-1}|$. Let us consider the influence of the different terms. The Jacobian determinant measures the volume spanned by the endmembers \mathbf{S} . Maximizing its inverse will therefore try to shrink the simplex spanned by \mathbf{S} . The term $p_a(\mathbf{X}\mathbf{S}^{-1})$ should guarantee that all data points map into the inside of the simplex, since the term should contribute zero or low probability for points that violate the constraint. Note that these two terms, in principle, define the same objective as the simplex envelope fitting algorithms previously mentioned [2].

In the present work we are more interested in the algorithm that results from removing \mathbf{S} and finding the MAP estimate of \mathbf{A} . We obtain (cf. Eq.(7))

$$p(\mathbf{A}|\mathbf{X}) \propto \int d\mathbf{S} \delta(\mathbf{X} - \mathbf{A}\mathbf{S}) p_a(\mathbf{A}) p_s(\mathbf{S}) = |\mathbf{A}^{-1}| p_s(\mathbf{A}^{-1}\mathbf{X}) p_a(\mathbf{A}). \quad (8)$$

For now we assumed $N = M$.¹ If $p_s(\mathbf{S})$ factors over m , i.e. endmembers are independent, maximizing the first two terms represents the ICA algorithm. However,

¹In practice more frequently we have $N > M$. In that case the observations \mathbf{X} can be mapped into a M dimensional subspace using the singular value decomposition (SVD), $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. The discussion applies then to the reduced observations $\tilde{\mathbf{X}} = \mathbf{U}_M^T \mathbf{X}$ with \mathbf{U}_M being the first M columns of \mathbf{U} .

the prior on \mathbf{A} will restrict the solutions to satisfy the abundance constraints and bias the result depending on the detailed choice of $p_a(\mathbf{A})$, so we are led to *constrained ICA*.

In summary, depending on which variable we integrate out we obtain two methods for solving the spectral unmixing problem: the known technique of simplex fitting and a new constrained ICA algorithm.

2.4 MAP Solution for the Noisy Case

Combining the choices for the priors made in section 2.2 (Eqs.(4), (5) and (6)) with (2) and (3) we obtain

$$\mathbf{A}_{\text{MAP}}, \mathbf{S}_{\text{MAP}} = \arg \max_{\mathbf{A}, \mathbf{S}} \prod_{\lambda=q}^L \left\{ \prod_{i=1}^N \mathcal{N}(x_{i\lambda} - \mathbf{a}_i \mathbf{s}_\lambda, \sigma_\lambda) \prod_{m=1}^M p_e \left(\sum_{\lambda'=1}^L s_{m\lambda'} t_{\lambda\lambda'} \right) \right\}, \quad (9)$$

subject to $\mathbf{A} \mathbf{1}_M = \mathbf{1}_L$, $\mathbf{A} \geq \mathbf{0}$. The logarithm of the cost function in (9) is denoted by $L = L(\mathbf{A}, \mathbf{S})$. Its gradient with respect to the hidden variables is

$$\frac{\partial L}{\partial \mathbf{s}_m} = -\mathbf{A}^T \mathbf{n}_m \text{diag}(\boldsymbol{\sigma})^{-1} - f_s(\mathbf{s}_m) \quad (10)$$

where $\mathbf{N} = \mathbf{X} - \mathbf{A}\mathbf{S}$, \mathbf{n}_m are the M column vectors of \mathbf{N} , $f_s(s) = -\frac{\partial \ln p_e(s)}{\partial s}$. In (10) f_s is applied to each element of \mathbf{s}_m .

The optimization with respect to \mathbf{A} for given \mathbf{S} can be implemented as a standard *weighted least squares* (LS) problem with a linear constraint and positivity bounds. Since the constraints apply for every pixel independently one can solve N separate constrained LS problems of M unknowns each. We alternate between gradient steps for \mathbf{S} and explicit solutions for \mathbf{A} until convergence. Any additional parameters of $p_e(e)$ such as scale and mean may be obtained in a maximum likelihood (ML) sense by maximizing L . Note that the nonlinear optimization is not subject to constraints; the constraints apply only in the quadratic optimization.

3 Experiments

3.1 Zero Noise Case: Artificial Mixtures

In our first experiment we use mineral data from the United States Geological Survey (USGS)² to build artificial mixtures for evaluating our unsupervised unmixing framework. Three target endmembers were chosen (Almandine WS479, Montmorillonite+Illite CM42 and Dickite NMNH106242). A spectral scene of 100 samples was constructed by creating a random mixture of the three minerals. Of the 100 samples, there were no pure samples (i.e. no mineral had more than a 80% abundance in any sample). Figure 1A is the spectra of the endmembers recovered by the constrained ICA technique of section 2.3, where the constraints were implemented with penalty terms added to the conventional maximum likelihood ICA algorithm. These are nearly identical to the spectra of the true endmembers, shown in figure 1B, which were used for mixing. Interesting to note is the scatter-plot of the 100 samples across two bands. The open circles are the absorption values at these two bands for endmembers found by the MAP technique. Given that each mixed sample consists of no more than 80% of any endmember, the endmember points on the scatter-plot are quite distant from the cluster. A simplex fitting technique would have significant difficulty recovering the endmembers from this clustering.

²see <http://speclab.cr.usgs.gov/spectral.lib.456.descript/decript04.html>

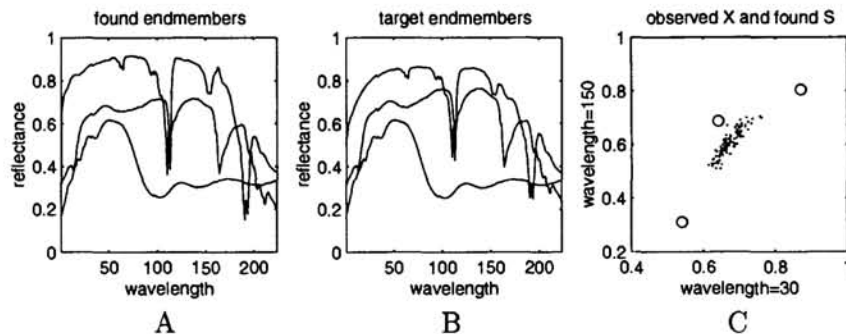


Figure 1: Results for noise-free artificial mixture. **A** recovered endmembers using MAP technique. **B** “true” target endmembers. **C** scatter plot of samples across 2 bands showing the absorption of the three endmembers computed by MAP (open circles).

3.2 Noisy Case: Real Mixtures

To validate the noise model MAP framework of section 2.4 we conducted an experiment using ground truthed USGS data representing real mixtures. We selected 10x10 blocks of pixels from three different regions³ in the AVIRIS data of the Cuprite, Nevada mining district. We separate these 300 mixed spectra assuming two endmembers and an AR detrending with 5 AR coefficients and the MAP techniques of section 2.4. Overall brightness was accounted for as explain in the linear modeling of section 1. The endmembers are shown in figure 2A and B in comparison to laboratory spectra from the USGS spectral library for these minerals [8]. Figure 2C shows the corresponding abundances, which match the ground truth; region (III) mainly consists of Muscovite while regions (I)+(II) contain (areal) mixtures of Kaolinite and Muscovite.

4 Discussion

Hyperspectral unmixing is a challenging practical problem for unsupervised learning. Our probabilistic approach leads to several interesting algorithms: (1) simplex fitting, (2) constrained ICA and (3) constrained least squares that can efficiently use multi-channel information. An important element of our approach is the explicit use of prior information. Our simulation examples show that we can recover the endmembers, even in the presence of noise and model uncertainty. The approach described in this paper does not yet exploit local correlations between neighboring pixels that are well known to exist. Future work will therefore exploit not only *spectral* but also *spatial* prior information for detecting objects and materials.

Acknowledgments

We would like to thank Gregg Swayze at the USGS for assistance in obtaining the data.

³The regions were from the image plate2.cuprite95.alpha.2um.image.wlocals.gif in <ftp://speclab.cr.usgs.gov/pub/cuprite/gregg.thesis.images/>, at the coordinates (265,710) and (275,697), which contained Kaolinite and Muscovite 2, and (143,661), which only contained Muscovite 2.

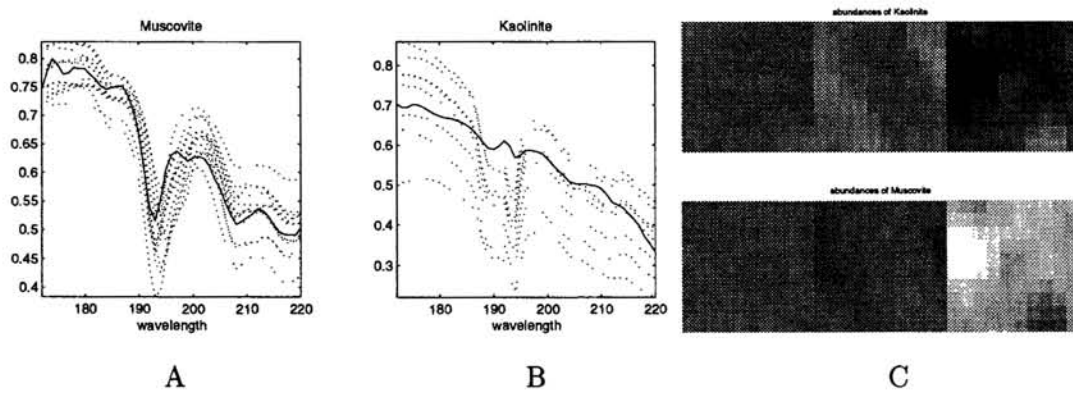


Figure 2: **A** Spectra of computed endmember (solid line) vs Muscovite sample spectra from the USGS data base library. Note we show only part of the spectrum since the discriminating features are located only between band 172 and 220. **B** Computed endmember (solid line) vs Kaolinite sample spectra from the USGS data base library. **C** Abundances for Kaolinite and Muscovite for three regions (lighter pixels represent higher abundance). Region 1 and region 2 have similar abundances for Kaolinite and Muscovite, while region 3 contains more Muscovite.

References

- [1] J. Bayliss, J. A. Gualtieri, and R. Crompton. Analyzing hyperspectral data with independent component analysis. In J. M. Selander, editor, *Proc. SPIE Applied Image and Pattern Recognition Workshop*, volume 9, P.O. Box 10, Bellingham WA 98227-0010, 1997. SPIE.
- [2] J.W. Boardman and F.A. Kruse. Automated spectral analysis: a geologic example using AVIRIS data, north Grapevine Mountains, Nevada. In *Tenth Thematic Conference on Geologic Remote Sensing*, pages 407–418, Ann Arbor, MI, 1994. Environmental Research Institute of Michigan.
- [3] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1991.
- [4] F. Maselli, M. Pieri, and C. Conese. Automatic identification of end-members for the spectral decomposition of remotely sensed scenes. *Remote Sensing for Geography, Geology, Land Planning, and Cultural Heritage (SPIE)*, 2960:104–109, 1996.
- [5] B. Pearlmutter and L. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 613–619, Cambridge MA, 1997. MIT Press.
- [6] J.J. Settle. Linear mixing and the estimation of ground cover proportions. *International Journal of Remote Sensing*, 14:1159–1177, 1993.
- [7] M.O. Smith, J.B. Adams, and A.R. Gillespie. Reference endmembers for spectral mixture analysis. In *Fifth Australian remote sensing conference*, volume 1, pages 331–340, 1990.
- [8] U.S. Geological Survey. USGS digital spectral library. Open File Report 93-592, 1993.
- [9] H. Szu and C. Hsu. Landsat spectral demixing via superresolution of blind matrix inversion by constraint MaxEnt neural nets. In *Wavelet Applications IV*, volume 3078, pages 147–160. SPIE, 1997.

Invariant Feature Extraction and Classification in Kernel Spaces

Sebastian Mika¹, Gunnar Rätsch^{1,2}, Jason Weston²,
Bernhard Schölkopf³, Alex Smola⁴, and Klaus-Robert Müller¹

1 GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany

2 Barnhill BioInformatics, 6709 Waters Av., Savannah, GR 31406, USA

3 Microsoft Research Ltd., 1 Guildhall Street, Cambridge CB2 3NH, UK

4 Australian National University, Canberra, 0200 ACT, Australia

{mika, raetsch, klaus}@first.gmd.de, jasonw@dcs.rhnc.ac.uk

bsc@microsoft.com, Alex.Smola.anu.edu.au

Abstract

We incorporate prior knowledge to construct nonlinear algorithms for invariant feature extraction and discrimination. Employing a unified framework in terms of a nonlinear variant of the Rayleigh coefficient, we propose non-linear generalizations of Fisher's discriminant and oriented PCA using Support Vector kernel functions. Extensive simulations show the utility of our approach.

1 Introduction

It is common practice to preprocess data by extracting linear or nonlinear features. The most well-known feature extraction technique is principal component analysis PCA (e.g. [3]). It aims to find an orthonormal, ordered basis such that the i -th direction describes as much variance as possible while maintaining orthogonality to all other directions. However, since PCA is a linear technique, it is too limited to capture interesting nonlinear structure in a data set and nonlinear generalizations have been proposed, among them Kernel PCA [14], which computes the principal components of the data set mapped nonlinearly into some high dimensional feature space \mathcal{F} .

Often one has prior information, for instance, we might know that the sample is corrupted by noise or that there are invariances under which a classification should not change. For feature extraction, the concepts of known noise or transformation invariance are to a certain degree equivalent, i.e. they can both be interpreted as causing a change in the feature which ought to be minimized. Clearly, invariance alone is not a sufficient condition for a good feature, as we could simply take the constant function. What one would like to obtain is a feature which is as invariant as possible while still covering as much of the information necessary for describing the particular data. Considering only one (linear) feature vector w and restricting to first and second order statistics of the data one arrives at a maximization of the so called *Rayleigh* coefficient

$$J(w) = \frac{w^T S_I w}{w^T S_N w}, \quad (1)$$

where \mathbf{w} is the feature vector and S_I , S_N are matrices describing the desired and undesired properties of the feature, respectively (e.g. information and noise). If S_I is the data covariance and S_N the noise covariance, we obtain *oriented PCA* [3]. If we leave the field of data description to perform supervised classification, it is common to choose S_I as the separability of class centers (between class variance) and S_N to be the within class variance. In that case, we recover the well known Fisher Discriminant [7]. The ratio in (1) is maximized when we cover much of the information coded by S_I while avoiding the one coded by S_N . The problem is known to be solved, in analogy to PCA, by a generalized symmetric eigenproblem $S_I \mathbf{w} = \lambda S_N \mathbf{w}$ [3], where $\lambda \in \mathbb{R}$ is the corresponding (biggest) eigenvalue.

In this paper we generalize this setting to a nonlinear one. In analogy to [8, 14] we first map the data via some nonlinear mapping Φ to some high-dimensional feature space \mathcal{F} and then optimize (1) in \mathcal{F} . To avoid working with the mapped data explicitly (which might be impossible if \mathcal{F} is infinite dimensional) we introduce support vector kernel functions [11], the well-known kernel trick. These kernel functions $k(\mathbf{x}, \mathbf{y})$ compute a dot product in some feature space \mathcal{F} , i.e. $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$. Formulating the algorithms in \mathcal{F} using Φ only in dot products, we can replace any occurrence of a dot product by the kernel function k . Possible choices for k which have proven useful e.g. in Support Vector Machines [2] or Kernel PCA [14] are Gaussian RBF, $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/c)$, or polynomial kernels, $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$, for some positive constants $c \in \mathbb{R}$ and $d \in \mathbb{N}$, respectively.

The remainder of this paper is organized as follows: The next section shows how to formulate the optimization problem induced by (1) in feature space. Section 3 considers various ways to find Fisher's Discriminant in \mathcal{F} ; we conclude with extensive experiments in section 4 and a discussion of our findings.

2 Kernelizing the Rayleigh Coefficient

To optimize (1) in some kernel feature space \mathcal{F} we need to find a formulation which uses only dot products of Φ -images. As numerator and denominator are both scalars this can be done independently. Furthermore, the matrices S_I and S_N are basically covariances and thus the sum over outer products of Φ -images. Therefore, and due to the linear nature of (1) every solution $\mathbf{w} \in \mathcal{F}$ can be written as an expansion in terms of mapped training data¹, i.e.

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}_i). \quad (2)$$

To define some common choices in \mathcal{F} let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_\ell\}$ be our training sample and, where appropriate, $\mathcal{X}_1 \cup \mathcal{X}_2 = \mathcal{X}$, $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$, two subclasses (with $|\mathcal{X}_i| = \ell_i$). We get the full covariance of \mathcal{X} by

$$C = \frac{1}{\ell} \sum_{\mathbf{x} \in \mathcal{X}} (\Phi(\mathbf{x}) - \mathbf{m})(\Phi(\mathbf{x}) - \mathbf{m})^\top \text{ with } \mathbf{m} = \frac{1}{\ell} \sum_{\mathbf{x} \in \mathcal{X}} \Phi(\mathbf{x}), \quad (3)$$

¹ S_B and S_W are operators on a (finite-dimensional) subspace spanned by the $\Phi(\mathbf{x}_i)$ (in a possibly infinite space). Let $\mathbf{w} = \mathbf{v}_1 + \mathbf{v}_2$, where $\mathbf{v}_1 \in \text{Span}(\Phi(\mathbf{x}_i) : i = 1, \dots, \ell)$ and $\mathbf{v}_2 \perp \text{Span}(\Phi(\mathbf{x}_i) : i = 1, \dots, \ell)$. Then for $S = S_W$ or $S = S_B$ (which are both symmetric)

$$\begin{aligned} \langle \mathbf{w}, S\mathbf{w} \rangle &= \langle (\mathbf{v}_1 + \mathbf{v}_2), S(\mathbf{v}_1 + \mathbf{v}_2) \rangle \\ &= \langle (\mathbf{v}_1 + \mathbf{v}_2)S, \mathbf{v}_1 \rangle \\ &= \langle \mathbf{v}_1, S\mathbf{v}_1 \rangle \end{aligned}$$

As \mathbf{v}_1 lies in the span of the $\Phi(\mathbf{x}_i)$ and S only operates on this subspace there exist an expansion of \mathbf{w} which maximizes $J(\mathbf{w})$.

which could be used as S_I in oriented Kernel PCA. For S_N we could use an estimate of the noise covariance, analogous to the definition of C but over mapped patterns sampled from the assumed noise distribution. The standard formulation of the Fisher discriminant in \mathcal{F} , yielding the *Kernel Fisher Discriminant* (KFD) [8] is given by

$$S_W = \sum_{i=1,2} \sum_{\mathbf{x} \in \mathcal{X}_i} (\Phi(\mathbf{x}) - \mathbf{m}_i)(\Phi(\mathbf{x}) - \mathbf{m}_i)^\top \quad \text{and} \quad S_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top,$$

the within-class scatter S_W (as S_N), and the between class scatter S_B (as S_I). Here \mathbf{m}_i is the sample mean for patterns from class i .

To incorporate a known invariance e.g. in oriented Kernel PCA, one could use the tangent covariance matrix [12],

$$T = \frac{1}{\ell t^2} \sum_{\mathbf{x} \in \mathcal{X}} (\Phi(\mathbf{x}) - \Phi(\mathcal{L}_t \mathbf{x}))(\Phi(\mathbf{x}) - \Phi(\mathcal{L}_t \mathbf{x}))^\top \text{ for some small } t > 0. \quad (4)$$

Here \mathcal{L}_t is a local 1-parameter transformation. T is a finite difference approximation t of the covariance of the tangent of \mathcal{L}_t at point $\Phi(\mathbf{x})$ (details e.g. in [12]). Using $S_I = C$ and $S_N = T$ in oriented Kernel PCA, we impose invariance under the local transformation \mathcal{L}_t . Crucially, this matrix is not only constructed from the training patterns \mathcal{X} . Therefore, the argument used to find the expansion (2) is slightly incorrect. Nevertheless, we can assume that (2) is a reasonable approximation for describing the variance induced by T .

Multiplying either of these matrices from the left and right with the expansion (2), we can find a formulation which uses only dot products. For the sake of brevity, we only give the explicit formulation of (1) in \mathcal{F} for KFD (cf. [8] for details). Defining $(\boldsymbol{\mu}_i)_j = \frac{1}{\ell_i} \sum_{\mathbf{x} \in \mathcal{X}_i} k(\mathbf{x}_j, \mathbf{x})$ we can write (1) for KFD as

$$J(\boldsymbol{\alpha}) = \frac{(\boldsymbol{\alpha}^\top \boldsymbol{\mu})^2}{\boldsymbol{\alpha}^\top N \boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}^\top M \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top N \boldsymbol{\alpha}}, \quad (5)$$

where $N = KK^\top - \sum_{i=1,2} \ell_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top$, $\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$, $M = \boldsymbol{\mu} \boldsymbol{\mu}^\top$, and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The results for other choices of S_I and S_N in \mathcal{F} as for the cases of oriented kernel PCA or transformation invariance can be obtained along the same lines. Note that we still have to maximize a Rayleigh coefficient. However, now it is a quotient in terms of expansion coefficients $\boldsymbol{\alpha}$, and not in terms of $\mathbf{w} \in \mathcal{F}$ which is a potentially infinite-dimensional space. Furthermore, it is well known that the solution for this special eigenproblem is in the direction of $N^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ [7], which can be solved using e.g. a Cholesky factorization of N . The projection of a new pattern \mathbf{x} onto \mathbf{w} in \mathcal{F} can then be computed by

$$(\mathbf{w} \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{\ell} \alpha_i k(\mathbf{x}_i, \mathbf{x}). \quad (6)$$

3 Algorithms

Estimating a covariance matrix with rank up to ℓ from ℓ samples is ill-posed. Furthermore, by performing an explicit centering in \mathcal{F} each covariance matrix loses one more dimension, i.e. it has only rank $\ell - 1$ (even worse, for KFD the matrix N has rank $\ell - 2$). Thus the ratio in (1) is not well defined anymore, as the denominator might become zero. In the following we will propose several ways to deal with this problem in KFD. Furthermore we will tackle the question how to solve the optimization problem of KFD more efficiently. So far, we have an eigenproblem of size $\ell \times \ell$. If ℓ becomes large this is numerically demanding. Reformulations of the original problem allow to overcome some of these limitations. Finally, we describe the connection between KFD and RBF networks.

3.1 Regularization and Solution on a Subspace

As noted before, the matrix N has only rank $\ell - 2$. Besides numerical problems which can cause the matrix N to be not even positive, we could think of imposing some regularization to control capacity in \mathcal{F} . To this end, we simply add a multiple of the identity matrix to N , i.e. replace N by N_μ where

$$N_\mu := N + \mu I. \quad (7)$$

This can be viewed in different ways: (i) for $\mu > 0$ it makes the problem feasible and numerically more stable as N_μ becomes positive; (ii) it can be seen as decreasing the bias in sample based estimation of eigenvalues (cf. [6]); (iii) it imposes a regularization on $\|\alpha\|^2$, favoring solutions with small expansion coefficients. Furthermore, one could use other regularization type additives to N , e.g. penalizing $\|w\|^2$ in analogy to SVM (by adding the kernel matrix $K_{ij} = k(x_i, x_j)$).

To optimize (5) we need to solve an $\ell \times \ell$ eigenproblem, which might be intractable for large ℓ . As the solutions are not sparse one can not directly use efficient algorithms like chunking for Support Vector Machines (cf. [13]). To this end, we might restrict the solution to lie in a subspace, i.e. instead of expanding w by (2) we write

$$w = \sum_{i=1}^m \alpha_i \Phi(z_i), \quad (8)$$

with $m < \ell$. The patterns z_i could either be a subset of the training patterns \mathcal{X} or e.g. be estimated by some clustering algorithm. The derivation of (5) does not change, only K is now $m \times \ell$ and we end up with $m \times m$ matrices N and M . Another advantage is, that it increases the rank of N (relative to its size) although there still might be some need for regularization.

3.2 Quadratic optimization and Sparsification

Even if N has full rank, maximizing (5) is underdetermined: if α is optimal, then so is any multiple thereof. Since $\alpha^\top M \alpha = (\alpha^\top \mu)^2$, M has rank one. Thus we can seek for a vector α , such that $\alpha^\top N \alpha$ is minimal for fixed $\alpha^\top \mu$ (e.g. to 1). The solution is unique and we can find the optimal α by solving the quadratic optimization problem:

$$\min \alpha^\top N \alpha \quad \text{subject to} \quad \alpha^\top \mu = 1. \quad (9)$$

Although the quadratic optimization problem is not easier to solve than the eigenproblem, it has an appealing interpretation. The constraint $\alpha^\top \mu = 1$ ensures, that the average class distance, projected onto the direction of discrimination, is constant, while the intra class variance is minimized, i.e. we maximize the *average* margin. Contrarily, the SVM approach [2] optimizes for a large *minimal* margin.

Considering (9) we are able to overcome another shortcoming of KFD. The solutions α are *not* sparse and thus evaluating (6) is expensive. To solve this we can add an l_1 -regularizer $\lambda \|\alpha\|_1$ to the objective function, where λ is a regularization parameter allowing us to adjust the degree of sparseness.

3.3 Connection to RBF Networks

Interestingly, there exists a close connection between RBF networks (e.g. [9, 1]) and KFD. If we add no regularization and expand in all training patterns, we find that an optimal α is given by $\alpha = K^{-1} \mathbf{y}$, where K is the symmetric, positive matrix of all kernel elements $k(x_i, x_j)$ and \mathbf{y} the ± 1 label vector². A RBF-network with the

²To see this, note that N can be written as $N = KDK$ where $D = I - \mathbf{y}_1 \mathbf{y}_1^\top - \mathbf{y}_2 \mathbf{y}_2^\top$ has rank $\ell - 2$, while \mathbf{y}_i is the vector of $1/\sqrt{\ell_i}$'s for patterns from class i and zero otherwise.

	RBF	AB	AB _R	SVM	KFD	
Banana	10.8±0.06	12.3±0.07	<i>10.9±0.04</i>	11.5±0.07	10.8±0.05	Table 1: Comparison between KFD, single RBF classifier, AdaBoost (AB), regul. AdaBoost (AB _R) and SVMs (see text). Best result in bold face, second best in italics.
B.Cancer	27.6±0.47	30.4±0.47	26.5±0.45	<i>26.0±0.47</i>	25.8±0.46	
Diabetes	24.3±0.19	26.5±0.23	23.8±0.18	<i>23.5±0.17</i>	23.2±0.16	
German	24.7±0.24	27.5±0.25	24.3±0.21	23.6±0.21	<i>23.7±0.22</i>	
Heart	17.6±0.33	20.3±0.34	16.5±0.35	16.0±0.33	<i>16.1±0.34</i>	
Image	3.3±0.06	2.7±0.07	2.7±0.06	<i>3.0±0.06</i>	4.8±0.06	
Ringnorm	1.7±0.02	1.9±0.03	<i>1.6±0.01</i>	1.7±0.01	1.5±0.01	
F.Sonar	34.4±0.20	35.7±0.18	34.2±0.22	32.4±0.18	<i>33.2±0.17</i>	
Splice	<i>10.0±0.10</i>	10.1±0.05	9.5±0.07	10.9±0.07	10.5±0.06	
Thyroid	4.5±0.21	<i>4.4±0.22</i>	4.6±0.22	4.8±0.22	4.2±0.21	
Titanic	23.3±0.13	<i>22.6±0.12</i>	<i>22.6±0.12</i>	22.4±0.10	23.2±0.20	
Twonorm	2.9±0.03	3.0±0.03	<i>2.7±0.02</i>	3.0±0.02	2.6±0.02	
Waveform	10.7±0.11	10.8±0.06	9.8±0.08	<i>9.9±0.04</i>	<i>9.9±0.04</i>	

same kernel at each sample and fixed kernel width gives the same solution, if the mean squared error between labels and output is minimized. Also for the case of restricted expansions (8) there exists a connection to RBF networks with a smaller number of centers (cf. [4]).

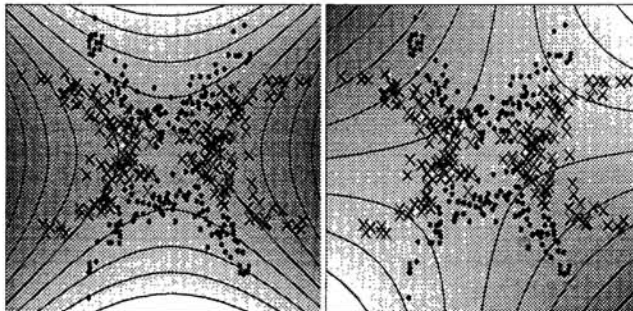
4 Experiments

Kernel Fisher Discriminant Figure 1 shows an illustrative comparison of the features found by KFD, and Kernel PCA. The KFD feature discriminates the two classes, the first Kernel PCA feature picks up the important nonlinear structure. To evaluate the performance of the KFD on real data sets we performed an extensive comparison to other state-of-the-art classifiers, whose details are reported in [8].³ We compared the Kernel Fisher Discriminant and Support Vector Machines, both with Gaussian kernel, to AdaBoost [5], and regularized AdaBoost [10] (cf. table 1). For KFD we used the regularized within-class scatter (7) and computed projections onto the optimal direction $w \in \mathcal{F}$ by means of (6). To use w for classification we have to estimate a threshold. This can be done by e.g. trying all thresholds between two outputs on the training set and selecting the median of those with the smallest empirical error, or (as we did here) by computing the threshold which maximizes the margin on the outputs in analogy to a Support Vector Machine, where we deal with errors on the training set by using the SVM soft margin approach. A disadvantage of this is, however, that we have to control the regularization constant for the slack variables. The results in table 1 show the average test error and the standard

If K has full rank, the null space of D , which is spanned by y_1 and y_2 , is the null space of N . For $\bar{\alpha} = K^{-1}y$ we get $\bar{\alpha}^T N \bar{\alpha} = 0$ and $\bar{\alpha}^T \mu \neq 0$. As we are free to fix the constraint $\alpha^T \mu$ to any positive constant (not just 1), $\bar{\alpha}$ is also feasible.

³The breast cancer domain was obtained from the University Medical Center, Inst. of Oncology, Ljubljana, Yugoslavia. Thanks to M. Zwitter and M. Soklic for the data. All data sets used in the experiments can be obtained via <http://www.first.gmd.de/~raetsch/>.

Figure 1: Comparison of feature found by KFD (left) and first Kernel PCA feature (right). Depicted are two classes (information only used by KFD) as dots and crosses and levels of same feature value. Both with polynomial kernel of degree two, KFD with the regularized within class scatter (7) ($\mu = 10^{-3}$).



deviation of the averages' estimation, over 100 runs with different realizations of the datasets. To estimate the necessary parameters, we ran 5-fold cross validation on the first five realizations of the training sets and took the model parameters to be the median over the five estimates (see [10] for details of the experimental setup).

Using prior knowledge. A toy example (figure 2) shows a comparison of Kernel PCA and oriented Kernel PCA, which used S_I as the full covariance (3) and as noise matrix S_N the tangent covariance (4) of (i) rotated patterns and (ii) along the x-axis translated patterns. The toy example shows how imposing the desired invariance yields meaningful invariant features.

In another experiment we incorporated prior knowledge in KFD. We used the USPS database of handwritten digits, which consists of 7291 training and 2007 test patterns, each 256 dimensional gray scale images of the digits 0...9. We used the regularized within-class scatter (7) ($\mu = 10^{-3}$) as S_N and added to it an multiple λ of the tangent covariance (4), i.e. $S_N = N_\mu + \lambda T$. As invariance transformations we have chosen horizontal and vertical translation, rotation, and thickening (cf. [12]), where we simply averaged the matrices corresponding to each transformation. The feature was extracted by using the restricted expansion (8), where the patterns z_i were the first 3000 training samples. As kernel we have chosen a Gaussian of width $0.3 \cdot 256$, which is optimal for SVMs [12]. For each class we trained one KFD which classified this class against the rest and computed the 10-class error by the winner-takes-all scheme. The threshold was estimated by minimizing the empirical risk on the normalized outputs of KFD.

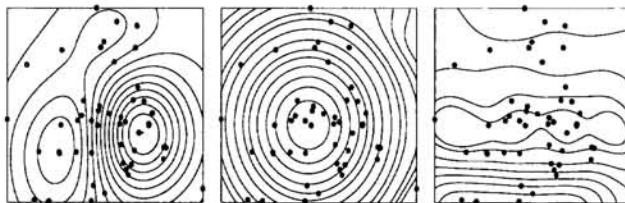
Without invariances, i.e. $\lambda = 0$, we achieved a test error of 3.7%, slightly better than a plain SVM with the same kernel (4.2%) [12]. For $\lambda = 10^{-3}$, using the tangent covariance matrix led to a very slight improvement to 3.6%. That the result was not significantly better than the corresponding one for KFD (3.7%) can be attributed to the fact that we used the same expansion coefficients in both cases. The tangent covariance matrix, however, lives in a slightly different subspace. And indeed, a subsequent experiment where we used vectors which were obtained by clustering a larger dataset, including virtual examples generated by the appropriate invariance transformation, led to 3.1%, comparable to an SVM using prior knowledge (e.g. [12]; best SVM result 2.9% with local kernel and virtual support vectors).

5 Conclusion

In the task of learning from data it is equivalent to have prior knowledge about e.g. invariances or about specific sources of noise. In the case of feature extraction, we seek features which are sufficiently (noise-) invariant while still describing interesting structure. Oriented PCA and, closely related, Fisher's Discriminant, use particularly simple features, since they only consider first and second order statistics for maximizing the Rayleigh coefficient (1). Since linear methods can be too restricted in many real-world applications, we used Support Vector Kernel functions to obtain nonlinear versions of these algorithms, namely oriented Kernel PCA and Kernel Fisher Discriminant analysis.

Our experiments show that the Kernel Fisher Discriminant is competitive or in

Figure 2: Comparison of first features found by Kernel PCA and oriented Kernel PCA (see text); from left to right: KPCA, OKPCA with rotation and translation invariance; all with Gaussian kernel.



some cases even superior to the other state-of-the-art algorithms tested. Interestingly, both SVM and KFD construct a hyperplane in \mathcal{F} which is in some sense optimal. In many cases, the one given by the solution w of KFD is superior to the one of SVMs. Encouraged by the preliminary results for digit recognition, we believe that the reported results can be improved, by incorporating different invariances and using e.g. local kernels [12].

Future research will focus on further improvements on the algorithmic complexity of our new algorithms, which is so far larger than the one of the SVM algorithm, and on the connection between KFD and Support Vector Machines (cf. [16, 15]).

Acknowledgments This work was partially supported by grants of the DFG (JA 379/5-2,7-1,9-1) and the EC STORM project number 25387 and carried out while BS and AS were with GMD First.

References

- [1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford Univ. Press, 1995.
- [2] B. Boser, I. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. COLT*, pages 144–152. ACM Press, 1992.
- [3] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks*. Wiley, New York, 1996.
- [4] B.Q. Fang and A.P. Dawid. Comparison of full bayes and bayes-least squares criteria for normal discrimination. *Chinese Journal of Applied Probability and Statistics*, 12:401–410, 1996.
- [5] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT 94*. LNCS, 1994.
- [6] J.H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.
- [7] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.
- [8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.
- [9] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.
- [10] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. Technical Report NC-TR-1998-021, Royal Holloway College, University of London, UK, 1998.
- [11] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.
- [12] B. Schölkopf. *Support vector learning*. Oldenbourg Verlag, 1997.
- [13] B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors. *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1999.
- [14] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [15] A. Shashua. On the relationship between the support vector machine for classification and sparsified fisher’s linear discriminant. *Neural Processing Letters*, 9(2):129–139, April 1999.
- [16] S. Tong and D. Koller. Bayes optimal hyperplanes \rightarrow maximal margin hyperplanes. Submitted to IJCAI’99 Workshop on Support Vector Machines (robotics.stanford.edu/~koller/), 1999.