PhD Thesis

Enhanced Multiple Output Regression based on Canonical Correlation Analysis with Applications in Computer Vision

Michael Reiter

Graz University of Technology
Institute for Computer Graphics and Vision

Thesis supervisors
Prof. Dr. Horst Bischof
Prof. Dr. Robert Sablatnig

Graz, June 2010

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Ort	Datum	Unterschrift

Abstract

Modeling the functional relation between high dimensional signals is a common task in computer vision. Just like natural perception systems are able to relate and combine impressions from different senses (speech, facial expression, gestures, haptics, etc.), simultaneous processing of signals of different sources and establishing a functional relation between these sources is an important issue in this research area.

In this thesis, we employ statistical regression models for prediction of high dimensional signals, such as images, where standard regression algorithms will cause overfitting to the training sample due to the large number of regression parameters to be estimated. We employ *canonical correlation analysis* (CCA) and its nonlinear generalization *kernel-CCA* for making explicit the regression relevant subspaces and to reduce the effective number of parameters of the regression model.

The proposed algorithms are successfully applied to 3D pose estimation, prediction of face depth maps from a single color image of the face, and fast matching of *active appearance models* and *active feature models*. Qualitative and quantitative results show that CCA-based methods outperform standard regression models because of their ability to exploit correlations in the input and output space.

Kurzfassung

Die Modellierung des funktionalen Zusammenhangs zwischen hochdimensionalen Signalen ist eine Problemstellung, die im Bereich der Bildverarbeitung und automatischen Objekterkennung häufig auftritt. Auch die natürliche Wahrnehmung beruht auf der Fähigkeit, Eindrücke verschiedener Quellen in Beziehung zu setzen (Gesichtsausdruck, Sprache, Gesten, Berührungen, u.s.w.). Die simultane Verarbeitung von Signalen aus verschiedenen Quellen und das Analysieren des funktionalen Zusammenhangs der Signale ist daher ein wichtiges Thema in diesem Forschungsbereich.

In dieser Doktorarbeit setzen wir statistische Verfahren der Regressionsanalyse ein, um ein hochdimensionales Ausgabesignal anhand eines hochdimensionalen Eingabesignals vorherzusagen. Eine besondere Herausforderung in der Bildverabreitung ist dabei das ungünstige Verhältnis der Kardinalität der Trainingsmenge zur Anzahl der zu schätzenden Parameter des Vorhersagemodels, da die Anzahl der Parameter in Zusammenhang zur Dimensionalität der Daten steht und meist nur relativ wenige Beobachtungen zur Verfügung stehen, anhand derer die Parameterwerte gelernt werden können. Hier kommt die *kanonische Korrelationsanalyse* (*canonical correlation analysis*, CCA) bzw. auch deren nicht-lineare Erweiterung durch Kernel-Methoden zum Einsatz, mit deren Hilfe sich regressionsrelevante Unterräume der Signalräume bestimmen lassen und dadurch die effektive Anzahl der Parameter reduziert werden kann.

Die vorgestellten Algorithmen werden erfolgreich für folgende Anwendungen eingesetzt: Lageschätzung von 3D Objekten, Vorhersage von 3D Struktur eines Gesichts anhand eines einzelnen RGB Farbbildes des Gesichts und schnelles *matching* von *active appearance models* und *active feature models*. Qualitative und quantitative Ergebnisse zeigen, dass CCA-basierte Verfahren durch die Eigenschaft, sowohl im Eingabesignalraum als auch im Ausgabesignalraum Korrelationen ausnutzen zu können, bessere Ergebnisse erzielen als Standardverfahren.

Acknowledgements

This dissertation would not have been possible without the help and support of a number of people.

First and foremost, I would like to thank my supervisor Horst Bischof for guiding and supporting me throughout the years, for encouraging me to develop my own ideas, for giving me the freedom to follow my own research path and not running out of patience, when things overran their time. I am grateful for the opportunity to finish this thesis at the ICG.

I would like to thank Robert Sablatnig, co-supervisor of this dissertation and head the Institute of Computer Aided Automation (CAA) at the Vienna University of Technology, where this most of this thesis was written. Thank you for your encouragement and support.

Thanks to my former colleagues Georg Langs and René Donner. I began to enjoy the strain of an approaching paper deadline. Our collaboration also greatly invigorated my own research. Thanks to all colleagues at CAA and ICG for the pleasant working atmosphere.

I also would like to thank my friend and former co-worker Thomas Melzer for the many fruitful discussions during lunch break and his advises and support, both as a scientist and friend.

Most of all, I would like to thank my family: my mother, father, sister and brother for their love, their support and patience.

Contents

Chapter 1

Introduction

This thesis deals with enhanced regression methods based on *canonical correlation analysis* (CCA) applied to machine vision problems. The term "regression" refers to the task of approximating a continuous, real-valued function from noisy observations. Here, we deal with vector-valued (i.e., multiple output variables) functions of vector arguments (i.e., multiple input variables). The proposed methods are used to model the functional relation between two high dimensional signal spaces (random vectors) $\mathbf{x}$ and $\mathbf{y}$, by learning from a set of observations, i.e., corresponding realizations of both vectors. The learned model will then be used for prediction of $\mathbf{y}$ from a new observation of $\mathbf{x}$.

The learning method will be applied to four vision tasks which are outlined in figure 1.1 (details will be given in chapter 4). The first application is an image-based face shape modeling approach using a linear regression model based on CCA. It does not employ an explicit illumination model (in contrast to several shape-from-shading approaches) and allows to recover the structure of the face surface from a single RGB image. The second example uses the same technique for the prediction of near infrared images from normal greyscale images.

In another task, CCA will be applied to non-linear feature extraction for pose estimation. Here, the input signal $\mathbf{x}$ is a greyscale image showing the object, whereas the output signal is low dimensional representation of the pose (for example pan and tilt angle of the camera w.r.t. the object's position in degrees). In the resulting feature space, which captures all regression relevant information, a low dimensional parametric manifold model is build up (see section 4.1). The quality of the predictions of an unseen pose depends on

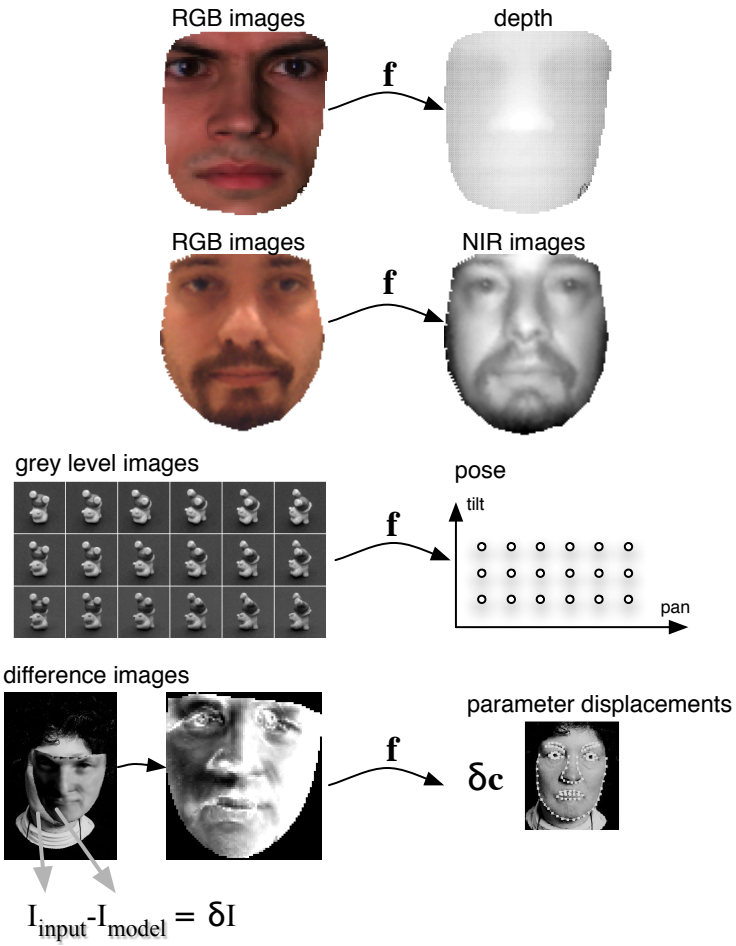


Figure 1.1: Four high dimensional regression problems. The goal is to estimate the from a sample of observations the predictive model $\mathbf{f}$ which allows to predict the signal $\mathbf{y}$ from the signal $\mathbf{x}$. Typically the number of observations in the training set is much smaller than the dimensionality of $\mathbf{x}$ and $\mathbf{y}$ which makes the learning problem ill-posed.

the choice of representation of pose. For example, when using a linear angular scale such as the degree or radian measure, the features extracted by CCA on periodic data perform relatively poor due to the discontinuity at 2π. We will show that regularized *kernel-CCA* - a non-linear generalization of CCA by the use of kernel-methods - can be employed to automatically find an optimal non-linear transformation of pose parameters. In this case the transformation results in a trigonometric representation of pose parameters with four instead of two parameters in the output space (corresponding approximately to sine and cosine of each of the two parameters).

A forth application is matching of an *active appearance model* (AAM) to an image, where regression of the texture residuals on the parameter displacements of the AAM is performed. The texture residuals results from the difference of a synthetic image generated by the AAM (with corresponding parameters) and the input image to which the AAM is matched. Instead of ordinary linear least squares regression, or numeric differentiation approaches to modeling the relation between texture residuals and parameter displacements, CCA is used to select a set of directions which are highly correlated between texture-residual and parameter spaces. Performing a reduced-rank regression on the signal subspaces thus obtained will reduce the variance of the estimator of the update matrix.

All these regression problems share the following characteristics:

- Processing of high dimensional signals: For example, a monochrome image with an image size of 128×128 pixels the dimensionality p of the signal becomes 16384.
- Learning of a functional relation between two high dimensional signals (e.g., two different images modes) from a sample of observations.
- A small number $N << p$ of available training images. N is typically in the range of a few hundred to a few thousand images. In this situation the sample covariance matrix, on which regression analysis is based, is likely to be singular. This makes the learning problem ill-posed, i.e., there are a possibly infinite number of solutions with zero training error, that however fail to capture the functional relation of input and output signal.

Due to the low "observation-to-variable ratio", a standard regression algorithm will cause *overfitting* to the training sample. This means that the resulting function will typically show a low training error rate (in fact zero training error if the system of linear equations is under-determined with $p < N$), but a large error on unseen data, i.e., new images which are not in the training set. In the case of high dimensional data, where many regression parameters have to be estimated from a relatively small sample, training error rate and prediction error rate on unseen samples will typically diverge.

Overfitting is avoided by standard *shrinking methods*, such as *ridge regression* (these methods will be reviewed in section 2.1) or techniques like *principle component regression* (see for example [33]). These methods exploit correlations in the input space, but

neglect correlations in the output space (in the case of multiple output variables). This is disadvantageous if the output space is high dimensional.

In order to improve the prediction error, we will employ CCA for making explicit the regression relevant low-dimensional subspaces and to reduce the effective degrees of freedom. In doing so, we perform model complexity control and avoid overfitting. CCA is a tool for finding directions in two signal spaces that yield maximum correlation between the projections of the original signals onto these directions (see figure 1.2). Thus, like *principal component analysis* (PCA), CCA can be used as a dimensionality reduction method yielding a small number (compared to the *superficial* dimensionality of the original signal space) of linear features. Unlike PCA, however, CCA takes into account the relation between the two signal spaces, which makes it better suited for regression tasks than PCA.

Regression can be performed on the reduced number of features extracted by CCA, thereby the number of independent parameters that are to be learned from the training data (*effective number of parameters*) is reduced in a sensible way. CCA does this by exploiting correlations in input and output variables. For example, the leading *canonical output variates* are those linear combinations of output variables that are best predicted by the input variables, because they show the highest correlation with canonical input variates. The trailing canonical output variates have low correlation with input variables and thus can not be predicted accurately. By dropping these variables, we reduce the variance of the predicted values, and hence may improve the overall prediction accuracy of the model.

There are several strongly related regression methods such as *principal component regression*, *partial least squares*, and especially *reduced-rank regression* (we will discuss the relation between CCA and the latter in section 2.4). An overview of these methods is given in [33] and [8] (where a unifying framework for these methods is presented). While the goal of these methods is inference of a predictive model (predictive function), CCA is a tool for inspection of linear relations between to random vectors. Unlike regression methods, where $\mathbf{x}$ act as input (independent) variables and $\mathbf{y}$ as noisy output (dependent) variables, CCA is symmetric and looks for common latent variables of two (possibly noisy) signals, i.e., $\mathbf{x}$ and $\mathbf{y}$ take on the same role. Thus, CCA can not only be used

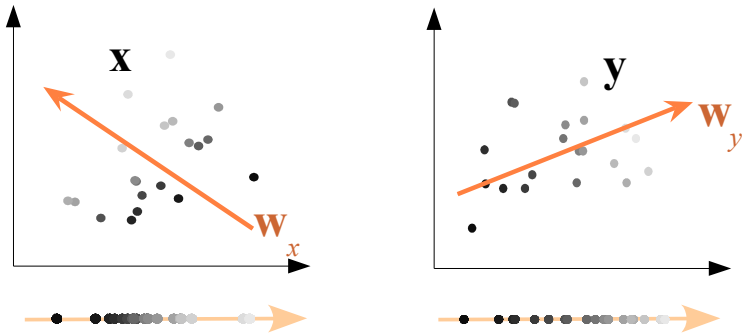


Figure 1.2: CCA finds directions in two signal spaces $\mathbf{x}$ and $\mathbf{y}$, such that the projections onto these directions have maximum correlation. In this illustration the (empirical) CCA is performed on a sample of 20 two-dimensional observations of $\mathbf{x}$ and $\mathbf{y}$. Corresponding observations share the same grey value. The canonical directions found are shown as arrows (vectors) in the original signal space. Projections of the sample onto the one-dimensional subspaces are shown below the 2d plots. The illustration can be interpreted as a schematic plot, where the original signal space is high dimensional ($\gg 2d$) and $\mathbf{w}_x$ and $\mathbf{w}_y$ are the basis of a low-dimensional subspace spanned by a basis of canonical factors successively found by CCA (see section 2.2 for details).

for regression purposes, but whenever we need to establish a relation between two high dimensional signals or sets of measurements. This is particularly beneficial if we assume (in contrast to the standard regression model) that the input signal is also noisy.

1.1 Contributions

The most important individual contributions are:

- Non-linear extension of canonical correlation analysis by the use of kernel methods (kernel-CCA) and enhancement of manifold models for appearance based pose estimation (Sections 3.1 and 4.1).

- Application of CCA for fast matching of active appearance models (see Section 4.2). The proposed method is an alternative training strategy for the update matrix used in the active appearance model.

- Application to predicting depth maps of facial surfaces from RGB color images using regression on feature measurements determined by CCA (Section 4.3). An

experimental comparison of (kernel-)CCA-based regression and standard enhanced regression methods, such as the curds & whey procedure [10] or regression on sparse CCA features [68] is conducted.

1.2 Overview of the document

In chapter 2.2, we review canonical correlation analysis and its relation to reduced-rank regression and ridge regression. In chapter 3.1 we introduce the kernel-based non-linear generalization of CCA (*kernel*-CCA) and discuss the effect of ridge-penalty regularization. In section 4.2 we suggest an enhanced regression method based on CCA which exploits correlations within and between the input and output signal to matching of active appearance models as an alternative training strategy for calculating the update matrix (see [17]). It will be shown, that compared to the standard regression based matching approach, we obtain a speed-up factor of approximately 4. As will be shown in section 4.1, appearance models based on kernel-CCA (manifold models) can be employed for the task of estimating the pose of a 3D object relative to the camera. In section 4.3, we use CCA-based regression for prediction of depth maps of facial surfaces from color images. Conclusions are drawn in section 5.

1.3 Notation

The following uniform notation will be used throughout this thesis. Scalars are indicated by italic letters such as x. Vectors are indicated by lowercase bold letters, such as $\mathbf{w}$ or $\mathbf{x}$. Matrices are indicated by uppercase bold letters, such as $\mathbf{W}$. Elements of vectors or matrices are given using the corresponding italic lowercase letters and the indices of the element. For example, the (i, j) element of matrix $\mathbf{W}$ is accessed by w_{ij}. The same notation will also be used for random quantities (provided the meaning is clear from the the context). Sometimes indices like in $\mathbf{x}_0$ are used to distinguish observations (realizations) from the random vectors $\mathbf{x}$.

Chapter 2

Linear Regression and CCA

2.1 Learning Models of High Dimensional Data

Consider two random vectors $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$ with a joint probability

$$p(\mathbf{x}, \mathbf{y}) \quad = \quad p(\mathbf{x})p(\mathbf{y}|\mathbf{x}). \tag{2.1}$$

The regression model assumes that $\mathbf{y}$ depends on $\mathbf{x}$ by

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{2.2}$$

where $\mathbf{g} : \mathbb{R}^p \rightarrow \mathbb{R}^q$ is a (deterministic) vector-valued function and $\boldsymbol{\epsilon} \in \mathbb{R}^q$ is a random noise vector with $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{\epsilon}) = \Sigma$. It relates the dependent variables $\mathbf{y}$ to a function of the independent variables (*regressors*) $\mathbf{x}$, i.e., a parameterized model for the conditional probability of the form (see [5])

$$\mathbf{g}(\mathbf{x}) = \mathrm{E}_{\mathbf{y}}(\mathbf{y}|\mathbf{x}) = \int \mathbf{y} \mathrm{p}(\mathbf{y}|\mathbf{x}) d\mathbf{y} \quad = \quad \mathbf{f}(\mathbf{x}, \mathbf{w}) \tag{2.3}$$

is deployed, where $\mathbf{f}$ is a parameterized vector-valued function which is completely determined by the choice of the parameter vector $\mathbf{w}$. Given a training set $\mathcal{T} = \{\mathbf{x}_i, \mathbf{y}_i\}, i = 1, \ldots, N$ of N pairs of corresponding observations of the random variables $\mathbf{x}$ and $\mathbf{y}$, the parameters $\mathbf{w}$ are adjusted by minimizing some error criterion on the training set. This error criterion reflects by a single positive number, the *training error*, how well the model fits the training sample (see below). Once the optimal $\mathbf{w}$ has been determined, predictions of $\mathbf{y}$ given a new value of $\mathbf{x}$ can be made by evaluating $\mathbf{f}(\mathbf{x}, \mathbf{w})$.

2.1.1 Loss function

In order to assess the quality of the model fit to the data, a *loss function*

$$L(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i, \mathbf{w})) \tag{2.4}$$

is used, which is defined pointwise and assigns a positive value to the deviation of the prediction $\mathbf{f}(\mathbf{x}_i, \mathbf{w})$ from the observed corresponding output $\mathbf{y}_i$. A common loss function is the squared error loss

$$L(\mathbf{y}, \mathbf{f}(\mathbf{x}, \mathbf{w})) = \|\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{w})\|^2, \tag{2.5}$$

which is the special case of a loss based on the likelihood of the response density of $\mathbf{y}$ at a given $\mathbf{x}$, i.e.

$$L(\mathbf{y}, \theta(\mathbf{x})) = -2 \log \mathrm{p}_{\theta(\mathbf{x})}(\mathbf{y}), \tag{2.6}$$

where θ is a parameter of a probability density depending (conditioned) on $\mathbf{x}$. For the case of the Gaussian additive error model of Eq. 2.2 we have

$$\mathrm{p}_{\theta(\mathbf{x})}(\mathbf{y}) = \mathrm{N}(\mathbf{f}(\mathbf{x}), \Sigma). \tag{2.7}$$

2.1.2 Risk

The expected loss of the trained model with a specific $\mathbf{w}$ on unseen data, i.e. new observations of pairs $\mathbf{x}, \mathbf{y}$ which are not in the training set, is sometimes referred to as *risk*

$$R(\mathbf{w}) = \mathrm{E}_{\mathbf{x}} \mathrm{E}_{\mathbf{y}|\mathbf{x}} L(\mathbf{y}, \mathbf{f}(\mathbf{x}, \mathbf{w})), \tag{2.8}$$

where the expectation is taken over $\mathbf{x}, \mathbf{y}$ and $\mathbf{w}$ is the (fixed) argument. The risk conditioned on a specific input position $\mathbf{x}_0$, i.e.,

$$R(\mathbf{x}_0, \mathbf{w}) = \mathrm{E}_{\mathbf{y}|\mathbf{x}_0} L(\mathbf{y}, \mathbf{f}(\mathbf{x}_0, \mathbf{w})) \tag{2.9}$$

is called *conditional risk*. The optimal approximating function is the one minimizing the risk and is given by parameters

$$\mathbf{w}^* \quad = \quad \arg\min_{\mathbf{w} \in \mathcal{F}} \mathrm{R}(\mathbf{w}). \tag{2.10}$$

In the case of squared error loss and if we use a completely flexible model, minimization of the risk results in $\mathbf{f}(\mathbf{x}, \mathbf{w}^*) = \mathrm{E}_{\mathbf{y}}(\mathbf{y}|\mathbf{x})$, i.e., the model implements the true regression function (see for example [5]).

2.1.3 Training error

Note that in a regression task Eq. 2.10 cannot be solved directly because the probability densities of $\mathbf{x}$ and $\mathbf{y}$ are unknown and thus the expectation of Eq. 2.8 can not be evaluated. However, given a sample, we can calculate the *training error*, which is the average loss on the sample $\mathcal{T}$:

$$R_{emp}(\mathbf{w}, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i, \mathbf{w})). \tag{2.11}$$

Because the training error can be seen as an estimate of the risk of $\mathbf{w}$, it is sometimes referred to as *empirical risk*. We can obtain estimates of $\mathbf{w}^*$ by minimization of the empirical risk, i.e.,

$$\hat{\mathbf{w}}^* \quad = \quad \arg\min_{\mathbf{w}} R_{emp}(\mathbf{w}, \mathcal{T}). \tag{2.12}$$

The estimator $\mathbf{f}(\mathbf{x}_0, \hat{\mathbf{w}}^*)$ of the output at an arbitrary position $\mathbf{x}_0$ is - as a function of the random sample $\mathcal{T}$ - a random variable. From now on, whenever we refer to predictions using a trained model $\mathbf{f}(\mathbf{x}_0, \hat{\mathbf{w}}^*)$, where its parameters have been optimized by minimization of Eq. 2.12 using a sample $\mathcal{T}$, we will denote it by $\mathbf{f}(\mathbf{x}_0; \mathcal{T})$.

The most common loss function is the squared error loss (cf. Eq. 2.5) which leads to the residual sum-of-squares error function (RSS)

$$\mathrm{RSS}(\mathbf{w}, \mathcal{T}) \quad = \quad \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i, \mathbf{w}))^2.$$

This criterion is motivated by the principle of maximum likelihood on the assumption that the training vectors $\mathbf{x}_i, \mathbf{y}_i$ have been drawn independently and that $\mathrm{p}(\mathbf{y}|\mathbf{x})$ is Gaussian (cf. Eq. 2.6). This leads to the *least squares estimator*

$$\hat{\mathbf{w}}^* \quad = \quad \arg\min_{\mathbf{w}} \mathrm{RSS}(\mathbf{w}, \mathcal{T}).$$

2.1.4 Linear regression and the Wiener filter

The linear regression model assumes that

$$\mathrm{E}(\mathbf{y}|\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{w}_0, \tag{2.13}$$

where $\mathbf{W} \in \mathbb{R}^{p \times q}$ is the matrix of regression coefficients and $\mathbf{w}_0 \in \mathbb{R}^q$ is a vector of parameters compensating the difference of the mean of the predictor variables and the

response variables. The model either assumes a linear (affine) regression function or that it can be approximated by a linear function. To simplify the following discussion, we assume $\mathrm{E}(\mathbf{x}) = 0$ and $\mathrm{E}(\mathbf{y}) = 0$ and consequently the vector $\mathbf{w}_0 = 0$.

If we assume a stationary ergodic environment in which $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian, such that the environment can be described by the second-order statistics

- $\mathbf{C}_{xx} = E(\mathbf{x}\mathbf{x}^T)$, which is the covariance of $\mathbf{x}$ and
- $\mathbf{C}_{xy} = E(\mathbf{x}\mathbf{y}^T)$, the cross-covariance of $\mathbf{x}$ and $\mathbf{y}$ and $\mathbf{C}_{yx} = \mathbf{C}_{xy}^T$,

the coefficients $\mathbf{W}$ are given by the *Wiener filter solution*

$$\mathbf{W} = \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}. \tag{2.14}$$

to the linear optimum filtering problem [36]. The Wiener solution corresponds to the *least mean square solution* in the sense that, if we are using squared loss, the risk reaches its minimum:

$$\begin{aligned}
\mathrm{R}(\mathbf{w}) &= \mathrm{E}_{\mathbf{x}}\mathrm{E}_{\mathbf{y}|\mathbf{x}}L(\mathbf{y}, \mathbf{f}(\mathbf{x}, \mathbf{w})) && (2.15)\\
&= \mathrm{E}_{\mathbf{x}}\mathrm{E}_{\mathbf{y}|\mathbf{x}}\|\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{w})\|^2 && (2.16)\\
&= \mathrm{trace}(\mathbf{C}_{yy} - \mathrm{E}(\mathbf{f}(\mathbf{x}, \mathbf{w})\mathbf{f}(\mathbf{x}, \mathbf{w})^T)) && (2.17)\\
&= \mathrm{trace}(\mathbf{C}_{yy} - \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}) && (2.18)\\
&= \mathrm{trace}(\mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T)) = q\sigma^2, && (2.19)
\end{aligned}$$

where $\mathbf{f}_{\mathbf{w}}$ denotes the linear model of Eq. 2.13 with $\mathbf{W}$ given by Eq. 2.14. Note that if $\mathbf{x}, \mathbf{y}$ are jointly Gaussian and $\boldsymbol{\epsilon}$ is uncorrelated then $\mathbf{f}_{\mathbf{w}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$. Otherwise $\mathbf{f}(\mathbf{x})$ is approximated by a linear (affine) function.

2.1.4.1 Canonical coordinates

The Wiener filter can be written in terms of canonical coordinates as follows:

$$\begin{aligned}
\mathbf{W} &= \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\\
&= \mathbf{C}_{yy}^{\frac{1}{2}}\mathbf{C}^T\mathbf{C}_{xx}^{-\frac{1}{2}}\\
&= \mathbf{C}_{yy}^{\frac{1}{2}}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{C}_{xx}^{-\frac{1}{2}}, && (2.20)
\end{aligned}$$

where $\mathbf{C}$ is the *coherence matrix* defined in Eq. A.21. Eq. 2.20 shows that the Wiener filter can be decomposed (left to right) into a whitening transform, a *coherence filter* [58] and a coloring transform which reconstructs the response signal. In the case of pre-whitened variables $\mathbf{x}$ and $\mathbf{y}$ the Wiener filter corresponds to the *coherence filter* $\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{U}^T$.

2.1.5 Linear estimator

Designing the Wiener filter requires knowledge of the second-order statistics (see section 2.1.4), which is normally not available in practice. An estimate of $\mathbf{W}$ can be obtained using the RSS criterion on a sample $\mathcal{T}$ of N observations (realizations) as follows: Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{p \times N}$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N) \in \mathbb{R}^{q \times N}$ be the data matrices containing the corresponding N observations of the sample $\mathcal{T}$ in their N columns. We seek an estimate of the true parameters $\mathbf{W}$ minimizing the residual sum-of-squares error criterion, i.e.,

$$\hat{\mathbf{W}} \;=\; \arg\min \mathrm{RSS}(\mathbf{W})$$

where

$$\begin{aligned} \mathrm{RSS}(\mathbf{W}) &= \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i))^2 \\ &= \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{W}\mathbf{x}_i)^T (\mathbf{y}_i - \mathbf{W}\mathbf{x}_i) \\ &= \mathrm{trace}\left((\mathbf{Y} - \mathbf{W}\mathbf{X})^T (\mathbf{Y} - \mathbf{W}\mathbf{X}) \right). \end{aligned} \tag{2.21}$$

The estimator $\hat{\mathbf{W}}$ is obtained by setting the derivative of Eq. 2.21 to zero and is given by

$$\hat{\mathbf{W}} = \mathbf{Y}\mathbf{X}^T (\mathbf{X}\mathbf{X}^T)^{-1}. \tag{2.22}$$

In the Gaussian setting $\hat{\mathbf{W}}$ corresponds to the maximum-likelihood estimate [5] of $\mathbf{W}$[1].

Eq. 2.22 is called the *ordinary least squares* (OLS) solution to the multivariate linear regression problem and states that in the case of multiple outputs (i.e. $q > 1$), the solution is obtained by separate univariate linear regression on each component of $\mathbf{y}$ (see for

[1] This is even true for non-diagonal noise covariance $\Sigma = \mathrm{E}(\epsilon\epsilon^T)$, as long as Σ does not change among the observations.

example [33]). The predicted values for the training data are

$$\hat{\mathbf{Y}} = \hat{\mathbf{W}}\mathbf{X} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}, \tag{2.23}$$

where the i-th column of $\hat{\mathbf{Y}}$ is $\hat{\mathbf{y}}_i = \hat{\mathbf{W}}\mathbf{x}_i$. The matrix $\mathbf{H} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$ in the above equation is called the "hat" matrix because it puts a hat on $\mathbf{Y}$. The matrix $\mathbf{H}$ corresponds to a projection onto the row space of $\mathbf{X}$ (for the geometrical interpretation see for example [5] or [33]).

2.1.6 Linear basis function models

The linear model shown in section 2.1.4 is a special case of models which are linear in their parameters $\mathbf{w}$. These models are also linear in their input variables $\mathbf{x}$ which imposes a limitation on the model. We can extend the class of models to *linear basis function models* (see for example [6]) by considering linear combinations of fixed nonlinear functions of $\mathbf{x}$, of the form

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}), \tag{2.24}$$

where $\mathbf{W}$ is a $q \times k$ matrix of parameters and

$$\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), ..., \phi_{m-1}(\mathbf{x}))^T \tag{2.25}$$

is the m-vector of basis function activations. Here we use the same set of basis functions to model all output components (which is the most common approach). By using a constant basis function $\phi_0(\mathbf{x}) = 1$ we can allow for any fixed offset in the data. For example in the case of polynomial regression we have

$$\boldsymbol{\phi}(\mathbf{x}) = \left(1, \mathbf{x}, \mathbf{x}^2, ..., \mathbf{x}^{m-1}\right)^T. \tag{2.26}$$

Although these models are linear in their parameters, they are able to implement nonlinear functions of $\mathbf{x}$ of arbitrary complexity by choosing a large number k of suitable basis functions. Such models are referred to as *universal approximators*. The $m \times N$ matrix

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_0(\mathbf{x}_2) & ... & \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & ... & \phi_1(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m-1}(\mathbf{x}_1) & \phi_{m-1}(\mathbf{x}_2) & ... & \phi_{m-1}(\mathbf{x}_N) \end{pmatrix}. \tag{2.27}$$

is called *design matrix* and takes on the role of the "transformed" data matrix holding the m-dimensional feature vectors in its columns. The least squares (maximum likelihood) estimator of $\mathbf{W}$ becomes

$$\hat{\mathbf{W}} = \mathbf{Y}\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}. \tag{2.28}$$

Predictions of training data are given by (cf. 2.23)

$$\hat{\mathbf{Y}} = \mathbf{Y}\mathbf{H}, \tag{2.29}$$

with $\mathbf{H} = \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}$ being the hat matrix.

2.1.7 Model selection

In most situations, the dependency of input $\mathbf{x}$ and output $\mathbf{y}$ is unknown and the parametric form of the model $\mathbf{f}(\mathbf{x}, \mathbf{w})$ has to be specified as part of the learning process prior to adapting its parameters. Once the parametric form is chosen the optimal $\mathbf{w}^*$ has to be determined from the sample. The trained model should offer good generalization, i.e., it should minimize the risk rather than the training error. Very flexible models (e.g., large m) can achieve a low (or zero) training error by fitting the noise in $\mathbf{y}$, but consequently will fail to capture the deterministic, functional dependency $\mathbf{g}(\mathbf{x}) = \mathrm{E}(\mathbf{y}|\mathbf{x})$ between inputs $\mathbf{x}$ and outputs $\mathbf{y}$. This phenomenon is called *overfitting* and occurs when we try to fit too a complex model (a set of functions with too large capacity) to a finite sample.

In theory, the problem of overfitting could be addressed by choosing an extremely flexible model (universal approximator) and providing an infinite amount of (iid) training data, which is equivalent to the case where the joint density function $p(\mathbf{x}, \mathbf{y})$ is known. Clearly, if $p(\mathbf{x}, \mathbf{y})$ is known, model selection could be performed by minimizing Eq. 2.37, because then all expectation operators can be evaluated. In fact, if $p(\mathbf{x}, \mathbf{y})$ was known the regression function can be determined immediately from Eq. 2.8. For example for squared loss and when using a completely flexible model, it can be shown (see [6]) that by minimizing the risk we obtain $\mathbf{f}(\mathbf{x}, \mathbf{w}^*) = \mathrm{E}(\mathbf{y}|\mathbf{x})$.

In practice, $p(\mathbf{x}, \mathbf{y})$ and thus the regression function are unknown and one is only given a finite sample. Without additional assumptions, the learning problem is inherently ill-posed, i.e., there is a possibly infinite number of functions of varying complexity with minimal (or zero) training error. To obtain a useful, unique solution, the model complexity has to be adapted to the size of the training set.

2.1.8 Regularization

This a priori knowledge may determine the choice of type and number m of basis functions (e.g., polynomials of degree $m-1$) or it can be given in form of smoothness assumptions. For example, in the regularization framework there are parameters that govern the strength or influence of such a priori assumptions (e.g., "how smooth") which are referred to as regularization parameters. Other methods try to estimate the expected risk from the sample itself in order to achieve effective model selection.

Regularization methods add a penalty functional term to the error function to be minimized during training:

$$\mathrm{R}_{pen}(\mathbf{w}, \lambda, \mathcal{T}) = \mathrm{R}_{emp}(\mathbf{w}, \mathcal{T}) + \lambda \mathrm{P}(\mathbf{w}) \tag{2.30}$$

This penalty associates large positive values to complex functions and small values to simple functions, such that solutions are restricted to functions of limited complexity. Penalty functionals can be constructed for a wide range of models in any dimension, imposing the desired structure on the set of functions that can be implemented by the model.

In a linear model framework, more complex functions typically have larger weight magnitudes, so the penalty term corresponds to a function of the norm of the parameter vector, as will be seen later.

2.1.9 Regularized least squares: ridge regression

Ridge regression [34] is a linear regression method for a single response which uses the penalized RSS criterion

$$\mathrm{RSS}(\lambda) \;=\; (\mathbf{y} - \mathbf{w}^T\mathbf{\Phi})(\mathbf{y} - \mathbf{w}^T\mathbf{\Phi})^T + \lambda \mathbf{w}^T\mathbf{w}, \tag{2.31}$$

where $\mathbf{y}$ is here the *row* vector of N sample responses and $\mathbf{w}$ is the p-vector of regression parameters. $\lambda \geq 0$ is the complexity parameter (*ridge parameter*) that controls the influence of the penalty term. The estimator $\hat{\mathbf{w}}_{ridge} = \arg\min_{\mathbf{w}} \mathrm{RSS}(\lambda)$ biases the coefficient estimates towards smaller absolute values and discourages dispersion among their values (see section 2.4.4). In this case the penalty term corresponds to the sum-of-squares of the components of $\mathbf{w}$. This form of regularizer has the advantage that the sum of the RSS

function and the penalty term is a quadratic function of $\mathbf{w}$, so that the solution is given in closed form

$$\hat{\mathbf{w}}_{ridge} = (\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{\Phi}\mathbf{y}. \tag{2.32}$$

In the case of multiple responses we can perform separate ridge regression on each individual response (i.e., using the q rows of the sample responses $\mathbf{Y}$ separately) obtaining the q rows of $\mathbf{W}$ with separate ridge parameters $\lambda_k, k = 1, \ldots, q$. Sometimes, a single common ridge parameter is used in which case the criterion can be written as

$$\text{RSS}(\lambda) = \text{trace}\left((\mathbf{Y} - \mathbf{W}\mathbf{\Phi})(\mathbf{Y} - \mathbf{W}\mathbf{\Phi})^T\right) + \lambda\text{trace}(\mathbf{W}\mathbf{W}^T). \tag{2.33}$$

Regularized least squares can be interpreted in the bayesian framework, where the penalized RSS criterion corresponds to the log of the posterior distribution given by the sum of the log likelihood (RSS function) and the log of the prior (penalty term) [5].

2.1.10 Effective number of parameters

In the case of linear basis function models, the parameters $\mathbf{W}$ are a linear combination of the training output data $\mathbf{y}_i$ (see Eq. 2.32). The predictions of the training predictors $\mathbf{x}_i$ are

$$\hat{\mathbf{Y}} = \mathbf{Y}\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{\Phi} \tag{2.34}$$

$$= \mathbf{Y}\mathbf{H}_\lambda, \tag{2.35}$$

where the hat matrix of Eq. 2.29, now becomes a $N \times N$ *smoother matrix* $\mathbf{H}_\lambda$ (see for example [33]).

The complexity of the linear basis function model is related to the number of its independent parameters, which are in turn related to the number of basis functions used. In the case of ordinary least squares linear regression on $\mathbf{x}$ the number of parameters depends on the superficial dimensionality of the data. For example, consider a multiple output regression model of Eq. 2.13. In this case the overall number of parameters clearly depends on the dimensionality of $\mathbf{x}$ and $\mathbf{y}$. When dealing with high dimensional data, where the sample size is typically small in relation to the number of parameters, we can search for and exploit correlations between the variables in order to reduce the number of parameters and thus adjust the model complexity.

For linear basis function models, the *effective number of parameters* (sometimes referred to as *effective degrees of freedom*, EDOF) corresponds to the trace of the smoother matrix

$$\text{trace}(\mathbf{H}_\lambda) \quad = \quad \text{trace}(\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{\Phi}). \tag{2.36}$$

It can be shown by eigen-decomposition of $\mathbf{\Phi}\mathbf{\Phi}^T$ (see for example [33]), that increasing the regularization parameter λ has the effect of shrinking coefficients in directions of small variance in the input feature space spanned by the columns of $\mathbf{\Phi}$, which results in a smoother fit. These directions are those for which the RSS function is relatively insensitive to variations of $\mathbf{W}$ and so - following the principle of *Ockham's razor*[1] - these parameters are set to a small value. It is easy to show that the quantity $\text{trace}(\mathbf{H}_\lambda)$ will lie in the range between 0 and k (the number of basis functions).

Note that the smoother matrix only takes into account the covariance of the input feature space. As will be shown in later sections, that canonical correlation analysis can be employed to find directions of maximum correlation between input and output space and that these correlations can be exploited to reduce the EDOF in a sensible way and thus improve the prediction accuracy.

2.1.11 Expected risk

In order to choose the right model complexity for a given sample size N, we have to consider the average performance of a model when it is repeatedly trained with different samples $\mathcal{T}$ of size N. More formally, given a estimation method for $\mathbf{w}$, the quantity to be minimized by the chosen class of functions is the *expectation* of the risk *taken over all possible training samples* of size N, i.e.,

$$\text{E}_\mathcal{T} R(\hat{\mathbf{w}}) = \text{E}_\mathbf{x}\text{E}_\mathbf{y}\text{E}_\mathcal{T} L(\mathbf{y}, \mathbf{f}(\mathbf{x}; \mathcal{T})), \tag{2.37}$$

where now the expectation is taken over any variable that is random, including $\hat{\mathbf{w}}$ as it depends on $\mathcal{T}$ via Eq. 2.12. The model complexity (appropriate class of functions) should

[1] *Pluralitas non est ponenda sine neccesitate* ("plurality should not be posited without necessity"): According to the principle of *Ockham's razor* we should eliminate all assumptions in an explanatory hypothesis which make no difference in its observable predictions. In the context of machine learning this translates as "models should be no more complex than is sufficient to explain the data", i.e., if we have more than one predictive function explaining the training data (making the same prediction in the mean, when trained with different samples), we should select the least complex function.

be chosen such that Eq. 2.37 is minimal. We can assess the expected prediction risk at a specific position $\mathbf{x}_0$ by conditioning on the input, i.e.,

$$\mathrm{E}_{\mathcal{T}}\mathrm{E}_{\mathbf{y}|\mathbf{x}_0} L(\mathbf{y}, \mathbf{f}(\mathbf{x}_0; \mathcal{T})). \tag{2.38}$$

2.1.12 Squared loss

If we use squared error loss (c.f. Eq. 2.5) we can decompose the expected conditional risk as

$$\mathrm{E}_{\mathcal{T}}\mathrm{E}_{\mathbf{y}|\mathbf{x}_0}(\|\mathbf{y} - \mathbf{f}(\mathbf{x}_0; \mathcal{T})\|^2) \;=\; \mathrm{E}_{\mathcal{T}}(\|\mathrm{E}(\mathbf{y}|\mathbf{x}_0) - \mathbf{f}(\mathbf{x}_0; \mathcal{T})\|^2) + \mathrm{trace}(\Sigma).$$

The first term of the right hand side corresponds to the *mean squared error* (MSE) of the estimator $\hat{\mathbf{f}}(\mathbf{x}_0, \mathcal{T})$. The second term is the variance of the target values $\mathbf{y}$ around its true mean $\mathrm{E}(\mathbf{y}|\mathbf{x}_0)$ and can not be avoided. It is therefore called *irreducible error*. The MSE is a pointwise measure, because we condition on $\mathbf{x}_0$. An optimal estimator is one for which the MSE becomes minimal at every given input position $\mathbf{x}$. This is accounted for by the *overall* expected prediction risk given by Eq. 2.37, which is a global error measure taking into account the density of input $\mathbf{x}$. Our goal is to find a model (estimator) of optimal complexity which minimizes this error measure.

2.1.13 Bias and variance

The MSE is of particular importance because it can be recast as

$$\mathrm{MSE} \;=\; \underbrace{\|\mathrm{E}_{\mathcal{T}}\mathbf{f}(\mathbf{x}, \mathcal{T}) - \mathbf{g}(\mathbf{x})\|^2}_{\mathrm{Bias}^2(\mathbf{f}(\mathbf{x},\mathcal{T}))} + \underbrace{\mathrm{E}_{\mathcal{T}}\|\mathbf{f}(\mathbf{x}, \mathcal{T}) - \mathrm{E}_{\mathcal{T}}\mathbf{f}(\mathbf{x}, \mathcal{T})\|^2}_{\mathrm{Var}(\mathbf{f}(\mathbf{x},\mathcal{T}))}. \tag{2.39}$$

The first term on the right side is the *squared bias* which is the amount by which the expected estimate differs from the true mean. The second term is the *variance*, the expected squared deviation of the estimate around its mean. When selecting the model of optimal complexity, there is a tradeoff between squared bias and variance. In figure 2.1 a biased model is used, since the regression function $\mathbf{g}$ is not included in the model space, meaning that the model is too simple to implement $\mathbf{g}$. The more flexible the model (estimator) is, the lower is its bias. At the same time, due to its flexibility it will fit the sample well, which generates higher variance (according to the variability between different samples).

On the other hand, if we use a highly biased model, the variance will be lower. Figure 2.2 shows the squared bias and variance using models of varying complexity (linear, quadratic and a polynomial of degree 4).

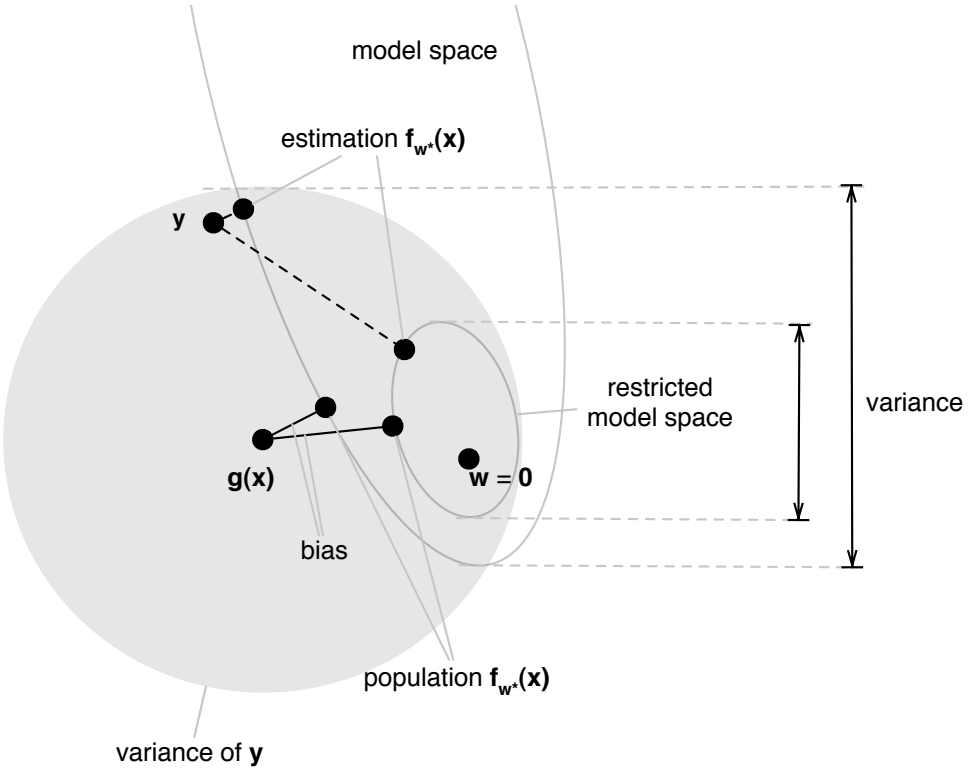


Figure 2.1: In this schematic figure (adopted from [33]), we assume the sample is fitted by the model with parameters $\mathbf{w}^*$ optimized by the training algorithm. We repeatedly take (iid) samples of size N. The output values $\mathbf{y}$ in the sample will vary within the gray circular area. The model is capable of implementing functions which allow predictions within the model space. Some of the samples may be fitted with zero training error, others may have positive training error. If the regression function, which equals the pointwise conditional mean $\mathbf{g}(\mathbf{x}) = \mathrm{E}(\mathbf{y}|\mathbf{x})$, can be fitted by the model, its estimates are unbiased, i.e., the $\mathrm{E}_{\mathbf{w}}\mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{g}(\mathbf{x})$ (population $\mathbf{f}(\mathbf{x}, \mathbf{w})$ denotes $\mathrm{E}_{\mathbf{w}}\mathbf{f}(\mathbf{x}, \mathbf{w})$). This figure shows a biased model, since $\mathbf{f}(\mathbf{x})$ lies outside the model space. It also shows the effect of regularization and shrinking methods: The model space shrinks towards smaller parameter values, whereby the bias is increased. On the other hand the expected loss of predictions $\mathbf{f}(\mathbf{x}, \mathbf{w}^*)$ of a trained model is reduced, due to smaller prediction variance.

2.1.14 Approximations of the expected risk of linear estimators

The prediction capability of the trained model is related to the risk given by Eq. 2.8, which quantifies for a specific vector of parameter values its performance on new test data not encountered in the training set. The expected risk given by Eq. 2.37 on the other hand quantifies the expected prediction performance on test data if the model is repeatedly trained with i.i.d. samples of size N. The latter is important to select the model of suitable complexity.

Since the expected risk can not be evaluated, practical methods for model selection rely on approximations of the expected risk based on the sample itself. For ordinary linear least squares regression, we can obtain an approximation as follows: Let

$$\begin{aligned} \mathbf{G} &= \mathrm{E}_{\mathbf{x}}(\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T) && (2.40)\\ &= \mathbf{C}_{xy}^T\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy} && (2.41)\\ &= \mathbf{C}_{yy} - \mathbf{\Sigma} && (2.42) \end{aligned}$$

and let $\mathbf{h}(\mathbf{x}) = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{x}$ so that $\hat{\mathbf{g}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{Y}\mathbf{h}(\mathbf{x})$. Then, if we condition on the design $\mathbf{X}$ and assume that only ϵ is random, we can write

$$\begin{aligned} \mathrm{E}_{\mathbf{Y}|\mathbf{X}}\left[\frac{1}{N}\sum_{i=1}^{N}\hat{\mathbf{g}}(\mathbf{x}_i)\hat{\mathbf{g}}(\mathbf{x}_i)^T\right] &= \frac{1}{N}\mathrm{E}_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}\mathbf{H}\mathbf{Y}^T)\\ &= \frac{p}{N}\mathbf{\Sigma}. && (2.43) \end{aligned}$$

Assuming that the sample mean and covariance of the input observations $\mathbf{x}_i$ are equal to the true mean and covariance, i.e.

$$\begin{aligned} \mathrm{E}_{\mathcal{T}}\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i &= \mathrm{E}(\mathbf{x}), && (2.44)\\ \mathrm{E}_{\mathcal{T}}\frac{1}{N}\mathbf{X}\mathbf{X}^T &= \mathrm{E}(\mathbf{x}\mathbf{x}^T) && (2.45) \end{aligned}$$

as a consequence of Eq. 2.43 we can write

$$\begin{aligned} \mathrm{E}(\hat{\mathbf{g}}(\mathbf{x})\hat{\mathbf{g}}(\mathbf{x})^T) &= \frac{p}{N}\mathbf{\Sigma} + \mathbf{G} && (2.46)\\ &= \mathbf{C}_{yy} + \left(\frac{p}{N} - 1\right)\mathbf{\Sigma} && (2.47) \end{aligned}$$

Finally, the overall expected risk (ER) of the OLS estimator can be written as (cf. Eq. 2.38)

$$\begin{aligned} \mathrm{ER}_{\hat{\mathbf{g}}} &= \mathrm{trace}\left[\boldsymbol{\Sigma} + \mathrm{E}(\hat{\mathbf{g}}(\mathbf{x})\hat{\mathbf{g}}(\mathbf{x})^T) - 2\mathrm{E}(\mathbf{g}(\mathbf{x})\hat{\mathbf{g}}(\mathbf{x})^T) + \mathrm{E}(\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T)\right] \\ &= \mathrm{trace}\left[\boldsymbol{\Sigma}\left(1 + \frac{p}{N}\right)\right], \end{aligned} \tag{2.48}$$

where we have used $\mathrm{E}(\mathbf{g}(\mathbf{x})\hat{\mathbf{g}}(\mathbf{x})^T) = \mathrm{E}(\hat{\mathbf{g}}(\mathbf{x})\mathbf{g}(\mathbf{x})^T) = \mathrm{E}(\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T) = \mathbf{G}$. Because the above assumption holds if we condition on the design (as in Eq. 2.43), but not in general, Eq. 2.48 is referred to as *in-sample* prediction error. If the assumption does not hold, then Eq. 2.48 can be regarded as a simplifying approximation of the true expected risk.

2.1.14.1 Optimism of the training error rate

The training error itself is not a good measure of generalization capability because it typically underestimates the ER due to the fact that the same data (sample) is used to fit the model and assess the prediction error. The discrepancy between the expected training error of the estimator and its ER can be approximated by

$$\mathrm{Op} = \mathrm{ER}(\hat{\mathbf{g}}) - \mathrm{E}_{\mathcal{T}}\mathrm{R}_{emp}(\hat{\mathbf{g}}), \tag{2.49}$$

where we can use Eq. 2.48 as an approximation of the ER. This quantity is called the *optimism* of the training error rate [33]. For the case of linear models and squared loss we have

$$\mathrm{E}_{\mathcal{T}}\mathrm{R}_{emp}(\hat{\mathbf{f}}) = \frac{1}{N}\mathrm{E}_{\mathcal{T}}\mathrm{trace}((\mathbf{Y} - \hat{\mathbf{W}}\mathbf{X})(\mathbf{Y} - \hat{\mathbf{W}}\mathbf{X})^T) \tag{2.50}$$

$$= \frac{1}{N}\mathrm{E}_{\mathcal{T}}\mathrm{trace}(\mathbf{Y}\mathbf{Y}^T - \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X})^{-1}\mathbf{X}\mathbf{Y}^T) \tag{2.51}$$

$$= \mathrm{Bias}^2 + (1 - \frac{p}{N})\mathrm{trace}(\boldsymbol{\Sigma}) \tag{2.52}$$

and thus the optimism is

$$\mathrm{Op} = \frac{2p}{N}\mathrm{trace}(\boldsymbol{\Sigma}). \tag{2.53}$$

Several methods for model selection rely on the in-sample approximation to assess analytically the prediction capability on independent (unseen) test data, among which are the C_p statistic, the *Akaike information criterion*(AIC) and the *Bayesian information criterion*(BIC). These methods can be employed for the class of linear fitting methods for which the predictions can be written in the form of Eq. 2.34.

Thereby, $\mathbf{\Sigma}$ in Eq. 2.53 is estimated using a low-bias model, e.g. by $\frac{1}{N}\mathbf{Y}\mathbf{Y}^T$. The number of parameters p in Eq. 2.53 is replaced by $d = \text{trace}(\mathbf{H})$, which is referred to as *effective number of parameters* (see section 2.1.10). For OLS predictions, $\mathbf{H} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$. If a ridge-penalty with common ridge parameter is used (see section 2.1.9), i.e., $\mathbf{H}_\lambda = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{X}, \lambda > 0$ then $d = \text{trace}(\mathbf{H}_\lambda) < p$.

We have derived the optimism as Eq. 2.53 for a linear fit under squared error loss. For a general (nonlinear) fitting method (and different loss functions), it can be shown easily that the optimism becomes

$$\text{Op} = \sum_{i=1}^{N} \text{Cov}_{\mathcal{T}}(\mathbf{y}_i, \hat{\mathbf{g}}(\mathbf{x}_i)), \tag{2.54}$$

which shows that the optimism becomes larger the stronger the training sample affects its own prediction [33].

2.1.15 Bayesian regression

In the last years, regression methods based on Bayesian inference have become increasingly popular. In the Bayesian inference paradigm the parameters $\mathbf{w}$ are treated as random variables. The distribution of $\mathbf{w}$ is inferred using Bayes' rule. The hyperparameters which control the model complexity (e.g., the ridge parameter λ) emerge naturally as parameters of the prior distribution which expresses the "degree of belief" over the values that $\mathbf{w}$ might take. The Bayesian approach allows marginalization, i.e., integrating out all irrelevant parameters, and thus determine models which generalize well, without having to cross-validate the hyperparameters. This is done by using proper priors for these parameters. Even in the case where *uninformative priors* are used (flat priors), the Bayesian approach automatically avoids models which are too complex [67].

One disadvantage is that the calculation of the integrations over the irrelevant variables is in most cases analytically intractable. Thus, practical Bayesian approaches rely on approximation strategies, e.g., by using a maximum likelihood approximation for those integrations, which are not analytically tractable [67].

Further, it is possible to obtain sparsity within the Bayesian framework, by using multiple independent hyperparameters for each component of $\mathbf{w}$. This results in a "sparse" prior which is equivalent to regularization with the term $\sum_k \log|w_k|$. A special sparse

Bayesian regression model is the *Relevance Vector Machine* [66], which uses the parameterization (in dual space) together with kernel functions like the support vector machine.

2.2 Canonical Correlation Analysis

Canonical correlation analysis (CCA) is a very powerful and versatile tool that is especially well suited for relating two sets of measurements (signals). Like *principal components analysis* (PCA), CCA also reduces the dimensionality of the original signals, since only a few factor-pairs are normally needed to represent the relevant information; unlike PCA, however, CCA takes into account the relationship between the two signals (in the correlation sense), which makes them better suited for regression tasks than PCA. Furthermore, CCA takes advantage of the correlations between the response variables to improve predictive accuracy [10].

CCA, in particular, has some very attractive properties (for example, it is invariant w.r.t. affine transformations - and thus scaling - of the input variables) and can not only be used for regression purposes, but whenever one needs to establish a relation between two sets of measurements (e.g., finding corresponding points in stereo images [8]). In signal processing, CCA is used for optimal reduced-rank filtering [36], where the goal is data reduction, robustness against noise and high computational efficiency. Geometrically interpreted, CCA measures the angles between two linear subspaces and canonical correlations play the same role as cosines of principal angles [63] between the subspaces (see for example [57]). In [2] it is shown that CCA reveals how well two input variables (i.e. two sets of vectors) are represented by a common source variable (latent variable). CCA has been successfully applied to pattern classification [51], appearance based 3D pose estimation [47] and stereo vision [8]. In [39] CCA is used for image-set classification with a discriminative transformation for images-set based object recognition. Extensions of CCA to that of high-order tensors with applications to video sequence analysis have been proposed in [40]. This latter approach allows a pair-wise analysis of holistic action volumes in which both spatial and temporal information are important.

There are a couple of extensions of CCA used in the computer vision community, among which are kernel-based nonlinear generalizations which will be which will discussed thoroughly in section 3.1, tensor versions and sparse-CCA. The latter two exten-

sions will be reviewed in section 2.3.

2.2.1 Definition

Given two zero-mean random variables $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$, CCA finds pairs of directions $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projections $x = \mathbf{w}_x^T\mathbf{x}$ and $y = \mathbf{w}_y^T\mathbf{y}$ (in the context of CCA, the projections x and y are also referred to as *canonical variates*). More formally, the directions can be found as maxima of the function

$$\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} = \frac{E[\mathbf{w}_x^T\mathbf{x}\mathbf{y}^T\mathbf{w}_y]}{\sqrt{E[\mathbf{w}_x^T\mathbf{x}\mathbf{x}^T\mathbf{w}_x]E[\mathbf{w}_y^T\mathbf{y}\mathbf{y}^T\mathbf{w}_y]}},$$

$$\rho = \frac{\mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y}{\sqrt{\mathbf{w}_x^T\mathbf{C}_{xx}\mathbf{w}_x\mathbf{w}_y^T\mathbf{C}_{yy}\mathbf{w}_y}}. \tag{2.55}$$

whereby $\mathbf{C}_{xx} \in \mathbb{R}^{p\times p}$ and $\mathbf{C}_{yy} \in \mathbb{R}^{q\times q}$ are the *within-set covariance matrices* of $\mathbf{x}$ and $\mathbf{y}$, respectively, while $\mathbf{C}_{xy} \in \mathbb{R}^{p\times q}$ denotes their *between-set covariance matrix*. A number of at most $k = \min(p, q)$ factor pairs $\langle \mathbf{w}_x^i, \mathbf{w}_y^i \rangle, i = 1, \ldots, k$ can be obtained by successively solving

$$\mathbf{w}^i = (\mathbf{w}_x^{iT}, \mathbf{w}_y^{iT})^T = \arg\max_{(\mathbf{w}_x^i, \mathbf{w}_y^i)} \{\rho\} \tag{2.56}$$

subject to

$$\rho(\mathbf{w}_x^j, \mathbf{w}_y^i) = \rho(\mathbf{w}_x^i, \mathbf{w}_y^j) = 0 \;\; j = 1, \ldots, i-1$$

2.2.2 Rayleigh quotient formulation of CCA

The solution to this optimization problem can be found using a formulation of Eq. 2.56 by a Rayleigh quotient [8]. Let

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{0} \end{pmatrix}, \; \mathbf{B} = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix}. \tag{2.57}$$

It can be shown [8] that the stationary points $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ of ρ (i.e., the points satisfying $\nabla\rho(\mathbf{w}^*) = \mathbf{0}$) coincide with the stationary points of the *Rayleigh quotient*:

$$r = \frac{\mathbf{w}^T\mathbf{A}\mathbf{w}}{\mathbf{w}^T\mathbf{B}\mathbf{w}}, \tag{2.58}$$

and thus, by virtue of the *generalized spectral theorem* [22], can be obtained as solutions (i.e., eigenvectors) of the corresponding generalized eigen-problem:

$$\mathbf{Aw} = \mu\mathbf{Bw}. \tag{2.59}$$

The extremum values $\rho(\mathbf{w}^*)$, which are referred to as *canonical correlations*, are equally obtained as the corresponding extremum values of Eq. 2.58 or the eigenvalues of Eq. 2.59, respectively, i.e., $\rho(\mathbf{w}^*) = r(\mathbf{w}^*) = \mu(\mathbf{w}^*)$.

Given n pairs of mean-normalized observations $(\mathbf{x}_i^T, \mathbf{y}_i^T)^T \in \mathbb{R}^{p+q}$, and data matrices $\mathbf{X} = (\mathbf{x}_1..\mathbf{x}_n) \in \mathbb{R}^{p\times n}$, $\mathbf{Y} = (\mathbf{y}_1..\mathbf{y}_n) \in \mathbb{R}^{q\times n}$, we obtain the estimates for the covariance matrices $\mathbf{A}, \mathbf{B}$ in Eq. 2.57 as

$$\hat{\mathbf{A}} = \frac{1}{n}\begin{pmatrix} \mathbf{0} & \mathbf{XY}^T \\ \mathbf{YX}^T & \mathbf{0} \end{pmatrix},\ \hat{\mathbf{B}} = \frac{1}{n}\begin{pmatrix} \mathbf{XX}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{YY}^T \end{pmatrix} \tag{2.60}$$

If the mean was estimated from the data, we have to replace n by $n-1$ in both equations.

2.2.3 CCA and linear regression

It is instructive to compare CCA to the full-rank solution (the OLS solution, cf. 2.22) of standard multivariate linear regression (MLR), *ordinary (linear) least squares regression* (OLS), where the regression parameters $\mathbf{W}$ are given by the *Wiener filter* (cf. Eq. 2.14): $\mathbf{W} = E[\mathbf{xx}^T]^{-1}E[\mathbf{xy}^T] = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}$.

When comparing the Wiener filter with the derivation of CCA by *singular value decomposition* (see appendix A.4), we see that in contrast to MLR, the CCA solution is computed using only the leading singular vectors of the cross-correlation matrix of pre-whitened variables $\mathbf{x}, \mathbf{y}$ which are made explicit by SVD. Thus, CCA can be used to compute a (reduced) rank-n regression parameter matrix by using only $n < k$ factor pairs. Thereby, in contrast to standard multivariate regression CCA takes advantage of the correlations between the response variables to improve predictive accuracy [10]. Note also that in contrast to the Wiener filter the additional pre-whitening of $\mathbf{y}$ makes CCA invariant w.r.t. scaling of $\mathbf{x}, \mathbf{y}$.

The relation to MLR and how CCA can be used to enhance standard MLR procedures will be discussed in detail in section 2.4.

2.3 Extensions of CCA

2.3.1 CCA in tensor space

While conventional CCA makes explicit the correlation between two sets of vectors (observations), i.e. matrices with common set of columns, Harshman [32] considers the generalization of CCA to that of general N-way arrays that share one or more subscripts in common. This idea was later used by Kim et al. [39] [40] for the task of video volume tensor analysis for action categorization. In contrast to the method proposed in [32], which obtains canonical weight vectors (referred to as single-shared-mode by Kim et al.), in [39] [40] a general concept of multiple-shared-modes (joint-shared-modes) is proposed, which allows to obtain canonical tensors as well.

The basic idea of the CCA generalization to tensors (tensor CCA, TCCA) is as follows: If we interpret for example image sequences as 3D video cubes (3-way tensors), where two axis represent image coordinates (spatial domain) and the third axis represents the time domain, then we can calculate measures for the similarity of two sequences by calculating canonical vectors along all three axis. The corresponding canonical factors are measures of similarity (cosines of the canonical angles between the respective subspaces). In the analysis of actions captured in image sequences, the ordering of the images in the video volume is of particular importance. This temporal information is lost, if we perform standard CCA of the set of images, because it is invariant w.r.t. the ordering of the observations.

In [42], CCA of tensor spaces is used for the recovery of facial depth maps (similar to the application presented in section 4.3). Experimental results, which are superior to that of standard CCA, are reported. A possible explanation is, that the number overall parameters estimated by tensor CCA is smaller than the number of parameters obtained by CCA (lower dimensionality), which might lead to improved predictive accuracy in the case of a relatively small training set. For details, the reader is referred to the publications mentioned above.

2.3.2 Sparse CCA

As we have discussed in Section A.3, when estimating the canonical factors from data matrices $\mathbf{X} = (\mathbf{x}_1..\mathbf{x}_n) \in \mathbf{R}^{p\times n}$, $\mathbf{Y} = (\mathbf{y}_1..\mathbf{y}_n) \in \mathbf{R}^{q\times n}$ with N observations, with $N < p+q$, there are $p+q-N$ linearly independent solution vectors, making CCA ill-posed. A possible remedy is ridge regularization, which will be discussed in more detail in section 2.4.4. Ridge regularization shrinks the solution vectors $\mathbf{w}_x$ and $\mathbf{w}_y$ by imposing a penalty on their size. This involves the use of the l^2 norm in the penalty term. Here, we will consider cardinality constraints in the formulation of CCA leading to sparse solution vectors $\mathbf{w}_x$ and $\mathbf{w}_y$ in the sense that only some of the coefficients of the solution are non-zero. Sparsity is an attractive concept, allowing to control model complexity and perform implicit feature selection, i.e. finding a small number of the most meaningful input variables.

In figure 2.3(a) a straight line describes all $\mathbf{w}_x$ satisfying Eq. A.15 (for the case of $N < p+q$) for a fixed $\mathbf{w}_y$. Ridge penalization leads to a unique solution with minimal $\|\mathbf{w}_x\|^2$ (black dot), where all coefficients are shrunk and the energy of $\mathbf{w}_y$ is spread over all coefficients. A cardinality constraint penalizing nonzero components is implemented using the l^0-norm $\|\mathbf{w}_x\|^0$(the number of nonzero coefficients of $\mathbf{w}_x$). However, the variational formulation of this CCA problem [59] given by

$$\max(\mathbf{w}^T\mathbf{A}\mathbf{w} : \mathbf{w}^T\mathbf{B}\mathbf{w} = 1, \|\mathbf{w}\|^0 \leq k), \tag{2.61}$$

with $\mathbf{A}, \mathbf{B}$ given by Eq. 2.60, is non-convex, NP-hard and thus intractable. To make this problem feasible, usually the l^1-norm approximation is used, where $\|\mathbf{w}\|^0$ is replaced by $\|\mathbf{w}\|^1$ (see figure 2.3(c)). In the context of regression this kind of penalization is called the *lasso* (see, e.g., [33], page 64).

There are several formulations of *sparse* generalized eigen-problem solvers based on the l^1-norm approximation, e.g., an algorithm using elastic net [75], or d.c. (difference of convex functions) programming [59], which can be employed for sparse CCA. In [38], the non-convex optimization problem is broken into a large number separate convex problems. The algorithm is used for the localization of visual events associated with sound in a video, where the assumption is, that these visual events are spatially sparse, i.e. a relatively small group of pixels. In [68], sparse CCA is employed for building a vocabulary of predictive semantic concepts.

2.4 Enhanced Regression Methods and Canonical Coordinates

The ordinary least squares (OLS) estimator has several (related) drawbacks, which were discussed in the previous sections:

- Correlations in the response variables are ignored. The EDOF grows with the number of predictor variables and the optimism grows with the EDOF and the number of response variables. This leads to overfitting in the case of limited and noisy training data. As the example in section 2.4.1 shows, this might lead to poor prediction accuracy, when training data is limited.

- When the number of predictors is large, the outcome is difficult to interpret. We are often interested in finding a smaller number of parameters with the strongest effect.

- Often the functional relation between two high dimensional signals is inherently lower dimensional. Section 2.4.1 gives an example, where the intrinsic dimensionality of the response signal is one. The knowledge of the intrinsic dimensionality k allows to discard meaningless dimensions (caused by noise) by using a rank-k regression model, leading to higher predictive accuracy.

- Channel noise reduces the true dimensionality of the regression function: Assuming that the noise is zero-mean and uncorrelated with the input signal, it is shown in [21] that the rank of the optimal regression matrix decreases as the noise variance increases, so that in the presence of channel noise $k < \min(p, q)$, even if the rank of the original $\mathbf{f}$ is higher than k.

- In image processing applications the case $k << N$ is very common, also because there are pixel correlations in images of objects of a certain class (within one signal space). For example, in the case of face images these correlations are due to homogeneous skin regions, symmetry, etc.

Ridge regression (see section 2.1.9) exploits correlations in the input space, however, it ignores correlations in the output space, because the regression parameters are calculated separately for each output variable. The methods reviewed in the following sections

also take into account multiple output variables. Thereby, canonical coordinates allow to combine response variables.

2.4.1 Exploiting correlations of response variables

If there are correlations in the output data better estimates of the regression function are found by combining the variables by exploiting correlations among the responses. As an example, suppose that all components of $\mathbf{f}(\mathbf{x})$ share the same structural part, e.g.,

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \boldsymbol{\epsilon} = g(\mathbf{x})\mathbf{c} + \boldsymbol{\epsilon} = \begin{pmatrix} c_1\mathbf{w}^T\mathbf{x} + \epsilon_1 \\ \vdots \\ c_q\mathbf{w}^T\mathbf{x} + \epsilon_q \end{pmatrix}, \tag{2.62}$$

where each component is a multiple of the same scalar-valued function $g : \mathbb{R}^p \rightarrow \mathbb{R}^1, g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ and $\mathbf{c} = (c_1, \ldots, c_q)^T$ is a vector with multipliers.

According to Eq. 2.14 the theoretical optimum is

$$\mathbf{W} = \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1} = E[(\mathbf{c}\mathbf{w}^T\mathbf{x} + \boldsymbol{\epsilon})\mathbf{x}^T]\mathbf{C}_{xx}^{-1} = \mathbf{c}\mathbf{w}^T\mathbf{C}_{xx}\mathbf{C}_{xx}^{-1} = \mathbf{c}\mathbf{w}^T, \tag{2.63}$$

i.e., the true regression matrix $\mathbf{W}$ is of rank one. As $N \rightarrow \infty$ the OLS solution given by Eq. 2.22 approximates the Wiener filter solution (cf. Eq. 2.14). However in the case of limited data the full-rank OLS (using Eq. 2.22) is prone to modeling the noise in all remaining $q - 1$ dimensions, i.e. it is sensitive to variations in the training set and may be rendered full rank by noise in the data.

For example, if $\mathbf{c} = (1, 1, ..., 1)^T$, it is obvious that the prediction $\tilde{\mathbf{y}} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_q)^T$ at a given input is improved by using for each response component the "average" of the separate OLS estimates, i.e.,

$$\tilde{y}_i = \frac{1}{q}\left(\hat{y}_1 + \hat{y}_2 + \ldots + \hat{y}_q\right). \tag{2.64}$$

which corresponds to the rank-one estimate

$$\hat{\mathbf{W}} = \frac{1}{q}\mathbf{I}_{q\times q}\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \tag{2.65}$$

For a general known $\mathbf{c}$ we can use the estimate

$$\hat{\mathbf{W}} = \mathbf{P}_c\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}, \tag{2.66}$$

where $\mathbf{P}_c$ is a projection onto the one-dimensional space spanned by $\mathbf{c}$. Clearly, $\hat{\mathbf{W}}$ is of rank one. How can correlations be used when $\mathbf{c}$ is not known? It turns out that canonical coordinates obtained by CCA are the right coordinate system to perform reduced rank regression or proportional shrinkage of coordinates in order to reduce the MSE [10].

Fig. 2.4 shows an example of two inherently one dimensional sets of data points corrupted with additive Gaussian noise with high isotropic variance in two dimensions. Since the reduced-rank solution given by Eq. 2.65 models the signal subspace and neglects (orthogonal) noise components it is also less sensitive to noise in the input data.

Compared to full-rank OLS the low-rank solution introduces bias. However, it will in many cases perform better (with respect to the true risk) in the case of limited training data, because it has less degrees of freedom to fit the noise in the training data. Moreover, if we have a-priori knowledge about the rank of the regression function, we might get a better estimate of the signal subspace.

In fact it has been shown in [20] (although for the case of channel noise) that as the noise variance increases in relation to the signal variance the rank of the optimal linear channel $\mathbf{W}$ decreases. A similar result for parallel additive Gaussian noise channels is described in [20] from the information theoretical viewpoint.

In the remainder of this section we will survey various methods which allow sufficient improvements over ordinary multivariate regression introduced above in the case of correlated input resp. response variables. We will also discuss the special case $p, q > N$ which is the typical situation in image processing applications.

We will review enhanced methods for regression and show how canonical coordinates can be used for combining response variables to obtain improved regression estimates, which yield better performance in the case of correlated response variables and limited training data. The methods are based on the concept of effective degrees of freedom and its relation to the theoretical in-sample prediction error (see Section 2.1.14). Estimates of this error measure can be used to select the optimal number of (effective) parameters of the regression model. Additionally we will discuss the effect of ridge regularization of both, the input and output space, to obtain improved estimates of canonical factors in the case of poor sample support.

The improved predictions have the general form

$$\hat{\mathbf{y}} \;=\; \mathbf{T}\hat{\mathbf{g}}(\mathbf{x}) = \mathbf{T}\mathbf{Y}\mathbf{h}(\mathbf{x}), \tag{2.67}$$

where $\mathbf{h}(\mathbf{x})$ is a N-vector of linear weights producing the OLS fit $\hat{\mathbf{g}}(\mathbf{x}) = \mathbf{Y}\mathbf{h}(\mathbf{x})$. For example (cf. Eq. 2.23), in the case of OLS regression $\mathbf{h}(\mathbf{x}) = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{x}$. For ridge-regression, $\mathbf{h}(\mathbf{x}) = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{x}$ with $\lambda > 0$ being the common ridge parameter.

$\mathbf{T}$ is a shrinking matrix used to pool the observations on the *response* variables and thereby exploiting correlations between the responses. In the case of OLS estimates $\mathbf{T}$ is simply the identity matrix. In the case of reduced-rank regression, which will be discussed in section 2.4.2, $\mathbf{T}$ is an orthogonal projector truncating dimensions in which estimates are less reliable. Shrinkage resp. truncation is performed in the CCA response space as discussed next.

$\mathbf{T}$ is a linear least-squares regression of $\mathbf{y}$ on the sample-based OLS predictions over the population distribution, i.e.,

$$\mathbf{T} \;=\; \mathrm{E}(\mathbf{y}\hat{\mathbf{g}}(\mathbf{x})^T)\mathrm{E}(\hat{\mathbf{g}}(\mathbf{x})\hat{\mathbf{g}}(\mathbf{x})^T)^{-1}, \tag{2.68}$$

where $\mathrm{E} \equiv \mathrm{E}_{\mathbf{x},\mathbf{y}}\mathrm{E}_{\mathcal{T}}$. Using Eq. 2.46 and Eq. 2.68

$$\begin{aligned}
\mathbf{T} &= \mathbf{G}(\mathbf{G} + \frac{p}{N}\mathbf{\Sigma})^{-1} && (2.69)\\
&= \mathbf{G}(\mathbf{G} + \frac{p}{N}(\mathbf{C}_{yy} - \mathbf{G}))^{-1} && (2.70)\\
&= (\mathbf{I}_q + \frac{p}{N}(\mathbf{G}^{-1}\mathbf{C}_{yy} - \mathbf{I}_q))^{-1} && (2.71)\\
&= (\mathbf{I}_q + \frac{p}{N}((\mathbf{C}^T\mathbf{C})^{-1} - \mathbf{I}_q))^{-1} && (2.72)\\
&= (\mathbf{I}_q + \frac{p}{N}((\mathbf{V}\mathbf{D}^{-2}\mathbf{V}^T) - \mathbf{I}_q))^{-1} && (2.73)\\
&= \mathbf{V}\mathbf{D}^*\mathbf{V}^T && (2.74)
\end{aligned}$$

where $\mathbf{D}^*$ is a diagonal matrix with diagonal elements

$$d_i^* = \frac{d_i^2}{d_i^2 + \frac{q}{N}(1 - d_i^2)}. \tag{2.75}$$

This result shows that the matrix $\mathbf{T}$ is diagonal in the (population) $\mathbf{y}$ canonical coordinate system (see Breiman and Friedman [10]).

2.4.2 Truncating the response canonical space: reduced-rank regression

Reduced rank regression [34] uses the criterion

$$\text{RSS}(\mathbf{W}) = \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{W}\mathbf{x}_i)^T \mathbf{\Sigma}^{-1} (\mathbf{y}_i - \mathbf{W}\mathbf{x}_i) \tag{2.76}$$

which is minimized subject to $\text{rank}(\mathbf{W}) = k$ with $\mathbf{\Sigma} = \text{E}(\epsilon\epsilon^T)$ (population noise covariance). The solution is the rank-k matrix

$$\hat{\mathbf{W}} = \mathbf{V}^{-1}\mathbf{I}_k\mathbf{V}\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}, \tag{2.77}$$

where $\mathbf{V}$ is the matrix containing the (population) left canonical vectors in its columns and $\mathbf{I}_k = diag\{\mathbf{1}(i \leq k)\}_1^k$.

Reduced rank regression performs a linear regression on the pooled response variables $\mathbf{Y}^T\mathbf{V}^T\mathbf{I}_k$ by discarding the directions of trailing canonical correlation. These are those linear combinations of response variables which have least prediction accuracy. Finally, multiplying from the left by $\mathbf{V}^{-1}$ then maps the fits back to the original response space. $\mathbf{\Sigma}$ can be replaced by the estimate $\hat{\mathbf{\Sigma}} = \mathbf{Y}\mathbf{Y}^T$, in which case $\mathbf{V}$ is replaced by the empirical canonical response coordinates.

2.4.3 Shrinking in the response canonical space: Curds & Whey procedure

Canonical coordinates are also the right coordinates for performing multivariate shrinking in the case of prediction of multiple outputs with limited training data. Methods for shrinking in canonical response coordinates are proposed in [69] (filtered canonical y-variate regression) and in [10] (curds and whey method). These methods represent smooth versions of reduced rank regression, just like ridge-regression can be regarded as a smooth version of principal component regression.

In [10], Breiman and Friedman propose simultaneous shrinking in input and output space. In their formulation the regression parameters are

$$\mathbf{W}_k = \mathbf{V}^{-1}\mathbf{D}^*\mathbf{V}\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}, \tag{2.78}$$

where $\mathbf{D}^*$ is a diagonal matrix with diagonal elements given by Eq. 2.75.

In practice the population canonical vectors $\mathbf{V}$ have to be estimated from the sample by empirical canonical vectors. In this case, $\mathbf{D}^*$ has to be estimated by generalized cross validation. As shown in [10], this leads to the estimate $\hat{\mathbf{D}}^*$ with diagonal elements

$$\hat{d}_i^* = \frac{(1-r)(\hat{\rho}_i^2 - r)}{(1-r)^2\hat{\rho}_i^2 + r^2(1-\hat{\rho}_i^2)}. \tag{2.79}$$

where $\hat{\rho}_i$ are the empirical canonical correlations and $r = \frac{p}{N}$. In the case of simultaneous shrinking in input and output space the regression parameters are

$$\mathbf{W} = \hat{\mathbf{V}}^{-1}\mathbf{D}^*\hat{\mathbf{V}}\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda_x\mathbf{I}_p)^{-1}, \tag{2.80}$$

where $\hat{\mathbf{V}}$ are the canonical vectors obtained by canonical correlation analysis of the sample responses $\mathbf{Y}$ and the ridge regression estimates $\hat{\mathbf{Y}}$. $\mathbf{D}^*$ is obtained by Eq. 2.79 using the corresponding empirical canonical correlations $\hat{\rho}_i$ and the effective degrees of freedom $r = \text{trace}(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda_x\mathbf{I}_p)^{-1}\mathbf{X})$.

2.4.4 Ridge regularization for CCA

In previous sections we have discussed ways to improve the performance of a linear least-squares estimator in the case of limited and noisy training data. The methods involve the usage of the $\mathbf{y}$ canonical coordinate system (i.e., the basis $\mathbf{V}$). However, we have not discussed how we can improve the estimates of $\mathbf{U}$ and $\mathbf{V}$.

In the case of limited sample support or even singular $\hat{\mathbf{C}}_{xx} = \mathbf{X}\mathbf{X}^T$ resp. $\hat{\mathbf{C}}_{yy} = \mathbf{Y}\mathbf{Y}^T$ we can use the respective generalized inverse and the canonical correlation analysis is confined to the non-zero variance subspace of inputs resp. responses. However, the estimates of $\mathbf{U}$ and $\mathbf{V}$ will still be poor. In fact, when the number of samples $N < p + q$ and there are possible additional row degeneracies ($\text{rank}(\mathbf{X}^T) < p$ or $\text{rank}(\mathbf{Y}^T) < q$) there at least $p + q - N$ canonical correlations of 1 and as many factor pairs having high arbitrariness.

In the case of $p > N$ we can use a positive ridge-penalty parameter λ_x to avoid a singular or badly conditioned $\mathbf{X}\mathbf{X}^T$. This approach protects against potentially high variance of regression parameters corresponding to directions of small variance in the input space at the expense of increasing the bias of the estimator. In doing so, we implicitly assume that the gradient of response is highest in directions of high variance in the input

space and that the noise rate is higher in directions of small variance (for instance in the case of additive isotropic noise).

If $q > N$, the situation is similar. Canonical correlation analysis of responses $\mathbf{Y}$ and regression estimates $\hat{\mathbf{Y}}$ fitted by ridge regression with $\lambda_x > 0$, will obtain $\min(q, N)$ canonical correlations $\hat{\rho}_i = 1$. Consequently, $\mathbf{D}^* = \mathbf{I}_q$ such that the resulting regression is equivalent to ridge regression without shrinkage in the response canonical space (cf. Eq. 2.80).

Ridge regularization for CCA has originally been proposed in [70]. To gain a better understanding of the effect of the regularization term, we consider the standard (primal) definition of CCA

$$\rho_{CCA} = \frac{\mathbf{w}_x^T \hat{\mathbf{C}}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \hat{\mathbf{C}}_{xx} \mathbf{w}_x \mathbf{w}_y^T \hat{\mathbf{C}}_{yy} \mathbf{w}_y}}, \tag{2.81}$$

where $\hat{\mathbf{C}}_{xy}$ is the estimated between-set covariance matrix and $\hat{\mathbf{C}}_{xx}$, $\hat{\mathbf{C}}_{yy}$ are estimated within-set covariance matrices. We compare Eq. 2.81 with the defining equations for partial least squares (PLS) and multivariate linear regression (MLR) [8]. PLS, which maximizes the covariance between $\mathbf{x}$ and $\mathbf{y}$, replaces both $\hat{\mathbf{C}}_{xx}$ and $\hat{\mathbf{C}}_{yy}$ in the denominator by the unit matrix,

$$\rho_{PLS} = \frac{\mathbf{w}_x^T \hat{\mathbf{C}}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{w}_x \mathbf{w}_y^T \mathbf{w}_y}}, \tag{2.82}$$

while MLR, which performs a least squares regression onto $\mathbf{y}$, retains the normalization by the variance of the predictor variable $\mathbf{x}$, but discards the variance-normalization w.r.t. $\mathbf{y}$ (where the square error is defined), i.e.,

$$\rho_{MLR} = \frac{\mathbf{w}_x^T \hat{\mathbf{C}}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \hat{\mathbf{C}}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{w}_y}}. \tag{2.83}$$

Thus, as also pointed out in [8], all three approaches effectively solve the same problem, namely maximization of the covariance, but are subject to different scalings of the variables.

As mentioned above, the regularization term $\lambda \mathbf{I}$ can be used to render singular covariance matrices positive definite. If λ is increased even further, the matrices will eventually become isotropic. Hence, for sufficiently large λ, regularized CCA becomes equivalent to PLS in the sense that both approaches will yield the same extremum points (the extremum

values, however, will differ approximately by a factor $\frac{1}{\lambda}$). By the same argument, we can transform CCA into MLR; if we use different regularization parameters λ_x and λ_y for $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$, respectively, their relative magnitude determines whether (or, more precisely: to which extent) we perform a regression onto $\mathbf{x}$ or onto $\mathbf{y}$. As mentioned above solutions orthogonal to the signal variance are not always desirable; in such cases the regularization parameter λ can be used to adjust the influence of signal variance on the solutions $\mathbf{w}_x, \mathbf{w}_y$ [33].

2.4.5 Input noise

The standard regression model of Eq. 2.2 assumes a noiseless input signal and that only the output is contaminated with additive Gaussian noise. Now, let us assume that the input as well as the output are noise-contaminated signals, i.e., both signals are related to noiseless variables $\mathbf{s}$ by

$$\mathbf{x} = \mathbf{W}_{xs}\mathbf{s} + \boldsymbol{\epsilon}_{xs}, \tag{2.84}$$

$$\mathbf{y} = \mathbf{W}_{ys}\mathbf{s} + \boldsymbol{\epsilon}_{ys}, \tag{2.85}$$

where we assume $\mathbf{s} \in \mathbb{R}^m$, $\mathbf{W}_{xs} \in \mathbb{R}^{m\times p}$, $\mathbf{W}_{ys} \in \mathbb{R}^{q\times m}$ and that $\boldsymbol{\epsilon}_{xs}$ and $\boldsymbol{\epsilon}_{ys}$ are normally distributed with zero mean. Assuming that $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian, because of $\mathrm{Cov}(\boldsymbol{\epsilon}_{xs}, \boldsymbol{\epsilon}_{ys}) = 0$ and because all components of $\mathbf{y}$ that are uncorrelated with $\mathbf{x}$ can not be predicted, we can set $m = \min(p, q)$.

In the case of $\mathrm{Cov}(\boldsymbol{\epsilon}_{xs}) = 0$ (noiseless input) the optimal regression parameters are given by the Wiener solution

$$\begin{aligned}
\mathbf{W} &= \mathrm{E}(\mathbf{y}\mathbf{x}^T)\mathrm{E}(\mathbf{x}\mathbf{x}^T)^{-1} && (2.86)\\
&= \mathrm{E}(\mathbf{y}\mathbf{s}^T\mathbf{W}_{xs}^T)\mathrm{E}(\mathbf{W}_{xs}\mathbf{s}\mathbf{s}^T\mathbf{W}_{xs}^T)^{-1} && (2.87)\\
&= \mathbf{W}_{ys}\mathrm{E}(\mathbf{s}\mathbf{s}^T)\mathbf{W}_{xs}(\mathbf{W}_{xs}\mathrm{E}(\mathbf{s}\mathbf{s}^T)\mathbf{W}_{xs})^{-1} && (2.88)\\
&= \mathbf{W}_{ys}\mathbf{W}_{xs}^{\dagger} && (2.89)
\end{aligned}$$

Obviously, in the case of noisy input, i.e. $\mathrm{Cov}(\boldsymbol{\epsilon}_{xs}) > 0$, the OLS regression underesti-

mates $\mathbf{W}$, because it approximates the following Wiener solution

$$\begin{aligned} \mathbf{W} &= \mathrm{E}(\mathbf{y}\mathbf{x}^T)\mathrm{E}(\mathbf{x}\mathbf{x}^T)^{-1} && (2.90)\\ &= \mathbf{W}_{ys}\mathrm{E}(\mathbf{s}\mathbf{s}^T)\mathbf{W}_{xs}(\mathbf{W}_{xs}\mathrm{E}(\mathbf{s}\mathbf{s}^T)\mathbf{W}_{xs} + \mathrm{E}(\boldsymbol{\epsilon}_{xs}\boldsymbol{\epsilon}_{xs}^T))^{-1} && (2.91)\\ &< \mathbf{W}_{ys}\mathbf{W}_{xs}^{\dagger}, && (2.92) \end{aligned}$$

and thus the OLS estimator (cf. Eq. 2.22) produces biased predictions.

Given a sample of N observation pairs with noisy input and output, we are interested in an estimate of the true regression matrix that predicts the response from noiseless inputs. In [65] this problem is tackled by variational Bayesian algorithm based on Factor Analysis and assuming that $\mathbf{W}_{xs}$ is diagonal. Here we alternatively employ canonical correlation analysis to obtain an unbiased estimate of $\mathbf{W}$ in the presence of input noise. Thereby, in a first step the regression relevant subspaces are identified by CCA. In the second step we regress $\mathbf{Y}$ on the projections of $\mathbf{X}$ onto its canonical subspace. It is easy to show, that the column space of $\mathbf{W}_{sx}$ is equal to the columns space of the m left population canonical vectors $\mathbf{U}_m$, i.e.,

$$\mathbf{P}_{sx} = \mathbf{W}_{sx}\mathbf{W}_{sx}^{\dagger} = \mathbf{U}_m\mathbf{U}_m^T. \quad (2.93)$$

The same holds analogously for the column space of $\mathbf{W}_{sy}$ and the m right population canonical vectors $\mathbf{V}_m$. Thus, the improved estimate of $\mathbf{W}$ is given by

$$\hat{\mathbf{W}} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\hat{\mathbf{U}}_m\hat{\mathbf{U}}_m^T, \quad (2.94)$$

where $\hat{\mathbf{U}}_m$ are the first m empirical right canonical vectors. Note that this estimate requires knowledge of m. If m is unknown, it can be made a model selection parameter to be estimated through cross-validation. Experiments in which this procedure is applied to matching of Active Appearance Models are described in Section 4.2. Results indicate an improved performance compared to standard regression.

2.5 Summary

In this chapter we discussed linear models for regression, particularly for regression between two high dimensional signal spaces. We have reviewed relevant concepts of machine learning, the notion of effective number of parameters in the context of linear models and methods to approximate the expected risk, which are needed for model selection.

We have seen, that when using linear models, the model complexity is related to the effective number of parameters. While standard shrinking methods (e.g., ridge regularization) exploit correlations only in the predictors to reduce the effective number of parameters, enhanced regression methods based on CCA allow to pool the response variables and thus further improve the predictive accuracy. We have introduced regularized CCA, where ridge penalty terms are added to the CCA criterion. This allows to determine the (biased) empirical canonical factor pairs from a limited sample of high dimensional observations.

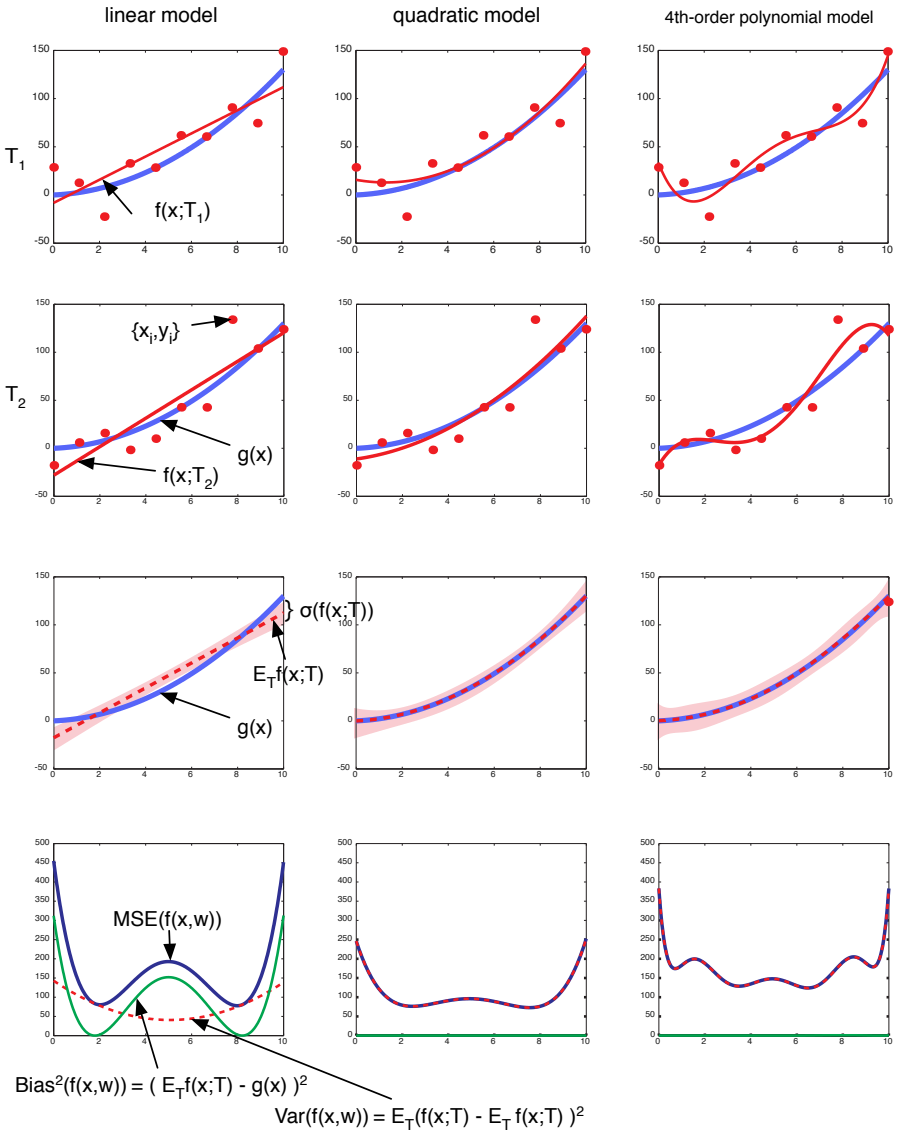


Figure 2.2: The trade-off between bias^2 and variance demonstrated on a regression example with a single input resp. output variable. The upper two rows show models of different complexity fitted to a sample of 10 data points (red points). The blue curve depicts the true deterministic function from which the data was generated. The red line depicts the fitted model. The left column shows a linear regression model, the middle column a quadratic regression model and the right column shows regression with a polynomial of degree 4. The third row shows the mean (red dotted curve) and standard deviation (red area) of the predictions $f(\mathbf{x}; \mathcal{T})$. The plots in the lowest row show the MSE of the predictions in dependence of $\mathbf{x}$ as a sum of bias^2 and variance. In this example, the MSE and its decomposition into bias^2 and variance was estimated by drawing 2000 samples.

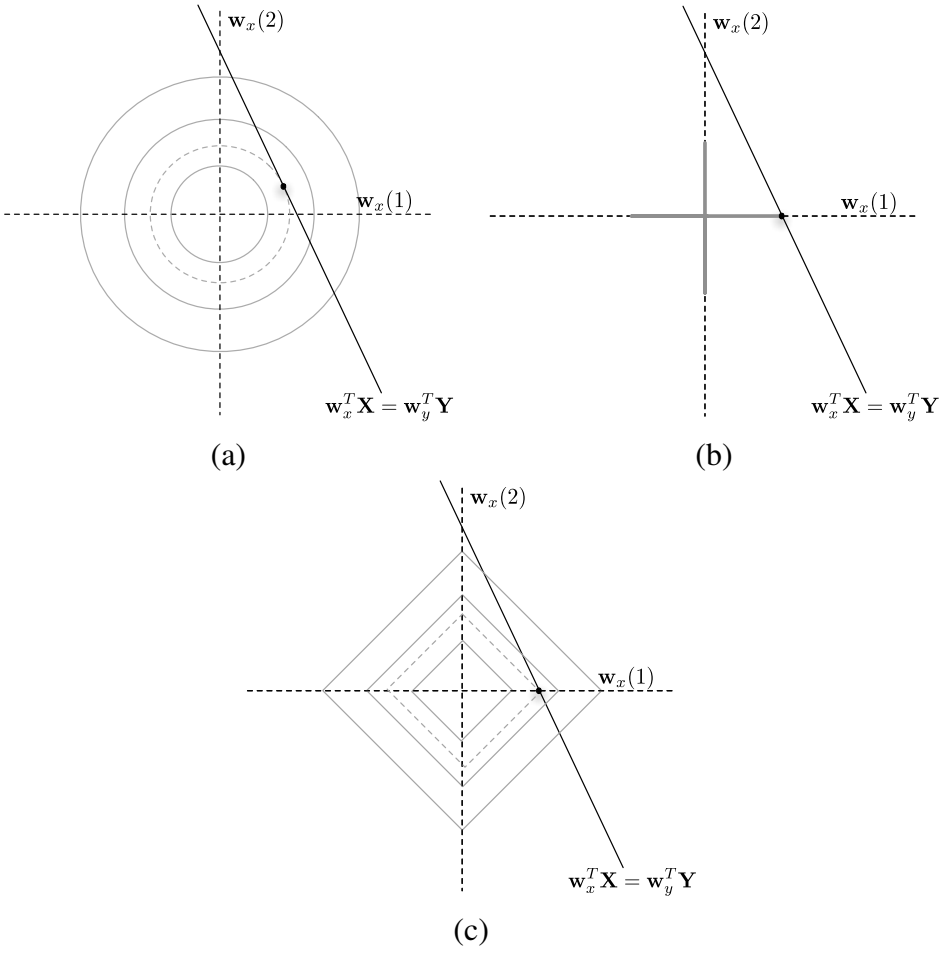


Figure 2.3: CCA with $N < p + q$ and fixed $\mathbf{w}_y$ is an underdetermined linear system. Ridge regularization yields a unique solution $\mathbf{w}$ with minimal energy (a). This energy is spread over all coefficients. A sparse solution, where the energy is concentrated in a few non-zero coefficients, is obtained using the l^0-norm penalty (b). However, the resulting optimization problem is non-convex and NP-hard. Therefore, the l^1-norm approximation is used, also yielding a sparse solution, with a convex criterion (c).

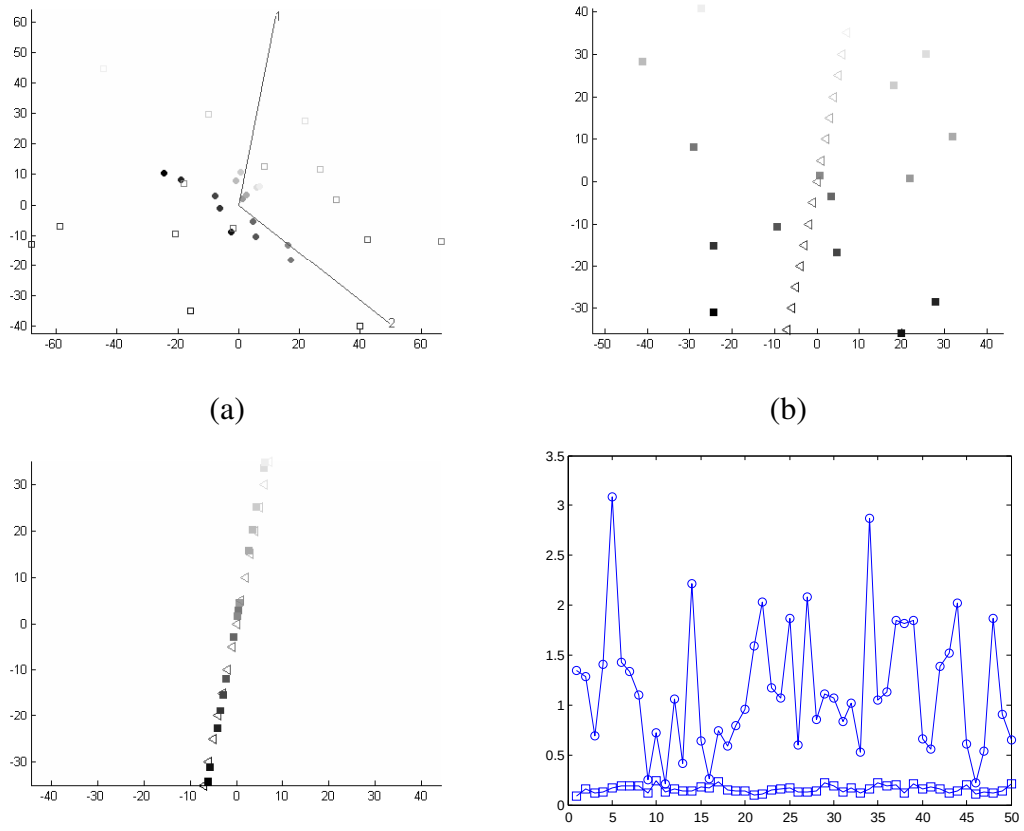


Figure 2.4: Example showing how CCA can be used to perform reduced-rank regression: (a) Two sets of noisy training data points (15 points in each set). Filled circles depict data points of input data and squares depict points of the response data set. Corresponding points have the same gray value. The lines indicate the directions of first and second empirical canonical factor of the response variables; (b) OLS regression response estimates (squares) on an independent test set: Target values (noise-free data points) are indicated as triangles; (c) Rank-1 predictions with test input data in the principal correlation subspace obtained by CCA, (d) Average prediction error for OLS regression (circles) and rank-1 regression (squares) for 50 test runs.

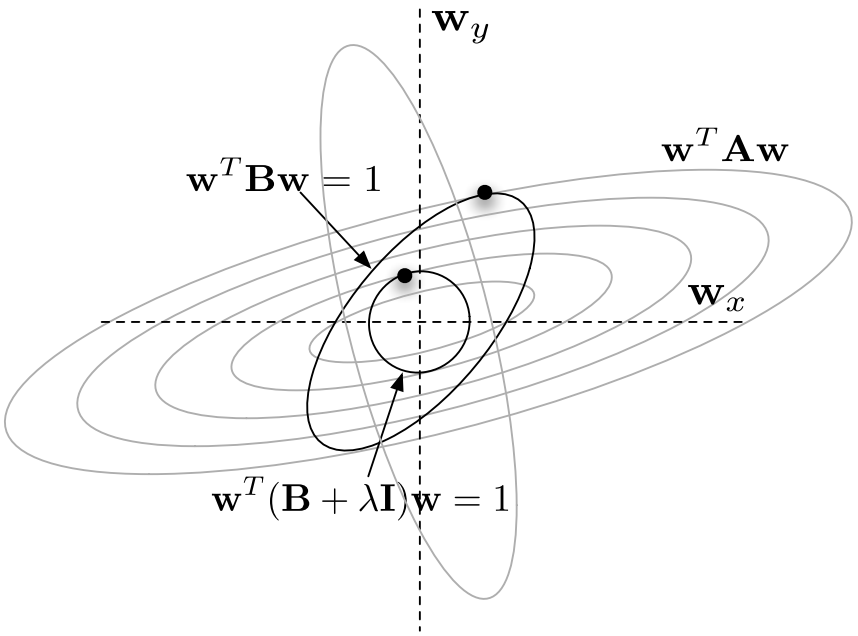


Figure 2.5: A schematic illustration of the space of coefficients $\mathbf{w} = (\mathbf{w}_x^T, \mathbf{w}_y^T)^T$and the effect of ridge penalty regularization. The concentric grey ellipses indicate the contours of the quadratic form $\mathbf{w}^T\mathbf{A}\mathbf{w}$ (the numerator of the rayleigh quotient), while the black ellipse indicates all points satisfying $\mathbf{w}^T\mathbf{B}\mathbf{w} = 1$. Maximization of ρ is achieved at the upper right black dot. If ridge regularization is performed, then with growing λ the ellipse $\mathbf{w}^T(\mathbf{B} + \lambda\mathbf{I})\mathbf{w} = 1$ becomes more and more circular with smaller radius (the coefficients $\mathbf{w}$ are shrunk), leading to solutions maximizing solely $\mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y$, regardless of $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ in the denominator. The gray ellipse with main axis perpendicular to $\mathbf{w}^T\mathbf{A}\mathbf{w}$ indicates the contour of the joint density of $\mathbf{x}$ and $\mathbf{y}$ (i.e., where $\mathbf{w}^T\mathbf{A}^{-1}\mathbf{w}$ is constant).

Chapter 3

Kernel-CCA and Regularization

3.1 Kernel-CCA

The goal of this section is to introduce a nonlinear generalization of CCA based on a kernel formulation. Kernel formulations allow to introduce nonlinearity to linear algorithms while avoiding a nonlinear optimization problem. The key idea behind kernel methods is that a linear algorithm can be employed on nonlinearly transformed input data. The transformation of input data is performed by a mapping from the original input space to a high-dimensional feature space.

If the linear algorithm can be formulated only in terms of inner products of the input data, the explicit computation of the high dimensional mapping is avoided by evaluating a kernel function instead of computing the mapping itself. In the field of pattern recognition, kernel-methods were originally proposed as a nonlinear extension of the *support vector machine* (SVM) classifier [9].

3.1.1 Formulation of nonlinear CCA

We rewrite the CCA criterion by introducing general (nonlinear) transformations $u : \mathbb{R}^p \mapsto \mathbb{R}, u \in \mathcal{H}_1$ and $v : \mathbb{R}^q \mapsto \mathbb{R}, v \in \mathcal{H}_2$. We start our formulation of nonlinear CCA by considering hypothesis spaces of square integrable functions, i.e. $\mathcal{H}_1$ and $\mathcal{H}_2$ are closed subspaces of $L_2(\mu)$ and $L_2(\nu)$ respectively, where μ and ν are corresponding probability measures (i.e., for $\mathcal{A} \subseteq \mathbb{R}^p$, $\mu(\mathcal{A}) = P(X \in \mathcal{A})$ and for $\mathcal{B} \subseteq \mathbb{R}^p$, $\nu(\mathcal{B}) = P(X \in \mathcal{B})$).

The goal of generalized CCA is to find the maximum w.r.t. $u \in \mathcal{H}_1$ and $v \in \mathcal{H}_2$ of the

functional

$$R(u,v) = \mathrm{Corr}^2(u(X), v(Y)) \tag{3.1}$$

$$= \frac{\mathrm{E}\left(u(X)v(Y)\right) - \mathrm{E}\left(u(X)\right)\mathrm{E}\left(v(Y)\right)}{\sqrt{\left(\mathrm{E}\left(u(X)^2\right) - \mathrm{E}^2\left(u(X)\right)\right)\left(\mathrm{E}\left(v(Y)^2\right) - \mathrm{E}^2\left(v(Y)\right)\right)}} \tag{3.2}$$

where E is the expectation.

Equivalently, we can formulate nonlinear CCA as a contrained optimization problem: Maximize

$$\mathrm{E}\left(u(X)v(Y)\right), \tag{3.3}$$

subject to

$$\mathrm{E}(u(X)) = \int_{\mathbb{R}^p} u(\mathbf{x}) d\mu(\mathbf{x}) = 0, \tag{3.4}$$

$$\mathrm{E}(v(Y)) = \int_{\mathbb{R}^p} v(\mathbf{y}) d\nu(\mathbf{y}) = 0, \tag{3.5}$$

$$\mathrm{E}(u(X)^2) = \int_{\mathbb{R}^p} u^2(\mathbf{x}) d\mu(\mathbf{x}) = \|u(\mathbf{x})^2\|_\mu = 1, \tag{3.6}$$

$$\mathrm{E}(v(Y)^2) = \int_{\mathbb{R}^q} v^2(\mathbf{y}) d\nu(\mathbf{y}) = \|v(\mathbf{y})^2\|_\nu = 1 \tag{3.7}$$

Finally, we write generalized CCA in terms of the conditional expectation operator as follow: Let $\mathbf{P} : L^2(\mathbb{R}^p) \mapsto L^2(\mathbb{R}^q), \mathbf{P}u = \mathrm{E}(u(X)|Y = \mathbf{y})$ be the conditional expectation operator and $\tilde{\mathbf{P}} : L^2(\mathbb{R}^q) \mapsto L^2(\mathbb{R}^p)$ be the adjoint operator of $\mathbf{P}$. In the following we assume $\mathbf{P}$ to be compact. Note that due to the finite range of data in practical applications this assumption is no restriction.

The maximum of $R(u,v)$ can be written as

$$\arg \max_{\substack{\|u^2\|_\mu = \|v^2\|_\nu = 1 \\ \|u\|_\mu = \|v\|_\nu = 0}} \langle \mathbf{P}u, v \rangle_\nu \tag{3.8}$$

which is equal to

$$\arg \max_{\substack{\|u^2\|_\mu = \|v^2\|_\nu = 1 \\ \|u\|_\mu = \|v\|_\nu = 0}} \langle u, \tilde{\mathbf{P}}v \rangle_\mu \tag{3.9}$$

Then the maximal value of $L(u,v)$ is given by the largest eigenvalue λ_0 of $\tilde{\mathbf{P}}\mathbf{P}$ (or $\mathbf{P}\tilde{\mathbf{P}}$, which has

the same eigenvalues), i.e.

$$\lambda_0 = L(e_o, e_o'), \tag{3.10}$$

where e_0 is any eigenfunction belonging to the largest eigenvalue λ_0 and $e_0' = \mathbf{P}e_0$. Note that the eigenspace of λ_0 is at most finite dimensional.

3.1.1.1 Example: $\mathbf{P}$ is compact (Hilbert-Schmidt)

Let the joint probability

$$P(X \in \mathcal{A} \wedge Y \in \mathcal{B}) = \int_{\mathcal{A}} \int_{\mathcal{B}} p(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \tag{3.11}$$

with the square integrable density $p(\mathbf{x}, \mathbf{y})$. The probability measures are given by

$$\begin{aligned} \mu(\mathcal{A}) &= P_X(X \in \mathcal{A}) && (3.12) \\ &= \int_{\mathcal{A}} p_x(\mathbf{x}) d\mathbf{x} && (3.13) \end{aligned}$$

and

$$\begin{aligned} \nu(\mathcal{B}) &= P_y(Y \in \mathcal{B}) && (3.14) \\ &= \int_{\mathcal{B}} p_y(\mathbf{y}) d\mathbf{y} && (3.15) \end{aligned}$$

for any $\mathcal{A} \subseteq \mathcal{X}$ and $\mathcal{B} \subseteq \mathcal{Y}$, i.e., $d\mu(\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$ and $d\nu(\mathbf{y}) = p(\mathbf{y})d\mathbf{y}$. In this situation our operator $\mathbf{P}$, which has the explicit form

$$(\mathbf{P}u)(\mathbf{y}) = \int_{\mathbf{R}^p} K(\mathbf{x}, \mathbf{y}) u(\mathbf{x}) d\mu(\mathbf{x}) \tag{3.16}$$

with kernel

$$K(\mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p_x(\mathbf{x}) p_y(\mathbf{y})}, \tag{3.17}$$

is known to be compact (Hilbert-Schmidt).

3.1.1.2 Example: Finite-dimensional case

Consider $X \in \mathcal{X} = \{1, \ldots, N\}$ and $Y \in \mathcal{Y} = \{1, \ldots, M\}$ and the probabilities $\mathbf{P} = (p_{ij})_{i,j} = \mathrm{E}(X = i \wedge Y = j)$. Then

$$Pu = \mathbf{P}u = \sum_{i \in \mathcal{X}} \frac{p_{ij}}{\sum_{i \in \mathcal{X}} p_i j} u_i \tag{3.18}$$

If e_0 is the eigenvector of $\mathbf{P}^T\mathbf{P}$.

Given a training set of N observations $\{\mathbf{x}_i, \mathbf{y}_i\}, i = 1, \ldots, N$ the corresponding empirical risk functional is

$$R(u, v)_{emp} = \frac{1}{2} \sum_{i=1}^{N} (u(\mathbf{x}_i) - v(\mathbf{y}_i))^2 \tag{3.19}$$

In contrast to the standard nonlinear regression problem, where the solution approximates the true regression function for $N \to \infty$, the problem of minimizing Eq. 3.19 is ill-posed even if the joint probability densitiy $p(x, y)$ is known, since there are generally infinitly many pairs u, v with the same value of the true risk $R(u, v)$.

The Problem can again be remedied by using a regularized risk functional

$$R_\lambda(u, v) = R(u, v) + \lambda \Omega(u, v), \tag{3.20}$$

where $\Omega(u, v)$ is a regularization functional. Ridge penalty regularization for kernel-CCA discussed in section 3.1.8.

3.1.2 Reproducing kernel Hilbert space

The function u (resp. v) of section 3.1.1 can be regarded as single output nonlinear transformation, which can be written in the form

$$u(\mathbf{x}) \quad = \quad \sum_{i=1}^{\infty} c_i \phi_i(\mathbf{x}), \tag{3.21}$$

i.e., as a linear combination of a (a possibly infinite number of) nonlinear basis functions ϕ_i The functions are elements of a nonlinear feature transformation $\boldsymbol{\phi}(\mathbf{x})$, where

$$\mathbf{x} = (x_1, \ldots, x_d) \mapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_i(\mathbf{x}), \ldots), \tag{3.22}$$

which maps the original data into a infinite dimensional feature space $\mathbf{F}$. A possibe way to define a space of functions, which can be represented by Eq. 3.21 is a *reproducing kernel Hilbert space* (RKHS), i.e. a function space generated by a kernel function $k(.,.)$. In the following will give a short overview of the properties of RKHS. More details can be found in [71] and [30]. A good survey of these models is given in [28].

A RKHS is a Hilbert space $\mathcal{H}$ of functions defined over a bounded domain $\mathcal{D} \subseteq \mathbb{R}^d$ with the property that, for each $\mathbf{x} \in \mathcal{D}$, the evaluation functionals $\mathcal{F}_\mathbf{x}$ which associate

$f \in \mathcal{H}$ with $f(\mathbf{x})$, are linear bounded functionals, i.e.

$$\mathcal{F}_{\mathbf{x}}[\lambda_1 f + \lambda_2 g] = \lambda_1 f(\mathbf{x}) + \lambda_2 g(\mathbf{x}) \quad \forall f, g \in \mathcal{H}, \lambda_1, \lambda_2 \in \mathbb{R} \tag{3.23}$$

and there exists a $U = U_{\mathbf{x}} \in \mathbb{R}^+$ such that

$$|\mathcal{F}_{\mathbf{x}}| = |f(\mathbf{x})| \leq U\|f\|, \tag{3.24}$$

where $\|.\|$ is the norm in $\mathcal{H}$.

A RKHS is induced by a *(Mercer) kernel*, which is a symmetric real valued continuous function $k(.,.) : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ of two variables, which is positive definite, i.e. for all $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathcal{D}$ and $\{c_1, \ldots, c_N\} \subset \mathbb{R}$, $k(.,.)$ satisfies

$$\sum_{i,j=1}^{m} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{3.25}$$

Under general conditions, if $k(.,.)$ is a Mercer kernel and satisfies

$$\int_{\mathcal{D}} \int_{\mathcal{D}} k^2(\mathbf{x}, \mathbf{x}') d\mathbf{x} d\mathbf{x}' < \infty \tag{3.26}$$

there exists an eigen-expansion in a in $\mathcal{D} \times \mathcal{D}$ uniformly convergent series of continuous eigenfunctions ϕ_i and positive associated eigenvalues $\gamma_i \geq 0$, i.e.,

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \gamma_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}'), \int_{\mathcal{D}} \int_{\mathcal{D}} k^2(\mathbf{x}, \mathbf{x}') d\mathbf{x} d\mathbf{x}' = \sum_{i=1}^{\infty} \gamma_i < \infty. \tag{3.27}$$

The set of eigenfunctions $\{\phi_i\}$ of $k(.,.)$ is a basis for the RKHS $\mathcal{H}_K$, i.e., all elements $f \in \mathcal{H}_k$ have an expansion

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} c_i \phi_i(\mathbf{x}) \tag{3.28}$$

with the contraint that

$$\|f\|^2 = \sum_{i=1}^{\infty} \frac{c_i^2}{\gamma_i} < \infty, \tag{3.29}$$

where $\|.\|$ is the norm induced by k. In particular, the coefficients c_i for $f = k(\mathbf{x}, .) \in \mathcal{H}_K$ are

$$c_i = \gamma_i \phi_i(\mathbf{x}). \tag{3.30}$$

The scalar product in $\mathcal{H}_K$ is defined by

$$\langle \sum_{i=1}^{\infty} c_i \phi_i(.), \sum_{i=1}^{\infty} d_i \phi_i(.) \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{c_i d_i}{\gamma_i} \tag{3.31}$$

It is easy to see that under the constraint of Eq. 3.29 the following properties hold:

$$\langle f, k(., \mathbf{x}) \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{c_i \gamma_i \phi_i(\mathbf{x})}{\gamma_i} = \sum_{i=1}^{\infty} c_i \phi_i(\mathbf{x}) = f(\mathbf{x}) \tag{3.32}$$

$k(., \mathbf{x})$ is therefore known as the *representer of evaluation* at $\mathbf{x}$. Further, for all $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$

$$\langle k(\mathbf{x}, .), k(\mathbf{x}', .) \rangle_{\mathcal{H}_K} \quad = \quad k(\mathbf{x}, \mathbf{x}') \tag{3.33}$$

which is called the *reproducing property*.

The eigen-decomposition can be understood as a generalization of the matrix-eigenvalue problem to kernel functions, i.e., for a discrete set points $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathbb{R}^d$, we can define the *kernel matrix* $\mathbf{K} = k(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \ldots, m$.

The matrix-eigenvalue problem can be recovered by choosing f to be the weighted sum of delta functions at each $\mathbf{x}_i$. In this case f is a limit of functions in $L_2(\mathcal{D})$, i.e.,

$$\mathbf{K} = k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{E} \mathbf{\Lambda} \mathbf{E}^T = \sum_{i=1}^{m} \gamma_i \mathbf{e}_i \mathbf{e}_i^T, \tag{3.34}$$

where $\mathbf{E}$ is the matrix with eigenvectors of $\mathbf{K}$ in its columns and Λ is the diagonal matrix of eigenvalues.

Condition Eq. 3.25 implies, that for any finite subset the corresponding kernel matrix is positive semi-definite, i.e., if $\mathbf{f} = (f(\mathbf{x}'_1), \ldots, f(\mathbf{x}'_m))^T$, then $\mathbf{f}^T \mathbf{K} \mathbf{f} \geq 0$ for all $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_m\} \in \mathcal{D}$. Conversely, approximating the integral with a finite sum and if the sample is chosen sufficiently finely, a negative value will be obtained if Eq. 3.25 does not hold for function f.

Thus, the conditions for Mercer's theorem are equivalent to requiring that for any finite subset of $\mathbf{D}$, the corresponding kernel matrix is positive semi-definite [12].

3.1.3 Feature space induced by Mercer kernel

Consider the feature mapping

$$\mathbf{x} = (x_1, \ldots, x_d) \mapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_i(\mathbf{x}), \ldots), \tag{3.35}$$

where ϕ_i are the basis functions obtained by the eigen-decomposition of the kernel $k(.,.)$ given by Eq. 3.27. The feature vector $\boldsymbol{\phi}(\mathbf{x})$ lives in a hilbert space $\mathbf{F}$ with the inner product given by

$$\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle = \sum_{i=1}^{\infty} \gamma_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}'), \tag{3.36}$$

where the eigenvalues γ_i serve as a weighting for each dimension.

3.1.4 Hypothesis space for learning from a finite sample

For a general nonlinear regression problem, the regularized empirical risk functional has the form

$$\begin{aligned} \mathrm{R}(f) &= \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda \|g\|^2_{\mathcal{H}_k} & (3.37) \\ &= \frac{1}{N} \sum_{i=1}^{N} L(y_i, \sum_{j=1}^{\infty} c_j \phi_j(\mathbf{x}_i)) + \lambda \sum_{j=1}^{\infty} \frac{c_j^2}{\gamma_j}. & (3.38) \end{aligned}$$

It can be shown (Wahba, 1990) that minimization of Eq. 3.38 yields a solution of the form

$$f^* = \arg\min_f \mathrm{R}(f) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, .), \tag{3.39}$$

so that the solution f^* lies in the linear span of functions $k(\mathbf{x}_i, .), i = 1, \ldots, N$, i.e., the representers of evaluation at $\mathbf{x}_i$ (cf. Eq. 3.32), and due to the reproducing property of k we have

$$\begin{aligned} \|f^*\| &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) & (3.40) \\ &= \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}, & (3.41) \end{aligned}$$

where $\boldsymbol{\alpha}$ is the N-vector with components α_i and $\mathbf{K}$ is the $N \times N$ kernel matrix (see Eq. 3.34). Thus, the criterion of Eq. 3.37 becomes a finite-dimensional criterion

$$\mathrm{R}(f) = L(\mathbf{y}, \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{3.42}$$

3.1.5 Duality

The solution f^* (cf. 3.39) is a linear function in $\mathbf{F}$ and can be represented by

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \gamma_i \varphi_i \phi_i(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j k(\mathbf{x}_j, \mathbf{x}), \tag{3.43}$$

where the first term is the primal representation of f and the second is the dual, the relation between the two being

$$\varphi_i = \sum_{j=1}^{N} \alpha_j \boldsymbol{\phi}(\mathbf{x}_j). \tag{3.44}$$

In the primal representation the number of terms is equal to the dimensionality of the feature space (infinite dimensional), while in the dual representation the number of terms is equal to the sample size. Due to this duality the infinite-dimensional problem of Eq. 3.38 reduces to a finite optimization problem as shown in section 3.1.4. This property is also referred to as the *kernel property* in the support-vector machine literatur. Any linear algorithm, that can be formulated only in terms of inner products of the input vectors, can be generalized to run on nonlinear transformed input data, without having to evaluate the nonlinear feature mapping explicitly, by using a kernel function.

3.1.6 Bayesian interpretation

If f is interpreted as a realization of a stationary Gaussian process with zero mean, we can interpret k as the prior covariance function. Then the eigen-decomposition of k yields the eigenfunctions and associated eigenvalues, which correspond to the variances. Looking at Eq. 3.38, we see that functions with small variance are penalized more. These functions have more high-frequency components, and thus "smooth" functions are less penalized. The penalty is the contribution of the prior to the likelihood.

3.1.7 Kernel CCA

In this section we will briefly summarize the formulation of kernel-CCA, which can be used to find nonlinear dependencies between two sets of observations.

It can be shown [46], that for all solutions $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ of Eq. 2.59, the component vectors $\mathbf{w}_x^*, \mathbf{w}_y^*$ lie in the span of the training data (i.e., $\mathbf{w}_x^* \in span(\mathbf{X})$ and

$\mathbf{w}_y^* \in span(\mathbf{Y})$). Under this assumption for each eigenvector $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ solving Eq. 2.59, there exist vectors $\mathbf{f}, \mathbf{g} \in \mathbb{R}^n$, so that $\mathbf{w}_x^* = \mathbf{X}\mathbf{f}$ and $\mathbf{w}_y^* = \mathbf{Y}\mathbf{g}$. Thus, CCA can completely be expressed in terms of dot products. This allows us to reformulate the *Rayleigh Quotient* of Eq. 2.58 using only inner products:

$$\frac{\begin{pmatrix} \mathbf{f}^T & \mathbf{g}^T \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{KL} \\ \mathbf{LK} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}}{\begin{pmatrix} \mathbf{f}^T & \mathbf{g}^T \end{pmatrix} \begin{pmatrix} \mathbf{K}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^2 \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}}, \tag{3.45}$$

where $\mathbf{K}, \mathbf{L}$ are Gram matrices defined by $\mathbf{K}_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ and $\mathbf{L}_{ij} = \mathbf{y}_i^T \mathbf{y}_j$, $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{n \times n}$. The new formulation makes it possible to compute CCA on nonlinearly mapped data without actually having to compute the mapping itself. This can be done by substituting the gram matrices $\mathbf{K}, \mathbf{L}$ by kernel matrices ${\mathbf{K}^{\phi}}_{ij} = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = k_\phi(\mathbf{x}_i, \mathbf{x}_j)$ and ${\mathbf{L}^{\theta}}_{ij} = \langle \theta(\mathbf{y}_i), \theta(\mathbf{y}_j) \rangle = k_\theta(\mathbf{y}_i, \mathbf{y}_j)$. $k_\phi(.,.), k_\theta(.,.)$ are the kernel functions corresponding to the nonlinear mappings $\phi : \mathbb{R}^p \to \mathbb{R}^\infty$ resp. $\theta : \mathbb{R}^q \to \mathbb{R}^\infty$.

The function u of section 3.1.1 corresponds to a linear projection onto $\mathbf{w}_\phi^*$ in feature space and can be computed using only the kernel function, without having to evaluate ϕ resp. θ itself (cf. Eq. 3.43):

$$u(\mathbf{x}) = \sum_{i=1}^{\infty} w_{\phi i}^* \phi_i(\mathbf{x}) = \sum_{i=1}^{n} f_{\phi i} \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i) \rangle = \sum_{i=1}^{n} f_{\phi_i} k(\mathbf{x}, \mathbf{x}_i). \tag{3.46}$$

The projections of $\mathbf{y}$ onto $\mathbf{w}_\theta^*$ (i.e., $v(\mathbf{y})$) are obtained analogously.

Kernel-CCA can be perfomed using singular value decomposition akin to Eq. A.21 as follows: Let

$$\mathbf{C} = \mathbf{K}^\dagger \mathbf{K} \mathbf{L} \mathbf{L}^\dagger, \tag{3.47}$$

where $\mathbf{K}^\dagger$ resp. $\mathbf{L}^\dagger$ is the generalized inverse of $\mathbf{K}$ resp. $\mathbf{L}$. Let

$$\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{3.48}$$

be the SVD of $\mathbf{C}$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_q)$ are orthogonal matrices and $\mathbf{D}$ is a diagonal matrix with singular values. The jth canonical factor pair in the dual space can be obtained as $\mathbf{f}_j = \mathbf{K}^\dagger \mathbf{u}_j$ and $\mathbf{g}_j = \mathbf{L}^\dagger \mathbf{v}_j$. Eq. 3.46 can be used to obtain the projections of the original data $\mathbf{x}_i$ resp. $\mathbf{y}_i$ onto the primal $\mathbf{w}_\phi$ and $\mathbf{w}_\theta$.

Note that using kernel-CCA, we can compute more than $min(p, q)$ (p, q being the dimensionality of the variable $\mathbf{x}$ and $\mathbf{y}$, respectively) factor pairs, which is the limit imposed by classical CCA.

Note also that in the case of nonsingular $\mathbf{K}$ and $\mathbf{L}$ we have all N canonical correlations being equal to one. This is true even in the case of a two-channel process with mutually independent elements of $\mathbf{x}$ and $\mathbf{y}$. In this case the empirical canonical factors $\mathbf{w}_\phi$ and $\mathbf{w}_\theta$ are arbitrary and will not explain the functional relation between the signals $\mathbf{x}$ and $\mathbf{y}$. Under smoothness assumptions meaningful results can be obtained by regularized kernel-CCA as explained in the next section.

3.1.8 Regularization

Ridge-penalty regularization can be incorporated into the kernel-CCA criterion in the same way as in the standard CCA approach by adding a multiple of the identity matrix to the kernel matrices before calculating their inverse, i.e., instead of Eq. 3.47 we can use

$$\mathbf{C} = (\mathbf{K} + \lambda_\phi \mathbf{I})^{-1} \mathbf{K}\mathbf{L} (\mathbf{L} + \lambda_\theta \mathbf{I})^{-1}, \tag{3.49}$$

where λ_ϕ and λ_θ are two separate regularization parameters. For example, if $\lambda_\phi > 0$, the dual coefficients $\mathbf{f}$ are shrunk, whereby in the primal space a greater amount of shrinkage is applied to coefficients $w_{\phi i}$ of basis functions ϕ_i with smaller associated eigenvalues γ_i, i.e., smaller variance. In the case of a Gaussian kernel, these are those basis functions with high frequencies. Besides the regularization effect, a positive regularization parameter value also helps in the case of singular $\mathbf{K}$ resp. $\mathbf{L}$. λ_ϕ and λ_θ can be made model selection parameters to be estimated by cross-validation.

3.2 Summary

We discussed how to nonlinearly extend CCA by using kernelfunctions. Extensions are based on dual CCA formulations. We have shown extensions based on the rayleigh quotient and on SVD. In section 4.1 we will see that Kernel- CCA is a efficient nonlinear feature extractor, which also overcomes some of the limitations of classical CCA. We will apply kernel-CCA to build appearance object models for pose estimation. There, it

will also be shown that kernel-CCA will automatically find an optimal, periodic representation for a training set containing object views ranging from 0 to 360 degrees (i.e., for periodic data).

Chapter 4

Applications

In the previous sections we have introduced enhanced linear and nonlinear regression methods based on CCA and kernel-CCA. In contrast to standard shrinking methods for regression (e.g., ridge regression), which reduce the model complexity (in terms of effective degrees of parameters) by exploiting correlations in the input space, CCA allows to account for correlations in the output space. This is particularly useful when dealing with high dimensional input *and* output data, because correlations in the output space are not exploited by standard regression methods. This is the case in the following applications, with exception of the first application, where the dimensionality of the original output space is 2 and we use kernel-CCA to expand the dimensionality of the output space by a nonlinear transformation.

Because CCA allows to identify the regression relevant low-dimensional subspaces and thus to reduce the effective number of parameters of the regression model, overfitting is avoided despite the small "sample-to-observation" ratio. The meta-parameters (number of factor pairs used, ridge parameter) will be determined by cross validation and generalized cross validation (see Appendix B).

4.1 Manifold Models for Pose Estimation

In this section, we employ nonlinear features obtained by kernel-CCA to build up manifold models for appearance-based pose estimation, i.e., for the task of estimating an object's pose from raw brightness images. The results of this section have been published in [46], [47]. Here, we give a brief summary of the experiments (for details see [46], [47]).

Fig. 4.1 shows some example views of an object, which was acquired with two varying pose parameters (pan and tilt angle of the objects pose relative to the camera).

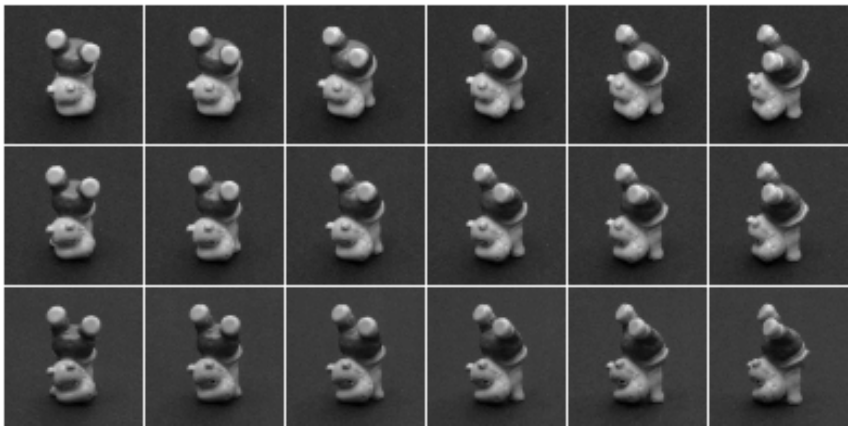
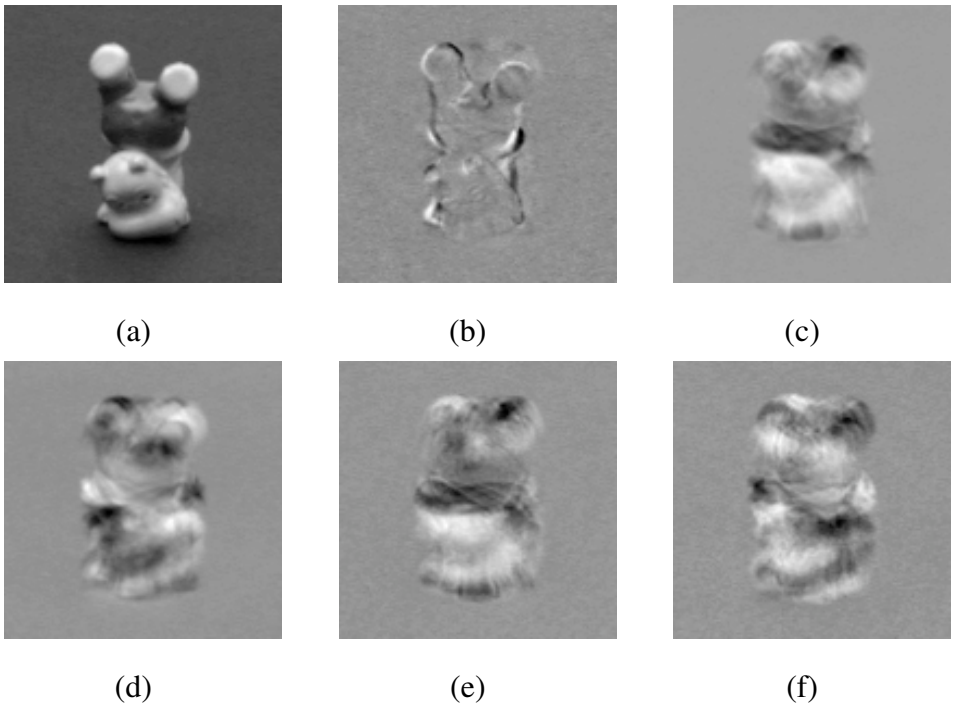


Figure 4.1: A subset of the images that were used in the pose estimation experiment taken with varying pan and tilt angle.

Experiments for appearance based pose estimation were conducted on 7 test objects where the image sets were acquired with two controlled DOFs (pan and tilt).

Figure 4.2 shows the canonical factors obtained in a 1-DOF experiment (the experiment was carried out with varying pan, but a fixed tilt angle). In this case, we had only one pose parameter $y_i, i = 0, \ldots, 179$ (pan angle in the range 0 to 358 degrees, where images were taken in two degrees steps). Figure 4.3 show the projections of the images onto the factor obtained by CCA using the scalar pose representation y_i in degrees resp. the $2d$ trigonometric pose representation $[sin(y_i), cos(y_i)]^T$, as well as the projections onto 2 factors obtained by kernel-CCA. It can be seen from Figure 4.3(d), that kernel-CCA will automatically find an optimal, periodic representation for a training set containing object views ranging from 0 to 360 degrees (i.e., for *periodic* data) with corresponding pose parameters. In this experiment a radial basis function (RBF) kernel was used. The optimal kernel parameter as well as the ridge parameter value was determined by five-fold cross validation.

In a second 2-DOF experiment, for each object, the pose parameters were in range



(a) (b) (c)

(d) (e) (f)

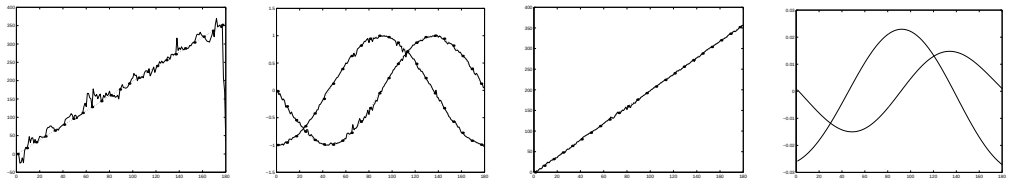
Figure 4.2: (a) Image $\mathbf{x}_1$ of our test object for the 1-DOF case (rotation through 360 degrees) (b) Canonical factor $\mathbf{w}_x^*$ computed a training set which was obtained by taking every 8th image from the original data set. (c),(d) Canonical factors obtained on the same set using a nonlinear (trigonometric) representation of the output parameter space. (e),(f) 2 factors with largest canonical correlation obtained by kernel-CCA.

0 to 90 degrees (pan) and 15 to 43 degrees (tilt), resulting in 690 images per object; the images were of size 64×64.

The visualization of the manifold given in Fig. 4.4 is obtained by plotting the projections of the training set onto the first three empirical canonical factors obtained by kernel-CCA, whereby neighboring (w.r.t. the pose parameters) projections are connected. Fig. 4.4 shows the manifold superimposed by projections of independent test data.

The parametric manifold serves as a starting point for computing pose estimates for new input images. The standard-approach for retrieving these estimates is to resample the manifold using, e.g., bicubic spline interpolation and then to perform a nearest neighbor search on the coefficients obtained for each new image [49].

A comparison of pose estimation performance of regularized kernel-CCA and PCA



(a) (b) (c) (d)

Figure 4.3: Output parameter estimates obtained from feature projections by the linear regression function of the training set. Horizontal axes correspond to the image indices, vertical axes to estimated output parameter values. The dotted line indicates the true pose parameter values. Parameter values of the training set are marked by filled circles. (a) shows estimates using scalar representation of orientation (b) using trigonometric representation. (c) Estimated pose obtained from trigonometric representation using the four-quadrant arc tangent. Note that the accuracy of estimated pose parameters can be improved considerably. (d) Projections onto factors obtained using kernel-CCA: optimal factors can be obtained automatically.

is given in table 4.1 for increasingly smaller training sets. The size of the training set (s) is given as approximate proportion of the whole data set. The optimal parameters for RBF-kernel-width and regularization were determined by five-fold cross-validation.

object	algorithm	$s = 25\%$		$s = 11\%$		$s = 4\%$		$s = 2\%$		$s = 1\%$	
		avg	std	avg	std	avg	std	avg	std	avg	std
(a)	kCCA	0.32	0.61	0.96	1.17	2.64	2.75	3.49	3.51	8.61	7.61
	PCA	3.54	4.76	5.48	6.60	6.79	8.41	4.59	4.89	10.66	9.72
(b)	kCCA	0.22	0.60	0.65	1.02	1.91	2	2.40	1.77	3.09	2.63
	PCA	1.48	1.59	2.28	2.42	2.58	2.84	2.55	1.90	3.48	3.67
(c)	kCCA	0.32	0.57	0.83	1.02	2.68	3.36	2.08	1.74	4.48	4.31
	PCA	2.49	4.27	2.91	3.98	3.74	4.68	4.96	7	7.35	8.46
(d)	kCCA	0.26	0.45	1.36	2.28	3.82	4.01	3.99	4.09	5.62	6.64
	PCA	4.11	5.04	4.59	4.74	7.28	7.65	7	6.02	8.09	9.15
(e)	kCCA	0.14	0.28	0.70	0.66	1.58	1.25	2.05	1.32	2.79	2.16
	PCA	2.43	1.66	2.92	2.43	2.67	2.16	3.12	3.35	5.37	4.36
(f)	kCCA	0.22	0.61	0.65	1.02	1.91	2	2.40	1.77	3.09	2.63
	PCA	2.15	2.15	2.28	2.42	2.58	2.84	2.55	1.90	3.48	3.67
(g)	kCCA	0.02	0.13	0.22	0.44	1.41	1.47	1	1.20	2.56	2.10
	PCA	1.23	1.09	1.63	1.68	2.08	2.01	1.62	1.39	2.68	2.19

Table 4.1: Mean (avg) and standard deviation (std) of the pose estimation error distribution for 7 test objects using training sets of different size.

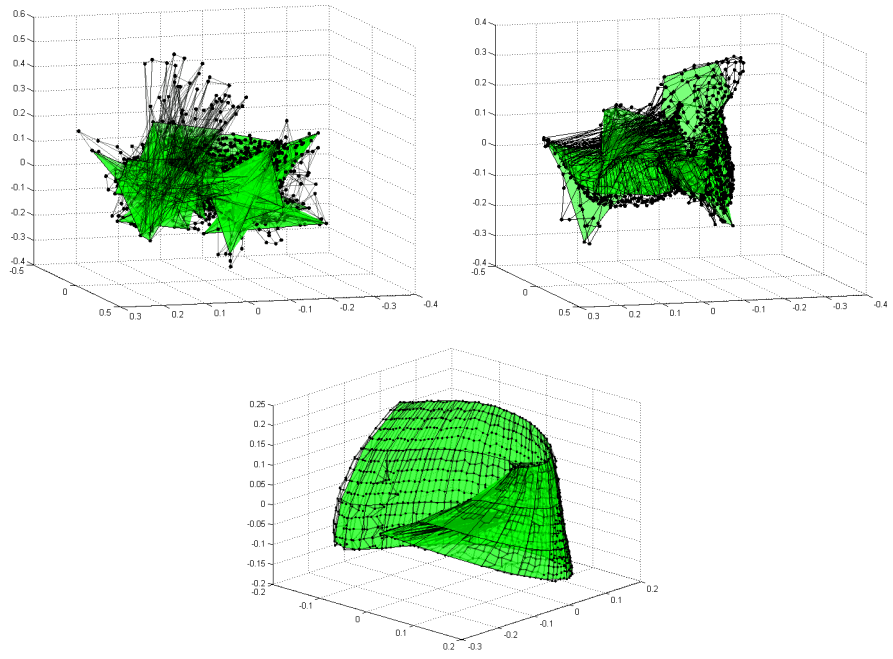


Figure 4.4: The effect of the ridge penalty on the parametric manifold representation obtained by regularized kernel-CCA. The graphs show projections of training and test images onto the first 3 canonical factors, whereby neighboring images w.r.t. to the 2 pose parameters are connected by an edge. The green surface depicts projections of the training images, while the grid depicts projections of the independent test sample. The upper left model was obtained using no ridge penalty at all, the upper right manifold was obtained using a too small ridge value, the lower manifold was obtained using the optimal ridge penalty determined by cross-validation.

4.2 Fast Active Appearance Model matching

In this section, we propose a fast matching method for active appearance models, which is based on CCA. The method was published in [24] and [23].

Active appearance models (AAMs) [17] learn characteristics of objects during a training phase by building a compact statistical model representing shape and texture variation of the object. The use of this a priori knowledge enables the AAM search to yield good results even on difficult and noisy data. AAMs have been employed in various domains like face modeling [27], studying human behavior [37], and medical imaging tasks, like segmentation of cardiac MRIs [48] or the diaphragm in CT data [4], and registration in functional heart imaging [61]. In [60] an extensive overview of existing applications is

given.

4.2.1 AAM search

The goal of the AAM search is to find the model parameters that generate a synthetic image as close as possible to a given input image and to use the resulting AAM parameters for interpretation [17]. Matching the model and target image is treated as an optimization problem, i.e., the problem of minimizing the texture residual w.r.t. model parameters. Provided that the model is roughly aligned with the target image, the relation of texture residuals and parameter updates can be modeled a priori (by offline training) within a certain class of objects [17].

In the original AAM search approach proposed by Cootes et al. [14] the mapping from error images to AAM parameters is modeled by a linear regression approach (linear least squares estimates). In the later proposed optimization approach [17] the regression estimates were replaced by a simplified Gauss-Newton procedure, where the Jacobian matrix is evaluated only once (offline) by numerical differentiation from training data. Throughout this paper we will refer to this as standard approach. Both approaches are similar in the sense that they assume that the error surface can be approximated reasonably well by a quadratic function. The main advantage of the latter approach [17] is that during training not all difference images have to be stored in memory.

Various approaches to increase convergence speed and AAM search result accuracy have been proposed. A review of various search techniques is given in [13]. ShapeAAMs [16] update only pose and shape parameters during search, while gray-level parameters are computed directly from the sample. They converge faster but the failure rate increases. Direct appearance models (DAMs) [35] predict shape parameters directly from texture. The convergence speed of AAMs for tracking applications is investigated in [25].

These modifications of the original approach improve the convergence speed and the quality of the results by either reducing the number of parameters that are to be optimized in a sensible way (DAMs, ShapeAAMs), by saving computation time by reducing the synthesizing steps necessary for the error function calculation (AAM tracking) or by reducing noise in the regression training images (DAMs). However, the basic parameter updating scheme during search is based on the standard approach [17] for all of these

methods. They represent heuristic approaches to AAM fitting, which trade accuracy for efficiency by assuming a constant Jacobian during search.

An alternative is to perform analytic gradient descent. To avoid the resulting inefficiency during search in [44] an inverse compositional approach (ICIA) was proposed. It treats shape and appearance variation independently and is based on a variant of the Lucas-Kanade image alignment algorithm [43], which performs a Gauss-Newton optimization. The texture warp is composed of incremental warps, and thus the assumption that the Jacobian used for parameter prediction stays constant becomes valid. In [31] an extension to this approach (simultaneous ICA) was proposed for combined AAMs. The analytical calculation of the Jacobian is based on the mean and partial derivatives of the AAM parameters.

The approach presented in this paper utilizes training images allowing CCA to extract additional regression relevant information which may be discarded by the purely generative PCA based model.

4.2.2 A fast CCA based search

Our approach follows the original AAM training procedures proposed by Cootes et al. [14] [17]. Essentially, we use a linear regression model of the texture residual vector $\mathbf{r} \in \mathbb{R}^p$ and corresponding AAM parameter displacements $\delta\mathbf{p} \in \mathbb{R}^q$ (p is the size of the synthetic image and q is the number of parameters used in the model). In our approach, however, we use reduced-rank estimates obtained by CCA, instead of ordinary linear least squares regression estimates.

The motivation of CCA is twofold. First, in the standard approach the regression matrix consisting of a large number ($p \times q$) of parameters has to be estimated from a limited number of (noisy) training images. The rank constraint therefore leads to more reliable regression parameter estimates [10, 20].

Second, we assume that the true regression matrix is of lower rank than $\min(p, q)$, because the texture residuals contain regression-irrelevant components, including noise and uncorrelated (higher order) components, which can not be captured by the linear model.

We will show experimentally on different types of data that indeed CCA provides more accurate parameter updates that lead to faster convergence of the AAM search.

4.2.3 Active appearance models

The concept of active appearance models as described in [17] is based on the idea of combining both shape and texture information of the objects to be modeled. First the shape vectors $\mathbf{x^i} = (x_1^i, \ldots, x_n^i, y_1^i, \ldots, y_n^i)^T, i = 1 \ldots j$ of the j training images are aligned using Procrustes analysis. The images are warped to the mean shape $\overline{\mathbf{x}}$ and normalized, yielding the texture vectors $\mathbf{g^i}$. By applying principal component analysis to the normalized data linear models are obtained for both shape, $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s$, and texture, $\mathbf{g} = \overline{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g$, where $\overline{\mathbf{x}}, \overline{\mathbf{g}}$ are the mean vectors, $\mathbf{P}_s, \mathbf{P}_g$ are sets of orthogonal modes of variation (the eigenvectors resulting from PCA) and $\mathbf{b}_s, \mathbf{b}_g$ are sets of model parameters.

A given object can thus be described by $\mathbf{b}_s$ and $\mathbf{b}_g$. As $\mathbf{P}_s, \mathbf{P}_g$ may still be correlated PCA is applied once more using the following concatenated vector

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \overline{\mathbf{x}}) \\ \mathbf{P}_g^T(\mathbf{g} - \overline{\mathbf{g}}) \end{pmatrix}$$

where $\mathbf{W}_s$ is a diagonal scaling matrix derived from the value ranges of the eigenvalues of the shape and texture eigenspaces. This yields the final combined linear model $\mathbf{b} = \mathbf{P}_c\mathbf{c}$, where $\mathbf{P}_c = (\mathbf{P}_{cs}^T, \mathbf{P}_{cg}^T)^T$.

Shape free images and the corresponding shapes defining the deformation of the texture can be expressed directly using $\mathbf{c}$ by $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{W}_s^{-1}\mathbf{P}_{cs}\mathbf{c}$ and $\mathbf{g} = \overline{\mathbf{g}} + \mathbf{P}_g\mathbf{P}_{cg}\mathbf{c}$. To enable the model to deal with rotation, scaling and translation the additional model parameters $\mathbf{t}$, capturing scaling and rotation and $\mathbf{u}$, modeling image contrast and brightness, are introduced. The resulting AAM model represents shape and texture variation of image content utilizing a single parameter vector $\mathbf{p} = (\mathbf{c}^T|\mathbf{t}^T|\mathbf{u}^T) \in \mathbb{R}^q$.

4.2.4 Standard AAM search approach

Provided we have a trained AAM where model parameters $\mathbf{p}$ generate synthetic images $\mathbf{I}_{model}(\mathbf{p})$. The standard search for an optimal match minimizes the difference between a given image $\mathbf{I}_{image}$ and the reconstructed image $\mathbf{I}_{model}(\mathbf{p})$. The search for the model parameters $\mathbf{p}$ can be guided by using knowledge about how the difference images correlate with the parameter displacements. This knowledge is obtained during training.

During each search step the current image residual between the model texture $\mathbf{g}_m(\mathbf{p})$

and the sampled image patch $\mathbf{g}_s(\mathbf{p})$ (warped to the mean shape) is computed using

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s(\mathbf{p}) - \mathbf{g}_m(\mathbf{p}). \tag{4.1}$$

The search procedure aims at minimizing the sum of square (pixel) error

$$\frac{1}{2}\mathbf{r}(\mathbf{p})^T\mathbf{r}(\mathbf{p}). \tag{4.2}$$

Following the standard Gauss-Newton optimization method one approximates (linearizes) Eq. 4.1 using the first-order Taylor expansion

$$\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) \approx \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p},$$

with the ij^{th} element of matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ being $\frac{\partial r_i}{\partial p_j}$.

Building the derivative of Eq. 4.2 w.r.t. $\mathbf{p}$ and setting it to zero gives

$$\delta\mathbf{p} = -\mathbf{R}\,\mathbf{r}(\mathbf{p}), \tag{4.3}$$

where

$$\mathbf{R} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{\dagger},$$

with † denoting the pseudo-inverse and $\mathbf{R}$ has size $q \times k$. Instead of recalculating $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ at every step it is computed once during training using numeric differentiation.

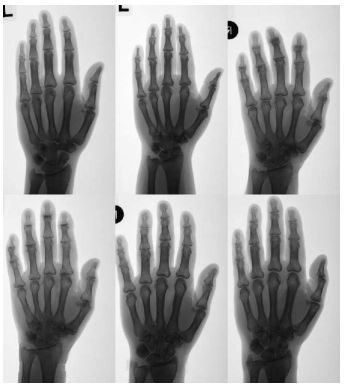
During training each parameter is displaced from its optimal value in h steps from -1 to +1 standard deviations, and a weighted average of the resulting difference images over the training set is built:

$$\frac{dr_i}{dp_j} = \sum_h \omega(\delta p_{jh}) \frac{(r_i(\mathbf{p} + \delta p_{jh}) - r_i(\mathbf{p}))}{\delta p_{jh}}$$

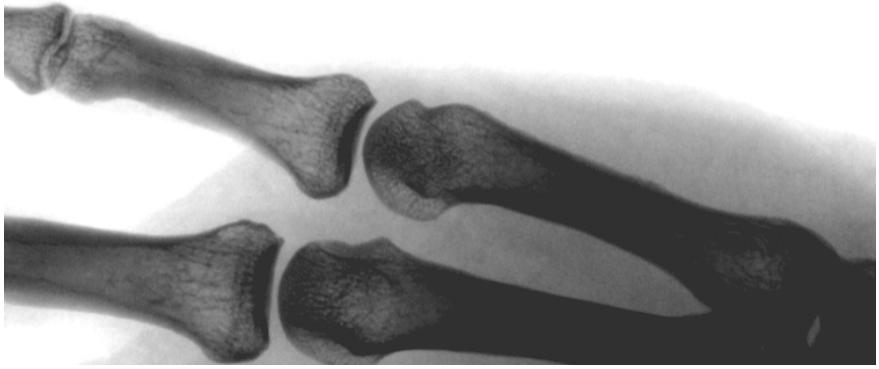
During the actual search, each iteration updates the model parameters using $\mathbf{p}_{next}(s) = \mathbf{p}_{current} + s\,\delta\mathbf{p}_{predicted}$, with $\delta\mathbf{p}_{predicted} = -\mathbf{R}\,\mathbf{r}_{current}$ and s being a scaling factor sequentially chosen from $\mathbf{s}_{steps} = \langle 1, 0.5, 1.5, 0.25, 0.1, 2, 0.025, 0.01\rangle$, as proposed in [17]. At each of these scaling steps, the image patch is compared to the synthesized image $\mathbf{I}_{model}$, which is computationally expensive.

Let $E(\mathbf{p}_{current}) = |\mathbf{r}(\mathbf{p}_{current})|^2 = |\mathbf{g}_s - \mathbf{g}_m|^2$ be the error of the current model. An iteration is declared successful for the first step s to produce an error $E(\mathbf{p}_{next}(s)) <$

(a)



(b)



(c)

Figure 4.5: Datasets used for evaluation: (a) face data, (b) hand data (c) detail of hand data used

$E(\mathbf{p}_{current})$. $\mathbf{p}_{current}$ is then set to $\mathbf{p}_{next}(s)$ and the search continues with the next iteration. If no $\mathbf{p}_{next}(s)$ better then $\mathbf{p}_{current}$ can be found, convergence is declared and $\mathbf{p}_{current}$ is the best estimate for the model parameters. As will be shown in Sec. 4.2.6 our approach eliminates the need for using different step sizes, as the parameter predictions are more accurate.

4.2.5 A fast AAM search based on CCA

In the standard AAM search algorithm a linear function (cf. Eq. 4.3) is used to map texture residuals (difference images) $\mathbf{r}(\mathbf{p}) \in \mathbb{R}^p$ to corresponding parameter displacements $\delta\mathbf{p} \in$

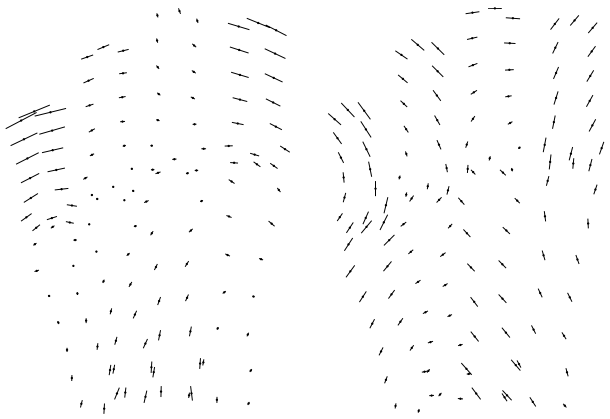


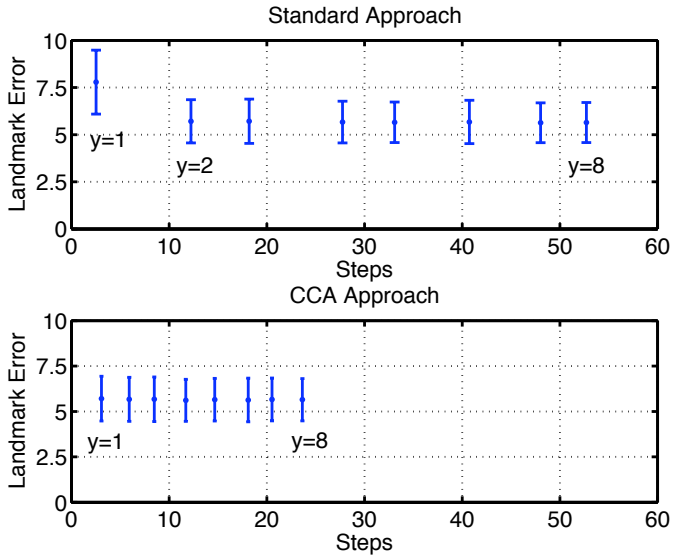
Figure 4.6: First two modes of shape variation for hand bone data.

$\mathbb{R}^q$ (approximating $\mathbf{r}(\mathbf{p})$ by a first-order Taylor series expansion). In our algorithm we extract linear features of $\mathbf{r}(\mathbf{p})$ by CCA of $\mathbf{r}(\mathbf{p})$ and $\mathbf{p}$.

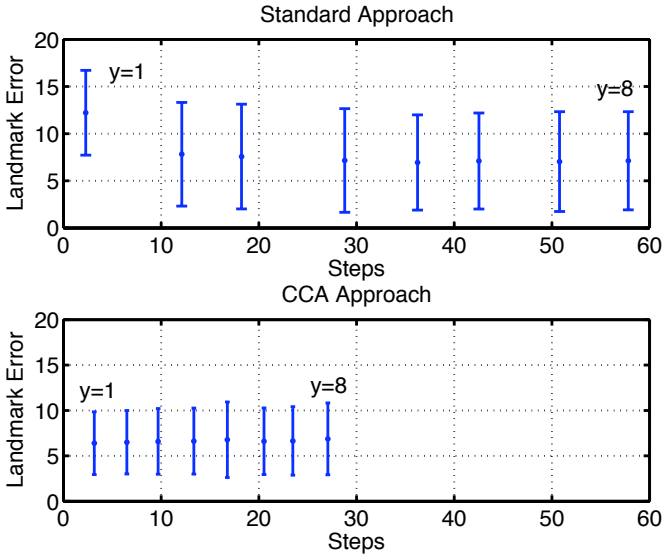
CCA-AAM training During training, instead of computing $\mathbf{R}$ by numeric differentiation, we create training data for CCA. More precisely, the training data is generated as follows: Given the original training images that were used to build the AAM, for each training image we generate a set of synthetic images by perturbing the optimal AAM match, i.e., $\mathbf{r}\left(\mathbf{p}_{opt} + \delta\mathbf{p}\right)$, where the optimum parameter vector $\mathbf{p}_{opt}$ is obtained by mapping the training image texture and shape into the model eigenspace and the components of $\delta\mathbf{p}$ are randomly drawn from uniform distributions from -1 to +1 standard deviation. An overall number of m residual vectors with m corresponding parameter displacement vectors is obtained. We denote the set of random displacement vectors by $\mathbf{P} \in \mathbb{R}^{q\times m}$ and the set of corresponding texture residuals by $\mathbf{G} \in \mathbb{R}^{p\times m}$.

Applying CCA to these two data sets yields empirical canonical factors pairs $\mathbf{W}_g = (\mathbf{w}_g^1, \ldots, \mathbf{w}_g^{k^*})$ and $\mathbf{W}_p = (\mathbf{w}_p^1, \ldots, \mathbf{w}_p^{k^*})$, respectively, where $i = 1 \ldots k^* \leq k$.

These are the (derived) linear combinations which are best predicted by $\mathbf{r}$. By discarding directions with low canonical correlation, i.e., those variates which are poorly predicted by $\mathbf{r}$ we expect to improve overall predictive accuracy and robustness against noise [10] (see also chapter 2). The optimal number of factors k^* is estimated from a separate validation set. After employing CCA, we perform regression on the leading canonical projections $\mathbf{G}_{proj} = \mathbf{W}_g^T\mathbf{G}$ and $\mathbf{P}$. These projections are then used to compute the $p \times k^*$ transformation matrix $\mathbf{l} = \mathbf{P}\mathbf{G}_{proj}^{\dagger}$, where $\mathbf{G}_{proj}^{\dagger} = (\mathbf{G}_{proj}^T\mathbf{G}_{proj})^{-1}\mathbf{G}_{proj}^T$.



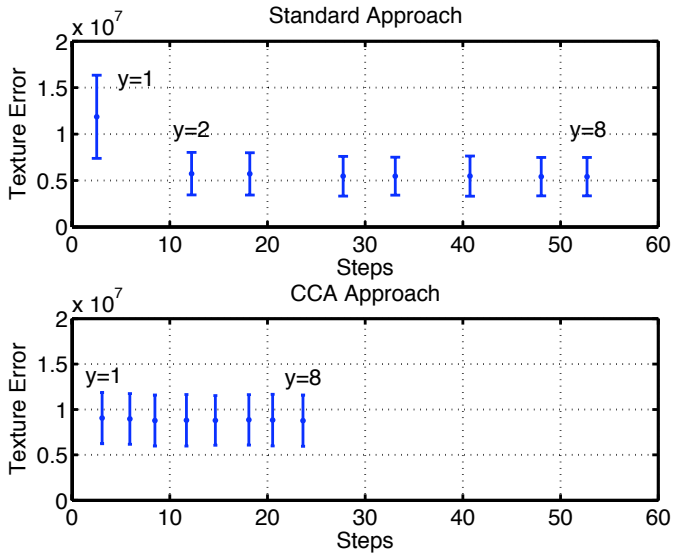
(a) Faces



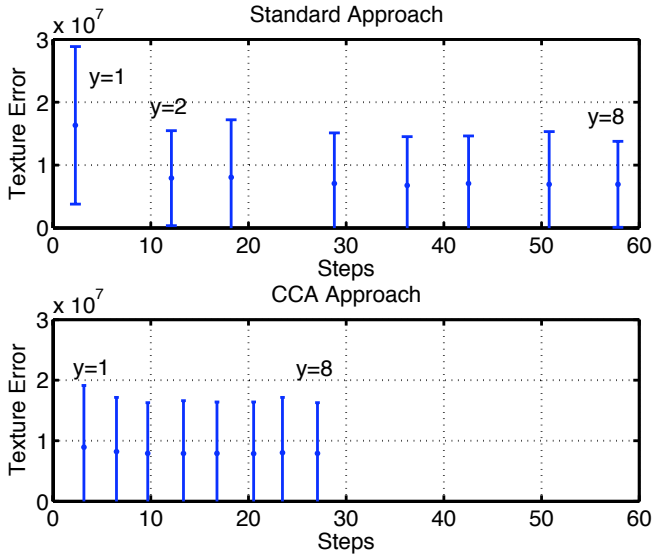
(b) Metacarpal bones

Figure 4.7: Comparison of landmark errors. The 8 bars correspond to y, i.e. the length of s_{steps}, ranging from 1 to 8, from left to right. Note how the CCA yields better (bones) or equal (faces) results faster (at ≈3 steps) than the standard approach (at ≈12 steps).

CCA-AAM search During search, a new displacement prediction has to be obtained at each iteration. Instead of using Eq. 4.3, the prediction $\delta\mathbf{p}_{predicted}$ can be obtained as $\delta\mathbf{p}_{predicted} = \mathbf{l}\mathbf{r}_{proj}$ where $\mathbf{r}_{proj} = \mathbf{W}_g^T\mathbf{r}_{current}$.



(a) Faces



(b) Metacarpal bones

Figure 4.8: Comparison of texture errors. The 8 bars correspond to y (length of s_{steps}), ranging from 1 to 8, from left to right. Again, the CCA approach yields its best results already at $y = 1$ at ≈3 steps while the standard approach needs ≈12 steps for equal error levels.

As $\mathbf{R}_{cca} = \mathrm{l}\mathbf{W}_g^T$ can be pre-computed during training the final formulation of the

prediction function is

$$\delta \mathbf{p}_{predicted}(\mathbf{r}_{current}) = \mathbf{R}_{cca}\mathbf{r}_{current}, \tag{4.4}$$

allowing for an AAM search utilizing the correlations between parameter displacement and image difference as captured by CCA. The computation of these predictions is as fast as for the standard approach, therefore one step of an iteration of CCA search is as fast as one step using Eq. 4.3. For practical application incremental PCA can be used to lower memory requirements. An appealing side-effect is that over-fitting is avoided, which would otherwise be accomplished using regularization techniques for CCA [47]. The outlined approach can also be applied to other kinds of image features (e.g., gradient images or edges).

4.2.6 Experiments

Setup Experiments were conducted on 36 face images [50] and 36 metacarpal bone images manually annotated by a medical expert (Fig. 4.5).The first two modes of shape variation for the bone images are shown in Fig. 4.6. The algorithm was evaluated using 4-fold cross validation. Following the standard AAM training scheme, a set of difference images and corresponding parameter displacements were obtained by randomly perturbing the AAM modes in the interval -1 to +1 standard deviation. While the calculation of $\mathbf{R}$ (cf. Eq. 4.3) by numerical differentiation requires separate variation of each AAM mode, CCA-AAM training allows simultaneous variation of all modes.

To compare search performance in both cases AAM search was performed on the test data using varying lengths of $\mathbf{s}_{steps}$. Scaling factors available during search are chosen by using the first y elements of $\mathbf{s}_{steps}$. As a performance measure we use the total number of steps accumulated over all iterations (cf. Sec. 4.2.4).

Searches were initialized using equal initialization (randomly generated by translations of up to 10 pixels and mean shape and texture) for both approaches. 180 search results provide the data for each of the result bars plotted.

Faster training CCA-AAM training needs fewer synthetic difference images. Using 24 modes for face data and 18 for bone data, 6480 synthetic face images and 4860 synthetic bone images were generated for standard training. For CCA training no improvement could be observed when using more than 200 synthetic difference images.

Thus, although the computation of the CCA is expensive, training is still considerably faster than standard training. For a Matlab implementation on a PowerMac G5 1.8GHz a speed-up factor of 4.9 and 3.5 was achieved.

Faster convergence with equal accuracy In Fig. 4.7 the mean landmark error (point to point distance) over the corresponding number of overall search steps until convergence is depicted. Error bars are 1 standard deviation. The 8 results plotted correspond to y ranging from 1 to 8 as stated above.

In contrast to full rank of 24 (faces) and 18 (bones) CCA employed ranks of 10 and 9 respectively. It can be observed that the CCA convergence speed with almost equal final accuracy is considerably better than the one of the standard approach. The time for an iteration is dominated by the warping in each of the texture comparison steps. The calculation of the parameter prediction (Eq. 4.4 or Eq. 4.3, resp.) amounts for only 9% of the calculation time for a single step. We thus utilize the necessary number of steps as the distinctive measure of convergence speed.

Already with $\mathbf{s}_{steps} = \langle 1 \rangle$, i.e. no scaling of $\delta\mathbf{p}$, the CCA approach yields its best result, in 3.07 steps for the faces data and 3.16 steps for the metacarpals, respectively. The standard approach needs at least $\mathbf{s}_{steps} = \langle 1, 0.5 \rangle$ to perform equally well, requiring 12.23 and 12.11 steps. The CCA approach is thus 3.98 and 3.83 times faster. The mean texture errors in Fig. 4.8 show a similar picture. Again, the CCA approach yields its best results already at $y = 1$. The standard approach is dependent on the availability of further scaling factors ($\mathbf{s}_{steps} = \langle 1, 0.5 \rangle$) to equal this performance. The results are summarized in Tab. 4.2.

Influence of rank reduction In a separate experiment the influence of rank reduction by CCA was investigated. In Fig. 4.9 the dependency of the mean landmark errors after search convergence is depicted for rank k set to $1, 4, \ldots, 24$ for the face data set. It can be seen that for $k = 7$ the search yields the lowest landmark errors, and for $k = 13$ the lowest texture errors. The number of necessary steps is lower than for full rank in both cases.

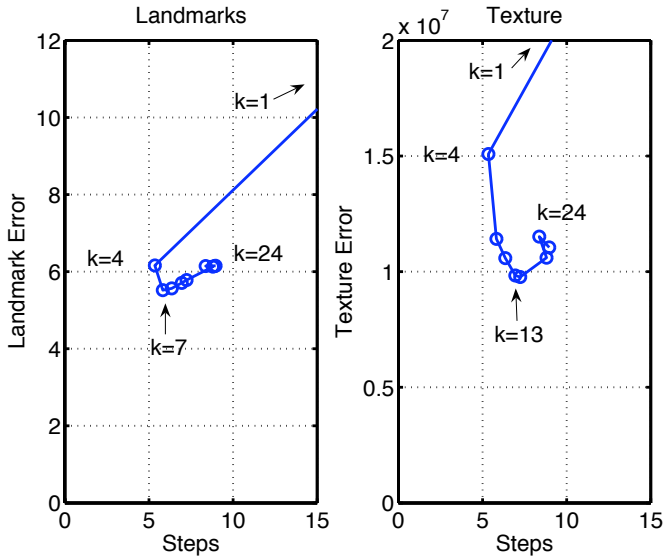


Figure 4.9: Influence of CCA regression rank. Mean landmark error against the number of steps for different choices of k (number of factors for CCA regression) for the face data set.

Faces	Standard	CCA
Training samples	6480	200
Mean landmark error	5.7	5.7
Mean texture error ($\cdot 10^6$)	5.7	9.1
Necessary search steps	12.23	3.07
Search speed-up	1.00	**3.98**

Bones	Standard	CCA
Training samples	4860	200
Mean landmark error	7.8	6.4
Mean texture error ($\cdot 10^6$)	7.9	8.9
Necessary search steps	12.11	3.16
Search speed-up	1.00	**3.83**

Table 4.2: Result summary. Mean landmark and texture errors and corresponding number of search steps for both data sets.

4.2.7 Active feature models

Like Active Appearance models (AAM, see section 4.2.3), *Active feature models* (AFM), which were presented in [41], build a statistical model of shape and texture based on a set

of training images. The model is related to the standard AAM, but instead of representing the entire texture in the images, only local patches are used for landmark localization. Corresponding positions of a set of landmarks are known on all of these images, and a statistical shape model is built based on the training shapes. In contrast to AAM, where the texture is modelled globally, AFM capture texture only locally instead of encoding the entire appearance of the objects in the training set. The texture is not used directly but represented by means of features extracted by descriptors. These values are integrated in a fast search scheme based on CCA, which works in the same way as in the case of AAM.

In addition to restricting the computation to the vicinity of the landmarks - assuming there is a high chance of relevant image content in these regions - it thereby takes advantage from descriptor properties like higher specifity that might be better suited to represent a certain class of images. Instead of performing pairwise matching of points in the images, the behavior of the descriptors stemming from landmark displacements is trained utilizing CCA in order to allow for a fast and efficient search during application. The shape variation of landmark configurations in the training set is captured by a statistical shape model. During search i.e. the identification of the landmarks in a new image, pose and shape parameters of the model are updated according to the trained relationship between displacement and descriptor responses. This results in robust localization of the structure represented by the model. Experimental results for two different data sets are reported to illustrate the effect. Hand/wrist radiographs and face images were used for evaluation (see figure 4.5).

4.2.8 Local features

The image texture is captured by means of local descriptors. Any descriptor can be used, allowing for straightforward adaptation of the algorithm to different data, if descriptors with favorable specificity and robustness with respect to the application are known.

For local image texture description in the experiments reported in [41], steerable filters [29] were employed due to their reliability and low dimensionality. For a given position in the image they extract a feature vector $\hat{\mathbf{f}}$ describing local frequency and directional behavior of the texture. For the steerable filters, among others jets comprising filters with frequencies $\theta \in \{0.3, 0.6, 0.9\}$ and directions $\alpha \in \{0, \pi/4, pi/2, 3\pi/4\}$ (Fig. 4.10) proved to give reliable descriptions. By utilizing complex modulus and argument the

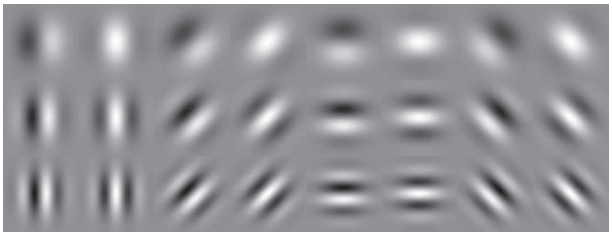


Figure 4.10: Filters for local description of texture.

algorithm works with feature vectors $\hat{\mathbf{f}}_i \in \mathbb{R}^{24}$ for each of n landmark positions that are concatenated to $\mathbf{f} \in \mathbb{R}^{24n}$ used for the model search. Note that the argument gives a good approximation of the shift of edges in the image, a property that will be inherently exploited by the regression described below, resulting in a behavior similar to *active shape models* [15] if strong edges are present in the data.

4.2.9 AFM training

Given a statistical shape model build from training data, AFM search has to fit an instance of the model to a new input image. The search uses knowledge about how the local descriptor responses correlate with the displacement of the model parameters $\mathbf{p}$. During training model parameters are perturbed randomly generating a large number of *displaced* model instances. A functional relation can then be learned from the resulting feature vectors $\mathbf{f}$ and the corresponding parameter displacement $\delta\mathbf{p}$. The basic idea of AFM training and search is similar to the AAM approach. The functional relation is modelled by CCA. In [24] a CCA based AAM search approach was proposed that outperforms the original Gauss-Newton procedure.

Connecting landmark displacements and feature variation with CCA Given a set of training images that were used to build the shape model, for each training image we generate a set of synthetic shape instances by displacing the known correct parameter vector. Let $\mathbf{f}\left(\mathbf{p}_{opt} + \delta\mathbf{p}\right)$ denote the feature vector consisting of descriptor responses at the landmark positions in the image after diplacing the parameter vector $\mathbf{p}_{opt}$ by $\delta\mathbf{p}$, where $\delta\mathbf{p}$ is randomly drawn from uniform distributions from -1 to +1 standard deviations in the model parameter space. An overall number of m feature vectors with m corresponding

parameter displacement vectors is obtained. We denote the set of random displacement vectors by $\mathbf{P} \in \mathbb{R}^{q \times m}$ and the set of corresponding feature vectors by $\mathbf{F} \in \mathbb{R}^{p \times m}$. In our case $q = n_p + 4$ and $p = 24n$.

To learn the functional relation of these two *signals* $\mathbf{P}$ and $\mathbf{F}$ CCA is employed as described in section 4.2.5 for AAM training.

4.2.10 AFM search

AFM search determines an optimal fit of the model to new image data. Instead of a straight-forward matching of descriptor responses AFMs use the shape model to cope with ambiguous and repetitive image content, like medical data, resulting in a reliable identification of landmarks.

Initialization The AFM search is initialized with a rough estimate of the object position in the image. Shape parameters are initialized with $\mathbf{0}$, corresponding to the mean shape.

Search During search, a new prediction for parameter correction is calculated at each iteration. The prediction $\delta\mathbf{p}_{predicted}$ can be obtained as $\delta\mathbf{p}_{predicted} = \mathbf{l}\mathbf{f}_{proj}$ where $\mathbf{f}_{proj} = \mathbf{W}_g^T\mathbf{f}_{current}$.

As $\mathbf{R}_{cca} = \mathbf{l}\mathbf{W}_g^T$ can be pre-computed during training the final formulation of the prediction function is

$$\delta\mathbf{p}_{predicted}(\mathbf{f}_{current}) = \mathbf{R}_{cca}\mathbf{f}_{current}. \tag{4.5}$$

The prediction of landmark positions in the input image is refined by iterating the search procedure, until a convergence criterion is met (e.g., the change of landmark positions falls below a certain treshold).

4.2.11 Experiments

Setup Experimental results are reported for two data sets: 1. For 36 hand/wrist radiographs metacarpal bones and proximal phalanges were annotated by an expert. Correspondences of 128 landmarks on the bone contours were then established by an MDL

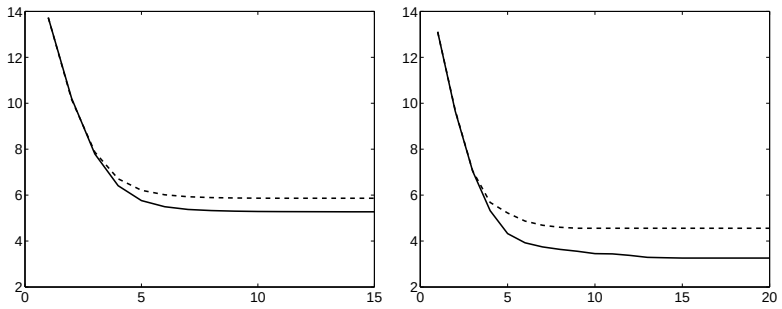


Figure 4.11: Mean landmark error for each iteration; dashed line: AAM, solid line: AFM; (a) face data, (b) hand data.

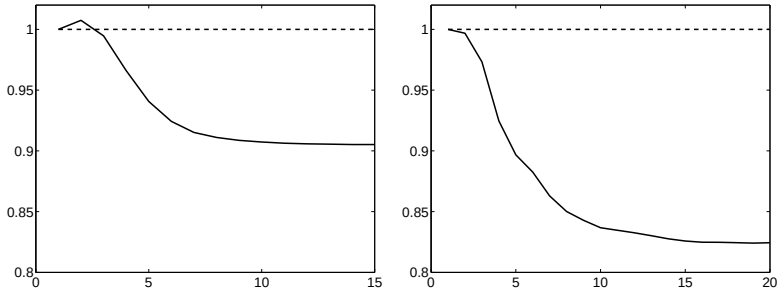


Figure 4.12: Comparison of mean landmark errors for each iteration, AAM performance equals to 1, solid line: AFM search; (left) face data, (right) hand data.

based method [19, 64]. 2. In a set of 36 face images 58 landmarks were annotated manually [**?**, 62]. Search results for AFMs and CCA based AAMs were compared. In order to evaluate the performance of the AFMs to AAMs 4-fold cross validation was performed on the sets. For each image 5 different random initializations were generated, resulting in a total number of 180 searches. Fig. 4.5 shows examples of the two data sets.

Results For evaluation the mean landmark error during search for AFMs and AAMs was compared. Fig. 4.11 shows the mean error over all 180 searches. Since searches differ in length, the error value was set to a fixed value after the minimum was reached, in order to allow for comparison of the performance. For the hand data the mean error after AAM convergence at approx. 8 iterations is 4.56 pixels while for AFMs it is 3.56 px with

convergence at 3.26 px. After 4 iterations AFMs already fall below the final error level of AAMs. For the face data AAMs reach a mean error of 5.87 px after 9 iterations, while AFMs slightly improve this result to 5.28 px. However AFMs fall below 5.86 px with the 4th iteration.

Fig. 4.12 shows a direct comparison between the two error developings. 1 corresponds to the error during AAM search and the AFM error is given as fraction of this value. After 10 iterations the error resulting from AFM search is 83% for the hand data and 90% for the face data.

An interesting observation is the higher performance advantage of AFMs in comparison to AAMs for the hand data as opposed to face images. The appearance of the hand radiographs with landmarks situated on high contrast bone contours benefits the compact steerable filter based texture representation and brings the approach close to ASMs. In the case of face images, the improvement is less pronounced, however, accuracy and search speed are better than with the use of the entire texture. This indicates that a successful application of AFMs is plausible on a variety of image data.

4.3 Recovery of Face Depth Maps from Single Color Images

The results presented in this section were published in [54], [52] and [53].

The recovery of depth and shape information from 2D-face images allows to deal with effects of changing illumination conditions and viewing angle [73]. It can be used, for example, to remove or reduce the effects of illumination and thereby increase the recognition accuracy in complex lighting situations. As another example, consider a face image acquired by a surveillance camera showing the face in a arbitrary viewing angle. Matching with a frontal view image stored in a database could be performed by transforming the stored image, i.e. rendering a synthetic face image with corresponding viewing angle and lighting conditions using a 3D-depth map of the face.

Different approaches exist for recovering shape from 2D face images. Yuille et al. [73] use Singular Value Decomposition (SVD) to reconstruct shape and albedo from multiple images under varying illumination conditions.

By isolating each of the factors that govern the face appearance, their model allows

to predict the shape of faces and generate face images under new illumination conditions with the use of less training data than standard appearance models. Shape from shading [26, 74] algorithms have been applied to face images, where different constraints such as symmetry or (piece-wise) constant albedo are used to render the problem well-posed.

In statistical approaches [1,3,7,11,55] the relationship of shape and intensity is learned by training from a set of examples, i.e. intensity images with corresponding shapes. In [7] a 3D-morphable model (3DMM) is learned from 3D scans, which can be matched with new input images by an iterative optimization procedure. The parameters of the matched model can be used to determine the shape of the input face. A modification of 3DMM is used in [72] in which the face normal vectors are recovered instead of the depth map.

Here, we propose a statistical method for predicting 3D depth maps of faces from frontal view color face images based on CCA. The basic idea of our approach is, that the relationship of depth and face appearance , as a combined effect of illumination direction, albedo and shape can be modeled effectively with a small number of factors pairs, i.e., correlated linear features in the space of depth images and color images. The method is not limited to face images but can generally applied to other classes of objects (surfaces) having similar structure, variability and shadows. A similar approach is taken [42], where the 3D surface of a face is recovered from a single near infrared image using an extension of CCA for that of tensors.

To demonstrate the broad applicability of the method we present a second application, where we use CCA to predict near-infrared (NIR) face texture from color face images, acquired by a standard color CCD camera. Here, we assume a high degree of correlation of the reflectance properties w.r.t the two spectra. In NIR face recognition, predicting NIR images is desirable when the identity of a person in the NIR database has to be determined from a standard color image or a person is to be added to the IR face database when there is no NIR face image available.

The CCA approach allows to take into account the vector spaces of color images and corresponding shapes simultaneously.

We regard the image vector of the RGB color image and the depth maps as two correlated random vectors $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$, where q and p corresponds to the dimensionality of the vector space of the RGB images and depth maps.

In [7] two separate eigenspaces are generated by PCA and a linear regression on the

eigenspace coefficients is used to model the relation of $\mathbf{x}$ and $\mathbf{y}$.

Predicting depth maps respectively NIR images from RGB face data In order to predict depth maps from RGB face images CCA is applied to a set of pairs of RGB image vectors and corresponding depth maps. This results in canonical factor pairs reflecting the inherent correlations between these two *signals*. A sub set of factor pairs $\mathbf{w}^i$ with $i = 1, \ldots, n < k$ is then used for the prediction of depth maps from new RGB images. The sub set corresponds to the canonical factors with the highest canonical correlations in the training set. Applying CCA to the two data sets yields empirical canonical factor pairs $\mathbf{W}_x = (\mathbf{w}_x^1, \ldots, \mathbf{w}_x^n)$ and $\mathbf{W}_y = (\mathbf{w}_y^1, \ldots, \mathbf{w}_y^n)$, respectively.

To predict a depth map in its vector representation $\mathbf{y}$ from a new RGB input image we represent the image as vector $\mathbf{x}$. The prediction $\mathbf{y}_{predicted}$ can be obtained as $\mathbf{y}_{predicted} = \mathbf{l}\mathbf{x}_{proj}$ where $\mathbf{x}_{proj} = \mathbf{W}_x^T\mathbf{x}$. As $\mathbf{R}_{cca} = \mathbf{l}\mathbf{W}_x^T$ can be pre-computed during training the final formulation of the prediction function is

$$\mathbf{y}_{predicted}(\mathbf{x}) = \mathbf{R}_{cca}\mathbf{x}. \tag{4.6}$$

For the application to NIR data we proceed analogously on shape-free NIR and RGB patches obtained by employing active appearance models [18]. Example patches are depicted in Fig. 4.19.

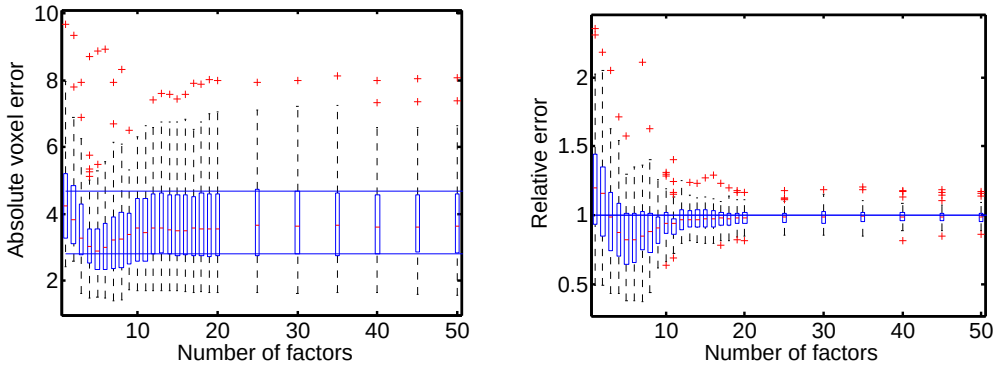
4.3.1 Experimental results

Setup Experiments were performed on two data sets: 1. For the evaluation of 3D Face reconstruction the USF Human-ID 3D Face Database [56] was used. The 3D form of the faces is acquired with a CyberWare Laser scanner. The faces were remapped from a cylindrical depth map with horizontal resolution 1° into a cartesian coordinate system with the z-direction parallel to the radiant going trough the center between the eyes and resolution corresponding the the vertical resolution of the scans. Regions not belonging to the face were discarded. The evaluation was performed on a set of 218 face images. During training, we use 5-fold cross-validation on the training set of 150 images to determine the optimal values for the number of factor pairs and the CCA regularization parameter (for details of how to perform regularization see [47]). The prediction is assessed qualitatively and quantitatively on the test set of the remaining 68 images. 2. For NIR face texture reconstruction we conducted experiments with 150 pairs of RGB face texture and

corresponding NIR data of overall 30 individuals. To obtain texture data the faces were registered based on 51 landmarks. 100 images were used for training and the remaining 50 images for testing the algorithm. To determine the optimal number of factor pairs and the optimal value for the regularization parameter needed for CCA 10-fold cross-validation is performed on the training set. The performance is evaluated on the test set by computing the average pixel error where we using gray-values between 0 and 255.

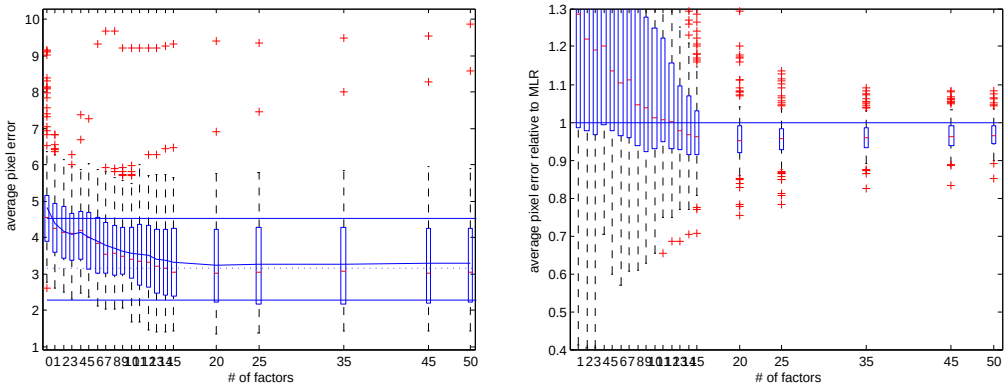
Results for 3D data The maximum possible number of factor pairs k is determined by the number of training data pairs: 150. The quantitative results given in Fig. 4.13 show, that the optimal number of factor pairs is much smaller than 150, namely only 5. The depth reconstruction error improves to 85% of the error of standard regression with 5 factor pairs. Only a fraction of the available factor pairs is sufficient for predicting the texture with higher accuracy than full-rank MLR can achieve. The resulting mean depth error is 6.93 voxels. Note that most of the error stems from the distortions at the boundary of the faces. Fig. 4.16 shows the first two factor pairs corresponding to the 2 largest canonical correlations. Qualitative reconstruction results for test data are shown in Fig. 4.18. In the first 2 columns 3D ground truth and reconstruction results are depicted. In the 3^{rd} and 4^{th} the RGB texture is mapped onto the 3D shapes and in the 5^{th} column the the depth value difference is visualized in the same scale.

Results for NIR data The maximum possible number of factor pairs k is 100. Fig. 4.14 b shows the predicted NIR pixel value error relative to standard regression. The error shows a minor improvement compared to full-rank MLR. However, only 20 factor pairs are sufficient for the predicting the texture with less error. A possible explanation is that due to the good registration of the training images much of the noise in the data has been removed and thus CCA rank-reduction yields only a minor improvement of the error. The optimum mean NIR pixel value error for CCA prediction, with 8 factor pairs, is 3.2 gray-values (95.61% of the MLR result). Qualitative results for test data are shown in Fig. 4.19.



(a) (b)

Figure 4.13: Prediction of 3D depth maps using CCA based regression: (a) Box plot statistics of the average 3D prediction error for a changing number of factor pairs. Each box plot involves the average error of of each of the 150 depth maps of the independent test set measured in voxels. The box blots show the average of the 150 errors (red bar) and quartiles (blue box). The straight solid lines show the quartile of the average error obtained by OLS regression. (b) The average 3D prediction error relative to the error produced by MLR.

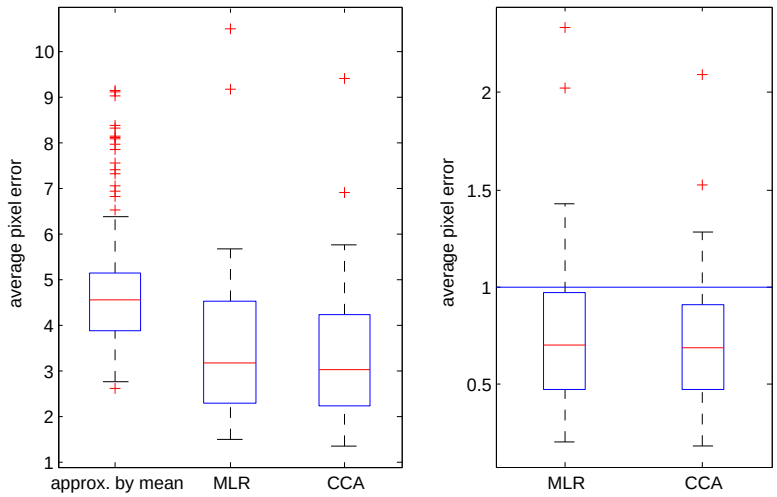


(a) (b)

Figure 4.14: Prediction of NIR face images: (a) Box plot statistics of the average pixel error for a changing number of factor pairs. The Each box plot involves the average error of of each of the 150 depth maps of the independent test set measured in voxels. The straight solid lines show the quartile of the average error obtained by MLR. (b) The average pixel error relative to the error produced by OLS regression.

4.3.2 Experimental comparison with competitors

In this section, we describe quantitative results of an experimental comparison of competing regression methods. The experiments were conducted on the facial scan data. The



(a) (b)

Figure 4.15: Prediction of NIR face images: (a) Box plot statistics of the average pixel error for a changing number of factor pairs. The Each box plot involves the average error of of each of the 150 depth maps of the independent test set measured in voxels. The straight solid lines show the quartile of the average error obtained by MLR. (b) The average pixel error relative to the error produced by OLS regression.

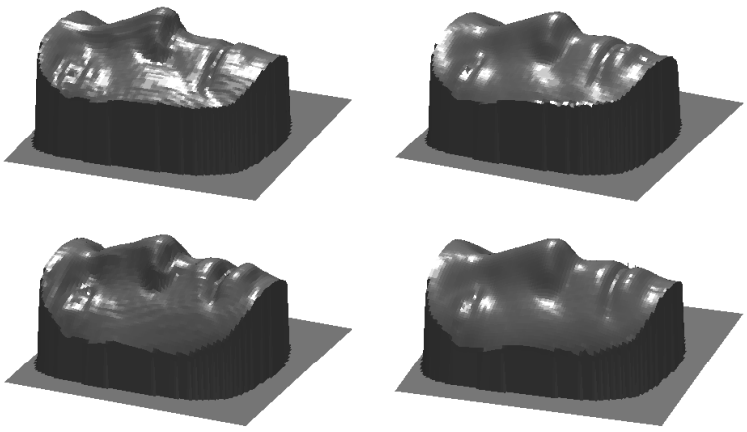
(a) (b) (c) (d)

Figure 4.16: First 2 canonical factor pairs for (a,b) RGB and (c,d) depth data.

regression methods which were tested are the following:

- Rank-k regression with k determined by generalized cross validation (RR-GCV). Training is relatively fast because in contrast to ordinary (v-fold) cross validation, the model has to be fixed only once with the whole sample.

- The Curds & Whey (C&W-GCV) procedure described in section 2.4.3: In a first step, the ridge estimates $\hat{\mathbf{Y}}$ are computed, whereby the ridge parameter value is obtained by generalized cross validation. Then, the optimal shrinkage matrix is



Original surface Reconstructed surface

Figure 4.17: 3D reconstruction: ground truth and CCA based prediction.

Original face patch Reconstructed face patch Difference surface

Figure 4.18: 3D reconstruction: ground truth and CCA based prediction.

calculated based on the canonical analysis of target outcomes $\mathbf{Y}$ and the ridge estimates $\hat{\mathbf{Y}}$. Because of $p, q > N$ we use the singular value decomposition of

$$(\mathbf{Y}\mathbf{Y}^T + \lambda_y \mathbf{I})^{-\frac{1}{2}} \mathbf{Y}\hat{\mathbf{Y}}^T (\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T + \lambda_{\hat{y}} \mathbf{I})^{-\frac{1}{2}} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{4.7}$$

i.e., instead of using the generalized inverse of $\mathbf{Y}\mathbf{Y}^T$ resp. $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$. In order to obtain more stable solutions, we introduce additional bias by the two parameters $\lambda_y, \lambda_{\hat{y}} > 0$, which are additional model parameters determined by five-fold cross

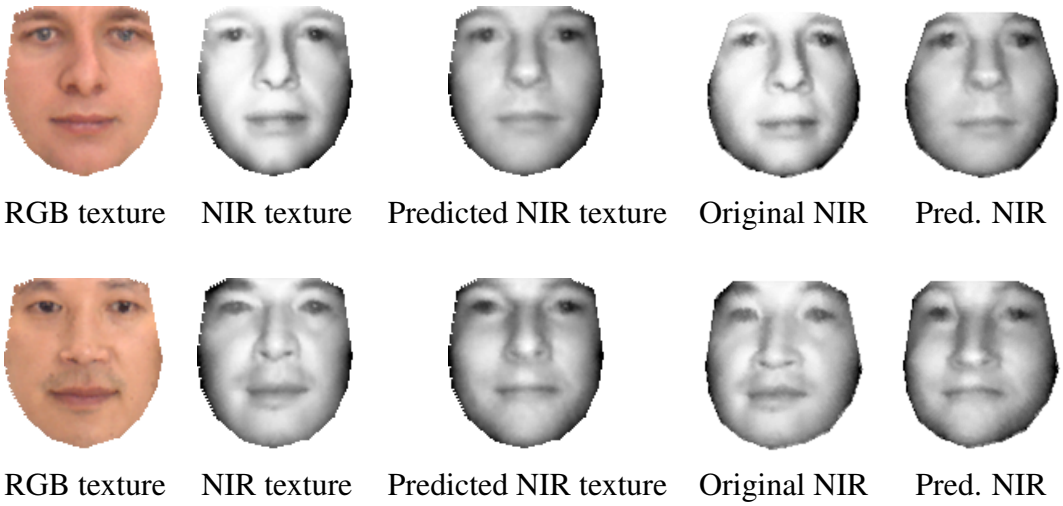


Figure 4.19: NIR texture prediction: ground truth, CCA based texture prediction and texture mapped onto true shape.

validation. For details of the Curds & Whey method, see section 2.4.3.

- Regression based on kernel-CCA (kCCA) as described in section 3.1. A radial basis function kernel was used to transform the inputs (implicit feature mapping ϕ), while the outputs were not transformed. Ridge-penalty regularization was used for shrinking in input (λ_ϕ) and output (λ_y) space. These model parameters as well as the optimal number of factor pairs k and the parameter σ_ϕ of the kernel function were determined by five-fold cross-validation.

- Regression based on standard (linear) CCA: Two independent ridge parameters were used to perform shrinking in the input resp. output space (see section 2.4.4). Their values were determined by five-fold cross validation.

- Regression based on sparse CCA (sCCA), where we have used the algorithm described in [59, 68]. The parameters λ_x and λ_y are the usual ridge parameters, while λ_s determines the regularization of the sparse eigenproblem (see [59]). k corresponds to the optimal number of factor pairs.

In all 5 cases, the criterion function (resp. eigen-problems) can be written in the dual formulation, such that their size depend on the size of the training sample N and not the dimensionality p resp. q. Figure 4.21 summarizes the quantitative results and shows

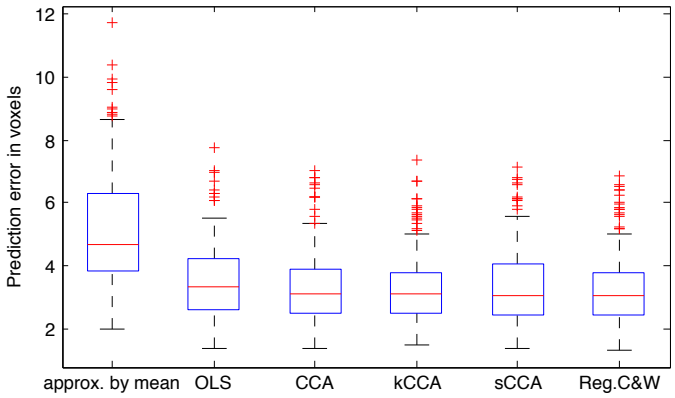


Figure 4.20: Boxplots show the distribution of the depth prediction errors measured in number voxels of deviation from the true facial surface. The errors have been computed on 150 test images. The average error and standard deviation of the errors is shown in figure 4.21.

the optimal model parameters determined by cross-validation as described above. The reduced-rank/shrinking methods (CCA, kCCA, sCCA, C&W) outperform separate OLS regression, but do not show any significant difference w.r.t. performance among each other.

4.4 Summary

In this chapter we have demonstrated the broad applicability of CCA on four different vision tasks. In all four applications a high dimensional response signal was regressed on a high dimensional input. CCA was employed to identify regression relevant subspaces, to reduce the effective number of parameters in a sensible way and thus to avoid over-fitting. Kernel-CCA allowed to automatically find an optimal nonlinear transformation of periodic pose parameters (sine and cosine of the original pose representation). The hyperparameters (number of factor pairs needed, regularization parameters) have to be estimated by cross-validation, which makes training a time-consuming task. Prospects of avoiding cross-validation are described in section 5.1.

	kCCA	CCA	Curds&Whey
average	3.3359	3.3621	3.3210
std. dev.	1.4026	1.3263	1.5967
parameters	$k = 30$, $\lambda_\phi = 10^{-6}$, $\lambda_y = 10^{-9}$	$k = 50$, $\lambda_x = 10^{-3}$, $\lambda_y = 10^{-9}$, $\sigma_\phi = 1.0$	$\lambda_x = 10^{-4}$, $\lambda_y = 10^{-16}$

	RRGCV	sCCA
average	3.3210	3.3704
std. dev.		1.2189
parameters	$\lambda_x = 10^{-4}$	$\lambda_x = 10^{-4}$, $\lambda_y = 10^{-4}$, $\lambda_s = 10^{-16}$, $k = 30$

Figure 4.21: The average and standard deviation of the voxel errors averaged over each of the 150 test images.

Chapter 5

Conclusions

In this thesis enhanced regression methods based on canonical correlation analysis and its non-linear generalization kernel-CCA were applied to object recognition, more specifically pose estimation, prediction of facial surfaces and fast matching of active appearance models to new input images. All of these tasks are regression problems with high dimensional input and output data, where the size of the training set is by far smaller than the number of regression parameters to be estimated. The effective number of parameters can be reduced by Canonical Correlation Analysis. CCA is a versatile tool, that is especially well suited to identify regression-relevant subspaces and can be used for dimensionality reduction and feature extraction.

The canonical coordinate system obtained by CCA is optimal for truncating or shrinking of the original coordinates of the signal spaces. The method works by regression of output variables on input features obtained by CCA. A kernel-version of CCA using a dual formulation of the rayleigh quotient was employed. The performance in various regression tasks was compared to other enhanced regression procedures, such as separate ridge regression, reduced-rank regression and the Curd&Whey method proposed in [10] which performs shrinking in canonical space (and can thus be regarded as smooth version of ridge regression).

The latter two methods are closely related to our proposed regression approach as it can be shown easily that regression parameters are obtained by performing truncation resp. shrinking in the canonical output space.

CCA-AAMs introduce a search algorithm based on CCA, which allows to model efficiently the dependencies between image residuals and parameter correction. Taking

advantage of the correlations between these two signal spaces CCA makes sensible rank reduction possible. It accounts for noise in the training data and thereby yields significant improvements of the AAM search performance in comparison to the standard search approach based on OLS. After computing CCA, linear regression is performed on a small number of linear features which leads to a more accurate parameter prediction during search, eliminating the need for the expensive variable step size search scheme employed in the standard approach.

Empirical evaluation on two data sets shows that the CCA-AAM search approach is up to 4 times faster than the standard approach. As fewer training samples are needed, training is up to 5 times faster. Our approach can be adopted in most of the existing extensions of the original AAM search approach based on linear regression.

In section 4.1, we applied kernel-CCA to an object pose estimation problem. **Manifold models** based on kernel-CCA show superior performance when compared to standard regression and CCA-based regression as well as traditional manifold models based on principal component analysis (parametric eigenspace models). It was also shown that kernel-CCA will automatically find an optimal, periodic representation for a training set containing object views ranging from 0 to 360 degrees (i.e., for *periodic* data).

5.1 Outlook

In the course of the experiments described in chapter 4 most of the ridge parameters where determined by cross validation, which is a time consuming task. A probabilistic interpretation of CCA akin to *probabilistic PCA* [6] allows to formulate CCA as a maximum log likelihood problem, and regularized CCA as the problem of maximization of a log posterior. In [2] a probabilistic interpretation of CCA as a latent variable model for two Gaussian random vectors is given. In [72] a similar formulation of CCA is employed for a variational bayesian approach to CCA. It is reported, that this bayesian version of CCA outperformes standard regularized CCA, due to the fact that model parameters and number of canonical correlations are determined automatically from the training data and that overfitting is avoided. Future research will focus on the use of probabilistic CCA, its combination with the shrinking methods presented in section 2.4 and the development of a kernel version of probabilistic CCA and its application to the vision tasks shown in

chapter 4.

Appendix A

Derivation of CCA

In section 2.2 we have given a formulation of CCA based on the Rayleigh quotient. In this section we will give four alternative derivations of CCA.

A.1 CCA by direct minimization of Eq. 2.55

The derivatives of ρ w.r.t. $\mathbf{w}_x$ are

$$\begin{aligned}
\frac{\partial \rho}{\partial \mathbf{w}_x} &= \frac{(\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y)^{-\frac{1}{2}} \mathbf{C}_{xy} \mathbf{w}_y}{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y} \\
&\quad - \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y (\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y)^{-\frac{1}{2}} \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y}{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y} \\
&= (\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y)^{-\frac{1}{2}} \left(\mathbf{C}_{xy} \mathbf{w}_y - \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x} \right) \\
&= \|\mathbf{w}_x\|^{-1} (\hat{\mathbf{w}}_x^T \mathbf{C}_{xx} \hat{\mathbf{w}}_x \hat{\mathbf{w}}_y^T \mathbf{C}_{yy} \hat{\mathbf{w}}_y)^{-\frac{1}{2}} \left(\mathbf{C}_{xy} \hat{\mathbf{w}}_y - \frac{\hat{\mathbf{w}}_x^T \mathbf{C}_{xy} \hat{\mathbf{w}}_y}{\hat{\mathbf{w}}_x^T \mathbf{C}_{xx} \hat{\mathbf{w}}_x} \right) \\
&= \frac{a}{\|\mathbf{w}_x\|} \left(\mathbf{C}_{xy} \hat{\mathbf{w}}_y - \frac{\hat{\mathbf{w}}_x^T \mathbf{C}_{xy} \hat{\mathbf{w}}_y}{\hat{\mathbf{w}}_x^T \mathbf{C}_{xx} \hat{\mathbf{w}}_x} \right), \qquad a \geq 0. \qquad \text{(A.1)}
\end{aligned}$$

By exchanging x and y we get

$$\frac{\partial \rho}{\partial \mathbf{w}_y} = \frac{a}{\|\mathbf{w}_y\|} \left(\mathbf{C}_{yx} \hat{\mathbf{w}}_x - \frac{\hat{\mathbf{w}}_y^T \mathbf{C}_{yx} \hat{\mathbf{w}}_x}{\hat{\mathbf{w}}_y^T \mathbf{C}_{yy} \hat{\mathbf{w}}_y} \right). \qquad \text{(A.2)}$$

Setting the derivatives to zero gives a system of equations

$$\mathbf{C}_{xy} \hat{\mathbf{w}}_y = \rho \lambda_x \mathbf{C}_{xx} \hat{\mathbf{w}}_x, \qquad \text{(A.3)}$$

$$\mathbf{C}_{yx} \hat{\mathbf{w}}_x = \rho \lambda_y \mathbf{C}_{yy} \hat{\mathbf{w}}_y \qquad \text{(A.4)}$$

where

$$\lambda_x = \lambda_y^{-1} = \sqrt{\frac{\hat{\mathbf{w}}_y^T \mathbf{C}_{yy} \hat{\mathbf{w}}_y}{\hat{\mathbf{w}}_x^T \mathbf{C}_{xx} \hat{\mathbf{w}}_x}}. \tag{A.5}$$

By substituting Eq. A.3 into Eq. A.4 (and vice versa) we can rewrite the equation system as

$$\begin{aligned} \mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\hat{\mathbf{w}}_y &= \rho^2 \hat{\mathbf{w}}_y & \text{(A.6)} \\ \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\hat{\mathbf{w}}_x &= \rho^2 \hat{\mathbf{w}}_x. & \text{(A.7)} \end{aligned}$$

Thus, the factor pairs $\mathbf{w}^i$ can be obtained as solutions (i.e., eigenvectors) of the generalized eigen-problem given by Eqs. A.6 and A.7. The extremum values $\rho(\mathbf{w}^i)$, which are referred to as *canonical correlations*, are obtained as the corresponding eigenvalues.

A.2 CCA by constrained optimization

The same result can be obtained by solving a constrained optimization problem: Note that ρ is not affected by the length of $\mathbf{w}_x$ and $\mathbf{w}_y$ and thus we can constrain x and y to have unit within set variance. Therefore, instead of maximizing Eq. 2.55 directly, we can maximize

$$\rho = \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y. \tag{A.8}$$

subject to the constraints

$$\begin{aligned} \mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x &= 1, \\ \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y &= 1. & \text{(A.9)} \end{aligned}$$

The Lagrangian function becomes

$$\begin{aligned} L(\mathbf{w}_x, \mathbf{w}_y, \alpha_1, \alpha_2) = \; & \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y + \\ & \frac{1}{2}\alpha_1(1 - \mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x) + \\ & \frac{1}{2}\alpha_2(1 - \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y) \end{aligned} \tag{A.10}$$

where we have multiplied the constraints by $\frac{1}{2}$ in order to simplify the subsequent calculations. The derivatives w.r.t. to $\mathbf{w}_x$ and $\mathbf{w}_y$ are

$$\frac{\partial L}{\partial \mathbf{w}_x} = \mathbf{C}_{xy}\mathbf{w}_y - \alpha_1 \mathbf{C}_{xx}\mathbf{w}_x \tag{A.11}$$

$$\frac{\partial L}{\partial \mathbf{w}_y} = \mathbf{C}_{yx}\mathbf{w}_x - \alpha_1 \mathbf{C}_{yy}\mathbf{w}_y. \tag{A.12}$$

Setting the derivatives to zero gives

$$\alpha_1 \mathbf{C}_{xx}\mathbf{w}_x - \mathbf{C}_{xy}\mathbf{w}_y = 0 \tag{A.13}$$

$$\alpha_2 \mathbf{C}_{yy}\mathbf{w}_y - \mathbf{C}_{yx}\mathbf{w}_x = 0 \tag{A.14}$$

which is equivalent to Eqs. A.3 and A.4. Multiplying the Eqs. A.13 and A.14 from left with $\mathbf{w}_x$ and $\mathbf{w}_y$ respectively gives $\alpha_1 = \alpha_2 = \rho = \mathbf{w}_x^T \mathbf{C}_{xy}\mathbf{w}_y$.

A.3 CCA as a linear least squares problem

Looking at Eqs. A.8 and A.9 we see that the empirical canonical correlation is one if the homogeneous linear system

$$\mathbf{w}_x^T \mathbf{X} = \alpha \mathbf{w}_y^T \mathbf{Y} \tag{A.15}$$

has a nontrivial solution. Eq. A.15 follows from the sample versions of Eqs. A.8 and A.9. To see this we formulate CCA as the following optimization problem: Minimize

$$\|\mathbf{w}_x^T \mathbf{X} - \mathbf{w}_y^T \mathbf{Y}\|^2 \tag{A.16}$$

subject to

$$\mathbf{w}_x^T \mathbf{X}\mathbf{X}^T \mathbf{w}_x = 1, \mathbf{w}_y^T \mathbf{Y}\mathbf{Y}^T \mathbf{w}_y = 1. \tag{A.17}$$

Using

$$\begin{aligned} \|\mathbf{w}_x^T \mathbf{X} - \mathbf{w}_y^T \mathbf{Y}\|^2 = {} & \mathbf{w}_x^T \mathbf{X}\mathbf{X}^T \mathbf{w}_x \\ & -2\mathbf{w}_x^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_y + \mathbf{w}_y^T \mathbf{Y}\mathbf{Y}^T \mathbf{w}_y \end{aligned} \tag{A.18}$$

we can replace Eq. A.16 (under the above constraints) by the criterion

$$1 - \mathbf{w}_x^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_y. \tag{A.19}$$

Thus, Eq. A.15 is a necessary and sufficient condition for

$$\min(1 - \mathbf{w}_x^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_y) = 0, \tag{A.20}$$

which is equivalent to the empirical canonical correlation being one ($\rho(\mathbf{w}^*) = 1$). Eq. A.15 might be satisfied only under certain conditions if $N \geq p + q$ (i.e., in the case of linear dependencies). However, if $N < p + q$, we have an underdetermined system and we will obtain $p + q - N$ linearly independent solution vectors. In other words, if $p + q$ is large relative to the sample size, a perfect correlation can always be found. If $p + q \sim N$, the solutions will exhibit high arbitrariness, meaning that minor deviations in the training data will have a huge impact on the empirical canonical factors. A demonstrative example of the influence of noise on the is given in [45].

A.4 CCA by singular value decomposition

Alternatively, $\mathbf{w}^i$ can be obtained by Singular Value Decomposition (SVD) of the cross-correlation matrix of pre-whitened input data, which is given by

$$\mathbf{C} = \mathbf{C}_{xx}^{-\frac{1}{2}} \mathbf{C}_{xy} \mathbf{C}_{yy}^{-\frac{1}{2}}. \tag{A.21}$$

$\mathbf{C}$ is referred to as *coherence matrix* by some authors in the signal processing community [36, 57, 58]. Let

$$\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{A.22}$$

be the SVD of $\mathbf{C}$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_q)$ are orthogonal matrices and $\mathbf{D}$ is a diagonal matrix with singular values. The canonical factors can be obtained as $\mathbf{w}_x^i = \mathbf{C}_{xx}^{-\frac{1}{2}} \mathbf{u}_i$ and $\mathbf{w}_y^i = \mathbf{C}_{yy}^{-\frac{1}{2}} \mathbf{v}_i$. The corresponding canonical correlations are the singular values.

Appendix B

Cross-Validation and Generalized Cross-Validation

K-fold cross validation is a widely used method for estimating the prediction error in the case if where available data is limited. It uses part of the available training data for training and the remaining part to test the fitted model, i.e.,, estimate its prediction error on unseen data. To this end, the available data is partitioned into K roughly equal-sized parts. Let $\kappa : \{1, \ldots, N\} \rightarrow \{1, \ldots, K\}$ be an indexing function mapping each index $i = 1, \ldots, N$ to one of K partitions. $f_\lambda^{-k}(\mathbf{x})$ denotes the function fitted to the training data with the kth part removed, where λ are the tuning parameters of the model. For example, in the case of ridge regression, λ is the ridge parameter (see Section 2.1.9) in the case of regression based on CCA we have $\lambda = (\lambda_x, \lambda_y)$. Then the cross-validated estimate of the prediction error is

$$\text{CV}(\lambda) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{y}_i, \hat{\mathbf{f}}_\lambda^{-\kappa(i)}(\mathbf{x}_i)). \tag{B.1}$$

The case where $K = N$ is referred to as leave-one-out cross-validation. In this case, CV(λ) is approximately unbiased for the true prediction error, but the variance of the estimator CV(λ) can be high. On the other hand, with $K = 5$, which is a typical choice (which has been used throughout the experiments in this thesis), the variance is lower, but the bias can be high.

In any case, λ is chosen according to

$$\lambda^* = \arg\min_\lambda \text{CV}(\lambda). \tag{B.2}$$

To compute the leave-one-out CV score, the learning method has to be applied N times, which is a computational burden. Generalized cross-validation is an approximation to leave-one-out CV and is based on the following observation: We replace the ith element of the training set by its prediction $\hat{\mathbf{y}}_i = \hat{\mathbf{f}}_{\mathbf{w}}^{-\kappa(i)}(\mathbf{x}_i)$ and use the training set $\mathcal{T}_i = \{\mathbf{X}, \mathbf{Y}^i\}$ with

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \tag{B.3}$$

$$\mathbf{Y}^i = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{i-1}, \hat{\mathbf{y}}_i, \mathbf{y}_{i+1} \ldots, \mathbf{y}_N). \tag{B.4}$$

Let $\hat{\mathbf{f}}_{\mathbf{w}}^{\mathcal{T}^i}$ be the estimate of $\mathbf{f}$ obtained by fitting the model to $\mathcal{T}^i$. Then by virtue of the "leave-out-one lemma" [71], we have

$$\hat{\mathbf{f}}_{\mathbf{w}}^{\mathcal{T}_i}(\mathbf{x}_i) = \hat{\mathbf{f}}_{\mathbf{w}}^{-i}(\mathbf{x}_i), \quad i = 1, \ldots, N. \tag{B.5}$$

For linear models we have

$$\hat{\mathbf{Y}} = \hat{\mathbf{f}}_\lambda = \mathbf{Y}\mathbf{H}(\lambda), \tag{B.6}$$

where λ is the parameter determining the complexity (effective degrees of freedom) of the model. Denote the elements of $\mathbf{H}$ by h_{ij}, then

$$\hat{\mathbf{f}}_\lambda(\mathbf{x}_i) = \sum_{j=1}^{N} h_{ij}\mathbf{y}_j, \tag{B.7}$$

and

$$\begin{aligned}
\hat{\mathbf{f}}_\lambda^{-i}(\mathbf{x}_i) &= \hat{\mathbf{f}}_\lambda^{\mathcal{T}_i}(\mathbf{x}_i) && \text{(B.8)}\\
&= \left[\mathbf{Y}^i\mathbf{H}(\lambda)\right]_{.i} && \text{(B.9)}\\
&= \sum_{j=1}^{N} h_{ij}\hat{\mathbf{y}}_j && \text{(B.10)}\\
&= \sum_{\substack{j=1,\ldots,N\\ j\neq i}} h_{ij}\hat{\mathbf{y}}_j + h_{ii}\hat{\mathbf{f}}_\lambda^{-i}(\mathbf{x}_i), && \text{(B.11)}\\
& && \text{(B.12)}
\end{aligned}$$

such that

$$\hat{\mathbf{f}}_\lambda(\mathbf{x}_i) - \hat{\mathbf{f}}_\lambda^{-i}(\mathbf{x}_i) = h_{ii}(\mathbf{y}_i - \hat{\mathbf{f}}_\lambda^{-i}(\mathbf{x}_i)). \tag{B.13}$$

After some rearrangement we have

$$\mathbf{y}_i - \hat{\mathbf{f}}_\lambda^{-i}(\mathbf{x}_i) \;=\; (\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i))(1 - h_{ii}). \tag{B.14}$$

and thus using squared error loss we have

$$\mathrm{CV}(\lambda) \;=\; \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i)}{1 - h_{ii}}\right)^2. \tag{B.15}$$

Replacing h_{ii} in Eq. B.15 results in the generalized cross-validation criterion

$$\mathrm{CV}(\lambda) \;=\; \frac{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i))^2}{(1 - \mathrm{trace}(\mathbf{H}(\lambda))/N)^2}. \tag{B.16}$$

Using the approximation $1/(1-x)^2 \approx 1 + 2x$ for small $\mathrm{trace}(\mathbf{H}(\lambda))/N$ we obtain

$$\begin{aligned}\mathrm{CV}(\lambda) \;=\; & \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i))^2 \\ & + \frac{2}{N}\mathrm{trace}(\mathbf{H}(\lambda))\left(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i))^2\right).\end{aligned} \tag{B.17}$$

Eq. B.17 reveals the similarity between GCV and the *Akaike information criterion* (AIC) (see e.g., [33])

$$\mathrm{AIC}(\lambda) = \mathrm{err}(\lambda) + 2\frac{d(\lambda)}{N}\hat{\sigma}_\epsilon^2, \tag{B.18}$$

where $\mathrm{err}(\lambda)$ is the training error and $d(\lambda)$ is the (effective) number of parameters for each model: Here, $\frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \hat{\mathbf{f}}_\lambda(\mathbf{x}_i))^2$ can be regarded as an estimate of σ_ϵ^2.

Bibliography

[1] JJ Atick, PA Griffin, and AN Redlich. Statistical approach to shape from shading: reconstruction of three- dimensional face surfaces from single two-dimensional images. *Neural Computation*, 1996.

[2] F.R. Bach and M.I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical report, Department of Statistics, University of California, Berkeley, 2005.

[3] Ronen Basri and David Jacobs. Photometric stereo with general, unknown lighting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[4] R. Beichel, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Diaphragm dome surface segmentation in CT data sets: A 3D active appearance model approach. In M. Sonka and J.M. Fitzpatrick, editors, *SPIE: Medical Imaging*, volume 4684, pages 475–484, 2002.

[5] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[7] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. *Auto. Face and Gesture Recognition*, 2002.

[8] Magnus Borga. *Learning Multidimensional Signal Processing*. Linköping Studies in Science and Technology, Dissertations, No. 531. Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1998.

[9] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[10] Leo Breiman and Jerome H. Friedman. Predicting multivariate responses in multiple linear regression. *J.R. Statist. Soc.*, 59(1):3–54, 1997.

[11] M. Castelan and E.R. Hancock. A simple coupled statistical model for 3d face shape recovery. volume 1, pages 231–234, 0-0 2006.

[12] Nello Christianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[13] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *Proc. of BMVC*, volume 2, pages 837–846, 2002.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision ECCV*, volume LNCS 1407, pages 484–498, 1998.

[15] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 266–275, Berlin, 1992. Springer.

[16] T.F. Cootes, G. Edwards, and C.J. Taylor. A comperative evaluation of active appearance model algorithms. In *Proceedings of BMVC*, volume 2, pages 680–689, 1998.

[17] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[18] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[19] Rhodri H. Davies, Carole j. Twining, Tim F. Cootes, John C. Waterton, and Chris J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging*, 21(5):525–537, May 2002.

[20] K.I. Diamantaras and K. Hornik. Noisy principal component analysis. In J. Volaufova and V. Witkowsky, editors, *Measurement '93*, pages 25–33. Institute of Measurement Science, Slovak Academy of Sciences, Bratislava, Slovakia, May 1993.

[21] K.I. Diamantaras and S.-Y. Kung. *Principal Component Neural Networks*. John Wiley & Sons, Inc., 1996.

[22] Konstantinos I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks*. John Wiley & Sons, 1996.

[23] René Donner, Georg Langs, Michael Reiter, and Horst Bischof. CCA-based active appearance model search. In *Computer Vision Winter Workshop*, 2005.

[24] René Donner, Michael Reiter, Georg Langs, Philipp Peloschek, and Horst Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE TPAMI*, 28(10):1690 – 1694, October 2006.

[25] F. Dornaika and J. Ahlberg. Efficient active appearance model for real-time head and facial feature tracking. In *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures AMFG03.*, pages 173 – 180, 2003.

[26] Roman Dovgard and Ronen Basri. Statistical symmetric shape from shading for 3d structure recovery of faces. In *European Conference on Computer Vision*, 2004.

[27] G. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting face images using active appearance models. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition 1998*, pages 300–305, 1998.

[28] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, pages 1–50, 2000.

[29] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[30] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural network architectures. *Neural Computation*, 7:219–269, 1995.

[31] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005.

[32] R. Harshman. Generalization of canonical correlation analysis to n-way arrays. In *Poster at Thirty-forth Annual Meeting of the Statistical Society of Canada*, May 2006.

[33] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. New York: Springer-Verlag, 2001.

[34] A. E. Hoerl and R. W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 8:27–51, 1970.

[35] X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *Proceedings Computer Vision and Pattern Recognition Conference*, volume 1, pages 828–833, 2001.

[36] Yingbo Hua, Maziar Nikpour, and Petre Stoica. Optimal reduced-rank estimation and filtering. *IEEETSP*, 49(3):457–469, 2001.

[37] N. Johnson, A. Galata, and D. Hogg. The acquisition and use of interaction behaviour models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR'98)*, pages 866–871, 1998.

[38] Einat Kidron and Yoav Y. Schechner Michael Elad. Pixels that sound. In *In Proc. Computer Vision and Pattern Recognition*, pages 88–95, 2005.

[39] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[40] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. of Conference on Computer Vision and Pattern Recognition*, 2007.

[41] G. Langs, P. Peloschek, R. Donner, M. Reiter, and H. Bischof. Active feature models. In *Proc. ICPR*, pages 417–420, 2006.

[42] Zhen Lei, Qinqun Bai, Ran He, and S.Z. Li. Face shape recovery from a single image using cca mapping between tensor spaces. pages 1–7, June 2008.

[43] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[44] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135 – 164, November 2004. In Press.

[45] Thomas Melzer. *Generalized Canonical Correlation Analysis for Object Recognition*. PhD thesis, Vienna University of Technology, September 2002.

[46] Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *Proc. of International Conference on Artificial Neural Networks (ICANN)*, pages 353–360, 2001.

[47] Thomas Melzer, Michael Reiter, and Horst Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 39(9):1961–1973, 2003.

[48] Steven C. Mitchell, Johan G. Bosch, Boudewijn P. F. Lelieveldt, Rob J. van der Geest, Johan H. C. Reiber, and Milan Sonka. 3-d active appearance models: Segmentation of cardiac MR and ultrasound images. *IEEE Trans. Med. Imaging*, 21(9):1167–1178, 2002.

[49] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[50] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, may 2004.

[51] A. Pezeshki, L. L. Scharf, M. R. Azimi-Sadjadi, and Y. Hua. Underwater target classification using canonical correlations. In *Proc. of MTS/IEEE Oceans '03*, 2003.

[52] Michael Reiter, Rene Donner, Georg Langs, and Horst Bischof. 3d and infrared face reconstruction from rgb data using canonical correlation analysis. In *Proc. International Conference on Pattern Recognition*. IAPR, 2006.

[53] Michael Reiter, Rene Donner, Georg Langs, and Horst Bischof. Estimation of face depth maps from color textures using canonical correlation analysis. In *Proceedings of Computer Vision Winter Workshop CVWW 2006*, 2006.

[54] Michael Reiter, Rene Donner, Georg Langs, and Horst Bischof. Predicting near infrared face texture from color face images using canonical correlation analysis. In *30th Workshop of the Austrian Association of Pattern Recognition*. AAPR2006, 2006.

[55] John A Robinson and Justen R Hyde. Estimation of face depths by conditional densities. In *British Mashine Vision Conference*, 2005.

[56] Sudeep Sarkar. 3d face database.

[57] L. L. Scharf and C. T. Mullis. Canonical coordinates and the geometry of inference, rate and capacity. *IEEETSP*, 46:824–831, March 2000.

[58] L. L. Scharf and J. K. Thomas. Wiener filter in canonical coordinates for transform coding, filtering, and quantizing. *IEEETSP*, 46:647–654, March 1998.

[59] Bharath K. Sriperumbudur, David Torres, and Gert Lanckriet. Sparse eigen methods by d.c. programming. In *International Conference on Machine Learning 2007*, 2007.

[60] Stegmann. Generative interpretation of medical images. Master's thesis, Technical University of Denmark, 2004.

[61] M. B. Stegmann, H. Ólafsdóttir, and H. B. W. Larsson. Unsupervised motion-compensation of multi-slice cardiac perfusion MRI. *Medical Image Analysis*, 9(4):394–410, aug 2005.

[62] M.B. Stegmann, B.K. Ersboll, and R. Larsen. FAME - a flexible appearance modelling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, 2003.

[63] G. W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory (Computer Science and Scientific Computing) (Computer Science and Scientific Computing)*. Academic Press, June 1990.

[64] Hans Henrik Thodberg. Minimum description length shape and appearance models. In *Proceedings of Information Processing and Medical Imaging; IPMI*, 2003.

[65] Jo-Anne Ting, Aaron D'Souza, and Stefan Schaal. Bayesian regression with input noise for high dimensional data. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[66] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 2001.

[67] Michael E. Tipping. *Bayesian inference: An introduction to principles and practice in machine learn- ing.*, pages 41–62. Advanced Lectures on Machine Learning. Springer, 2004.

[68] D. Torres, B. Sriperumbudur, and G. Lanckriet. Finding musically meaningful words by sparse cca. In *NIPS Workshop on Music, the Brain and Cognition*, 2007.

[69] A. van der Merwe and J. V. Zidek. Multivariate regression analysis and canonical variates. *Canadian Journal of Statistics*, 8:27–39, 1980.

[70] H. D. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 1976.

[71] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[72] C. Wang. Variational bayesian approach to canonical correlation analysis. *IEEE Trans. Neural Networks*, 18(3):905–910, 2007.

[73] A.L. Yuille, D. Snow, and R. Epstein. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability. *International Journal of Computer Vision*, 1999.

[74] W Zhao and R Chellappa. Illumination-insensitive face recognition using symmetric shape-from-shading. In *IEEE CVPR*, 2000.

[75] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:2006, 2004.