Statistical Shape Analysis of Anatomical Structures

by

Polina Golland

B.A., Technion, Israel (1993)
M.Sc., Technion, Israel (1995)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2001
[September 2001]

Author ....
Department of Electrical Engineering and Computer Science
August 10, 2001

Certified by ....
W. Eric L. Grimson
Bernard Gordon Professor of Medical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Statistical Shape Analysis of Anatomical Structures

by

Polina Golland

Submitted to the Department of Electrical Engineering and Computer Science on August 10, 2001, in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In this thesis, we develop a computational framework for image-based statistical analysis of anatomical shape in different populations. Applications of such analysis include understanding developmental and anatomical aspects of disorders when comparing patients vs. normal controls, studying morphological changes caused by aging, or even differences in normal anatomy, for example, differences between genders.

Once a quantitative description of organ shape is extracted from input images, the problem of identifying differences between the two groups can be reduced to one of the classical questions in machine learning, namely constructing a classifier function for assigning new examples to one of the two groups while making as few mistakes as possible. In the traditional classification setting, the resulting classifier is rarely analyzed in terms of the properties of the input data that are captured by the discriminative model. In contrast, interpretation of the statistical model in the original image domain is an important component of morphological analysis. We propose a novel approach to such interpretation that allows medical researchers to argue about the identified shape differences in anatomically meaningful terms of organ development and deformation. For each example in the input space, we derive a discriminative direction that corresponds to the differences between the classes implicitly represented by the classifier function. For morphological studies, the discriminative direction can be conveniently represented by a deformation of the original shape, yielding an intuitive description of shape differences for visualization and further analysis.

Based on this approach, we present a system for statistical shape analysis using distance transforms for shape representation and the Support Vector Machines learning algorithm for the optimal classifier estimation. We demonstrate it on artificially generated data sets, as well as real medical studies.

Thesis Supervisor: W. Eric L. Grimson
Title: Bernard Gordon Professor of Medical Engineering

Readers: Tomás Lozano-Pérez
William (Sandy) M. Wells III
Ron Kikinis

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Anatomical shape, and its variation, has always been an important topic of medical research, but only with the introduction of high resolution 3D imaging, such as MRI and CT, has it become possible to study morphology *in vivo*. Understanding morphological changes caused by a particular disorder can help to identify the time of onset of a disease, quantify its development and potentially lead to a better treatment. Other examples of morphological studies include investigating anatomical changes due to aging through a comparison of different age groups and identifying differences in anatomy between genders.

Originally, image-based statistical studies of morphology were based on simple measurements of size, area and volume. While these can provide some indication of normal variation and anomaly, they are fairly crude and do not capture the entire complexity of anatomical shape. If utilized properly, medical images can provide highly detailed shape information for analysis of morphological variability within a single population or among different groups of subjects. Such analysis is the main focus of this thesis. We consider different ways of extracting information from the images, extend existing classification techniques to yield an explicit description of shape differences between the classes and propose a visualization technique that depicts shape differences between classes as deformations of the original input shapes.

We start the introduction to the problem of statistical shape analysis by presenting example images and explaining the data acquisition procedures. We then discuss the general framework and the individual stages of the analysis, review related work and outline our approach. This chapter concludes with a preview of the results and a brief description of the original contributions in this thesis.

1.1 Data Example

Any statistical study of morphology starts with data collection. One or more volumetric scans are typically acquired for each subject in the study. Then the anatomical structures of interest are segmented, either manually or using automatic algorithms designed for this task [14, 16, 35, 36, 38, 50, 63, 64, 68]. All medical scans presented in this work were processed at the Surgical Planning Laboratory, Brigham and Women's Hospital, Harvard Medical School.

Figures 1-1 shows an example 3D MRI scan of a human head with a segmentation of the hippocampus-amygdala complex, as well as the surface model generated from the segmentation. This scan was acquired as part of a study that investigated morphological changes in the hippocampus-amygdala complex due to schizophrenia. Figure 1-2 shows examples of the hippocampus-amygdala complex for 10 schizophrenia patients and 10 normal controls from this study. Although statistically significant differences in the volume of the structure were previously reported in [59], the two groups look very similar. Our goal is to attempt to localize morphological differences using shape information. In other studies, we might be able to visually identify the shape differences in the images, but would still want to quantify them for assessing the severity of the disorder, effectiveness of the treatment, correlating with symptoms, etc.

While hundreds, or even thousands, of examples are typical in some classification problems, such as text classification or character recognition, data acquisition is an expensive and labor-consuming task in the medical imaging field. In addition to the cost and the time required to obtain the necessary scans, careful screening of the volunteers is needed to control for the factors that might influence the study's outcome, such as social status, education, age, gender, etc. This results in relatively

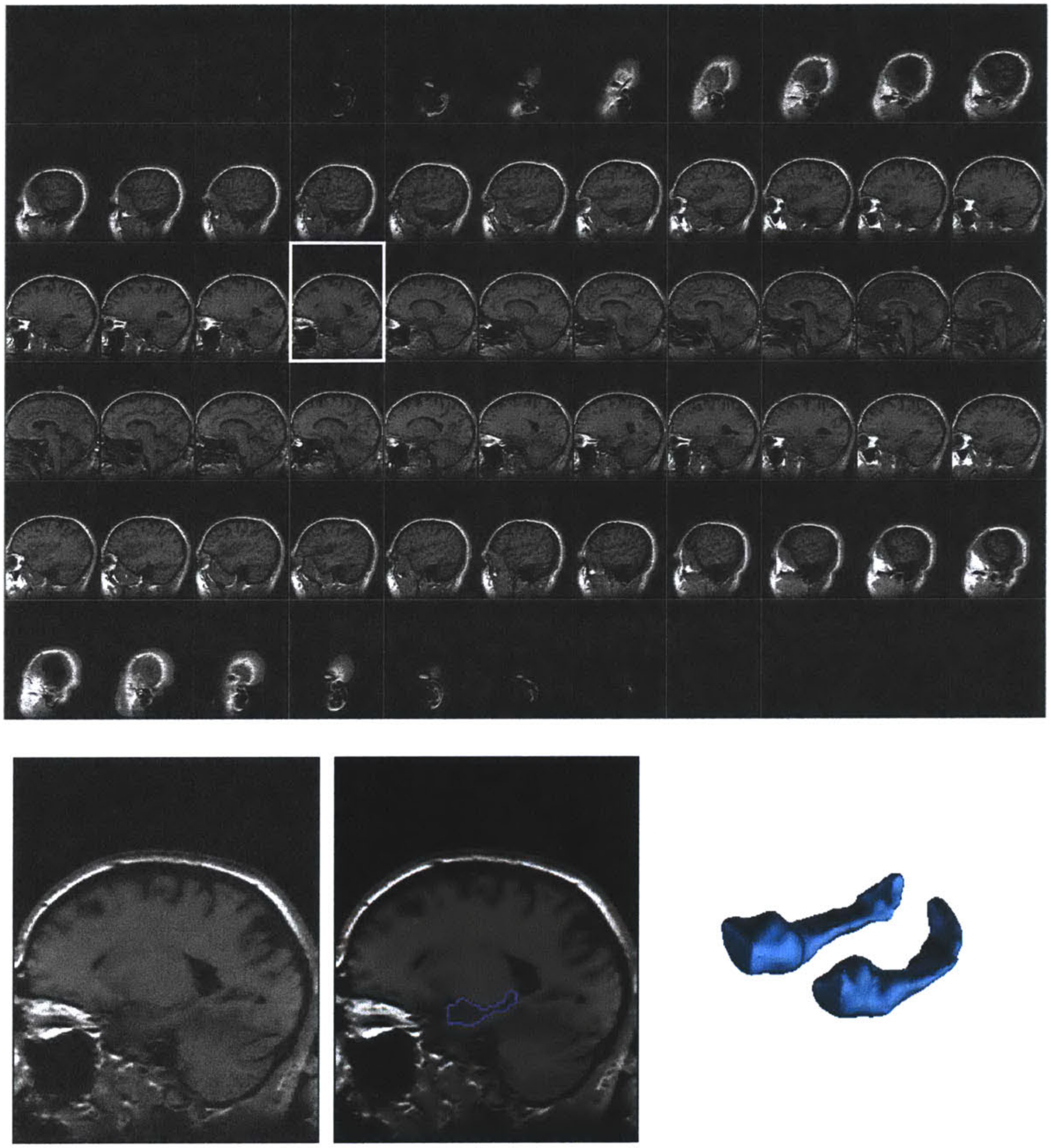
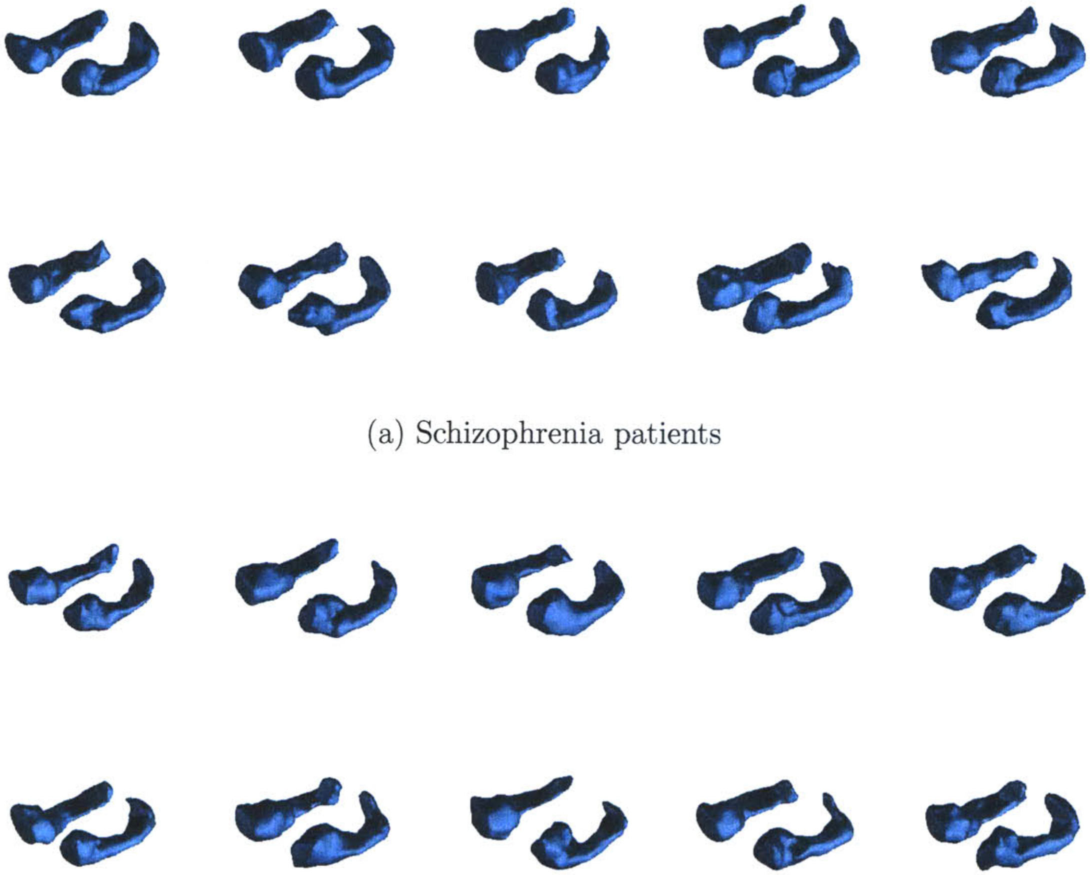


Figure 1-1: Input data example. The top picture shows a set of slices from a sagittal MRI scan of a head arranged in the left-to-right order; every third slice is shown. The bottom picture shows the highlighted grayscale slice (left), the segmentation of the hippocampus-amygdala complex in the same slice (middle) and the 3D model generated form the segmented scan (right).



(a) Schizophrenia patients

(b) Normal controls

Figure 1-2: Examples from the hippocampus-amygdala study in schizophrenia.

small training sets that represent a significant challenge for learning algorithms. All studies reported in this thesis contained fewer than one hundred subjects. Several projects have been started recently at medical centers and universities aiming to create large datasets by combining available medical scans from different groups into a single database (see, for example, [32]).

In the next section, we discuss how collections of input images like those presented in this section are used by statistical analysis techniques to learn about shape variability in the population.

1.2 The Problem of Statistical Shape Analysis

Image-based morphological analysis typically consists of three main steps. First, quantitative measures of shape are extracted from each input image and combined into a feature vector that describes the input shape. The set of feature vectors is then used to construct either a generative model of shape variation within one population or a discriminative model of shape differences between two populations, depending on the problem at hand. This is followed by interpretation of the statistical model in terms of the original shape and image properties. Such interpretation is necessary for expressing the identified shape differences in anatomically meaningful terms of organ development and deformation.

In this section, we describe each of the three stages of the analysis, provide a review of related work and outline our approach. Although our primary interest lies in the area of discriminative analysis, that is, analysis of differences among two or more populations, a lot of work in shape description has been motivated by applications of generative modeling, in which the statistical model of shape variability in a single population is constructed from a set of segmented images and is used either to study the details of shape distribution or to assist segmentation of new images [15, 36, 38, 64]. We include relevant work on generative analysis in our discussion, while pointing out the similarities and the differences between the two modeling techniques and the types of problems they are best suited to solve.

1.2.1 Feature Vector Extraction

Shape analysis starts with extraction of shape features from input images. A great number of shape descriptors have been proposed over the years for use in medical image analysis. They can be classified into several broad families, such as landmarks [7, 15], dense surface meshes [10, 36, 63, 64], skeleton-based representations [23, 27, 50], deformation fields that define a warping of a standard template to a particular input shape [13, 19, 44, 43] and the distance transforms that embed the outline of the object in a higher dimensional distance function over the image [29, 38].

The choice of shape representation depends crucially on the application. For statistical modeling, the two most important properties of a shape descriptor are its sensitivity to noise in the input images and the ease of establishing correspondences between different examples. These determine the amount of noise in the training data, and therefore the quality of the resulting statistical model.

In this work, we choose to use the distance transform for feature extraction, mainly because of its simplicity and its smooth dependence on the noise in the object's boundary and its pose.

1.2.2 Statistical Analysis

Once the feature vectors have been extracted from the images, they are used to construct an appropriate statistical model for the application in question. In the generative case, this is typically done by applying Principal Component Analysis (PCA) to the training data set. The mean feature vector is then considered a "typical shape", and the principal components are assumed to capture the variation within the class. This model has been demonstrated to work quite well for template based segmentation, where the mean is used as a template whose modes of deformation are determined by the principal components [15, 16, 36].

Earlier work on shape differences between populations extended this technique to the discriminative case by using PCA for dimensionality reduction, followed by training a simple (linear or quadratic) classifier in the reduced space [17, 44]. The main concern with this approach is that while PCA might be sufficient to constrain the template deformations in segmentation, the number of training examples is too small to model the probability densities of both classes accurately enough for building an effective maximum likelihood classifier. Furthermore, any PCA-based classification method implicitly assumes that the data was generated by a Gaussian probability distribution and is therefore suboptimal if this assumption is violated, even if the amount of data increases dramatically.

The dimensionality reduction step is typically performed to decrease the number of free parameters to be estimated from the data, based on the principle of choosing the simplest model that explains the data. Traditionally, the number of free parameters (e.g., the number of principal components used by the PCA-based model) has been used as a measure of the model's complexity. The more recently introduced concepts of a classifier capacity and its VC dimension led to a more precise estimation of the model complexity and its generalization performance [65]. The Support Vector Machines (SVMs) algorithm [11, 66] estimates the optimal classifier in the original space while controlling its complexity by using these tighter predictors of the generalization power of the model. In addition to the theoretical reasons for its asymptotic optimality, Support Vector learning has been empirically demonstrated to be robust to overfitting and to generalize well even for small data sets. We use SVMs to estimate the optimal classifier that discriminates between the two groups of input shapes. The classifier function implicitly represents the morphological differences between the two classes.

1.2.3 Model Interpretation In The Image Domain

The resulting statistical model has to be mapped back to the image domain, i.e., analyzed in terms of the input shape or image properties in order to provide the medical researchers with a comprehensible description of the structure in the training data that was captured by the model. In the generative case, this is often done by sampling the implied Gaussian distribution with the mean and the covariance estimated from the data. Alternatively, new examples can be created by varying one principal component at a time. The principal components form a basis in the space of shape deformations described by this linear model.

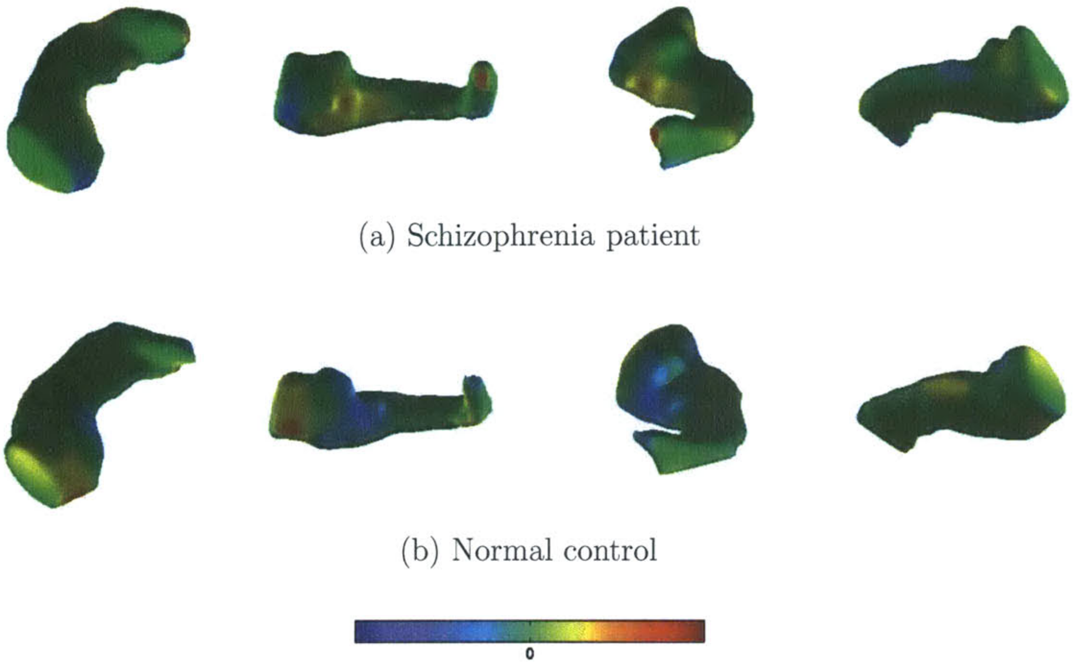
We previously used a similar approach for interpretation of a linear classifier by noting that only the vector's projection onto the normal to the separating hyperplane affects the value of the classification function. Thus, one can vary the projection of the vector while keeping its perpendicular component constant and create a sequence of shapes that look progressively more like the examples from the other class [27].

In some of the earlier work in discriminative modeling [17, 44], the resulting classifier was only used to establish statistical significance of morphological differences between the classes, and the generative models based on PCA were employed for visualization of the shape variation within each group. In this framework, one can investigate details of shape variation within each population, but not directly compare the populations.

In this work, we analyze the classifier in terms of the properties of the original feature vectors and their influence on the output of the classification function. For every input example, we solve for the direction in the feature space that maximizes the change in the classifier's value while introducing as little irrelevant changes into the input vector as possible. We use the intuition mentioned above that in order to understand the differences between the classes captured by the classification function, one should study the function's sensitivity to changes in the input along different directions in the feature space. We derive the sensitivity analysis for a large family of non-linear kernel-based classifiers. The results can be represented in the image domain as deformations of the original input shape, yielding both a quantitative description of the morphological differences between the classes and an intuitive visualization mechanism.

1.3 Example Results

In this section, we provide a preview of the visualization technique used in this work. Our approach yields a description of shape differences detected by the statistical analysis in a form of deformations of the original input shapes. For any shape, the resulting deformation indicates how this shape must be changed to make it more similar to the examples in the opposite population without introducing any irrelevant deformations. Figure 1-3 illustrates the results of the hippocampus-amygdala study mentioned in Section 1.1 for two different subjects, a schizophrenia patient and a normal control. The figure shows the right hippocampus with the estimated deformation "painted" on its surface. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards). We can see substantial deformation of the anterior



(a) Schizophrenia patient

(b) Normal control

Figure 1-3: Example of shape differences detected by the analysis in the right hippocampus. For each input shape, the analysis produces a deformation required to make the shape more similar to the examples in the opposite group. Four views of each structure are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

part of the structure, corresponding to "tucking-in" of the bulbous head (amygdala) in the schizophrenia patient and a similar deformation in the opposite direction in the normal control example.

We can also generate an animation of the deformation in order to help the medical researches to interpret the results of the analysis. Such detailed descriptions of the shape differences between two groups of subjects can help them understand the disease by studying its effects on the organ of interest.

1.4 Contributions

In this work, we present a framework for image-based statistical analysis of morphological differences between two groups of subjects. In addition to demonstration of a fully three-dimensional, automatic framework for feature extraction and statistical analysis, the contributions include

- explicit discriminative modeling of shape differences between two groups based on the Support Vector Machines algorithm, bypassing the dimensionality reduction step typically invoked in the analysis of high dimensional shape descriptors;
- a novel technique for interpretation of the resulting classifier in terms of deformation of the original input shapes;
- demonstration of the method on real morphological studies.

1.5 Outline

In the next chapter, we present a review of existing shape descriptors and their properties relevant to the problem of statistical shape modeling. The purpose of this chapter is to provide a general overview of commonly used descriptors, as well as guidelines for choosing a shape descriptor for a particular application. Chapter 3 explains our choice of the distance transform for extracting shape features and presents a local parameterization of the distance transform space which allows us to represent and visualize changes in the distance transform as deformations of the corresponding boundary surface.

Chapter 4 provides the background on Support Vector learning. We present and discuss the Support Vector Machines algorithm used in this work for construction of a statistical model of differences between the two classes. In Chapter 5, we introduce the discriminative direction as a representation for differences between two classes captured by the classification function. This chapter contains the main technical contribution of this thesis, namely, the analysis of the classifier in terms of changes in the original feature vectors.

In Chapter 6, we explain how to combine shape description with the statistical analysis, demonstrate the approach on artificial examples and report the experimental findings for the real medical studies. The dissertation concludes with a discussion of the lessons learned from the presented experiments and future research directions enabled by the results of this work.

Chapter 2

Shape Description

Image-based morphological analysis starts with extraction of a quantitative description of example shapes from the input images. This chapter reviews existing shape descriptors and their properties relevant to our application of statistical shape analysis. The three-dimensional nature of the images, the lack of occlusions and the high complexity of anatomical shape are among the factors that render this application specific enough that we limit our review to include only shape descriptors that have been successfully used in medical image analysis, leaving out some shape representations used in computer vision and other applications. The purpose of this chapter is to provide a brief overview of existing descriptors and a necessary background for our discussion on distance transforms in the next chapter.

2.1 Existing Descriptors

The wealth of shape descriptors used in medical image analysis includes both parametric models, such as Fourier descriptors or spherical harmonic functions, and numerous non-parametric models: landmark based descriptors, deformation fields, distance transforms and medial axes. In this section, we briefly describe these descriptors and explain how they are extracted from images.

Parametric descriptors. The methods in this family fit a parametric model to a boundary surface in a 3D image, or an outline curve in a 2D image, and use the model parameters as feature vector components. The best known parametric shape descriptors in medical imaging are based on decomposition of the object using a particular functional basis, such the Fourier series [63, 64] or the harmonic functions [10, 36]. The model parameters are typically extracted from segmented images, with an exception of model-based segmentation. In model-based segmentation, a set of segmented images is used to extract the shape parameters, forming a training set. Then a statistical model of the parameter distribution is constructed based on the training set and is used to assist the segmentation of a new image. The result of this procedure is a segmented scan and a set of shape parameters for the newly segmented shape. Thus, the shape parameters are extracted from the grayscale images as a segmentation by-product.

Landmarks. A landmark is a point on the object boundary – a surface in a 3D image, a curve in a 2D image – that can be reliably estimated from an image. Landmarks can be placed manually by the users who employ their knowledge of anatomy to identify "special locations" [7, 15, 16], or detected automatically using geometric properties of the outline surface, such as curvature [49]. Unfortunately, most anatomical shapes have smooth outlines that lack prominent singularity points. This causes an uncertainty in the position of manually placed landmarks and presents a challenge for automatic methods. Algorithms based on manual landmarks are typically limited to 2D images because it is difficult to visualize 3D images in a convenient way for the user to identify the landmarks. Landmarks can be extracted either from grayscale images, as a part of the segmentation process, or from previously segmented scans.

Deformation fields. Descriptors in this class are based on non-rigid matching of a template to an input image. Additional constraints on the resulting deformation field stabilize the inherently under-constrained matching problem. Examples of regularization models that ensure smoothness of the field include thin plate splines [7], elasticity constraints [19, 43, 44] and viscous fluid models [13, 17]. The output of the algorithm is a voxel-by-voxel field of displacements, which can be used as a feature vector describing the input shape. Some techniques in this class match the template to a previously segmented image [7, 43, 44], while others produce the deformation field as a result of the template-based segmentation [13, 19].

Distance transforms. A distance transform, or distance map, is a function that for each point in the image is equal to the distance from that point to the boundary of the object. The boundary is modeled implicitly as a zero level-set of the distance transform. A signed variant of the distance transform, which negates the values of the distance transform outside the object, eliminates the singularity at the object outline, changing linearly as we cross the boundary. Distance transforms have been used in computer vision for medial axis extraction [28, 40], and more recently, in medical image analysis for shape description [29, 38]. The distance transform is computed from a binary segmentation of the object.

Medial axes. A medial axis, or skeleton, of a shape is defined as a set of singularity points of the distance transform that lie inside the shape of interest. Any point in the medial axis has at least two closest points on the outline. Skeletons have been used extensively in computer vision since their introduction by Blum [5]. Shape modeling typically requires using a robust variant of the traditional medial axis that constrains changes in the topology of the skeleton induced by small changes in the boundary [23, 28, 50]. Most skeleton extraction algorithms require a segmented image as input [28, 37, 40, 41, 45]. One exception is the *medial cores* algorithm that simultaneously estimates the boundary of the object and its medial axis based on the intensity gradient in the original grayscale image [23, 50].

This concludes our brief review of shape representations used in medical image analysis. As we can see, many different descriptors have been developed over the years for use in various applications. The application often dictates a set of properties that the shape representation must posses. Such properties are the topic of the next section.

2.2 Taxonomy of Shape Descriptors

In this section, we discuss several important properties of shape descriptors that have to be considered when choosing shape representation for a particular application.

Automatic vs. Manual Feature Extraction. Using manually extracted features is infeasible for large statistical studies, and therefore we would like to concentrate on automatically extracted representations in this discussion. However, one has to be careful in classifying shape descriptors as fully automatic if they require segmented images as input. Since no universal segmentation algorithm has been demonstrated yet, extracting an organ's binary mask from grayscale images could require user involvement.

Raster vs. Vector Descriptors. Raster representations assign a value, or a set of values, to every voxel in a volume and construct feature vectors by concatenating the values into a single list. The simplest example of a volumetric descriptor is the binary segmentation mask [14, 54], other examples include the distance transform and the deformation fields. Vector descriptors are constructed by selecting a set of points to represent an input shape and combining their properties into a feature vector. The components can be point coordinates, intensity gradients, measurements of the shape related to the point of interest, etc. Landmarks, surface meshes and skeletons are examples of vector descriptors. Whether a descriptor is of a vector or a raster type affects how the correspondences among input examples are established, as we explain later in this section.

Coordinate-Based vs. Intensity-Based Descriptors. Intensity-based descriptors are constructed from a certain image function, for example, the binary segmentation mask or the distance transform. In contrast, coordinate-based descriptors use image coordinates to construct a feature vector. Examples of coordinate-based representations include deformation fields, landmarks and medial axes. The spatial structure of the feature space is significantly different for these two types of descriptors. The constraints enforced on the coordinate-based descriptors, e.g., a deformation field, are more directly related to the range of possible shape changes in the input, whereas the intensity based descriptors encode shape implicitly, for example, as a zero level set of the distance transform. A descriptor can be of a hybrid type, such as an appearance model that includes both the coordinates and the intensity at a fixed number of points along the boundary of the object [16, 36].

Closed Under Linear Operators. In many applications, we want to manipulate the feature vectors as part of the analysis. It is therefore important to know whether the feature vectors that describe possible input shapes populate the entire space, and weather such manipulation will always yield a new acceptable shape. For some shape representations, the resulting feature vectors form a linear vector space, i.e., any linear combination of feature vectors is also a valid feature vector that describes a shape that could occur in the image. Landmarks, some versions of medial axes, most parametric shape models and deformation fields belong to this category. Distance transforms do not populate the entire space of real-valued volumes, but rather form a manifold in that space. For some representations, for example, a set of nodes on the object outline, the operation of linearly combining different feature vectors is not even defined. In such cases, if a search in the feature space is required, it must be parametrized using an intermediate representation. Coordinate-based representations are typically closed under linear operations, while the intensity-based descriptors do not necessarily form a linear vector space.

Sensitivity to Noise. Shape description is an inherently noisy process, consisting of imaging, segmentation and feature extraction, with every step introducing errors. Small changes in the input image, such as changes in the patient's pose and errors in segmentation, cause the corresponding feature vector to change as well. Ideally, the representation should be insensitive to "small" noise in the input images. But defining what constitutes small amounts of noise that can be ignored in contrast to shape changes that must be modeled is difficult and often application dependent. For statistical shape analysis, a smooth behavior of the representation in the presence of noise can help us to bound the error in the constructed model. Such smooth behavior is exhibited by most descriptors described in the previous section. The medial axis in its original formulation is very sensitive to small changes in the outline of the object, which was remedied in several robust variants [28, 41, 23]. Another example of a highly sensitive descriptor is the segmentation mask: infinitesimal changes in the object can cause some of the components to change from zero to one.

Some level of insensitivity is often achieved by imposing smoothness constraints on the feature estimates, such as elasticity constraints for deformation fields. In parametric descriptors, the model's sensitivity can often be controlled explicitly by choosing the level of details to be represented, for example by limiting a number of basis functions in the Fourier decomposition.

Alignment and Correspondences. For shape studies, one would like the descriptors to be invariant under a family of rigid transformations, as the object's pose in the scanner should not affect its shape properties. Many raster representations are not invariant under rigid transformations, which is typically mitigated by aligning all shapes, or bringing them into a "canonical" pose, before extracting the features. The alignment also establishes implicit correspondences among features computed from different input images. Most vector descriptors are insensitive to the object's pose, but require establishing correspondences among "feature points" for different example shapes. The correspondences are established either manually [7, 15] or automatically. The automatic procedures for estimating correspondences typically set a reference frame on the surface of an object (or its medial axis) for ordering the points into a list, for example, using the principal axes of inertia [36]. This operation is similar to alignment of raster descriptors, as it often relies on the canonical pose of the objects to match feature points in different examples. Descriptors that produce variable length feature vectors, such as surface meshes that represent object boundaries, cause further complications in the feature matching problem.

The problem of establishing correspondences, or alignment, has not been solved satisfactory for raster or vector descriptors. One of the main obstacles is that it is not clear what the "correct correspondence" between two examples of an anatomical shape is. Most methods use rigid alignment, operating under the premise that the non-rigid differences among the examples are exactly the shape differences we would like to model and study. Since we cannot assume perfect alignment, it is important to understand the descriptor's behavior under noise in the object's pose. Raster descriptors, for which the changes in the feature values are determined by the transformation matrix and the gradient of the volumetric image function, lend themselves to such analysis easier than do vector descriptors, which give rise to a potentially exponential problem of matching among several discrete sets of points describing different example shapes.

To summarize, there are several important, sometimes conflicting, properties of shape descriptors that must be considered when choosing a suitable representation. Unfortunately, the perfect shape descriptor that posses all of them is yet to be demonstrated, and one has to carefully trade-off properties based on the application of interest. The next chapter contains a discussion on our choice of the distance transform as a shape representation and its properties relevant to the problem of statistical shape analysis.

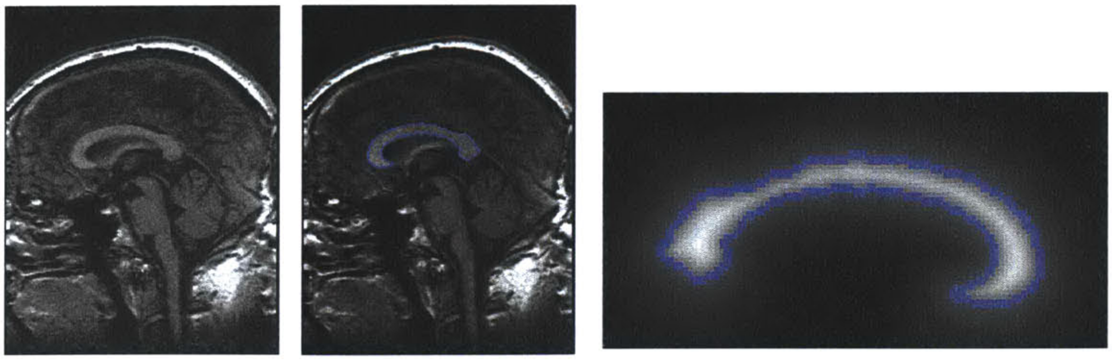
Chapter 3

Distance Transforms as Shape Descriptors

We use the distance transform for shape description in this work. Establishing correspondences between features from different examples in a reliable and well understood manner was important in our application, and therefore raster representations, such as the deformation fields or the distance transform, were more appropriate. We chose the distance transform mainly because of its relative simplicity: it uses only basic geometric relationships to construct the feature vector and can be implemented in a straightforward manner. The main drawback of the distance transform is that it produces feature vectors that do not form a linear vector space. In this chapter, we elaborate on the properties of the distance transform that make it attractive for shape description in statistical analysis and demonstrate a local parameterization of the distance transform space that eliminates the difficulty of manipulating feature vectors that do not populate the entire space.

3.1 Basic Definitions

To remind the reader, the distance transform is a function that for every point in the image is equal to the distance from that point to the object boundary. We use the signed distance transform which eliminates the singularity at the boundary by

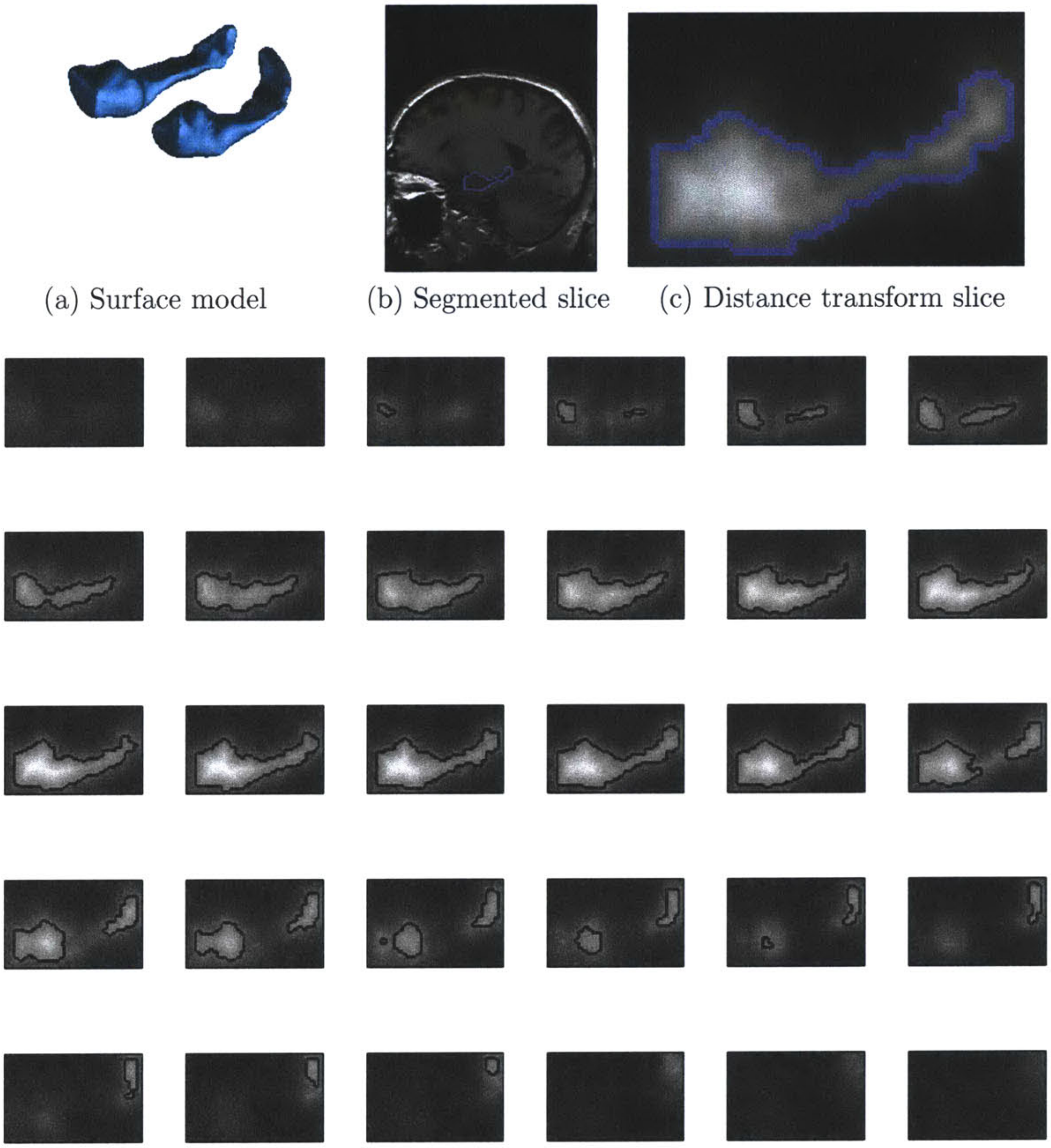


(a) Midsagittal slice (b) Segmentation (c) Distance transform

Figure 3-1: Two-dimensional distance transform for the corpus callosum. The modeling was performed entirely in 2D, as only a cross-section of the corpus callosum was considered in this study. Brighter intensities in the distance transform image correspond to higher values of the distance transform. The values outside the shape were log-scaled for visualization purposes.

negating distance values outside the object. Note that the definition of the distance transform can be applied both to 2D images, for which it was originally introduced, and to 3D volumetric scans. Figure 3-1 shows an example of the two-dimensional distance transform computed for a single slice of a volumetric scan of the corpus callosum, while Figure 3-2 illustrates the three-dimensional distance transform computed for the hippocampus-amygdala example from Figure 1-1. The distance transform and its properties, as well as efficient algorithms for computing it, have been investigated extensively in computer vision [8, 18, 39].

We will say that a point on the boundary *influences* a particular location in the volume if it is the closest boundary point to that location. Obviously, the value of the distance transform at any voxel is equal to the distance from the voxel to the point(s) on the boundary that influence that voxel. The distance transform is a piece-wise linear function. The singularity ridges of the distance transform form the object's skeleton, which is defined as a set of locations inside the shape that have two or more closest points on the boundary. Note that most voxels in the volume are influenced by one boundary point each, with an exception of the skeleton voxels. As we will see in the following sections, the influence relationship plays an important role in understanding the behavior of the distance transform feature vectors in the presence



(a) Surface model (b) Segmented slice (c) Distance transform slice

(d) Volumetric distance transform of the left hippocampus-amygdala.

Figure 3-2: Volumetric distance transform of the left hippocampus-amygdala complex. The distance transform was computed separately for the left and the right hippocampus. Brighter intensities in the distance transform image correspond to higher values of the distance transform. The values outside the shape were log-scaled for visualization purposes.

of noise or small changes in the original shape.

3.2 Sensitivity to Noise and Misalignment

Uncertainty in the feature vectors is caused by errors in the boundary location and inaccuracy of the alignment process. Note that these two factors are not independent, as the alignment procedure operates on the result of segmentation. In this section, we show that the distance transform changes smoothly as errors of both types are introduced into the process of feature vector extraction.

Small displacements of a boundary point cause changes of the distance transform values only at the voxels influenced by that point. Using the triangle inequality, we can show that the magnitude of the distance transform gradient is bounded by one. Consequently, the change in the distance transform values at the influenced voxels is bounded by the magnitude of the point's displacement. An alternative way to arrive at this conclusion is to observe that as a boundary point moves, the change in the distance between any voxel and this point cannot be greater than the magnitude of the point's displacement.

A similar argument can be used to bound errors due to misalignment. Small changes in the object's pose (translation, rotation) induce a rigid displacement field in the volume. Since the magnitude of the distance transform gradient is bounded by one, the change in the distance transform value at any voxel is bounded by the magnitude of the displacement at that voxel. This again places a linear upper bound on the uncertainty in the extracted feature vectors.

In order to reduce the sensitivity of the alignment process to the segmentation errors, we use moments of the distance transform inside the shape to align the images into a canonical representation. In contrast to the moments of shape that weigh all points equally, the moments of the distance transform assign weights proportionally to the distance from the boundary, reflecting our belief that the interior points of the distance transform are estimated more reliably than the points close to the boundary of the object.

3.3 Local Parameterization Using Surface Meshes

The main drawback of using the distance transforms as shape descriptors is that they do not form a linear vector space. Our analysis of shape differences in two populations requires performing local search in the space of input shapes, i.e., investigating a small neighborhood of shapes around a particular input example. Ideally, we would do it by searching a neighborhood of the corresponding feature vector in the space of distance transforms that form a manifold in the higher dimensional space of real-valued images. This manifold is fully determined by the local constraints on the distance transform, but unfortunately, it does not have a global parametric description, which makes local search on the manifold difficult. We solve this problem by using a local parameterization of the manifold around any particular shape example that makes use of the surface mesh of that shape.

A surface mesh is a graph that represents the object boundary. In a 3D image, a mesh contains nodes, edges and faces. In a 2D image, a mesh simplifies to a simple loop defined by its nodes and edges. The images in Figure 1-2 were rendered using surface meshes. Given a segmented scan, one can construct the corresponding surface mesh in two linear passes over the voxels of the scan (see, for example, the Marching Cubes algorithm [42]).

For shape analysis, the mesh node locations are used as features. One can think of a surface mesh as a dense version of a landmark based descriptor, where every boundary point becomes a landmark. Ease of generative modeling using surface meshes makes them an attractive choice for representing a family of possible deformations of the original shape. Meshes created by perturbing the node locations in the original mesh form a linear vector space, and are therefore perfect for exploring shapes close to any particular input example. This raises a question on the necessity of using other descriptors for shape analysis. Why not use surface meshes throughout? The main challenge in using surface meshes for statistical analysis is establishing correspondences. While it is trivial to generate new shapes of the same topology from a single mesh, it is difficult to reconcile several different examples: the number of nodes

Figure 3-3: Example shape, a simple ellipsoid.

and their connectivity can be arbitrarily different even for similar shapes.

Performing local search is fundamentally different from establishing a global correspondence framework because we assume that all small (infinitesimal in the limit) changes in the original shape can be generated by adjusting node positions, without changing the topology of the mesh. This assumption eliminates some of the shapes that are arguably close to the original one in the space of the distance transforms, but it provides a reasonable approximation to the set of the shapes that are close to the original input example in the image domain. We essentially use surface meshes to locally parameterize the manifold of the distance transforms in a way that incorporates our knowledge of the domain. In the remainder of this section, we present a formal analysis for such a parameterization, while demonstrating it on a simple example shape shown in Figure 3-3 whose distance transform is shown in Figure 3-4.

Let $\mathbf{x}$ be a feature vector formed by concatenating the distance transform values at all voxels in the image. Vector $\mathbf{x}$ can be thought of as a union of two sub-vectors: vector $\hat{\mathbf{x}}$ that contains the distance transform values at the non-skeleton voxels, and vector $\check{\mathbf{x}}$ that contains the distance transform values at the skeleton voxels. Let $\mathbf{s}$ be a vector of node displacements in the surface mesh of the same object: s_i is the displacement of node i along the normal to the boundary at that node. Since

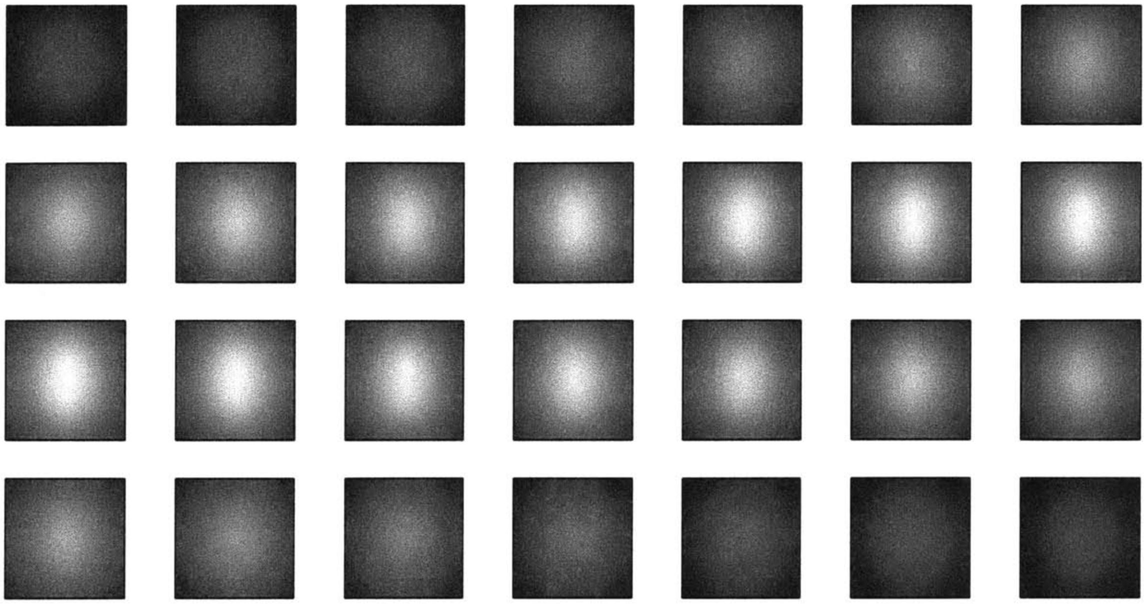


Figure 3-4: The distance transform of the example shape in Figure 3-3. Every second slice is shown.

moving nodes along the surface of the shape does not change the shape, we will only consider changes in the node positions along the normal to the boundary. Zero vector $\mathbf{s}$ corresponds to the original shape and its distance map $\mathbf{x}$. We fix the positive direction to correspond to the normal vector pointing outwards. And lastly, let $S(i)$ be a set of all mesh nodes that influence voxel i.

We start by considering the direct problem, i.e., given an infinitesimal displacement vector $d\mathbf{s}$ that defines a deformation of the original shape, what is the corresponding change $d\mathbf{x}$ in the distance transform feature vector? As shown in the previous section, the value of the distance transform in any non-skeleton voxel changes by the amount equal to the displacement of the boundary point that influences it. This implies a local linear parameterization of the distance transform manifold around the original point $\hat{\mathbf{x}}$:

$$d\hat{\mathbf{x}} = J_{\mathbf{x}} d\mathbf{s}, \tag{3.1}$$

where $J_{\mathbf{x}}$ is the Jacobian of the distance transform values at the non-skeleton voxels with respect to the node positions and contains exactly one non-zero element in every row:

$$J_{\mathbf{x}}(i,j) = \frac{\partial x_i}{\partial s_j} = \begin{cases} 1, & j \in S(i) \\ 0, & \text{otherwise} \end{cases} . \tag{3.2}$$

The change in the value of the distance transform in any skeleton voxel is equal to the minimum of the (signed) displacements of all the nodes that influence that voxel:

$$d\check{x}_i = \min_{j \in S(i)} ds_j, \tag{3.3}$$

exhibiting a non-linear behavior bounded by the linear model above[1].

Now we can solve the inverse problem, namely, estimating the deformation vector $d\mathbf{s}$ that best matches an arbitrary change in the distance transform vector $d\mathbf{x}$. As we change the original distance transform feature vector $\mathbf{x}$ by an infinitesimal displacement vector $d\mathbf{x}$, it will not necessarily stay on the manifold of valid distance transforms. In other words, there might not exist a shape deformation that changes the distance transform exactly as specified by the change vector $d\mathbf{x}$. We can solve this problem by projecting the result back onto the distance transform manifold. Using the linear model of Equation (3.1), we can find the deformation of the surface mesh that generates the closest point to the original discriminative change $d\mathbf{x}$ that still belongs to the distance transform manifold:

$$d\mathbf{s} \approx \left(J_{\mathbf{x}}^T J_{\mathbf{x}}\right)^{-1} J_{\mathbf{x}}^T d\hat{\mathbf{x}}, \tag{3.4}$$

where matrix $\left(J_{\mathbf{x}}^T J_{\mathbf{x}}\right)^{-1} J_{\mathbf{x}}^T$ is the generalized inverse of the Jacobian $J_{\mathbf{x}}$. It is easy to see that $J_{\mathbf{x}}^T J_{\mathbf{x}}$ is a diagonal matrix whose (j,j) entry is equal to the number of non-skeleton voxels influenced by node j.

The projection operation of Equation (3.4) is an approximation that ignores skeleton voxels. We argue that since the number of skeleton voxels is usually small com-

[1] Equation (3.1) defines an infinitesimal model of changes in the distance transform in non-skeleton voxels as a function of small deformations of the surface mesh. The nodes of the mesh define a Voronoi partition of the volume which determines the influence relationships between voxels and nodes [9, 46]. As long as the node displacements do not change voxels' membership in the partition, the linear model of change holds. Moreover, it provides an upper bound on the amount of change in the distance transform if the influence relationship changes as a result of the deformation.

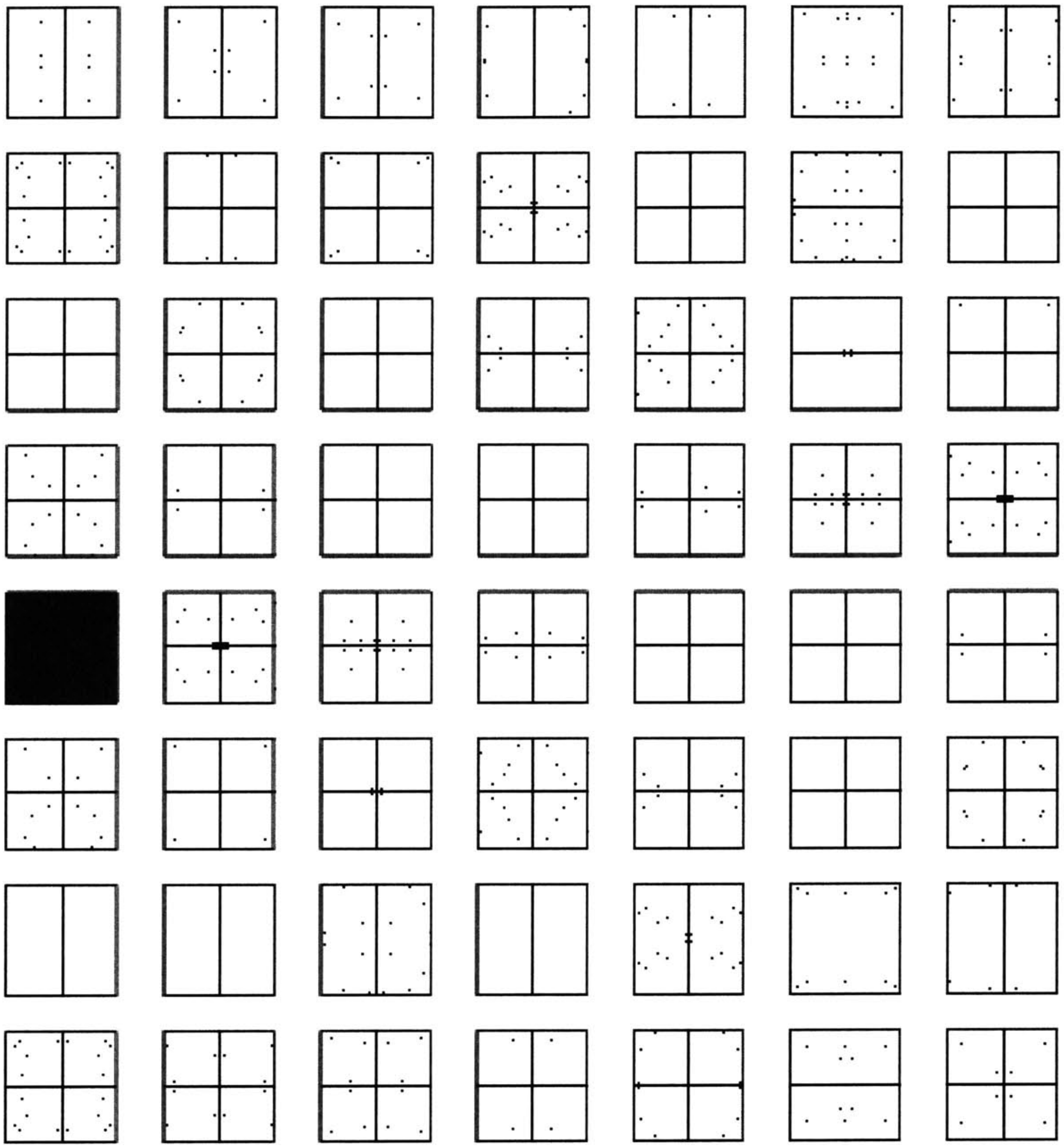


Figure 3-5: Skeleton voxels for the example shape in Figure 3-3. The skeleton voxels are shown in black, the non-skeleton voxels are shown in white. All slices are shown.

pared to the total number of voxels, they do not affect the results significantly. To illustrate this, Figure 3-5 shows a skeleton that was computed for the discrete version of the distance transform in Figure 3-4. Theoretically, a skeleton of an ellipsoid is a plane that passes through its largest principal axis and is perpendicular to its smallest principal axis. Additional single points, lines and gaps in the symmetry plane in Figure 3-5 are caused by the discretization errors in the distance estimates. The number of skeleton voxels is relatively small in this figure. In all our experiments, the number of skeleton voxels was 4-6% of the total number of voxels in the volume. In general, a skeleton of a shape is a surface of one dimension lower than the dimensionality of the image space (e.g., a skeleton of a 3D shape is a 2D surface) and therefore contains a negligible number of voxels compared to the shape itself.

Deformation $d\mathbf{s}$ of the original shape leads to the change of the distance transform that is linear for non-skeleton voxels:

$$d\hat{\mathbf{x}}^o = J_{\mathbf{x}} d\mathbf{s} \approx J_{\mathbf{x}} (J_{\mathbf{x}}^T J_{\mathbf{x}})^{-1} J_{\mathbf{x}}^T d\hat{\mathbf{x}} \tag{3.5}$$

and can be extended to a full volume vector $d\mathbf{x}^o$ by computing the change at the skeleton voxels as a minimum of the displacements of the influencing nodes:

$$dx_i^o = \begin{cases} \min_{j \in S(i)} s_j, & \|S(i)\| > 1 \\ d\hat{x}_i^o, & \|S(i)\| = 1 \end{cases} . \tag{3.6}$$

This operation projects the infinitesimal displacement vector $d\mathbf{x}$ onto the distance transform manifold so that the resulting change in the distance transform $d\mathbf{x}^o$ defines a valid deformation of the shape while minimizing the distance between the two vectors.

To illustrate our analysis on the shape in Figure 3-3, let's consider an example displacement vector $d\mathbf{x}$ shown in Figure 3-6. This vector was generated as a part of a simulated shape analysis study that we will describe later in this chapter. The colorbar indicates the intensity coding of the values in the vector $d\mathbf{x}$. Since we are using a linear approximation, the absolute values of the vector components are unimportant:

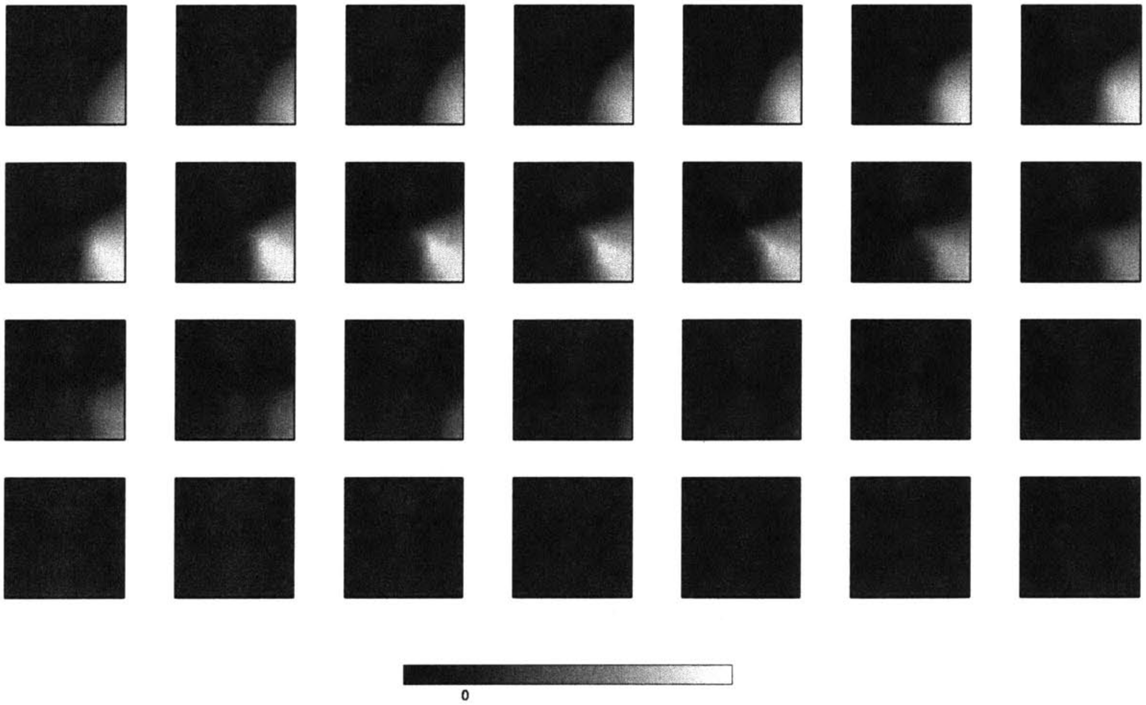
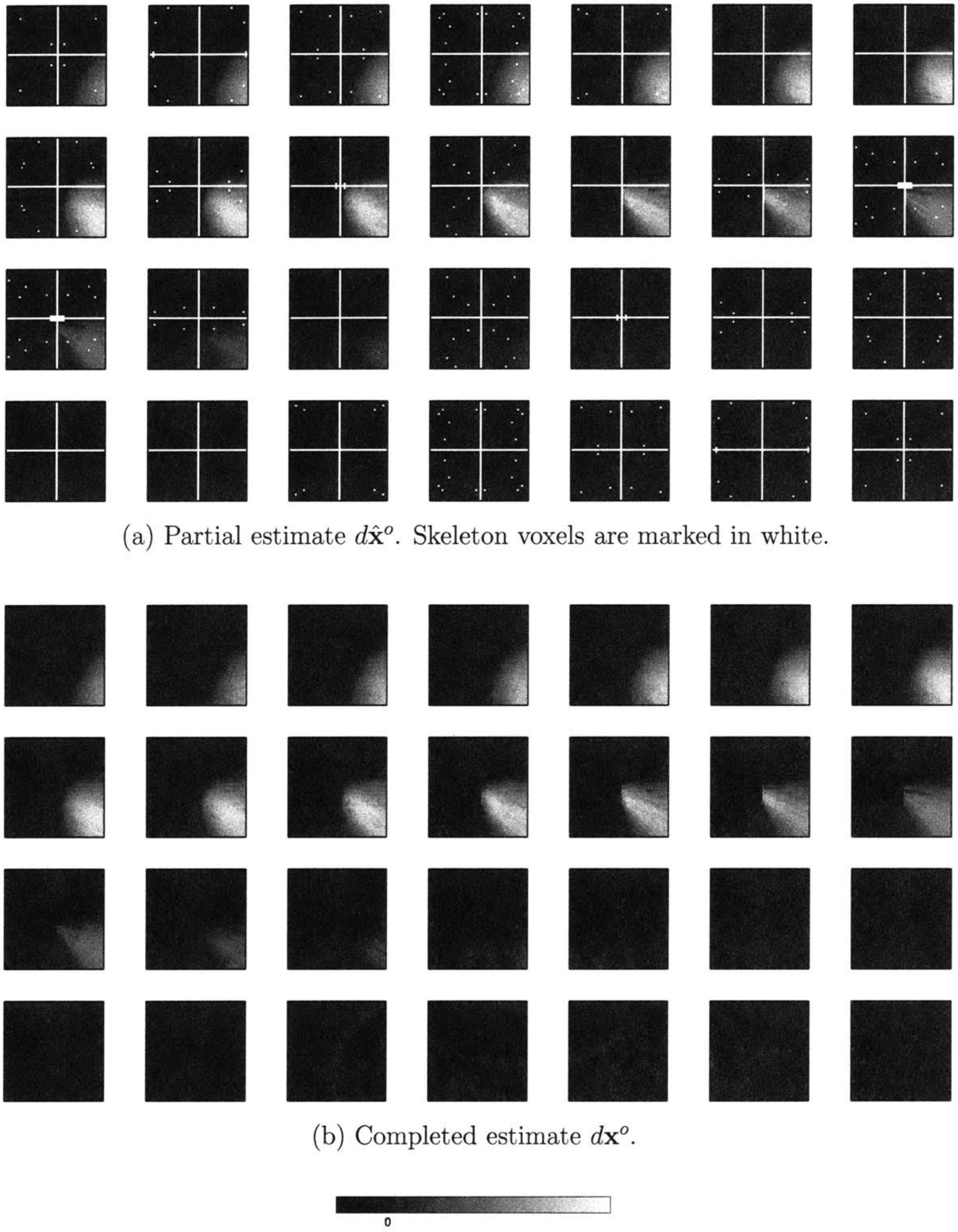


Figure 3-6: Example change vector $d\mathbf{x}$ for the distance transform $\mathbf{x}$ in Figure 3-4. Positive values (bright intensities) in the image correspond to increasing the values of the same voxels in the distance transform, negative values (dark intensities) correspond to reducing the values of the distance transform. The absolute scaling is omitted since we are interested only in the direction of the vector, not its magnitude. The intensity that corresponds to zero change in the distance transform is shown on the colorbar. Every second slice is shown.

scaling the vector by a constant factor will simply scale the resulting projection by the same factor. Furthermore, the infinitesimal analysis in our work concentrates on the direction of the change rather than its magnitude. We will come back to this point in Chapter 5 when we present the discriminative direction.

Figure 3-6 shows a consistent, localized increase in the values of the distance transform that corresponds to creating a protrusion on the surface of the original ellipsoid in the area of bright intensities. However, if we change the original distance transform $\mathbf{x}$ in Figure 3-4 by even an infinitesimal amount along the vector $d\mathbf{x}$, the result will not be a valid distance transform. To fix this problem, we can estimate the change in the distance transform $d\mathbf{x}^o$ that best approximates $d\mathbf{x}$. Figure 3-7 shows both the partial estimate for the non-skeleton voxels $d\hat{\mathbf{x}}^o$ and the final approxima-



(a) Partial estimate $d\hat{\mathbf{x}}^o$. Skeleton voxels are marked in white.

(b) Completed estimate $d\mathbf{x}^o$.

Figure 3-7: Optimal estimate of the changes in the distance transform that approximates the change vector $d\mathbf{x}$ in Figure 3-6. The color coding is identical to that in Figure 3-6. Every second slice is shown. Compare to Figure 3-6.

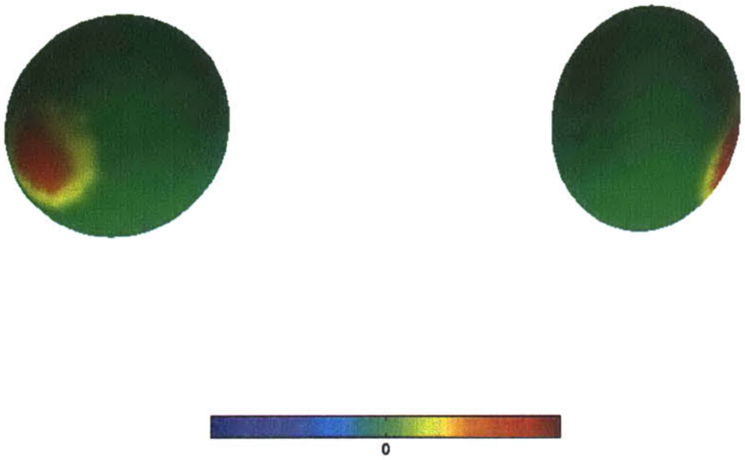


Figure 3-8: Two views of the example shape from Figure 3-3 with the deformation painted on its surface. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

tion $d\mathbf{x}^o$ of the original vector $d\mathbf{x}$ computed using Equations (3.4)-(3.6). The resulting vector $d\mathbf{x}^o$ is close to the original vector $d\mathbf{x}$, but has a structure similar to that of the distance transform.

In the process of projecting the vector $d\mathbf{x}$ onto the distance transform manifold, we also compute the mesh deformation vector $d\mathbf{s}$. We can then "paint" it on the surface of the shape by associating an appropriate color with each node in the mesh, as shown in Figure 3-8. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards). Similarly to the volumetric estimate $d\mathbf{x}^o$, the shape changes defined by the deformation $d\mathbf{s}$ are well localized and correspond to "growing a bump" on the original shape.

To remind the reader, the analysis presented in this section was developed in order to investigate the space of shapes in the infinitesimal neighborhood of the original examples. The local parameterization of the space through the deformations of the surface mesh enables such search. Moreover, Figure 3-8 demonstrates another

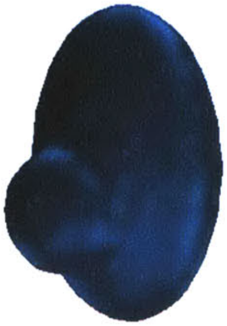


Figure 3-9: Another example shape, an ellipsoid with a bump.

important reason for using deformations to describe changes in the shape. In our experiments, we found that the surface-based representation was significantly more informative and easier to interpret than the volumetric results similar to those in Figure 3-6 and Figure 3-7. We will discuss the advantages of surface-based visualization in more detail in Chapter 6 when we present the entire analysis framework and explain how it generates a description of shape differences between the two example groups.

The example shape in Figure 3-3 comes from a simulated shape study presented in Chapter 6 in which the shapes in both classes were simple ellipsoids, but the examples in one class had a bump approximately in the same place. The goal of the experiment was to test if our technique could detect the bump as the main difference between the classes. The change in the distance transform $d\mathbf{x}$ in Figure 3-6 was automatically produced by the analysis to describe the necessary shape changes to cause the original shape to look more similar to the examples in the other class. Figure 3-8 shows a deformation that achieves this effect.

Before concluding this section, we demonstrate the projection operation for one more example. Figure 3-9 shows an example shape from the second class in the same study. The distance transform for this shape is displayed in Figure 3-10. Note the differences due to the bump between the distance transforms in Figure 3-10 and

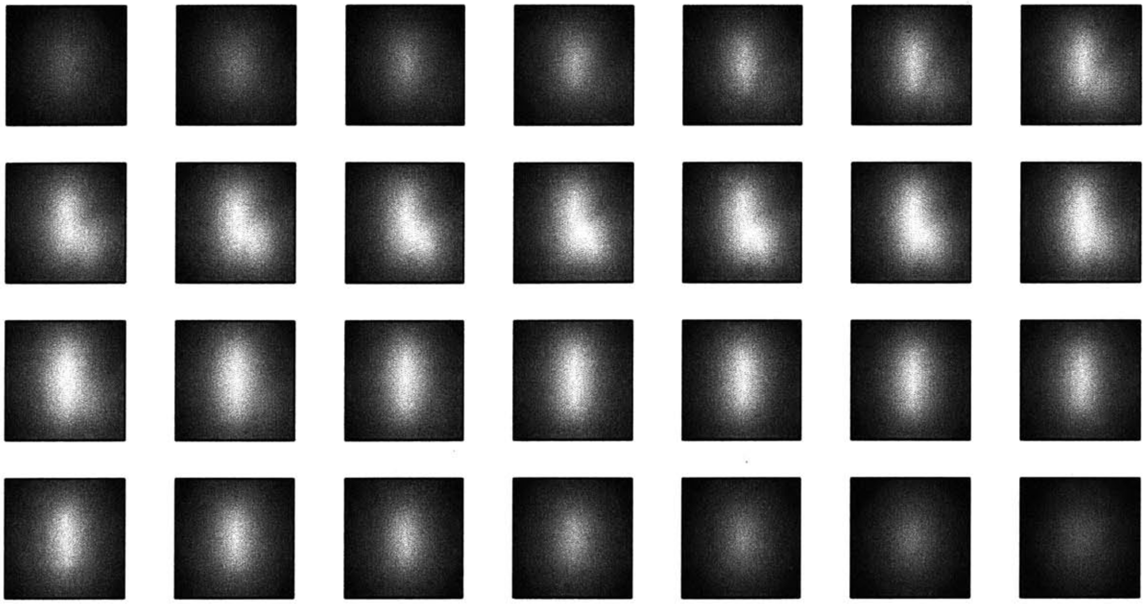
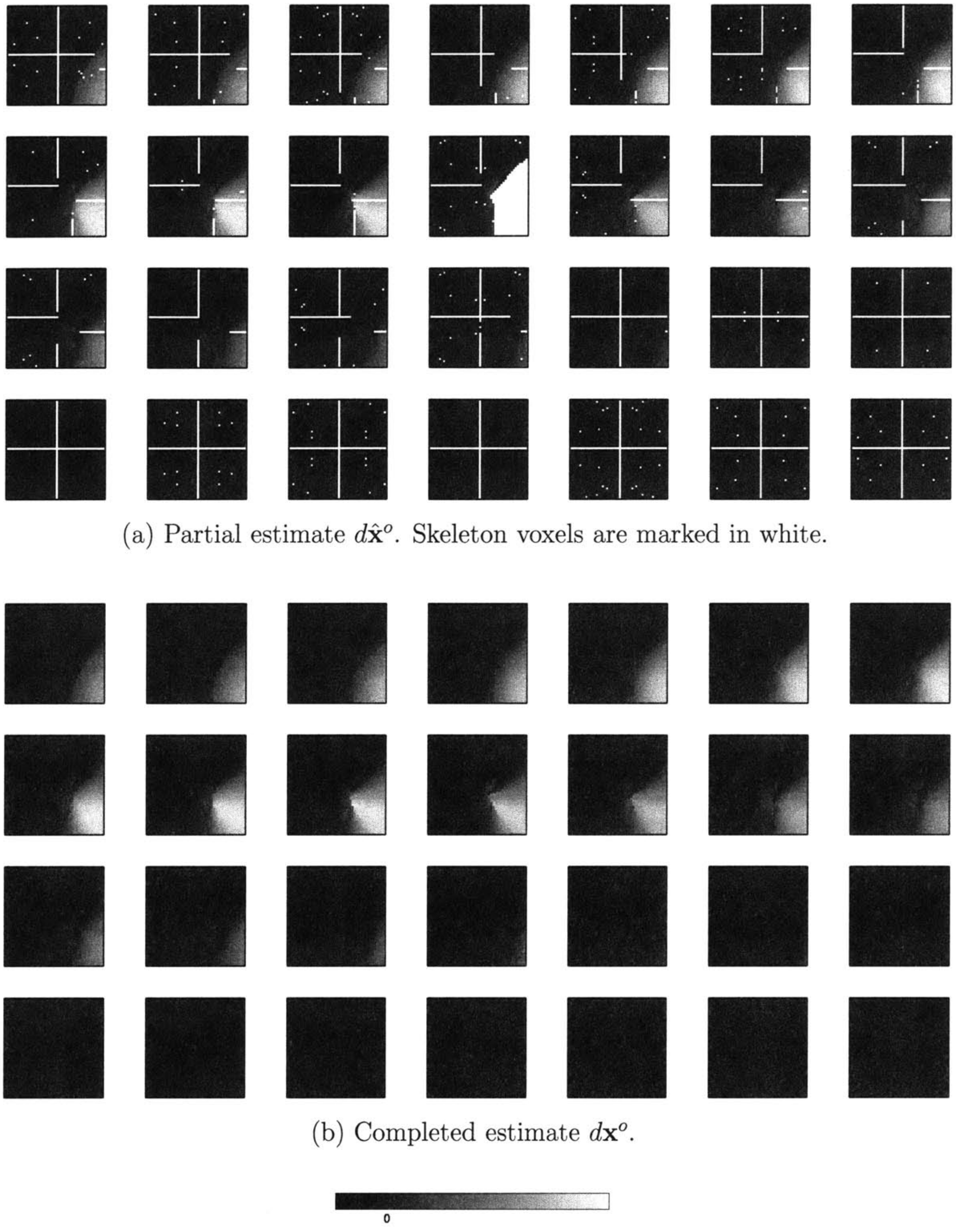


Figure 3-10: Distance transform of the shape in Figure 3-9. Every second slice is shown. Compare to Figure 3-4.

Figure 3-4. Figure 3-11 shows the estimated change in the distance transform of this shape $d\mathbf{x}^o$ for the displacement vector $d\mathbf{x}$ in Figure 3-6. In spite of the differences in the distance transforms themselves, the resulting change vector $d\mathbf{x}^o$ looks almost identical to that in Figure 3-7. This indicates that the local geometry of the manifold, captured by the Jacobian matrix $J_{\mathbf{x}}$, is similar at the two points. The discrepancies in the manifold geometry between the two examples cause the slight differences between the resulting vectors in Figure 3-7 and Figure 3-11.

The statistical analysis produced the change vector $d\mathbf{x}$ in Figure 3-6 for the first example shape. It turns out that the resulting change vector for the second shape is equal to $-d\mathbf{x}$. As we saw earlier, $d\mathbf{x}$ represents expanding of the object boundary outwards, and therefore $-d\mathbf{x}$ describes a change in the distance transform that corresponds to moving the boundary surface inwards. This is to be expected, as the deformations that bring the two shapes closer to each other must be of the opposite effects. Since the projection is a linear operation, the resulting projection vector is of the opposite sign to that shown in Figure 3-11. Figure 3-12 shows the corresponding deformation of the surface mesh $d\mathbf{s}$ that indeed corresponds to reducing the bump.



(a) Partial estimate $d\hat{\mathbf{x}}^o$. Skeleton voxels are marked in white.

(b) Completed estimate $d\mathbf{x}^o$.

Figure 3-11: Optimal estimate of the changes in the distance transform that approximates the change vector $d\mathbf{x}$ in Figure 3-6 for the shape in Figure 3-9. The color coding is identical to that in Figure 3-6. Every second slice is shown. Compare to Figure 3-6 and Figure 3-7.

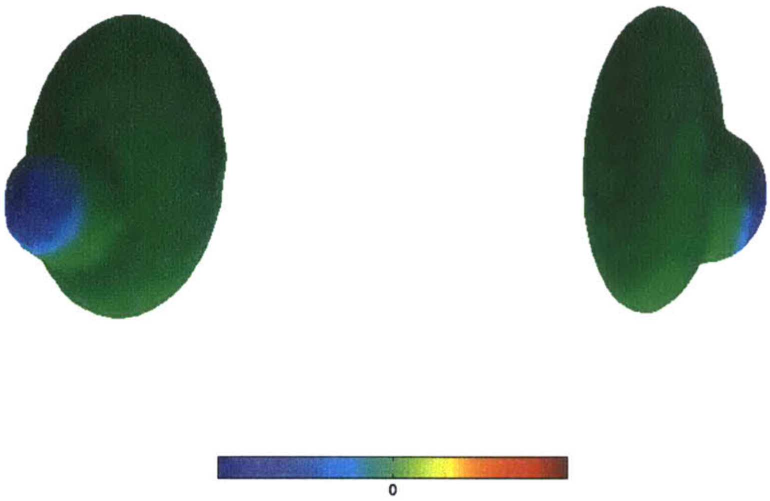


Figure 3-12: Two views of the example shape from Figure 3-9 with the deformation painted on the surface. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

3.4 Summary

We use the distance transform for shape description. In this chapter, we focused on the properties of the distance transform relevant for our application, while deferring the discussion on implementation details of the feature vectors extraction based on the distance transform representation until Chapter 6.

The distance transform is a dense descriptor capable of capturing an arbitrary shape whose bounded behavior in the presence of noise in the outline and in the object position makes it an attractive choice for image based statistical analysis. The main drawback of the distance transform is that it does not provide a way to search the space of possible deformations of the original shapes. We overcome this difficulty by using surface meshes for local parameterization of the distance transform manifold. Such a parameterization is necessary for statistical analysis of shape differences presented in the later chapters.

Chapter 4

Statistical Modeling Using Support Vector Machines

Once feature vectors have been extracted from the input images, the problem of shape analysis can be formulated in the traditional machine learning framework, where a set of training examples from two different classes is used to construct a classifier function that assigns new examples to one of the two classes while making as few mistakes as possible. In this work, we use the Support Vector Machines (SVMs) algorithm to estimate the optimal classifier.

SVMs were introduced by Vapnik [65] and have been used successfully in many classification and pattern recognition applications, such as digit recognition [66], text classification [33, 34], face detection [48] and others. In addition to its superior empirical performance, the algorithm can be proved to converge to the optimal solution as the amount of training data increases, with a distribution-independent bound on the rate of convergence [65]. Furthermore, the same theoretical framework provides a principled way to explore a hierarchy of increasingly complex classifier families, trading-off the training error and the complexity of the model.

The main practical limitation of the Support Vector Machines is that the algorithm requires solving quadratic optimization, which can be costly for large data sets. It is not a serious disadvantage for our application because we typically work with small training sets. Estimating the rate of convergence of the algorithm is another problem. Ideally, one would like to predict the size of the training set necessary to guarantee that with high probability the resulting classifier is close to the optimal one. Unfortunately, the distribution-free bound on the rate of convergence is too loose to be useful for this purpose, even in situations when empirical evidence suggests fast convergence. Deriving tighter bounds based on prior knowledge and better measures of the model complexity is an active research area in machine learning [31, 67].

The purpose of this chapter is to provide a brief overview of SVMs and the capacity analysis based on the VC dimension. We describe the algorithm and state without proof the main results from the statistical learning theory necessary for derivation of the sensitivity analysis presented in the next chapter. In this work, we follow closely the notation and the formalism introduced in [66]. Readers interested in more details on support vector learning are referred to the tutorials [11, 61] for an extensive introduction into kernel based classification and function regression, and to the work by Vapnik [65, 66] for a formal discussion on theoretical foundations of support vector methods.

4.1 Basic Definitions

Given a training set of l pairs $\{(\mathbf{x}_k, y_k)\}_{k=1}^{l}$, where $\mathbf{x}_k \in \mathbb{R}^n$ are observations and $y_k \in \{-1, 1\}$ are corresponding labels, and a family of classifier functions $\{f_\omega\}$ parametrized by ω,

$$f_\omega : \mathbb{R}^n \mapsto \{-1, 1\}, \tag{4.1}$$

the learning task is to select a member of the family that minimizes the expected error,

$$R(\omega) = \int \frac{1}{2}|y - f_\omega(\mathbf{x})| P(\mathbf{x}, y) d\mathbf{x}, \tag{4.2}$$

also called expected risk, when labeling new, unseen examples. $P(\mathbf{x}, y)$ is the probability distribution that generates the observations. In practice, however, Equation (4.2) is difficult to evaluate, as $P(\mathbf{x}, y)$ is unknown. Instead, the training error,

$$R\text{emp}(\omega) = \frac{1}{2l} \sum_{i=1}^{l} |y_k - f_\omega(\mathbf{x}_k)|, \tag{4.3}$$

also called empirical risk, can be computed. It can be shown that the minimum of the empirical risk over the family of classifiers $\{f_\omega(\mathbf{x})\}$ converges in probability to the minimum of the expected risk as the number of training examples grows. Consequently, one can perform well by collecting enough data to operate in the asymptotic range and using the empirical risk as the objective criterion to be minimized. This so-called *Empirical Risk Minimization Principle* underlies many learning algorithms in machine learning, starting with the basic perceptron [20].

Most learning algorithms search over on a set of real-valued classification functions, thresholding the function's value on a new example in order to assign it to one of the two classes:

$$\hat{y}(x) = \text{sign}(f(\mathbf{x})). \tag{4.4}$$

One of the simplest examples is a linear classifier,

$$f(\mathbf{x}) = \langle \mathbf{x} \cdot \mathbf{w} \rangle + b, \tag{4.5}$$

where $\langle \cdot \rangle$ denotes a dot product. The linear classifier uses the projection of input vector $\mathbf{x}$ onto the vector $\mathbf{w}$ to assign $\mathbf{x}$ to one of the classes. The separating boundary between the classes is a hyperplane whose normal is $\mathbf{w}$ and whose position is determined by the threshold b.

4.2 Linear Support Vector Machines

To find the optimal projection vector $\mathbf{w}$, most learning algorithms minimize a cost function that measures how well the data can be separated once projected onto $\mathbf{w}$. Well separated classes yield low empirical risk, which will lead to low expected risk as the number of examples grows. The difference between various methods is in

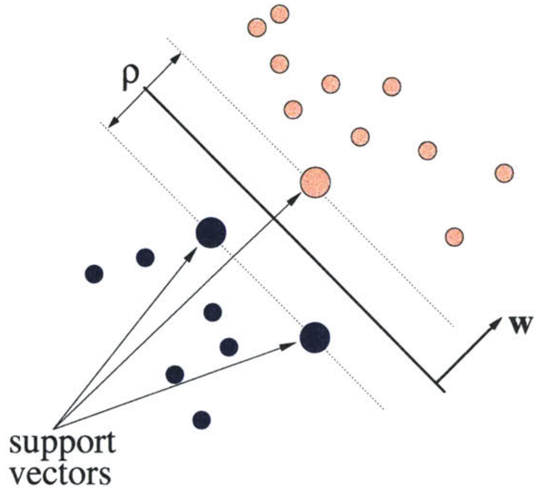


Figure 4-1: Linearly separable classes.

how they evaluate the separability of the projected data[1]. The resulting classifier's performance depends crucially on the cost function used to estimate the separability of the projected data. The Support Vector Machines learning algorithm maximizes the margin between the classes with respect to the projection vector. As we will see in Section 4.4.1, there is a fundamental relationship between generalization ability of a classifier and its margin.

Let's first consider a situation when the training data set is linearly separable, i.e., there exists a hyperplane defined by its normal vector $\mathbf{w}$ and constant b that separates the two classes (Figure 4-1). Note that for any hyperplane, $\mathbf{w}$ and b are unique up to a scale factor and can be normalized to satisfy

$$\forall k: \; y_k(\langle \mathbf{x}_k \cdot \mathbf{w} \rangle + b) \geq 1. \tag{4.6}$$

In this representation, the minimal distance from any data vector to the separating hyperplane is equal to $1/\|\mathbf{w}\|$, and the margin between the classes with respect to the separating hyperplane,

$$\rho(\mathbf{w}) = \min_{y_k=1} \frac{\langle \mathbf{x}_k \cdot \mathbf{w} \rangle}{\|\mathbf{w}\|} - \max_{y_k=-1} \frac{\langle \mathbf{x}_k \cdot \mathbf{w} \rangle}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{4.7}$$

[1] For example, the Fisher Linear Discriminant [20] uses the distance between the projected means normalized by the variance of the projected data.

is maximized when the length of the projection vector $\mathbf{w}$ is minimized subject to the constraints (4.6). Using Lagrange multipliers, we transform the original problem

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{w}) = \|\mathbf{w}\|^2 && (4.8) \\ \text{s.t.} \quad & \forall k: \; y_k(\langle \mathbf{x}_k \cdot \mathbf{w} \rangle + b) \geq 1. && (4.9) \end{aligned}$$

into its dual:

$$\begin{aligned} \text{maximize} \quad & W(\boldsymbol{\alpha}) = \sum_k \alpha_k - \frac{1}{2} \sum_{k,m} \alpha_k \alpha_m y_k y_m \langle \mathbf{x}_k \cdot \mathbf{x}_m \rangle && (4.10) \\ \text{s.t.} \quad & \forall k: \; \alpha_k \geq 0 && (4.11) \\ & \sum_k \alpha_k y_k = 0, && (4.12) \end{aligned}$$

where the α_k's are the multipliers for the inequalities (4.9). This is a well known constrained quadratic optimization problem that can be solved using a variety of numerical methods. Moreover, the optimal projection vector $\mathbf{w}^*$ is a linear combination of the training examples:

$$\mathbf{w}^* = \sum_k \alpha_k y_k \mathbf{x}_k. \tag{4.13}$$

The Kuhn-Tucker conditions imply that the non-zero α_k's in the solution correspond to the training vectors that satisfy the inequalities (4.9) with equality. These are called support vectors, as they "support" the separating boundary between the classes (Figure 4-1).

To extend this approach to a non-separable case, we would ideally modify the cost function $J(\mathbf{w})$ to maximize the margin while minimizing the number of misclassified training examples. Unfortunately, this leads to a discrete cost function, which in turn renders the problem NP-hard, and we have to resort to a linear approximation to the discrete penalty function by introducing non-negative slack variables ξ_k that

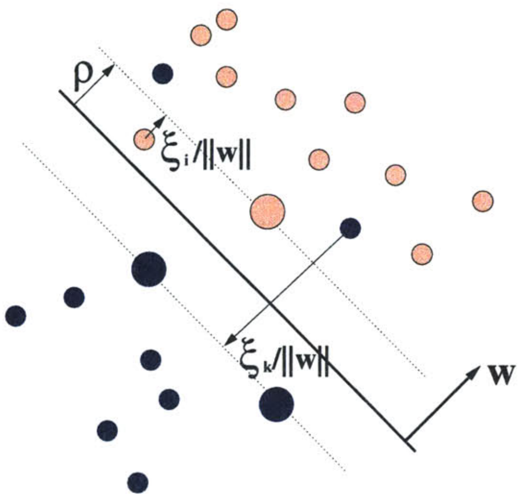


Figure 4-2: Imperfect separation using a hyperplane.

measure by how much each training example violates the margin constraint (4.9) (Figure 4-2). The optimization problem (4.8)-(4.9) changes to include a penalty term for misclassified examples:

$$\text{minimize} \qquad J(\mathbf{w}) = \|\mathbf{w}\|^2 + C\sum \xi_k \tag{4.14}$$

$$\text{s.t.} \qquad \forall k:\ y_k(\langle \mathbf{x}_k \cdot \mathbf{w}\rangle + b) \geq 1 - \xi_k, \tag{4.15}$$

where the constant C determines the trade-off between maximizing the margin and minimizing the number of errors. The resulting constrained quadratic optimization problem is very similar to the separable case:

$$\text{maximize} \qquad W(\boldsymbol{\alpha}) = \sum_k \alpha_k - \frac{1}{2}\sum_{k,m} \alpha_k \alpha_l y_k y_m \langle \mathbf{x}_k \cdot \mathbf{x}_m \rangle \tag{4.16}$$

$$\text{s.t.} \qquad \forall k:\ 0 \leq \alpha_k \leq C \tag{4.17}$$

$$\sum_k \alpha_k y_k = 0, \tag{4.18}$$

and can be solved using the same optimization techniques. The optimal projection vector $\mathbf{w}^*$ is still a linear combination of the training vectors in this case. The resulting classifier

$$f(\mathbf{x}) = \langle \mathbf{x} \cdot \mathbf{w}^* \rangle + b = \sum_k \alpha_k y_k \langle \mathbf{x} \cdot \mathbf{x}_k \rangle + b \tag{4.19}$$

is a linear function of dot products between the input vector $\mathbf{x}$ and the support vectors and defines a hyperplane whose normal is $\mathbf{w}^*$.

In practice, the algorithm is employed assuming that the data are not separable and adjusting the constant C until a desired (possibly full) level of separation is achieved.

4.3 Non-linear Support Vector Machines

Support Vector Machines can be extended to non-linear classification by observing that both the optimization problem (4.16)-(4.18) and the resulting classifier (4.19) use the input vectors only through the dot product operator. If we map the training points from the original space into a higher dimensional space in a non-linear fashion and apply a linear method in the higher dimensional space, the resulting classifier will be a non-linear function of the input examples. Kernel functions allow us to compute the dot products in the higher dimensional case without ever computing the mapping. Using kernel functions, we can explore complex non-linear classifier families that lead to exponential growth in the dimensionality of the resulting feature space, which would be computationally prohibitive if the mapping to that space had to be computed explicitly.

Formally, a function of two variables

$$K : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}, \tag{4.20}$$

is called a *kernel* if for some function

$$\Phi_K : \mathbb{R}^n \mapsto \mathbb{F} \tag{4.21}$$

that maps the data into a higher (or equal) dimensional space $\mathbb{F}$ (for example, $\mathbb{R}^m$ for $m \geq n$), the values of dot products in the space $\mathbb{F}$ can be computed by applying function K to vectors in $\mathbb{R}^n$:

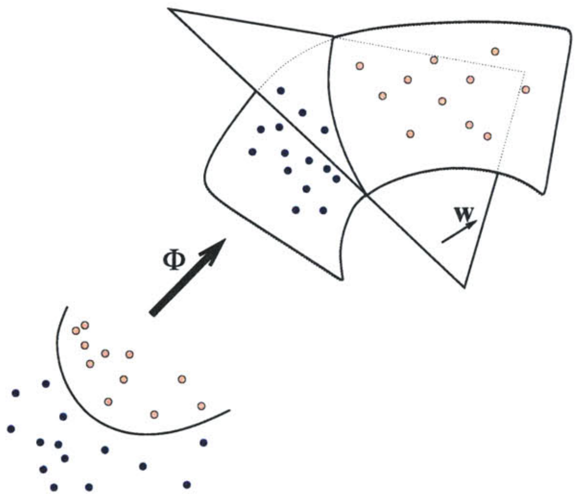


Figure 4-3: Kernel based classification.

$$\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n : \; K(\mathbf{u}, \mathbf{v}) = \langle \Phi_K(\mathbf{u}) \cdot \Phi_K(\mathbf{v}) \rangle \,. \tag{4.22}$$

Given vector $\mathbf{u} \in \mathbb{R}^n$ and vector $\mathbf{z} \in \mathbb{F}$ such that

$$\mathbf{z} = \Phi_K(\mathbf{u}), \tag{4.23}$$

we will say that $\mathbf{z}$ is an *image* of $\mathbf{u}$ and $\mathbf{u}$ is a *source* of $\mathbf{z}$.

According to Mercer's Theorem, a function is a kernel if and only is it is positive semi-definite. Kernel functions have been used for decades in linear systems analysis [47] and have gained increased popularity in the machine learning community as it became apparent that they allow the construction of non-linear algorithms by implicitly mapping the data to a higher dimensional space [57].

Different kernel functions have been proposed for use in classification and function regression [3, 12, 56]. The simplest example is the linear kernel

$$K(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u} \cdot \mathbf{v} \rangle \,, \tag{4.24}$$

for which there exists a mapping Φ_K that is the identity, and the resulting classifier is a linear function in the original space. This kernel is a special case of the polynomial family:

$$K(\mathbf{u},\mathbf{v}) = (1 + \langle \mathbf{u} \cdot \mathbf{v} \rangle)^d, \tag{4.25}$$

where d is the degree of the kernel.

Another commonly used kernel is the Gaussian kernel:

$$K(\mathbf{u},\mathbf{v}) = e^{\frac{-\|\mathbf{u}-\mathbf{v}\|^2}{\gamma}} \tag{4.26}$$

where the parameter γ defines how the influence of points on one another decreases with distance and is often called the width of the kernel. The range space of the corresponding mapping function is truly infinite-dimensional in this case.

Using kernel functions, we can effectively train a linear classifier in the higher dimensional space $\mathbb{F}$ without explicitly evaluating Φ_K, but rather using kernel K to compute the dot products in $\mathbb{F}$. This classifier produces a non-linear decision boundary back in the original space $\mathbb{R}^n$:

$$f_K(\mathbf{x}) = \sum_k \alpha_k y_k \langle \Phi_K(\mathbf{x}) \cdot \Phi_K(\mathbf{x}_k) \rangle + b = \sum_k \alpha_k y_k K(\mathbf{x},\mathbf{x}_k) + b. \tag{4.27}$$

The non-linearity arises from the kernel function $K(\mathbf{x},\mathbf{x}_k)$, which replaces the dot product in the linear classifier in Equation (4.19). In the higher dimensional space $\mathbb{F}$, the separating boundary is a hyperplane whose normal is a linear combination of the images of support vectors:

$$\mathbf{w} = \sum_k \alpha_k y_k \Phi_K(\mathbf{x}_k), \tag{4.28}$$

but it can be an arbitrarily complex surface in the original space (Figure 4-3). The coefficients α_k are computed by solving the constrained quadratic optimization problem (4.16)-(4.18), with the dot products evaluated in the space $\mathbb{F}$:

$$\text{maximize} \quad W(\boldsymbol{\alpha}) = \sum_k \alpha_k - \frac{1}{2}\sum_{k,m} \alpha_k \alpha_l y_k y_m K(\mathbf{x}_k,\mathbf{x}_m) \tag{4.29}$$

$$\text{s.t.} \quad \forall k: \; 0 \leq \alpha_k \leq C \tag{4.30}$$

$$\sum_k \alpha_k y_k = 0. \tag{4.31}$$

To summarize, the SVM learning algorithm searches over a family of classifier functions and selects the one that maximizes the margin between the two classes with respect to a separating boundary. In the linear case, the separating boundary is a hyperplane, and finding the best classifier requires solving a constrained quadratic optimization problem fully determined by the pairwise dot-products of the training vectors. This formulation lends itself naturally to a non-linear classification framework by using kernel functions to evaluate dot-products in the implied higher dimensional target space. Thus, the same quadratic optimization yields a classifier that is linear with respect to the image vectors in the higher dimensional space, and therefore is a non-linear function of the original data. In the next section, we discuss kernel selection that determines the complexity of the resulting model.

4.4 Model Selection

The Empirical Risk Minimization Principle states that the minimum of the empirical risk converges in probability to the minimum of the expected risk as the number of training examples approaches infinity. In practice, the rate of convergence can be slow enough that for any reasonable size of the training set, minimizing the empirical risk causes overfitting, i.e., creating a model that explains the training data, but does not generalize well on the test data. This leads to a notion of classifier (model) complexity: a simpler model can be trained reliably using less data, but is limited in the number of hypotheses it can express, while a more complex model is capable of explaining more hypotheses, but requires more data to train. This trade-off is the basis for such methods as Minimal Description Length [51, 52], regularization [26], and the more recently proposed *Structural Risk Minimization Principle* [65]. These techniques define a measure of the model complexity and use it to predict the expected generalization error of the classifier. Traditionally, the number of free parameters has been used to bound function complexity. More recently, the VC dimension was introduced as a better measure.

The VC dimension of a family of classifiers is an upper bound on the number of hypotheses the family members can generate on any given training set. One can show that the rate of convergence of the minimum of the empirical risk to the minimum of the expected risk for any particular classifier family can be bounded using the VC dimension. Formally, for any η $(0 \leq \eta \leq 1)$, with probability at least $1 - \eta$, the classifier $f_{\omega^*}(\mathbf{x})$ that minimizes the empirical risk $R_{\mathrm{emp}}(\omega)$ on the given training set satisfies

$$R(\omega^*) \leq R_{\mathrm{emp}}(\omega^*) + \sqrt{\frac{h}{l}\left(\log\frac{2l}{h} + 1\right) - \frac{1}{l}\log\frac{\eta}{4}}, \tag{4.32}$$

where h is the VC dimension of the classifier family $f_\omega(\mathbf{x})$, and l is the number of training examples. The right hand side of (4.32) is often referred to as the *VC bound* and its second term is called *VC confidence.* The Structural Risk Minimization Principle relies on the VC bound (4.32) to estimate the expected risk. Applying it to kernel-based SVMs, we consider a hierarchy of classifier families (e.g., the polynomial kernels of increasing degree or the Gaussian kernels or decreasing width), and for each family train a classifier using SVMs (select the support vectors and estimate their coefficients) and compute the upper bound on the expected risk. The optimal classifier is the one with the smallest upper bound.

The bound on the expected risk has to be predictive in order to be useful for model selection. Note however, that the VC bound is distribution-free, i.e., one does not need to know the distribution of the input data to estimate the convergence rate of the learning algorithm. This suggests that the bound is usually fairly loose for any particular distribution, and tighter bounds could be derived if the data distribution function were known. Investigating tighter bounds is an active area of research in the machine learning community [31, 67], and will hopefully lead to better estimates on the generalization ability of classifiers. There has also been some work relating the more traditional notion of regularization and the VC dimension [22, 60].

For problems with a small number of training examples, when the VC bound is too loose to be helpful for classifier selection, other methods, such as cross-validation, are employed [21]. The relationship between VC dimension and cross-validation is

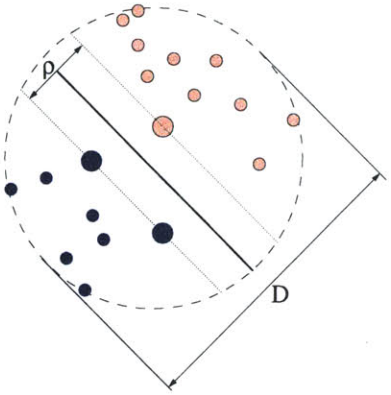


Figure 4-4: Bounding sphere.

discussed in [66]. The traditional approach to estimating the expected test error from the cross-validation is based on the Law of Large Numbers and De Moivre - Laplace approximation: with probability at least $1 - \eta$

$$|R(\omega^*) - \hat{R}| \leq \Phi_K^{-1}\left(\frac{1-\eta}{2}\right)\sqrt{\frac{\hat{R}(1-\hat{R})}{l}}, \tag{4.33}$$

where $\hat{R}$ is the error rate of the cross-validation and

$$\Phi_K(x) = \frac{1}{\sqrt{2\pi}}\int_0^x e^{-t^2/2}dt \tag{4.34}$$

is the standard error function.

4.4.1 VC dimension and Support Vector Machines

It can be shown that the VC dimension of a hyperplane that separates the data with margin ρ is bounded by

$$h \leq \min(D^2/\rho^2, n) + 1, \tag{4.35}$$

where D is the diameter of the smallest hyper sphere that contains all the training examples, and n is the dimensionality of the space (Figure 4-4). This bound can also be computed in the non-linear case, as the radius of the bounding sphere in the target space can be estimated using the kernel function.

Support Vector Machines have been demonstrated experimentally to be very ro-

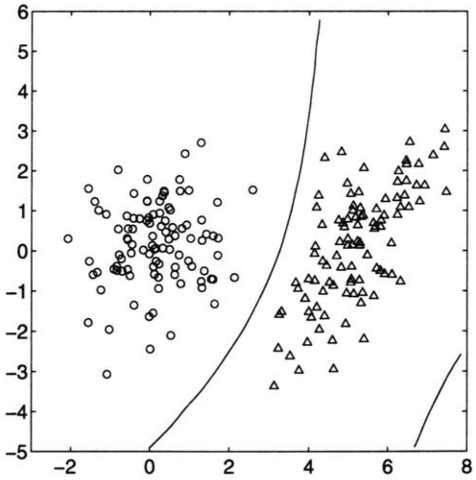


Figure 4-5: Example training set and the maximum-likelihood separating boundary (solid line) derived from the underlying probability distributions.

bust to overfitting, even in situations when the VC bound is too loose to be useful in predicting the generalization error reliably. It is believed that the classifier's margin is indicative of its capacity, and therefore its generalization power, much more so than the VC bound implies. While several tighter bounds have been demonstrated recently [31, 67], we do not have a formal explanation yet for SVMs' superior performance.

Since the concept of the VC dimension and its dependence on the margin of the support vector classifier was introduced by Vapnik, similar relationships have been shown for other learning techniques. For example, Schapire *et al.* [55] explained performance characteristics of the ADA Boosting algorithm in terms of VC dimension and the margin of the resulting learner.

4.5 Simple Example

We demonstrate the concepts presented in this chapter on a simple example shown in Figure 4-5. The training data were generated by sampling two different two-dimensional Gaussian distributions. The figure also shows the analytically derived maximum likelihood separating boundary.

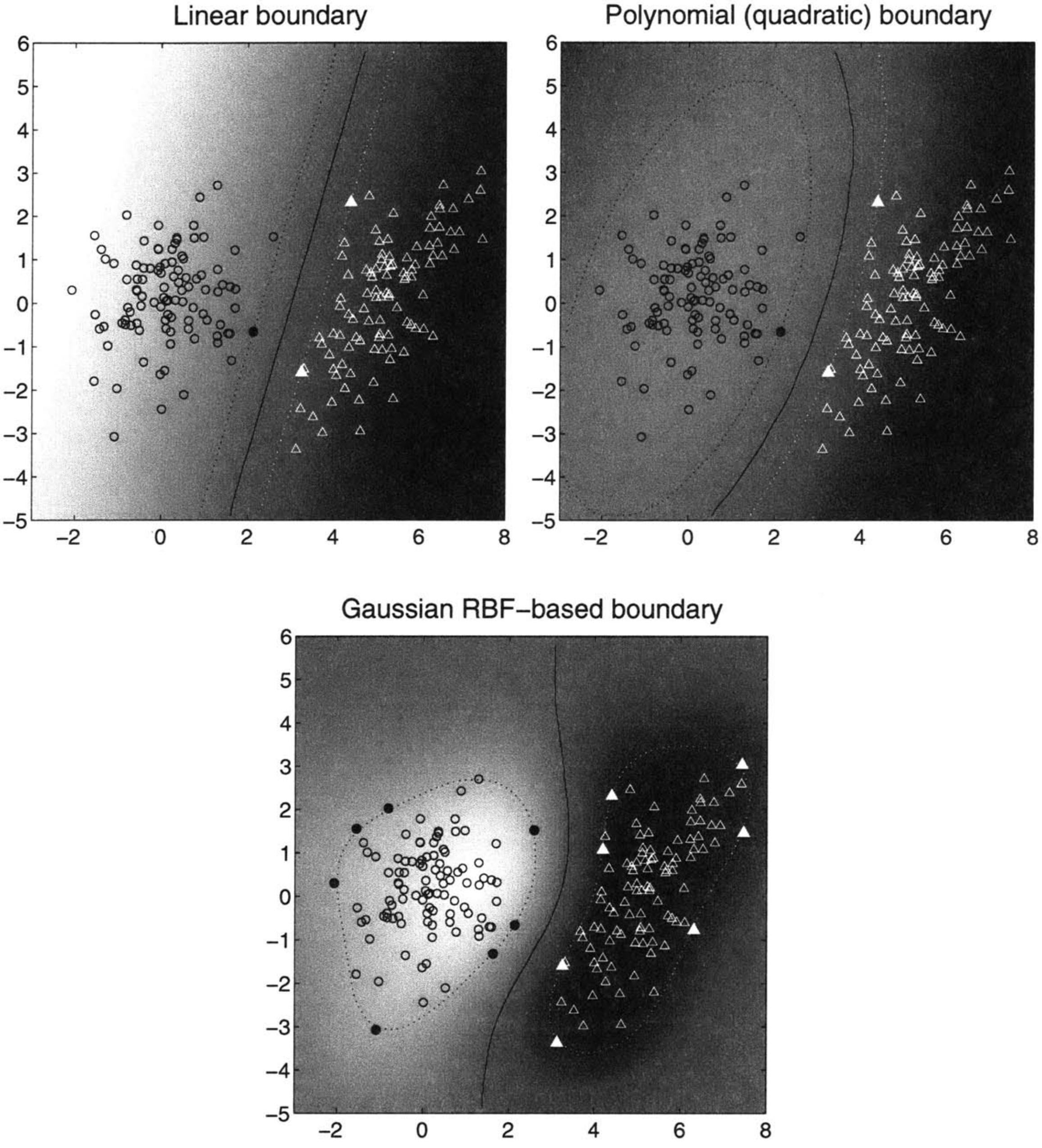


Figure 4-6: SVM training results for different kernels. The support vectors are shown as filled makers. The background is painted with the values of the classification function. The separating boundary (solid line) is the zero level set of the classification function, while the ±1 level sets define the margin (dotted lines).

We trained a linear, a quadratic and a Gaussian RBF classifiers on this data set. The width of the kernel for the RBF classifier ($\gamma = 6$) was chosen based on the VC-bound (4.32). Figure 4-6 shows the resulting classifiers. The background of each image is painted with the continuous values of the classifier function whose zero level set defines the separating boundary (solid line). We show the support vectors by drawing them as filled markers. The support vectors define a margin corridor, i.e., the ± 1 level sets of the classifier (dotted lines). Note that there are markers very close to the margin corridor that are not filled. The value of the classifier at these points was very close to ± 1, but not within the precision of the algorithm.

We observe that the shape of the classifier function, the separating boundary and the margin corridor depend on the kernel used by the algorithm. In places where the data provide enough evidence (lower cluster of support vectors), all three classifiers agree on the location of the separating boundary. However, in places where data are sparse, the shape of the separating boundary and the resulting support vectors are influenced heavily by the shape of the kernel function. This is especially prominent for the Gaussian RBF kernel that produces a margin corridor that is significantly different from the linear and the polynomial ones. However, as more training data become available, the estimated boundary will eventually converge to the maximum likelihood boundary if we allow kernels of arbitrary complexity.

4.6 Summary

We use the Support Vector Machines algorithm to train a classifier that captures differences between the two example groups. In addition to its theoretical foundation, the method has been demonstrated to be robust to overfitting in small data sets and to perform well in many applications. The Structural Risk Minimization Principle provides a framework for systematic exploration of increasingly complex classifier families, while optimizing for generalization performance of the resulting classifier.

For any kernel function, the algorithm produces a classifier function that is a linear combination of kernel instances located at the training examples. The coefficients in the linear combination are determined by solving a constrained quadratic optimization problem. The training examples that participate in the linear combination with non-zero coefficients are called support vectors, as they define, or support, the separating boundary between the two classes.

In application to the shape analysis problem, the algorithm uses the distance transform feature vectors extracted from the input images to estimate the best classifier for discriminating between the two groups of subjects based on the shape of the organ. We can also estimate the expected performance of the resulting classifier based on the statistical tests presented in this chapter. However, we are much more interested in extracting and understanding shape differences captured by the classifier function. The next chapter presents a novel analysis technique that yields an explicit description of the differences between the classes represented implicitly by the classification function.

Chapter 5

Discriminative Direction for Kernel Based Classifiers

The classifier function constructed during the training phase implicitly encodes the differences in data between the two classes. The classifier can be used to label new examples, and in many application domains, such as character recognition, text classification and others, this constitutes the final goal of the learning stage. In medical image analysis, we are far more interested in understanding the nature of the differences captured by the classifier than in using it for labeling new examples. These differences, expressed in terms of the original images or shapes, can provide an insight into the anatomical implications of shape differences detected by the learning algorithm. Furthermore, we would argue that studying the spatial structure of the data captured by the classification function is important in any application, as it illuminates the nature of the differences between the classes and can potentially help in improving the technique.

The analysis presented in this chapter addresses exactly this problem. We introduce and derive a *discriminative direction* at every point in the original feature space with respect to a given classifier. Informally speaking, the discriminative direction tells us how to change any input example to make it look more like an example from another class without introducing any irrelevant changes that possibly make it more similar to other examples from the same class. It allows us to characterize shape differences captured by the classifier and to express them as deformations of the original shapes.

This chapter is organized as follows. We start with a formal definition of the discriminative direction and explain how it can be estimated from the classification function. We then present some special cases, in which the computation can be simplified significantly due to a particular structure of the model. Summary and discussion on related work in the field of kernel methods conclude this chapter.

5.1 Discriminative Direction

Equations (4.27) and (4.28) imply that the classification function $f_K(\mathbf{x})$ is directly proportional to the signed distance from the input point to the separating boundary computed in the higher dimensional space defined by the mapping Φ_K. In other words, the function output depends only on the projection of vector $\Phi_K(\mathbf{x})$ onto $\mathbf{w}$ and completely ignores the component of $\Phi_K(\mathbf{x})$ that is perpendicular to $\mathbf{w}$. This suggests that in order to create a displacement of $\Phi_K(\mathbf{x})$ that corresponds to the differences between the two classes, one should change the vector's projection onto $\mathbf{w}$ while keeping its perpendicular component the same. In the linear case, we can easily perform this operation, since we have access to the image vectors, $\Phi_K(\mathbf{x}) = \mathbf{x}$. This is similar to visualization techniques typically used in linear generative modeling, where the data variation is captured using PCA, and new samples are generated by changing a single principal component at a time. However, this approach is infeasible in the non-linear case, because we do not have access to the image vectors $\Phi_K(\mathbf{x})$'s. Furthermore, the resulting vector might not even have a source in the original feature space (i.e., there might be no vector in the original spaces that maps into the resulting vector in the higher dimensional range space), as the image vectors $\Phi_K(\mathbf{x})$'s do not populate the entire space $\mathbb{F}$, but rather form a manifold of lower dimensionality whose geometry is fully defined by the kernel function K (Figure 5-1). We will refer to this manifold as the *target manifold* in our discussion.

Our solution is to search for the direction around the feature vector $\mathbf{x}$ in the

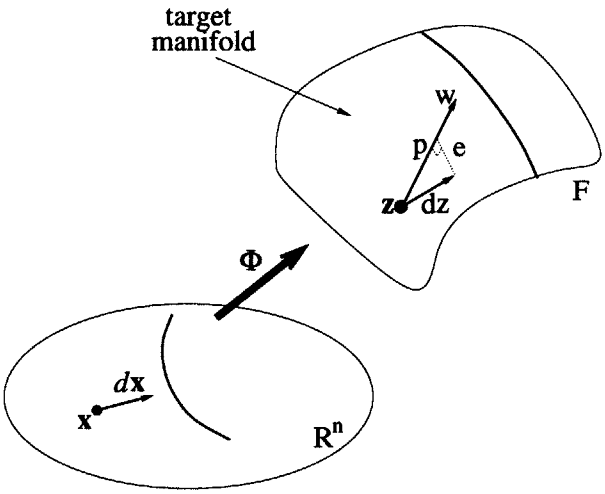


Figure 5-1: Discriminative direction.

original space that minimizes the divergence of its image $\Phi_K(\mathbf{x})$ from the direction of the projection vector $\mathbf{w}$. We call it a *discriminative direction*, as it represents the direction that affects the output of the classifier while introducing as little irrelevant change as possible into the input vector.

Formally, as we move from $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}$ in $\mathbb{R}^n$ (Figure 5-1), the image vector in space $\mathbb{F}$ changes by

$$d\mathbf{z} = \Phi_K(\mathbf{x} + d\mathbf{x}) - \Phi_K(\mathbf{x}). \tag{5.1}$$

This displacement can be thought of as a vector sum of its projection onto $\mathbf{w}$:

$$\mathbf{p} = \frac{\langle d\mathbf{z} \cdot \mathbf{w} \rangle}{\|\mathbf{w}\|} \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\langle d\mathbf{z} \cdot \mathbf{w} \rangle}{\langle \mathbf{w} \cdot \mathbf{w} \rangle} \mathbf{w}, \tag{5.2}$$

and its deviation from $\mathbf{w}$:

$$\mathbf{e} = d\mathbf{z} - \mathbf{p} = d\mathbf{z} - \frac{\langle d\mathbf{z} \cdot \mathbf{w} \rangle}{\langle \mathbf{w} \cdot \mathbf{w} \rangle} \mathbf{w}. \tag{5.3}$$

The discriminative direction is the solution of the following optimization problem:

$$\text{minimize} \qquad \mathcal{E}(d\mathbf{x}) = \|\mathbf{e}\|^2 = \langle d\mathbf{z} \cdot d\mathbf{z} \rangle - \frac{\langle d\mathbf{z} \cdot \mathbf{w} \rangle^2}{\langle \mathbf{w} \cdot \mathbf{w} \rangle} \tag{5.4}$$

$$\text{s.t.} \qquad \|d\mathbf{x}\|^2 = \epsilon. \tag{5.5}$$

Since the cost function depends only on dot products of vectors in the space $\mathbb{F}$, it can be computed using the kernel function K:

$$\begin{aligned}
\langle \mathbf{w} \cdot \mathbf{w} \rangle &= \left\langle \sum_k \alpha_k y_k \Phi_K(\mathbf{x}_k) \cdot \sum_m \alpha_m y_m \Phi_K(\mathbf{x}_m) \right\rangle & (5.6)\\
&= \sum_{k,m} \alpha_k \alpha_m y_k y_m \left\langle \Phi_K(\mathbf{x}_k) \cdot \Phi_K(\mathbf{x}_m) \right\rangle & (5.7)\\
&= \sum_{k,m} \alpha_k \alpha_m y_k y_m K(\mathbf{x}_k, \mathbf{x}_m), & (5.8)
\end{aligned}$$

$$\begin{aligned}
\langle d\mathbf{z} \cdot \mathbf{w} \rangle &= \sum_k \alpha_k y_k \left\langle (\Phi_K(\mathbf{x} + d\mathbf{x}) - \Phi_K(\mathbf{x})) \cdot \Phi_K(\mathbf{x}_k) \right\rangle & (5.9)\\
&= \sum_k \alpha_k y_k (K(\mathbf{x} + d\mathbf{x}, \mathbf{x}_k) - K(\mathbf{x}, \mathbf{x}_k)) & (5.10)\\
&= \sum_k \alpha_k y_k \sum_i \left. \frac{\partial K(\mathbf{u}, \mathbf{v})}{\partial u_i} \right|_{(\mathbf{u}=\mathbf{x}, \mathbf{v}=\mathbf{x}_k)} dx_i & (5.11)\\
&= \nabla f_K(\mathbf{x}) d\mathbf{x}, & (5.12)
\end{aligned}$$

where $\nabla f_K(\mathbf{x})$ is the gradient of the classifier function f_K evaluated at $\mathbf{x}$ and represented by a row-vector, and

$$\begin{aligned}
\langle d\mathbf{z} \cdot d\mathbf{z} \rangle &= \left\langle (\Phi_K(\mathbf{x} + d\mathbf{x}) - \Phi_K(\mathbf{x})) \cdot (\Phi_K(\mathbf{x} + d\mathbf{x}) - \Phi_K(\mathbf{x})) \right\rangle & (5.13)\\
&= K(\mathbf{x} + d\mathbf{x}, \mathbf{x} + d\mathbf{x}) + K(\mathbf{x}, \mathbf{x}) - 2K(\mathbf{x} + d\mathbf{x}, \mathbf{x}) & (5.14)\\
&= \sum_{i,j} \left(\frac{1}{2} \left. \frac{\partial^2 K(\mathbf{u}, \mathbf{u})}{\partial u_i \partial u_j} \right|_{(\mathbf{u}=\mathbf{x})} - \left. \frac{\partial^2 K(\mathbf{u}, \mathbf{v})}{\partial u_i \partial u_j} \right|_{(\mathbf{u}=\mathbf{x}, \mathbf{v}=\mathbf{x})} \right) dx_i \, dx_j & (5.15)\\
&= \sum_{i,j} \left. \frac{\partial^2 K(\mathbf{u}, \mathbf{v})}{\partial u_i \partial v_j} \right|_{(\mathbf{u}=\mathbf{x}, \mathbf{v}=\mathbf{x})} dx_i \, dx_j & (5.16)\\
&= d\mathbf{x}^T H_K(\mathbf{x}) d\mathbf{x}, & (5.17)
\end{aligned}$$

where matrix $H_K(\mathbf{x})$ is one of the (equivalent) off-diagonal quarters of the Hessian of the kernel function K, evaluated at $(\mathbf{x}, \mathbf{x})$. Substituting into Equation (5.4), we obtain

$$\text{minimize} \qquad \mathcal{E}(d\mathbf{x}) = d\mathbf{x}^T \left(H_K(\mathbf{x}) - \|\mathbf{w}\|^{-2} \nabla f_K^T(\mathbf{x}) \nabla f_K(\mathbf{x}) \right) d\mathbf{x} \tag{5.18}$$

$$\text{s.t.} \qquad \|d\mathbf{x}\|^2 = \epsilon. \tag{5.19}$$

The solution to this problem is the eigenvector of matrix

$$Q_K(\mathbf{x}) = H_K(\mathbf{x}) - \|\mathbf{w}\|^{-2} \nabla f_K^T(\mathbf{x}) \nabla f_K(\mathbf{x}) \tag{5.20}$$

that corresponds to the smallest eigenvalue. Note that in general, the matrix $Q_K(\mathbf{x})$ and its smallest eigenvector are not the same for different points in the original space, and need to be estimated for every input vector $\mathbf{x}$. Furthermore, each solution defines two opposite directions in the input space, corresponding to the positive and the negative projections onto $\mathbf{w}$. We want to deform the input example towards the opposite class and therefore assign the direction of increasing function value to the examples with label -1 and the direction of the decreasing function to the examples with label 1.

Obtaining a closed-form solution of this minimization problem could be desired, or even necessary, if the dimensionality of the feature space is high and computing the smallest eigenvector is computationally expensive. Below, we demonstrate how a particular form of the matrix $H_K(\mathbf{x})$ can lead to an analytical solution for a large family of kernel functions. While a very specialized structure of $H_K(\mathbf{x})$ in the example below is sufficient for simplifying the solution significantly, it is by no means necessary, and other kernel families might exist for which the estimation of the discriminative direction does not require solving the full eigenvector problem.

5.1.1 Special Cases. Analytical Solution

We first observe that the second component of the right hand side of Equation (5.20) is a matrix of rank one ($\nabla f_K(\mathbf{x})$ is a row-vector) whose only non-zero eigenvalue is equal to $\|\mathbf{w}\|^{-2} \|\nabla f_K(\mathbf{x})\|^2$ with the corresponding eigenvector $\nabla f_K^T(\mathbf{x})$. The rest of the eigenvectors have the eigenvalue zero and span the null-space of the matrix $\nabla f_K^T(\mathbf{x})\nabla f_K(\mathbf{x})$. Therefore, we might be able to infer some information about the eigenvectors and eigenvalues of $Q_K(\mathbf{x})$ if the matrix $H_K(\mathbf{x})$ is of special form.

Let's consider a case when $H_K(\mathbf{x})$ is a multiple of the identity matrix:

$$H_K(\mathbf{x}) = cI. \tag{5.21}$$

Since any vector is an eigenvector of the identity matrix with the eigenvalue one, adding cI to an arbitrary matrix does not change the eigenvectors of that matrix, but increments all of its eigenvalues by c. Consequently, the smallest eigenvector of matrix

$$Q_K(\mathbf{x}) = cI - \|\mathbf{w}\|^{-2}\nabla f_K^T(\mathbf{x})\nabla f_K(\mathbf{x}) \tag{5.22}$$

is equal to the largest eigenvector of the matrix $\nabla f_K^T(\mathbf{x})\nabla f_K(\mathbf{x})$ in this case:

$$\begin{aligned} d\mathbf{x}^* &= \nabla f_K^T(\mathbf{x}) && (5.23)\\ \mathcal{E}(d\mathbf{x}^*) &= c - \|\mathbf{w}\|^{-2}\|\nabla f_K^T(\mathbf{x})\|^2 && (5.24)\end{aligned}$$

We will show in this section that both for the linear kernel and, more surprisingly, for the Gaussian kernel, the matrix $H_K(\mathbf{x})$ is of the right form to yield an analytical solution. Furthermore, this solution is equal to the gradient of the classification function. It is well known that to achieve the fastest change in the value of a function, one should move along its gradient, but in the case of the linear and the Gaussian kernels, the gradient also corresponds to the direction that distinguishes between the two classes while minimizing inter-class variability.

Dot product kernels. This is a family of kernels of the form

$$K(\mathbf{u},\mathbf{v}) = k(\langle\mathbf{u}\cdot\mathbf{v}\rangle), \tag{5.25}$$

where k is a function of one variable. For any dot product kernel,

$$\left.\frac{\partial^2 K(\mathbf{u},\mathbf{v})}{\partial u_i \partial v_j}\right|_{(\mathbf{u}=\mathbf{x},\mathbf{v}=\mathbf{x})} = k'(\|\mathbf{x}\|^2)\delta_{ij} + k''(\|\mathbf{x}\|^2)x_i x_j, \tag{5.26}$$

and therefore $H_K(\mathbf{x}) = cI$ for all $\mathbf{x}$ if and only if $k''(\|\mathbf{x}\|^2) \equiv 0$, i.e., when k is a linear function. Thus the linear kernel is the only dot product kernel for which this simplification is relevant. In the linear case, $H_K(\mathbf{x}) = I$, and the discriminative direction is defined as

$$d\mathbf{x}^* = \nabla f_K^T(\mathbf{x}) = \mathbf{w} = \sum \alpha_k y_k \mathbf{x}_k \tag{5.27}$$

$$\mathcal{E}(d\mathbf{x}^*) = 0. \tag{5.28}$$

This is not entirely surprising, as the classifier is a linear function in the original space and we can move precisely along $\mathbf{w}$.

Polynomial kernels are a special case of dot product kernels. For polynomial kernels of degree $d \geq 2$,

$$\left.\frac{\partial^2 K(\mathbf{u},\mathbf{v})}{\partial u_i \partial v_j}\right|_{(\mathbf{u}=\mathbf{x},\mathbf{v}=\mathbf{x})} = d(1+\|\mathbf{x}\|^2)^{d-1}\delta_{ij} + d(d-1)(1+\|\mathbf{x}\|^2)^{d-2}x_i x_j. \tag{5.29}$$

$H_K(\mathbf{x})$ is not necessarily diagonal for all $\mathbf{x}$, and we have to solve the general eigenvector problem to identify the discriminative direction.

Distance kernels. This is a family of kernels of the form

$$K(\mathbf{u},\mathbf{v}) = k(\|\mathbf{u}-\mathbf{v}\|^2), \tag{5.30}$$

where k is a function of one variable. The members of this family are often called Radial Basis Functions (RBF) because of their radial symmetry. For a distance kernel,

$$\left.\frac{\partial^2 K(\mathbf{u},\mathbf{v})}{\partial u_i \partial v_j}\right|_{(\mathbf{u}=\mathbf{x},\mathbf{v}=\mathbf{x})} = -2k'(0)\delta_{ij}, \tag{5.31}$$

and therefore the discriminative direction can be determined analytically:

$$\begin{aligned} d\mathbf{x}^* &= \nabla f_K^T(\mathbf{x}) && (5.32) \\ \mathcal{E}(d\mathbf{x}^*) &= -2k'(0) - \|\mathbf{w}\|^{-2} \|\nabla f_K^T(\mathbf{x})\|^2. && (5.33) \end{aligned}$$

The Gaussian kernels are a special case of the distance kernel family, and yield a closed form solution for the discriminative direction:

$$\begin{aligned} d\mathbf{x}^* &= -\frac{2}{\gamma} \sum_k \alpha_k y_k e^{-\frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{\gamma}} (\mathbf{x} - \mathbf{x}_k) && (5.34) \\ \mathcal{E}(d\mathbf{x}^*) &= \frac{2}{\gamma} - \|\mathbf{w}\|^{-2} \|\nabla f_K^T(\mathbf{x})\|^2. && (5.35) \end{aligned}$$

Unlike the linear case, we cannot achieve zero error, and the discriminative direction is only an approximation. The exact solution is unattainable in this case, as it has no corresponding direction in the original space.

To summarize, the discriminative direction for the linear and the distance kernels is equal to the gradient of the classification function. It simultaneously maximizes the displacement towards the opposite class and minimizes irrelevant intra-class deformation of the input data. For other kernels, we might have to compromise between how fast we advance towards the opposite class and how much of irrelevant change we introduce in the feature vector. The deviation from the optimal direction is minimized by the smallest eigenvector of the matrix defined in Equation (5.20).

5.1.2 Geometric Interpretation

In this section, we provide a geometric intuition for the solution obtained in the previous section. We first remind the reader that we cannot explicitly manipulate elements of the space $\mathbb{F}$, but can only explore the target manifold through search in the original space. Ideally, if we could access vectors in $\mathbb{F}$ directly, we would move the input vector $\Phi_K(\mathbf{x})$ along the projection vector $\mathbf{w}$ and study the change in $\mathbf{x}$

introduced by this process. Moving along the projection vector might force us out of the target manifold, but we must stay on it to be able to perform this operation through manipulation of vectors in the original space. Therefore, every move along $\mathbf{w}$ has to be followed by a projection step that returns us back to the manifold. There are various ways to perform this projection, and in this work we chose to minimize the error between the approximation vector and the exact solution. We also note that different measures (e.g., the length of the projection of the resulting displacement vector onto $\mathbf{w}$) might be more appropriate for other applications.

We perform the search in the original space by considering all points on an infinitesimally small sphere centered at the original input vector $\mathbf{x}$. In the range space of the mapping function Φ_K, the images of points $\mathbf{x} + d\mathbf{x}$ form an ellipsoid defined by the quadratic form

$$d\mathbf{z}^T d\mathbf{z} = d\mathbf{x}^T H_K(\mathbf{x}) d\mathbf{x}. \tag{5.36}$$

For $H_K(\mathbf{x}) \sim I$, the ellipsoid becomes a sphere, all $d\mathbf{z}$'s are of the same length, and the minimum of error in the displacement vector $d\mathbf{z}$ corresponds to the maximum of the projection of $d\mathbf{z}$ onto $\mathbf{w}$. Therefore, the discriminative direction is parallel to the gradient of the classifier function. If $H_K(\mathbf{x})$ is of any other form, the length of the displacement vector $d\mathbf{z}$ changes as we vary $d\mathbf{x}$, and the minimum of the error in the displacement is not necessarily aligned with the direction that maximizes the projection[1].

5.2 Selecting Inputs

Given any input example, we can compute the discriminative direction that represents the differences between the two classes captured by the classifier in the neighborhood of the example. But how should we choose the input examples for which to com-

[1] One can show that our sufficient condition, $H_K(\mathbf{x}) \sim I$, implies that the target manifold is locally flat, i.e., its Riemannian curvature is zero. This follows directly from the definition of curvature [62]. Curvature and other properties of target manifolds have been studied extensively for different kernel functions [3, 12]. Understanding the geometry of the kernel spaces can provide a useful insight into the problem of selecting an appropriate kernel for a specific application.

pute the discriminative direction? We argue that in order to study the differences between the classes, one has to examine the input vectors that are close to the separating boundary, namely, the support vectors. The SVMs algorithm identifies the support vectors as training examples that are "the closest" to the separating boundary in terms of the classification function value. This definition can be naturally extended to other learning algorithms, for example, for the nearest neighbor classifier one should examine the training examples whose cells in the Voronoi partition of the space border with cells of examples from the opposite class. Note that this approach is significantly different from generative modeling, where a "typical" representative, often constructed by computing the mean of the training data, is used for analysis and visualization (e.g., to compare two different classes, one would compare their typical representatives [17, 43]). In the discriminative framework, we are more interested in the examples that lie close to the opposite class, as they define the differences between the two classes and the optimal separating boundary.

The separating boundary is implicitly estimated as a zero level set of the classification function. The structure of the resulting classifier depends on the classifier family used by the training algorithm, e.g., polynomial or RBF kernels. The zero level set of the function is the best approximation of the optimal separating boundary for this particular family. If the complexity (VC dimension) of the classifier family is sufficient to fully capture the properties of the data, the zero level set of the resulting classifier will converge to the optimal separating boundary as the number of the training examples increases.

Support vectors define a margin corridor whose shape is determined by the kernel type used for training (for example, see Figure 4-6). We can estimate the distance from any support vector to the separating boundary by examining the gradient of the classification function for that vector. A large gradient indicates that the support vector is close to the separating boundary and therefore can provide more information on the spatial structure of the boundary. This provides a natural heuristic for assigning importance weighting to different support vectors in the analysis of the discriminative direction.

5.3 Simple Example

In this section, we demonstrate the discriminative direction analysis on the simple example introduced in Section 4.5. Figure 5-2 shows the discriminative direction for the classifiers estimated for this example in the previous chapter. We display the discriminative direction vectors not just for the support vectors, but also for the training examples that are very close to the margin corridor. The length of the arrows in the plots is proportional to the magnitude of the gradient at the corresponding points.

Similarly to the training results discussed in Section 4.5, the estimates of the discriminative direction for the three different kernels agree in the areas where the data provide strong evidence for the separating boundary localization (the lower cluster of arrows). As we inspect the areas where insufficient data caused the regularization model supplied by the kernel function to drive the separating boundary estimation, we find larger discrepancies among the discriminative direction estimates in the three images.

The discriminative direction is equal to the gradient of the classifier function and could be computed analytically for the linear and the RBF classifiers, but we had to solve the eigenvector problem for the quadratic classifier. While the eigenvector problem is relatively simple in this two-dimensional example, the numerical stability of estimating the smallest eigenvector could be an obstacle for applying this analysis to polynomial classifiers when working with high-dimensional data. To relate this to the problem of shape analysis, the feature vectors based on the distance transforms can contain hundreds, or even thousands, of components. The features are highly redundant, with strong correlation among the values at neighboring voxels. As we will see in the next chapter, we can easily train a classifier that reliably captures the differences between the classes, even when the number of training examples is very small. High dimensionality of the data does not present a challenge to the training algorithm, but could render using polynomial kernels infeasible. This is one of the reasons for our choice of the Gaussian RBF kernels for our application.

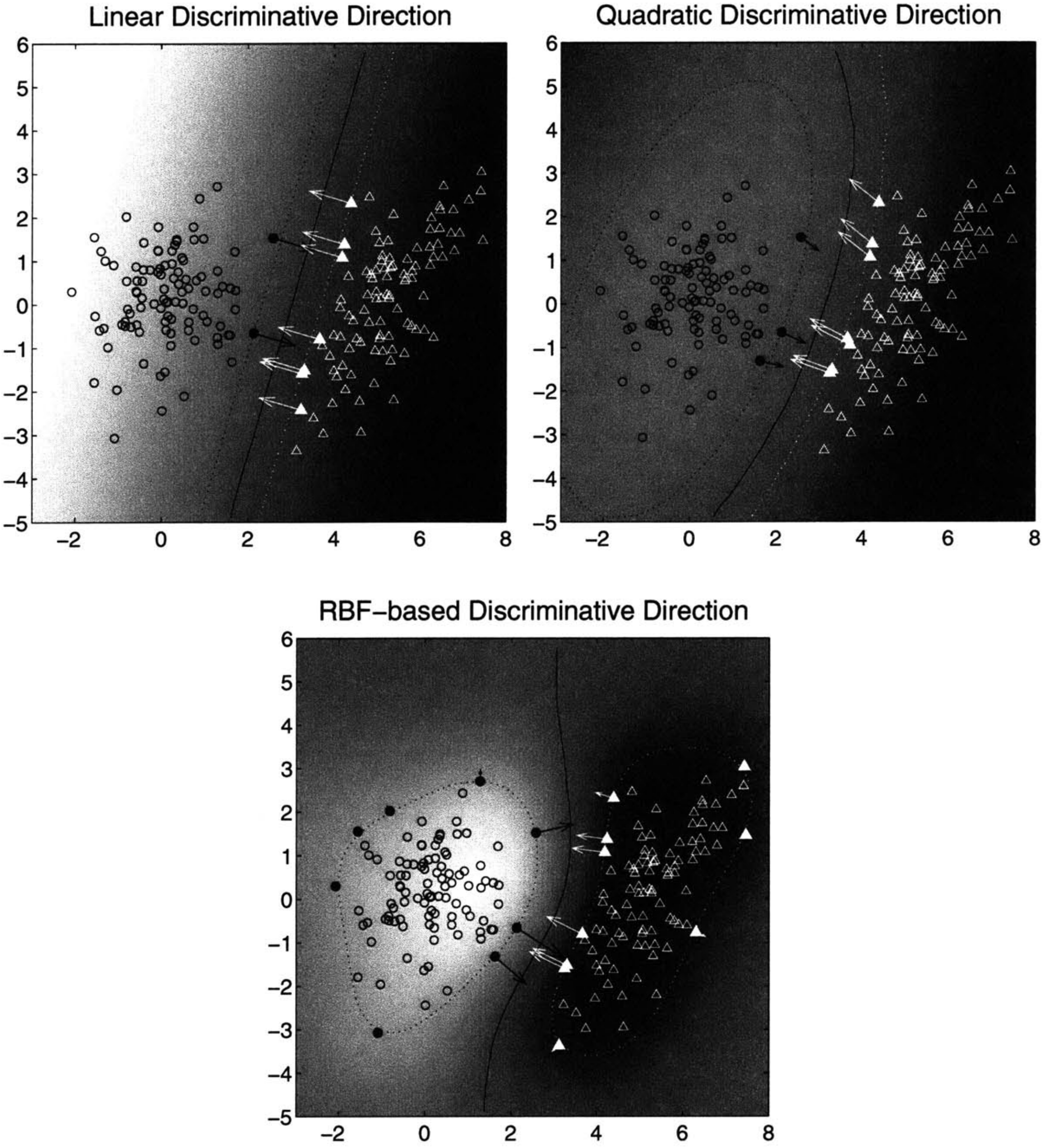


Figure 5-2: Discriminative direction for different kernel classifiers. The discriminative direction is shown for the vectors close to the boundary, including the support vectors. The support vectors are shown as filled makers. The background is painted with the values of the classification function. The separating boundary (solid line) is the zero level set of the classification function, while the ±1 level sets define the margin (dotted lines). The length of the vectors is proportional to the gradient of the classifier function.

5.4 Related Questions

Interpreting the results of kernel based learning in terms of the original features is challenging because the mapping function Φ_K implied by the kernel is not invertible, and therefore we cannot manipulate the vectors in the range space $\mathbb{F}$ and guarantee that there is always a vector in the original space that maps onto the newly generated vector in $\mathbb{F}$. Since only the points on the target manifold can be mapped back to the original feature space, approximation algorithms have to be used that define a way to project arbitrary vectors in the space $\mathbb{F}$ onto the target manifold. These algorithms effectively define an appropriate measure of proximity to the exact solution for a specific problem and then minimize it over the manifold.

A classification problem is the main focus of this work, but kernel methods have also been applied to generative modeling. Kernel-based PCA [56, 58] is a non-linear version of the Principal Component Analysis that uses kernels to implicitly map the original vectors to a higher dimensional space and perform linear modeling in that space. It turns out that computing principal components can be carried out entirely in terms of dot products, and therefore, the "kernel trick" can be applied. The most common use of PCA is for dimensionality reduction, when only first few principal components are used by the model. In this application, once the few corresponding coefficients are computed for a new input vector $\mathbf{x}$, its image vector $\Phi_K(\mathbf{x})$ is replaced by the linear combination of the first few principal components of the model. This new image vector then must be projected back to the original feature space to generate a new feature vector $\mathbf{x}'$ that represents an approximation of the original $\mathbf{x}$ within the reduced model. But if the resulting linear combination does not lie on the target manifold, the back-projection step cannot be performed exactly. Instead, an optimization is carried out to find a point in the original space that maps as close as possible to the newly computed result in the higher dimensional space. This technique was demonstrated in face tracking [53], successfully handling non-linear changes (occlusions, different views) in the input.

One might be able to construct a global optimization problem for the discriminative case as well. For example, we could reflect the image vector around the separating hyperplane and try to find the closest point on the target manifold to the exact solution. The main problem we encountered with such approach is constructing a good proximity measure for the search over the manifold. However, it is one of the interesting extensions to be investigated in the future.

As we mentioned before, the geometry of kernel spaces is a topic of active research in the machine learning community. Interestingly, it can be shown that the kernel function effectively induces a Riemannian metric on the original feature space [12]. Function optimization in arbitrary metric spaces was studied by Amari [2] who introduced a notion of natural gradient to adapt the traditional gradient descent algorithm to spaces with non-Euclidean metric. Similarly to the discriminative direction analysis presented in this chapter, the natural gradient is a direction estimated through an infinitesimal analysis. However, the goal of the analysis is to maximize the change in the function over an infinitesimal neighborhood whose geometry is determined by the metric tensor. In our analysis, $H_K(\mathbf{x})$ plays the role of the metric tensor at point $\mathbf{x}$.

5.5 Summary

We presented an approach to quantifying the classifier's behavior with respect to small changes in the input vectors, trying to answer the following question: what changes would make the original input look more like an example from the other class without introducing irrelevant changes? We introduced the notion of the discriminative direction, which corresponds to the maximum changes in the classifier's response while minimizing irrelevant changes in the input. In our application, this can be used to interpret the differences between the two classes as deformations of the original input shapes, as explained in the next chapter.

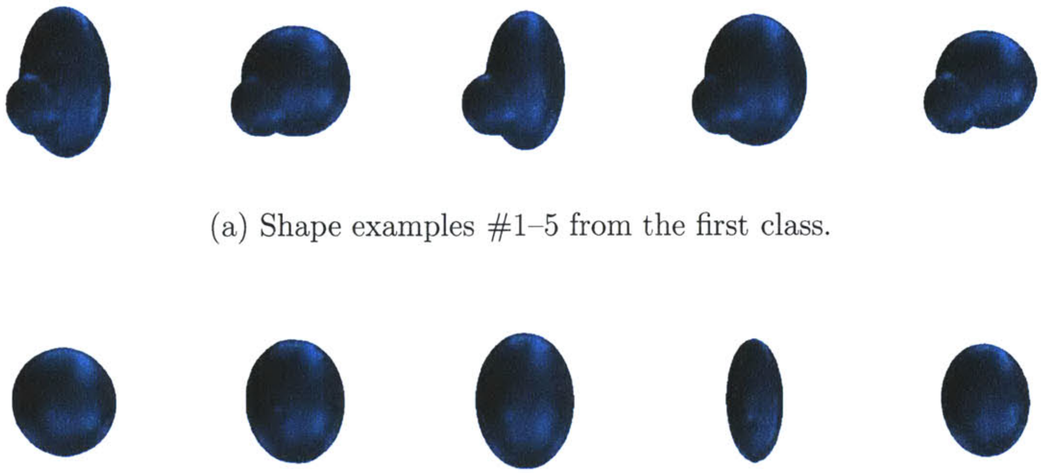
Chapter 6

Experimental Results

This chapter presents the results of applying our approach to several different shape studies. We first explain how the components of the analysis described in the previous chapters are combined into a system and provide implementation details while demonstrating the analysis on a simple artificial example. We then report experimental results for the medical studies on hippocampus and corpus callosum. We conclude this chapter with a discussion of the lessons learned from the experimental data.

6.1 System Overview

As we describe the steps of the algorithm, we will illustrate them on a simulated shape study that contains 30 volumetric images of ellipsoidal shapes of varying sizes. The width, height and thickness of the shapes were sampled uniformly out of a ± 10 voxel range centered at 20, 30 and 40 voxels respectively. We randomly divided the data set into two classes of 10 and 20 examples respectively and added a spherical bump to the shapes in the first class. The bump location was sampled out of a ± 3 voxel range centered on the side of the main ellipsoid. Figure 6-1 illustrates both types of shapes. The number of training shape examples corresponds to what we encounter in real morphological studies. Knowledge of the shape differences between the groups in this simulated experiment makes it is relatively easy to asses the effectiveness of the analysis. Evaluation of the results is more difficult in the real medical studies,



(a) Shape examples #1–5 from the first class.

(b) Shape examples #1–5 from the second class.

Figure 6-1: Artificial training set example. Five shapes are shown from each class.

where the true shape differences, if they exist, are unknown. We will come back to this problem later in the chapter, when discussing medical data.

6.1.1 Shape Representation

For every input scan, we compute the distance transform and use its moments to establish a new coordinate system in the volume, placing the origin at the center of mass and aligning the coordinate frame with the principal axes of inertia. The values of the distance transform are then sampled along the new axes at uniform intervals and concatenated into a feature vector in a row-by-row, slice-by-slice fashion. The result of this step is the training set $\{(\mathbf{x}_k, y_k)\}_{k=1}^{l}$, where $\mathbf{x}_k$ are the feature vectors obtained by sampling the distance transforms, and $y_k \in \{-1, 1\}$ are the corresponding labels defining the membership in one of the two populations.

Since we use the distance transform for feature extraction, the original scans must be segmented prior to the analysis. In all examples presented in this chapter, the computation was based only on the binary segmentation images (voxels inside the shape were assigned value 1, and voxels outside the shape were assigned value 0). We showed the original grayscale images in this dissertation only to illustrate the input data; they were not used by the algorithm.

6.1.2 Statistical Modeling

The training feature vectors and their labels are used by the Support Vector Machines learning algorithm to construct a classifier for labeling new examples. In each experiment we trained a linear and a non-linear classifier. We use the Gaussian RBF kernel SVMs for non-linear classification, largely because of their local nature: changing one of the support vectors by a small amount only affects the separating boundary in the vicinity of that vector. This is an important advantage in the presence of noise in the training examples. An additional, although not as important, reason for using RBF kernels is the ease of computing the discriminative direction.

In each study reported in this work, we systematically explore the space of parameters (the kernel width γ and the constant C) by sampling it on a logarithmic scale. To guarantee that we examine kernels of the width comparable with the pairwise distances between the example feature vectors, we set the lower and the upper ends of the range for γ to the one tenth of the smallest non-zero distance and the ten times the largest distance respectively. The range for the soft margin constant C is the same ($10^{-3} - 10^{3}$) in all our experiments. The training algorithm has been shown empirically to be quite robust to the setting of the parameter C. We have observed in our studies that in the optimal region for γ, the training results were very similar for a wide range of values for C.

For each setting of the parameters, we train a classifier as described in Section 4.3, compute an upper bound on its VC dimension and perform leave-one-out cross-validation for estimation of the generalization performance. We used both bounds to select the best classifier. The two bounds typically agree on the optimal parameter settings in successful studies, which provides an additional indicator for the robustness of the resulting classifier.

The shapes in our example were easily separable using a linear classification function, but we also trained a non-linear classifier for comparison. Both classifiers separate the data perfectly. The cross-validation accuracy is 100% as well. The number of support vectors is significantly higher for the RBF classifier (13) than for the linear one (6), but its estimated VC dimension (11.87) is lower than that for the linear classifier (31.16).

6.1.3 From The Classifier to Shape Differences

Once the optimal classifier has been constructed, we proceed to estimate the discriminative direction, as described in Chapter 5. The discriminative direction is equal to the gradient of the classification function for both the linear and the Gaussian kernels. We compute the discriminative direction $d\mathbf{x}^*$ at the support vectors identified in the training phase and use the magnitude of the classifier gradient to rank the vectors' importance in representing the shape differences captured by the classifier. For the linear classifier, the direction and the magnitude of the gradient is the same for all support vectors. For the non-linear classifier, we have to evaluate them separately for each support vector.

Figure 6-2 shows the discriminative direction $d\mathbf{x}^*$ estimated for the shapes in the second class based on the linear classifier. The change in the distance transform for the shapes in the first class can be obtained by changing the sign of all the components of $d\mathbf{x}^*$. Since we are interested only in the direction of the change, the magnitude of the discriminative direction vector $d\mathbf{x}^*$ is irrelevant. We consider the changes in the distance transform values at different voxels relative to each other, rather than on the absolute scale. Consequently, we omit absolute values from the figures and use colorbars to provide information on the relative values of the resulting feature vector components. We have used the discriminative direction vector shown in Figure 6-2 in Chapter 3 to demonstrate the local parameterization of the distance transform manifold based on the deformations of the surface meshes.

The volumetric image in Figure 6-2 is smooth[1], even though we did not augment the learning algorithm with any explicit information on the type of dependencies or

[1] Here, we refer to the correlation of the voxel values in the volumetric discriminative direction for one specific shape rather than the correlation of the discriminative direction vectors at close points in the shape space.

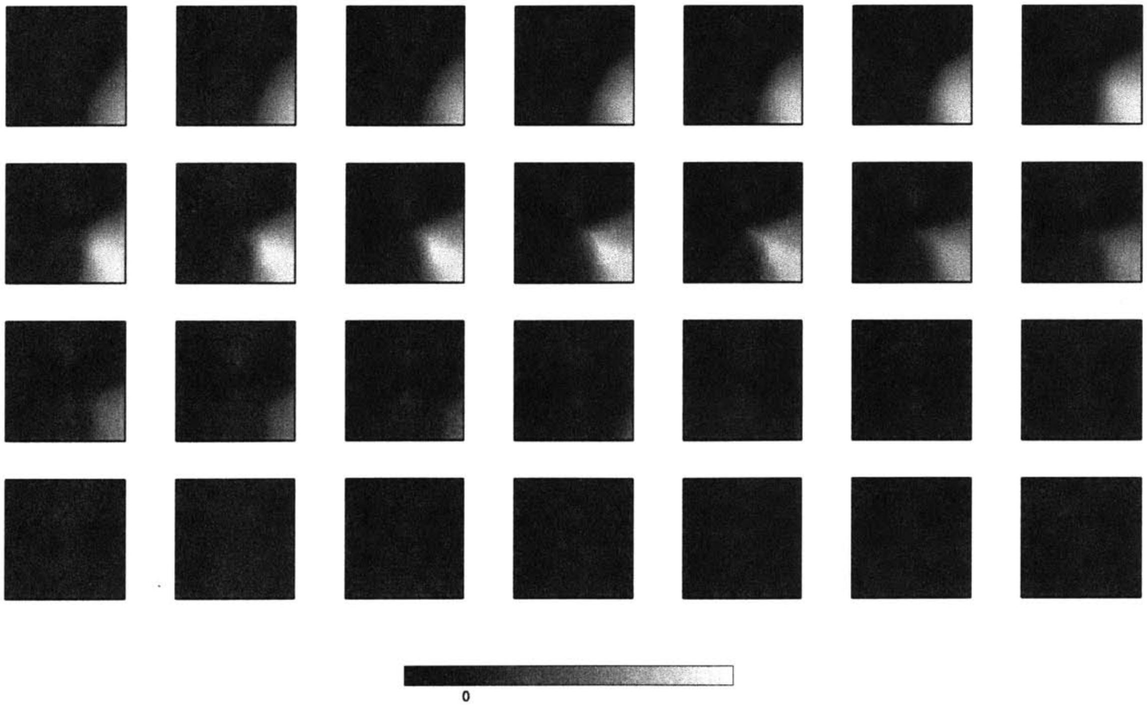


Figure 6-2: Volumetric discriminative direction $d\mathbf{x}^*$ for the shapes in the second class based on the linear classifier. Positive values (bright intensities) in the image correspond to increasing the values of the same voxels in the distance transform, negative values (dark intensities) correspond to reducing the values of the distance transform. The absolute scaling is omitted since we are interested only in the direction of the vector, not its magnitude. The intensity that corresponds to zero change in the distance transform is shown on the colorbar. Every second slice is shown.

spatial variation in the feature vectors. The resulting hypothesis is consistent with the smooth nature of the input distance transforms. Sharp discontinuities in the gradient of the classification function that do not coincide with the edges (skeleton branches) in the distance transform would indicate that the smooth nature of the input data did not get captured by the classifier from the small number of examples in the training set.

Most weight in the gradient image is concentrated in the area of the bump. Positive values of the discriminative direction indicate that the distance transform values should decrease in the examples from the first class and increase in the examples from the second class. Thus, the learning algorithm correctly identified the main difference between the classes. A secondary effect found in Figure 6-2 is a slight difference in

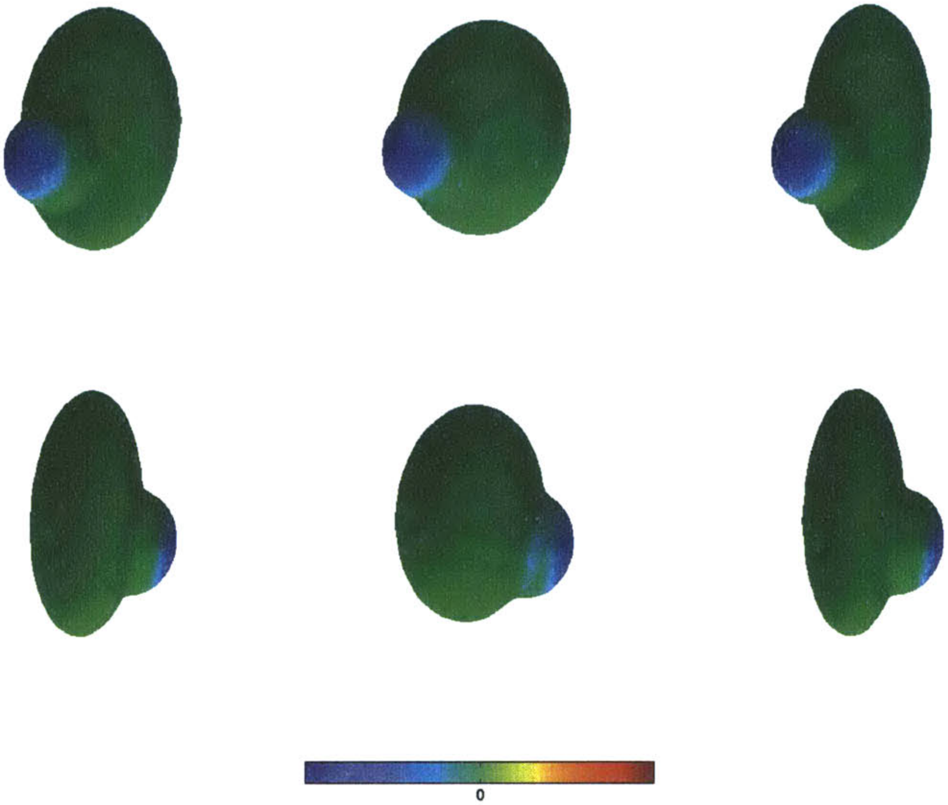


Figure 6-3: Deformation of the three support vectors from the first class computed using the discriminative direction for the linear classifier. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

the overall shape: a small enlargement in width and reduction in hight is required of shapes in the first class (the shapes in the second class have to undergo a change of the opposite sign). We believe this is an artifact caused by the small number of training examples. The same global size differences are detected if the learning is performed on the training data set without adding the bump to the shapes in the first class, but the cross-validation results indicate that this specific difference cannot be used to classify the shapes reliably. Furthermore, as we increased the number of training examples, the size-related effects disappeared from the resulting discriminative direction.

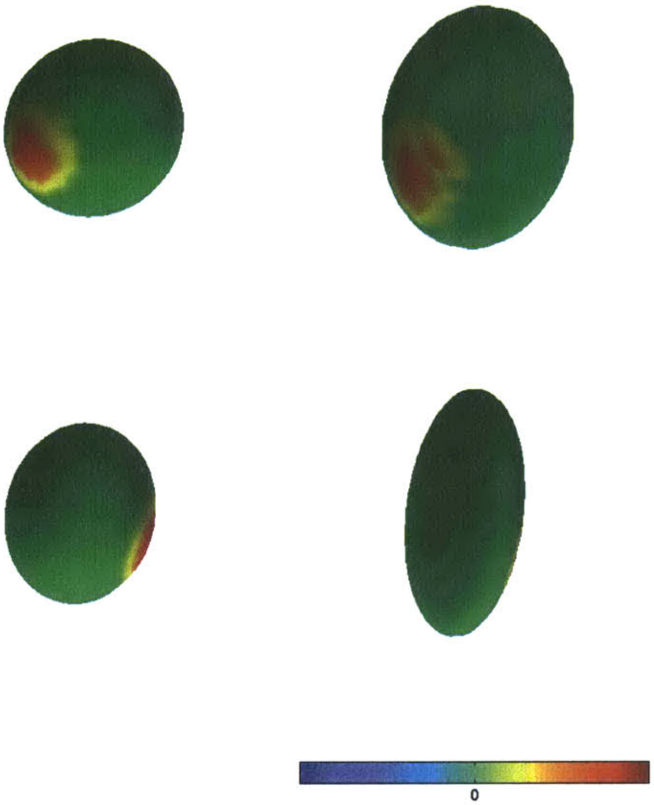


Figure 6-4: Deformation of the three support vectors from the second class computed using the discriminative direction for the linear classifier. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

6.1.4 Shape Differences As Deformations

The next step in the analysis is to compute the deformation of the support vectors $d\mathbf{s}$ that approximates the discriminative direction $d\mathbf{x}^*$, as explained in Chapter 3. To remind the reader, we project the vector $d\mathbf{x}^*$ onto the space of infinitesimal changes of the distance transform. The projection vector $d\mathbf{x}^o$ defines a deformation of the surface mesh $d\mathbf{s}$ that changes the original shape according to the discriminative direction.

Figure 6-3 and Figure 6-4 show the estimated deformation $d\mathbf{s}$ for the 6 support vectors of the linear classifier. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no

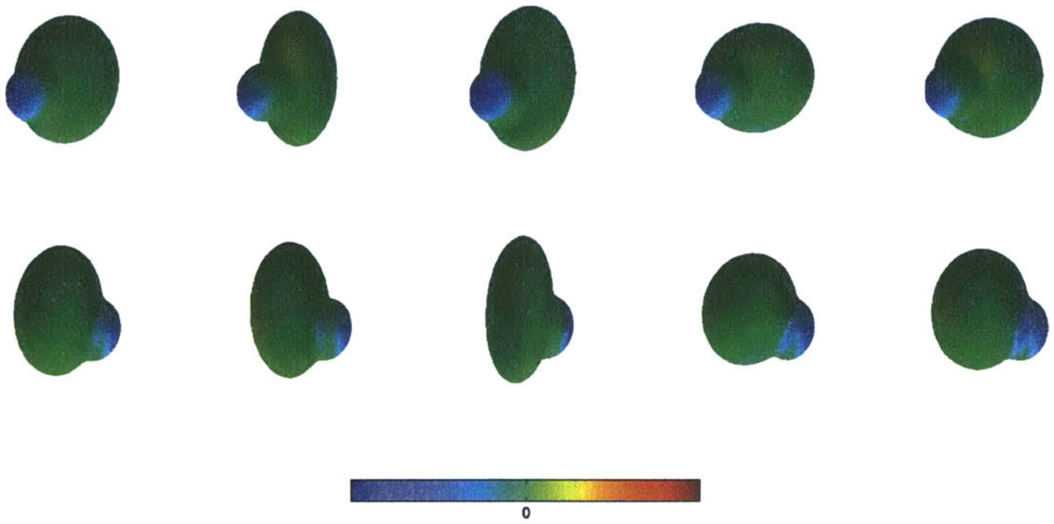


Figure 6-5: Deformation of the first 5 support vectors from the first class computed using the discriminative direction for the Gaussian RBF classifier. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

deformation) to red (outwards). The shape differences defined by the deformation $d\mathbf{s}$ are localized to the area of the bump, similarly to the volumetric discriminative direction $d\mathbf{x}^*$ in Figure 6-2. The support vectors are redundant, representing virtually identical deformation. We encountered this phenomenon in all our experiments: some support vectors are so close to each other in the feature space that they define very similar deformations.

As we mentioned before, the surface-based representation of the shape differences between the two classes is significantly easier to interpret than the volumetric results. The advantages of surface-based visualization become even more apparent when we work with real anatomical shapes and use non-linear classifiers that yield a different volumetric discriminative direction for every support vector. In order to infer a top-level description of shape differences from the changes in the distance transform, one has to argue about shape differences in a way that effectively reduces the volumetric representation to a surface-based deformation description. Our analysis presented in Section 3.3 formalizes this line of argument. We relied on the volumetric display in our

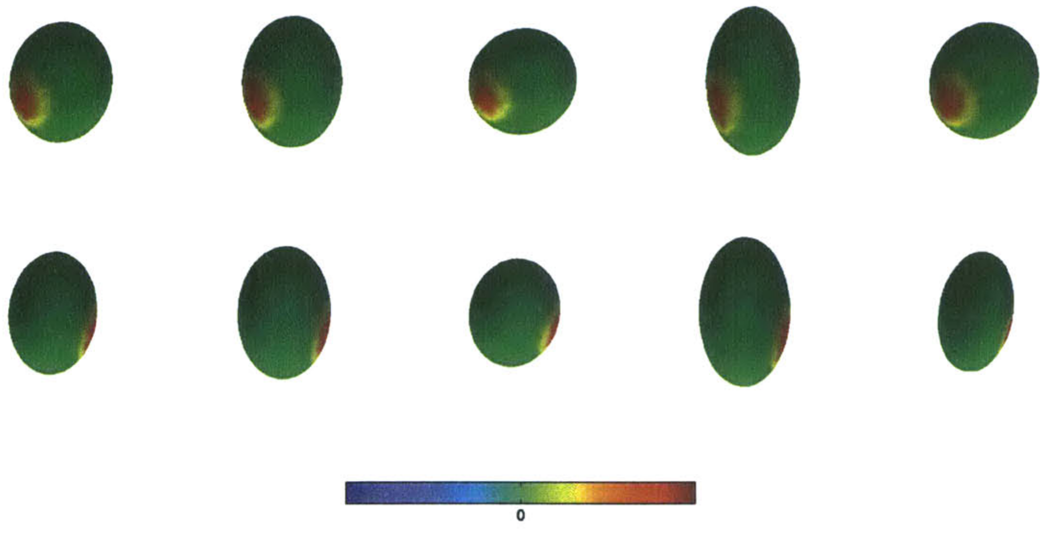


Figure 6-6: Deformation of the first 5 support vectors from the second class computed using the discriminative direction for the Gaussian RBF classifier. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

discussion in Chapter 3 and in this section to guide the reader through the analysis steps, but will use surface-based representation in the remainder of this chapter.

Figure 6-5 and Figure 6-6 present the results of the non-linear classification using Gaussian RBF kernels. The deformation was computed using the discriminative direction for the first 5 support vectors in each class. This classifier has 13 support vectors, 5 from the first class and 8 from the second class. We sorted the support vectors using the magnitude of the classifier gradient as an importance criterion. The magnitude of the gradient decreases by a factor of 7 from the first support vector to the last one. We can see that the shape differences captured by the non-linear classifier are very similar to the results of the linear classification, which is not surprising since the margin between the classes is wide enough for the linear classifier to separate the data perfectly. The non-linear classifier might be able to match the curvature of the margin corridor better, but it should identify a similar structure in the data. In cases of classes that are not linearly separable, we expect the differences between the linear and the non-linear classifiers to be more substantial, as they trade-off the training error and the model complexity in a different manner.

To summarize the steps of the algorithm, we extract the distance transforms from the segmented volumetric images in the training data set and pass them to the SVMs learning algorithm that produces a classification function for labeling new example images. We then compute the gradient of the classification function for the support vectors of the classifier and use its direction as the discriminative direction and its magnitude as the importance weighting for ranking the support vectors. The discriminative direction for each support vector is then projected onto the manifold of the valid distance transforms to produce a deformation that best approximates the discriminative direction. The deformation is visualized for inspection and can be analyzed further to identify the areas of differences between the classes.

6.2 Artificial Example 2

The simple example we used to illustrate the steps of the algorithm involved shape differences in a single spot on the surface of the training objects. In this section, we modify the shapes in the first class by adding an indentation in a different location (Figure 6-7). Thus, the shape differences consist of two individual deformations that have to be identified by the algorithm. The depth of the indentation was 4 pixels, which is smaller than the global size variability. The shapes in the second class were unchanged in this experiment.

Since the shapes in the two classes are even more distinct now, the training algorithm faces an easier learning task. We show the results only for the linear classifier in this section, as it was sufficient to capture the shape differences: both the training and the cross-validation accuracy was 100%.

Figure 6-8 shows the volumetric discriminative direction for the linear classifier. Note that in addition to the deformation in the area of the bump, a deformation of the opposite sign appears in the place corresponding to the indentation in the shapes from the first class. The intensity scaling in this figure is slightly different from that in Figure 6-2, as it contains a strong negative component.

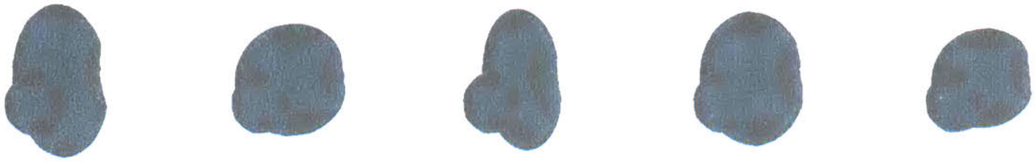


Figure 6-7: Additional shape change in the first class. Five example shapes are shown from the first class. The shapes in the second class were left unchanged. Compare to Figure 6-1.

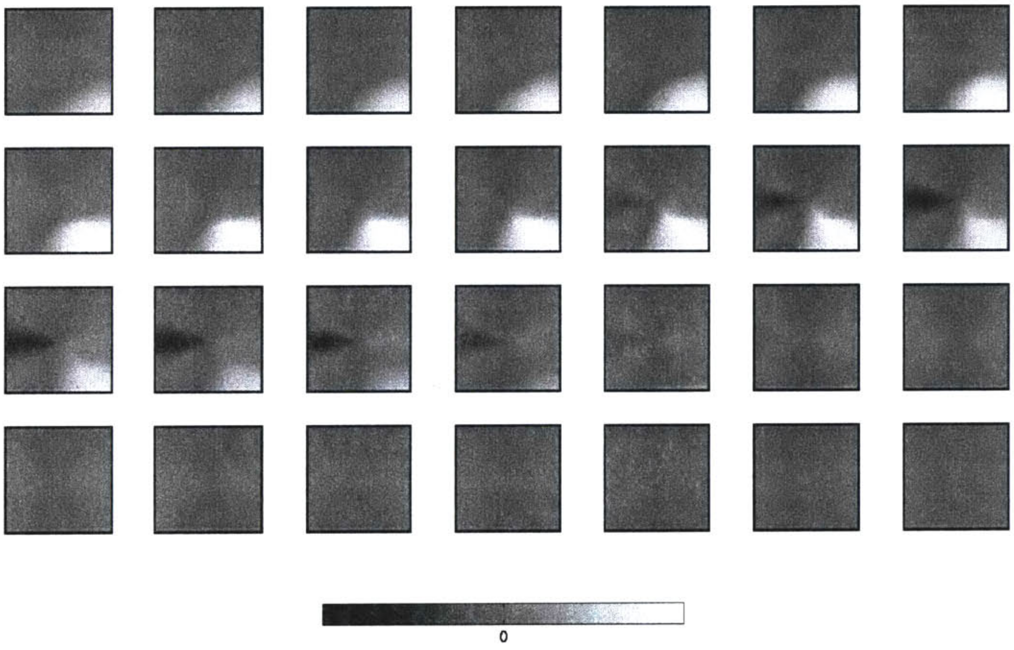


Figure 6-8: Volumetric discriminative direction for the shapes in the second class. Positive values (bright intensities) in the image correspond to increasing the values of the same voxels in the distance transform, negative values (dark intensities) correspond to reducing the values of the distance transform. The absolute scaling is omitted since we are interested only in the direction of the vector, not its magnitude. The intensity that corresponds to zero change in the distance transform is shown on the colorbar. Every second slice is shown. Compare to Figure 6-2, note that the intensity scaling has changed.

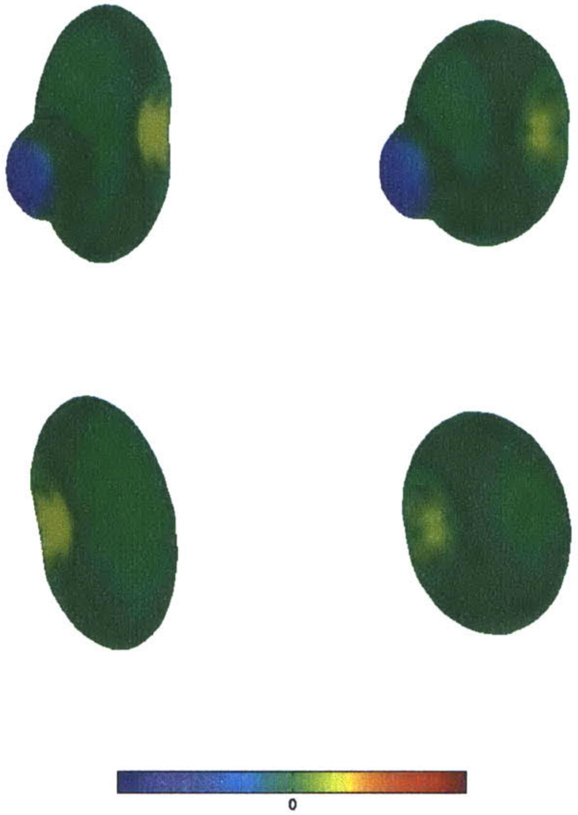


Figure 6-9: Deformation of the two support vectors from the first class. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards). Compare to Figure 6-3.

Figure 6-9 and Figure 6-10 show the deformation of the 5 support vectors that define the resulting classifier. These shapes were also identified as support vectors by the previous examples, while one of the support vectors in the first group did not appear as such in this experiment[2]. We can see that the technique identified both areas of differences and assigned the appropriate deformations to those areas. Furthermore, it could incorporate both deformations into a single resulting discriminative direction.

[2] The value of the classifier for this vector was 1.14, as compared to the range $[-1.7; 1.7]$ observed for the training set. Thus, it is still very close to the margin corridor.

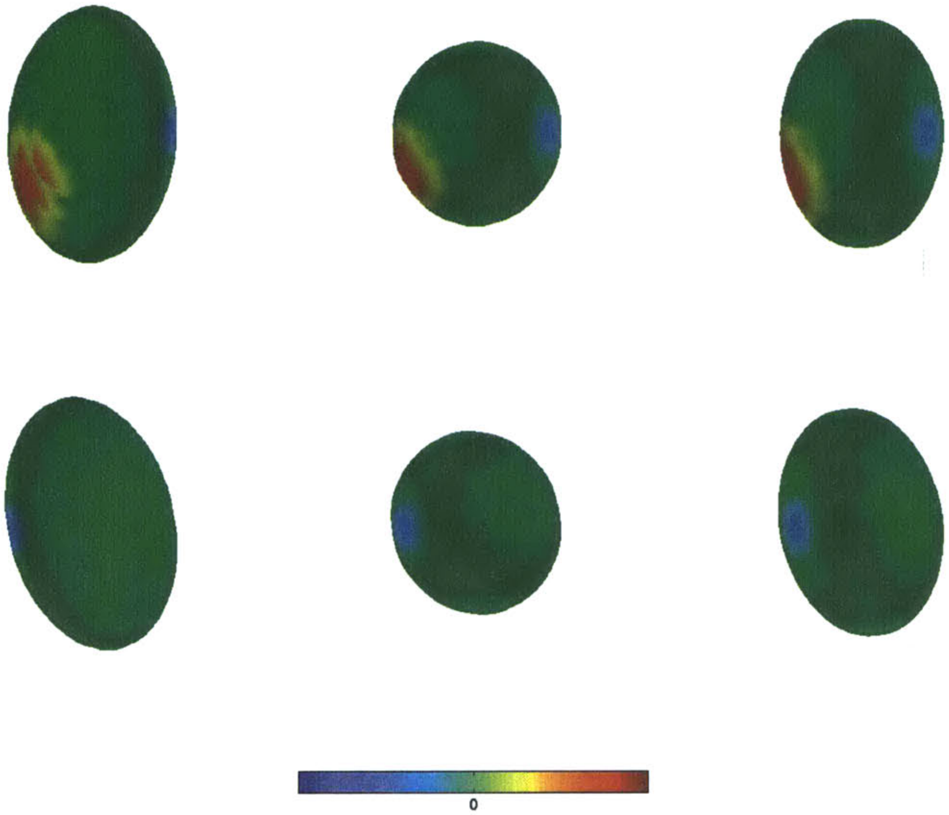


Figure 6-10: Deformation of the three support vectors from the second class. Two views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards). Compare to Figure 6-4.

6.3 Scaling

Volume and area measurements are extensively used in statistical studies of anatomical organs. In morphological studies, volume differences might be indicative of shape differences, and are therefore useful for fast pre-screening. However, using volume measurements by themselves has been questioned repeatedly in the medical research community because of wide variation in size among subjects. To alleviate this problem, volume-based statistical studies often use normalized volume measurements with the normalization factor chosen to reflect the global scale, e.g., the size of the brain can be used to normalize the hippocampus volume.

Volume		
Structure	Right hippocampus	Left hippocampus
Training accuracy (%)	60.0	63.3
Cross-validation accuracy (%)	60.0 ± 17.5	63.3 ± 17.2
Shape, Linear classification		
Structure	Right hippocampus	Left hippocampus
Training accuracy (%)	100	100
Cross-validation accuracy (%)	53.3 ± 17.8	56.7 ± 17.6
VC dimension	277	405
VC bound	1.79	1.86
Shape, RBF classification		
Structure	Right hippocampus	Left hippocampus
Training accuracy (%)	100	100
Cross-validation accuracy (%)	76.7 ± 15.1	70.0 ± 16.3
VC dimension	28	29
VC bound	1.34	1.35

Table 6.1: Performance estimates for the hippocampus study. The cross-validation confidence intervals and the VC bound were computed for 95% confidence level ($\eta = 0.05$). The training accuracy is reported for the parameter setting that yielded the best cross-validation results.

Whether the scale is part of the object's shape is a controversy of its own. One could argue that scaling the object uniformly does not change its shape. In this work, we scale the shapes to the same volume. Such scaling can be easily incorporated into the feature extraction step. In the following sections, unless specified otherwise, the training shapes are normalized with respect to their volume. Later in this chapter, we discuss possible ways to combine volume measurements with shape information to improve separation between the classes.

6.4 Hippocampus in Schizophrenia

In this section, we report the results of the method applied to a data set that contains MRI scans of 15 schizophrenia patients and 15 matched controls. In each scan, the hippocampus-amygdala complex was manually segmented. The examples of scans and segmentations from this study were shown in the previous chapters (see Figure 1-1, Figure 1-2 and Figure 3-2). Details on the subject selection and data acquisition can be found in [59]. The same paper reports statistically significant reduction in the relative volume of the left hippocampus (the volume of the structure was normalized by the total volume of intracranial cavity). This indicated that shape differences might also be present in this study.

In order to present and compare the results of different experiments in a uniform fashion, we first trained a classifier based on the volume measurements of the structure. The statistical significance test can be used only if the feature space is one-dimensional and is therefore not applicable to the case of multi-dimensional shape descriptors. Treating the one-dimensional volume descriptor similarly to the shape descriptors allows us to compare them directly. The probabilistic bounds, such as cross-validation accuracy and VC bound, are estimated for 95% confidence level ($\eta = 0.05$) for all the experiments in this work.

Table 6.1 contains the summary of performance estimates for this study. Note that statistical significance does not necessarily mean perfect separation: the volume-based leave-one-out cross-validation accuracy for the left hippocampus is 63.3% (±17.2%). By visually inspecting the shapes in Figure 1-2, we conclude that there are no distinct differences that would guarantee 100% classification accuracy. This is common in the medical studies, where the global anatomical shape is similar in both groups, and the small deformations due to a particular disorder, if such exist, are of interest.

Unlike the shape-based experiments, the cross-validation accuracy for volume-based descriptors is close to the training accuracy. Removing a single training example can only affect the training result if the example is close to the threshold, i.e., it's a support vector, and there could be only few such vectors in the low-dimensional space.

We conclude that the optimal non-linear classifier is likely to significantly outperform the linear classifier on this data, as indicated by the cross-validation results and the VC dimension estimates in Table 6.1. In fact, the cross-validation accuracy for the linear case is very close to the 50% baseline. In the remainder of this section, we present the discriminative direction analysis for the optimal RBF classifier. We present the results for the right and the left hippocampus next, followed by the

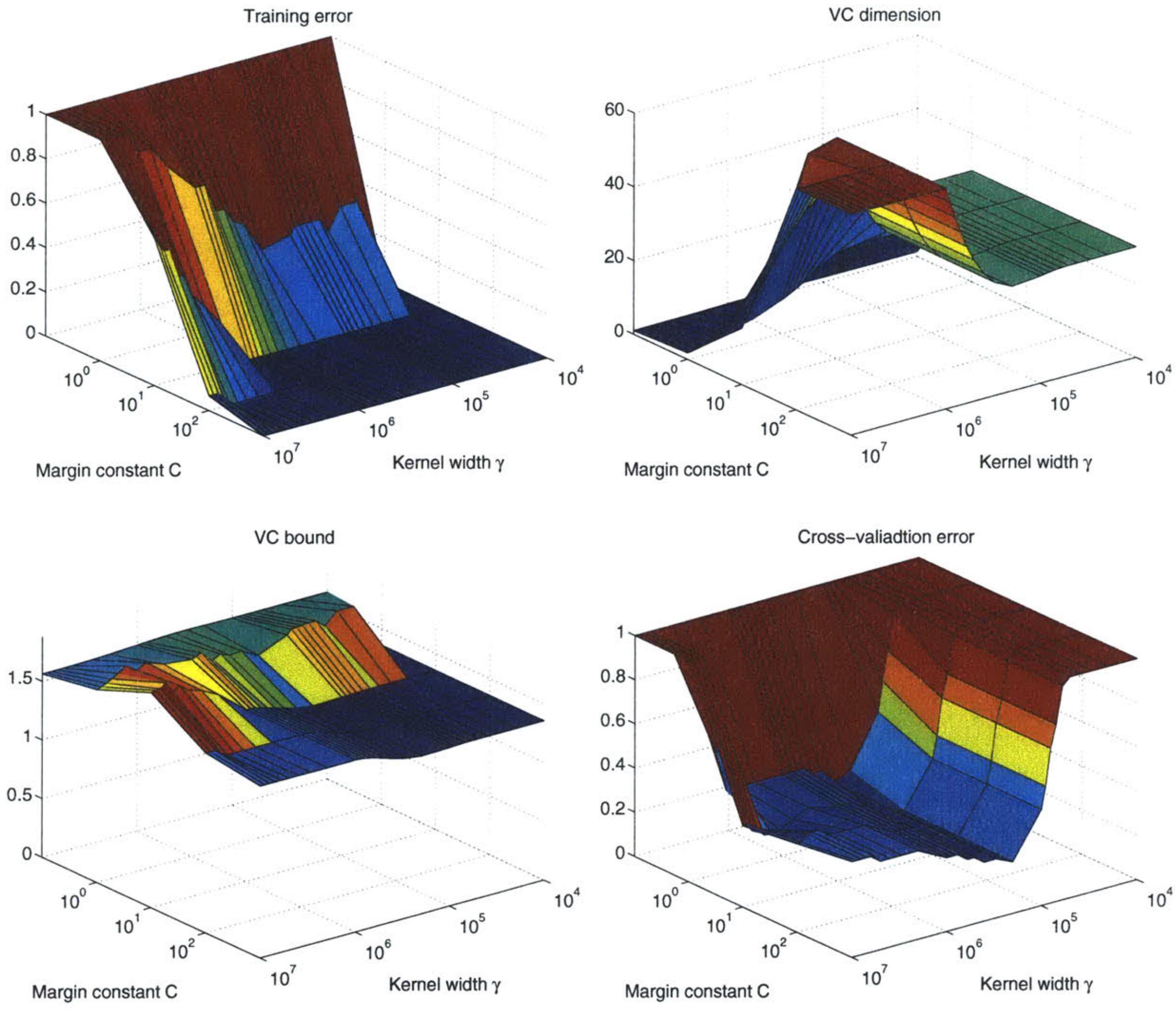


Figure 6-11: Training results for the right hippocampus. The graphs demonstrate the training error R_{emp}, the upper bound on the VC dimension (4.35), the VC bound (4.32) and the cross-validation error $\hat{R}$.

discussion on both structures, on the significance of the findings and the issues this experiment highlighted for the future extensions of the current analysis framework.

6.4.1 Right Hippocampus

Figure 6-11 shows detailed results of training and generalization performance estimation for the right hippocampus. Although the VC bound does not provide a useful estimate of the expected error, its spatial structure agrees with the cross-validation results in identifying the optimal setting of the parameters γ and C. Note that the performance estimates are very similar for a wide range of values of C $(10 \div 10^3)$ for the same value of γ. It has been commonly observed that the SVMs algorithm is fairly insensitive to changes in the parameter C, producing the same classifier for values of C several orders of magnitude apart [11].

We demonstrate the discriminative direction as deformations of the support vectors. Figure 6-12 and Figure 6-13 show three support vectors from the normal control group and the schizophrenia group respectively. Four views (front, center-out, back, outside-in) are shown for each shape. These shapes were chosen from the list of the support vectors sorted in the descending order of the magnitude of the classifier gradient. Similarly to the artificial example in the previous section, the algorithm produces several support vectors for the same type of deformation. We omit support vectors with very similar deformations to the ones shown in the figures. As a result, the shapes displayed in Figure 6-12 are support vectors 1, 3, and 6 from the normal control group, and the shapes in Figure 6-13 are support vectors 1, 3 and 5 from the schizophrenia group.

We can see that the deformations identified by the analysis are smooth and localized. Furthermore, the protrusions are separated from indentations by areas where no deformation is required[3].

We also note that the support vectors from different classes define deformations of very similar nature, but of opposite signs. We believe that such pairs of support vectors "oppose each other" across the separating boundary, but a more precise definition and analysis of this notion has to be developed before we can characterize it quantitatively.

A significant amount of deformation is localized in the anterior region of the structure, which indicates that the bulbous "head" of the amygdala is curved-in, or tucked-in, relative to the main body in normal controls more than in schizophrenia patients. This deformation is prominent in the first two support vectors from each

[3] A series of small changes of opposite sign close to each other would raise a concern that the structure captured by the classifier does not correspond to the sooth way the anatomical shapes deform and is induced by noise and errors in the boundary.

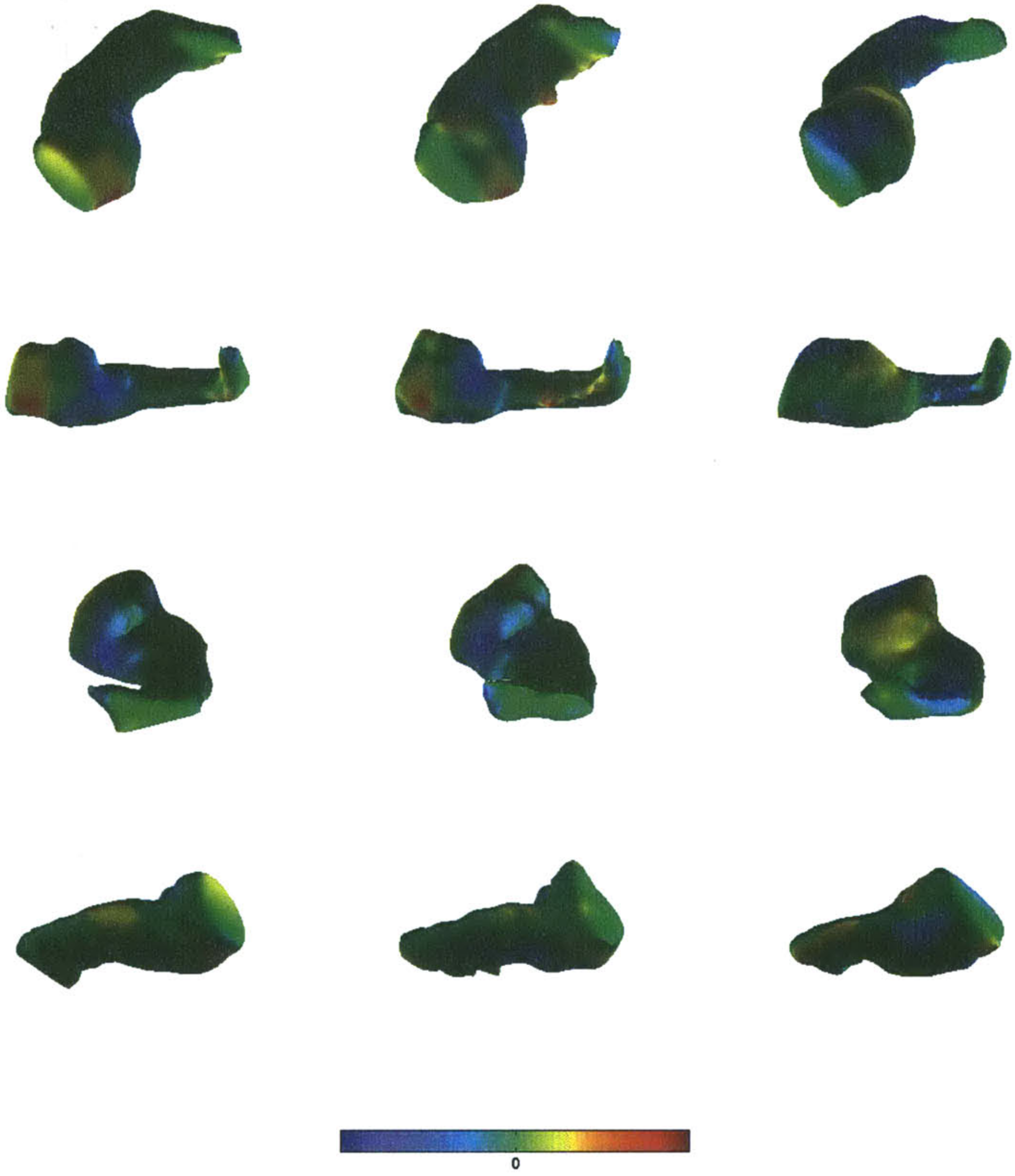


Figure 6-12: Discriminative direction for the right hippocampus shown as deformations of three support vectors from the normal control group. Four views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

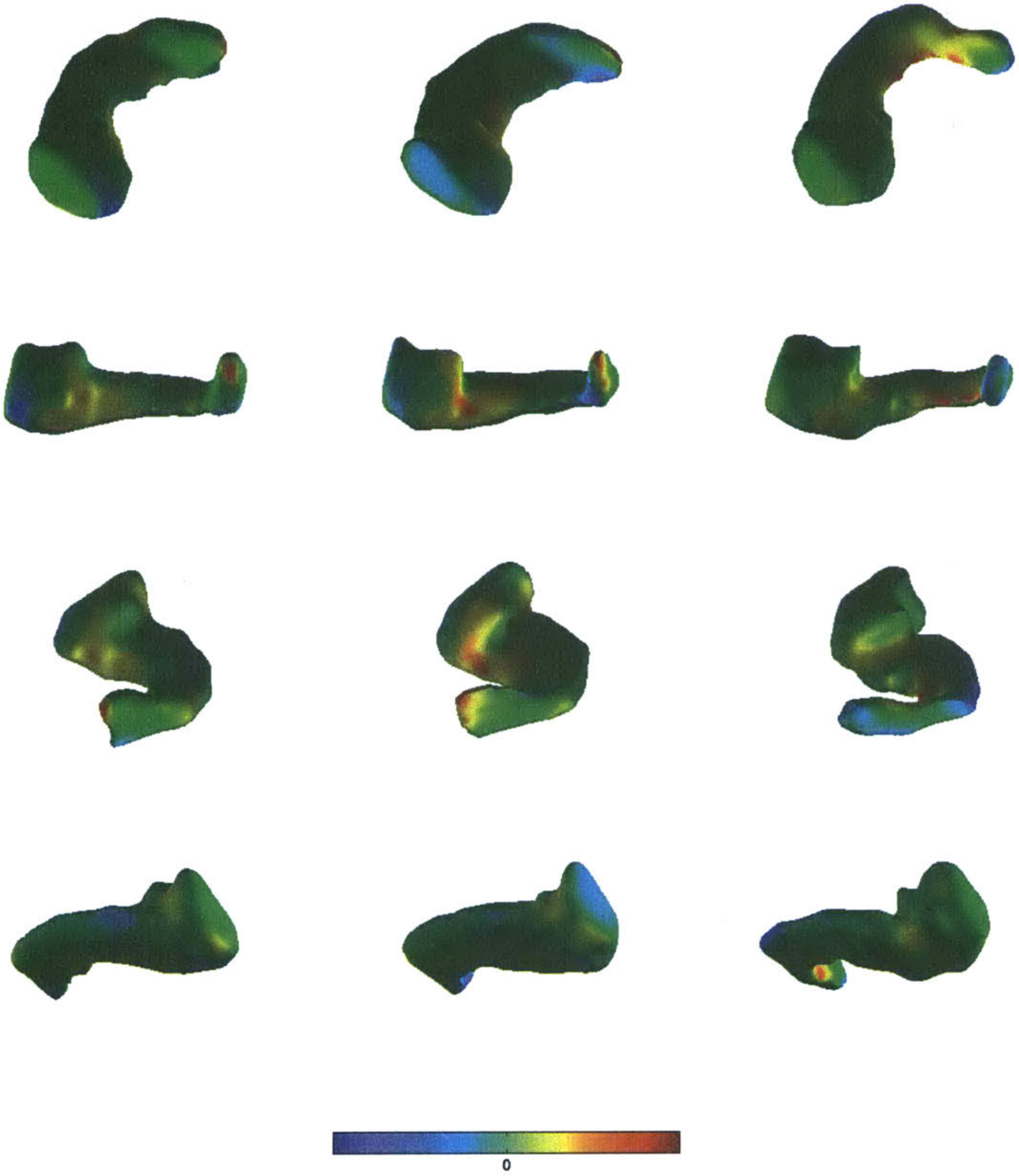


Figure 6-13: Discriminative direction for the right hippocampus shown as deformations of three support vectors from the schizophrenia group. Four views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

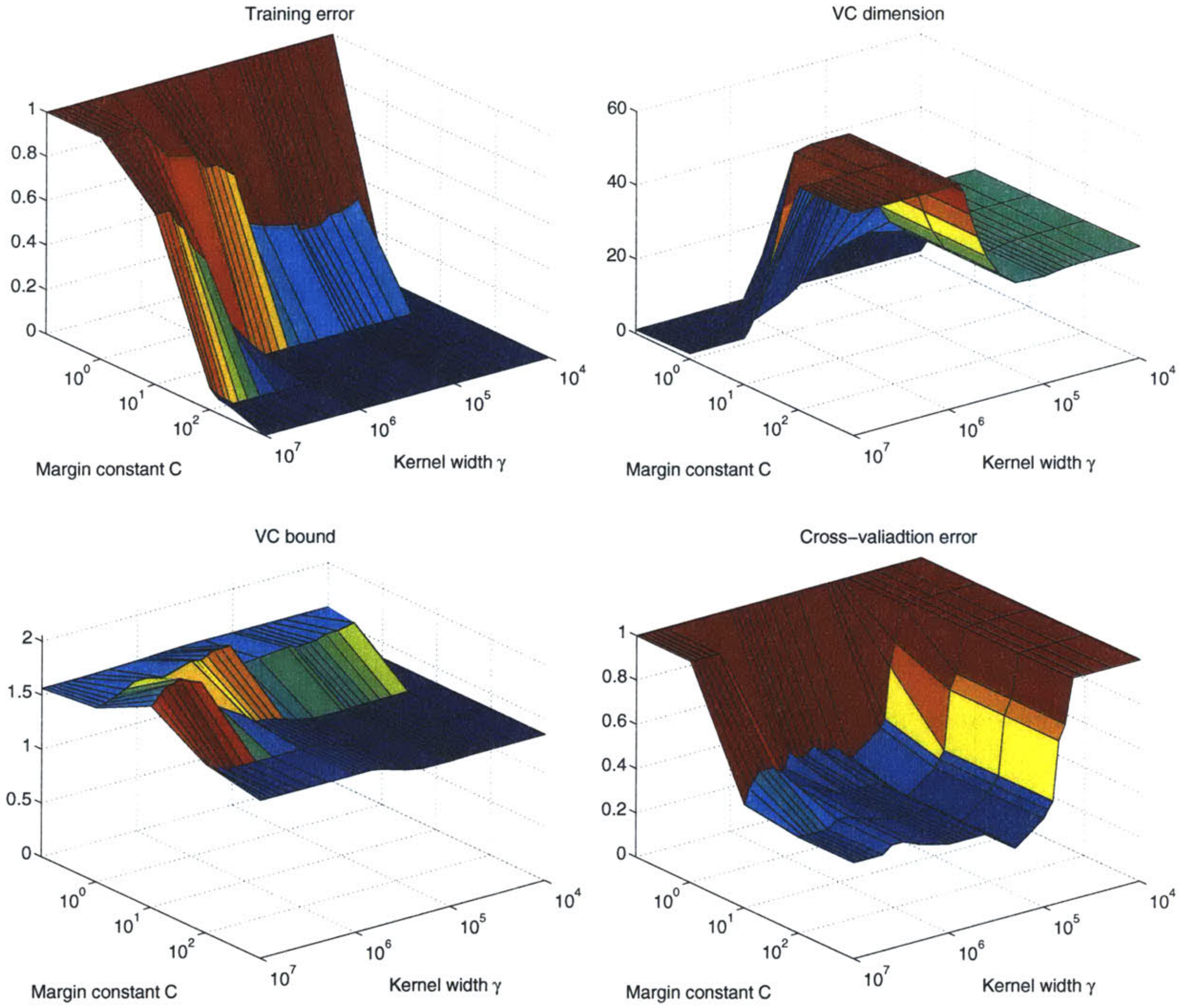


Figure 6-14: Training results for the left hippocampus. The graphs demonstrate the training error R_{emp}, the upper bound on the VC dimension (4.35), the VC bound (4.32) and the cross-validation error $\hat{R}$.

group. In normal controls, there is a significant deformation inward in the inner part of the amygdala and a corresponding expansion on the outside part of it. The second interesting shape difference is located in the posterior part of the hippocampus and is represented by the third support vector in the figures. It seems that the "tail" is thinner and possibly shorter in schizophrenics in the region of the shape space close to this support vector.

6.4.2 Left Hippocampus

While the volume-based differences are more prominent in the left hippocampus in this study, the shape-based performance estimates for the left hippocampus are lower than those for its right counterpart (Table 6.1). But since we scaled the structures to an identical volume, thus separating shape from size, we would not expect the volume-based results and the shape-based finding to be perfectly correlated.

Figure 6-14 illustrates the training results for the left hippocampus in detail. Similarly to the right hippocampus, the VC bound and the cross-validation estimates agree on the optimal range of parameters γ and C.

Figure 6-15 and Figure 6-16 show the discriminative direction as a deformation of the top support vectors from the normal control group and the schizophrenia group respectively. The first two support vectors in each group indicate that the posterior "tail" of the structure is folded-in, or curved, in normal controls more than in schizophrenics. In addition, the last three support vectors contain a deformation in the anterior part of the structure. The support vectors in the normal control group contain a slight deformation inward and a protrusion of a higher magnitude in the anterior part. This deformation is of a similar nature for the three support vectors, but it is localized in different parts of the bulbous head. Besides the obvious explanation that the location of this deformation is not fixed in the population, this could also correspond to a general enlargement of the anterior part relative to the whole structure in schizophrenics. Slight misalignments of the structures in the feature extraction step can cause such size differences to be detected in different areas of the affected surface. Since statistically significant volume reduction was detected in the left hippocampus, this could mean that the posterior part of the structure is affected by the volume reduction in a more significant way that the anterior part.

6.4.3 Discussion

The two previous sections demonstrate our technique on the real medical data. We obtained a detailed description of the shape differences between the schizophrenia

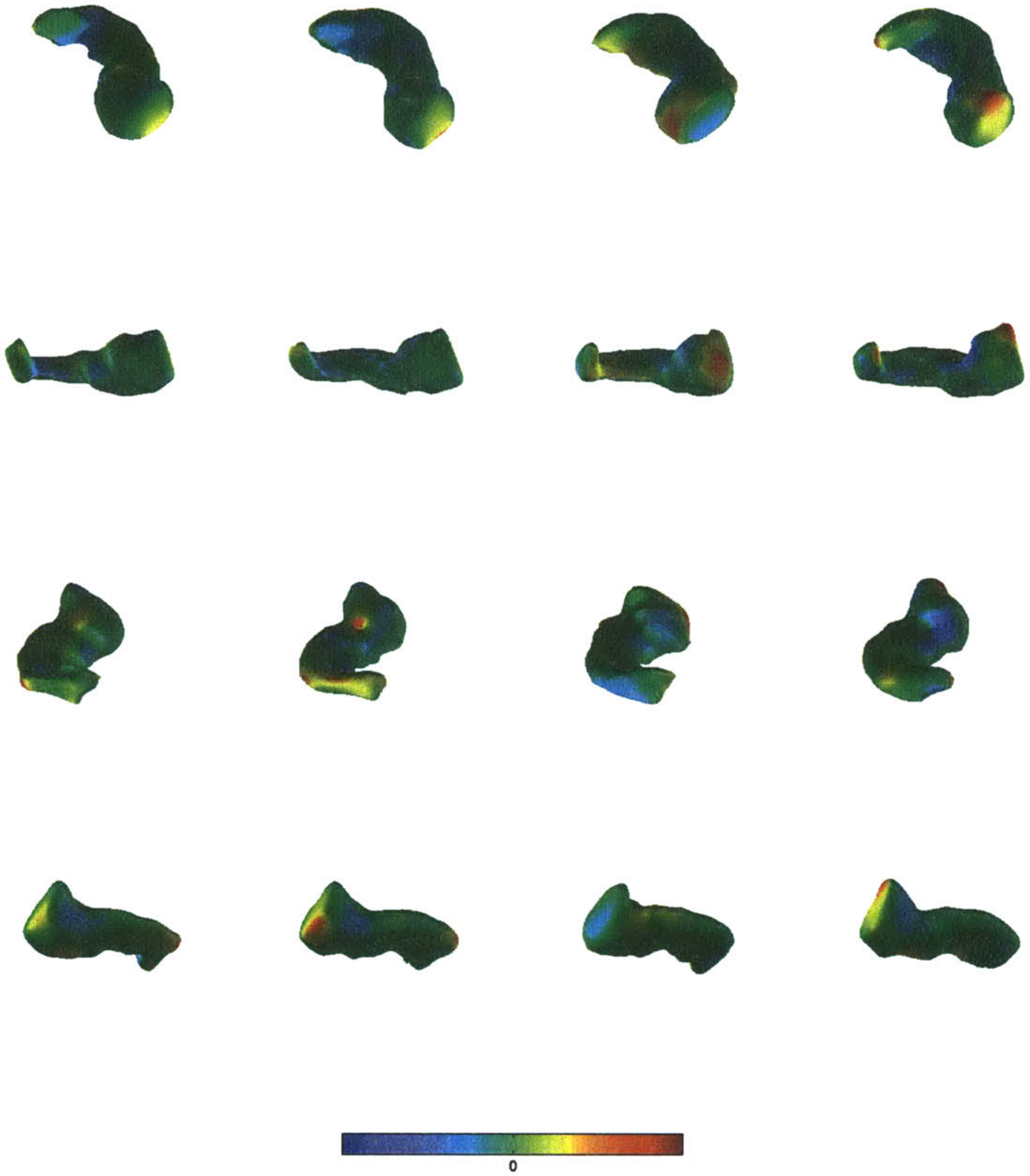


Figure 6-15: Discriminative direction for the left hippocampus shown as deformations of four support vectors from the normal control group. Four views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

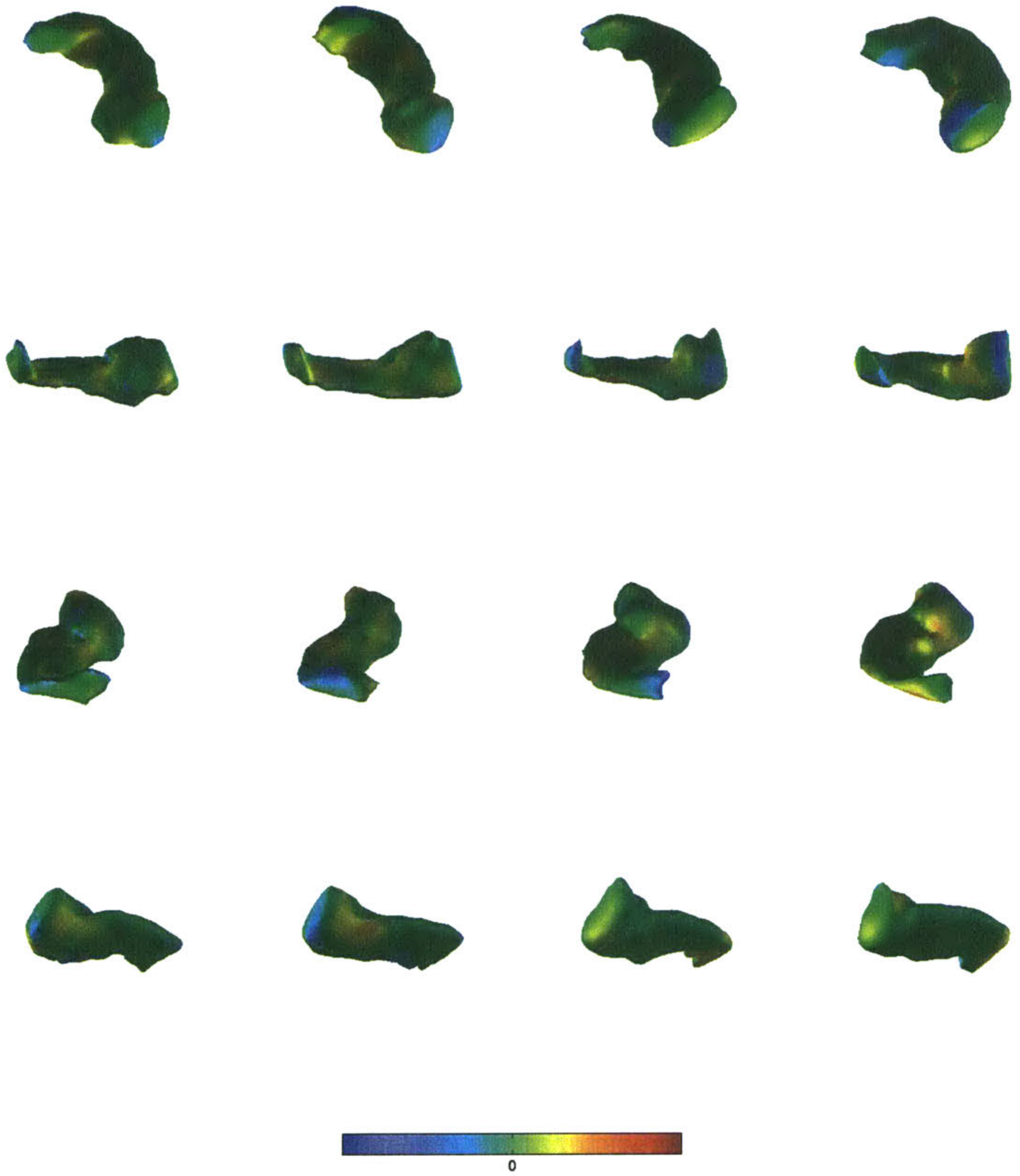


Figure 6-16: Discriminative direction for the left hippocampus shown as deformations of four support vectors from the schizophrenia group. Four views of each shape are shown. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

group and the normal control group expressed as deformations of the example shapes in the two groups. While the medical significance of these findings has to be further investigated by the neuroscientists, such visualizations can clearly facilitate their exploration of the shape differences in organs affected by the disease.

Note that the shape differences found in the two hippocampi are asymmetric. We considered the two structures independently of each other, but studying the relationship between the two sides of the brain could help us understand the effects of a disorder better. An even higher cross-validation accuracy (87%) was recently reported on the same data set in [25] based on the average distance between the aligned surfaces of the left hippocampus and a mirror image of the right hippocampus as an asymmetry measure. Unfortunately, such global asymmetry measures are too specific to be generally applicable, and furthermore, they do not provide information on the details of the shape differences. In this work, we concentrated on a single structure at a time and provided the analysis and the detailed interpretation of shape differences based on general descriptors. The separation between the classes could be improved by combining information from different structures (left and right) and different descriptors (volume and shape). We will come back to this question in the next chapter when we discuss possible future extensions of this work.

The results reported in this section indicate that the training data set is too small to provide accurate estimates of the generalization performance of the classifier. The gap between the training and the cross-validation errors, as well as wide confidence intervals, indicates that we are far from the asymptotic region where either of the estimated errors is a reliable predictor of the generalization performance. More data is needed to guarantee that the resulting classifier and the detected shape variation reflect the true differences in the population. Unfortunately, the ground truth, i.e., the true differences between the classes, or even whether such differences exist, is not known for the medical studies we are working with. Anatomical shape analysis is a relatively new field, and not much is known about the deformations caused by the disorders of interest. We therefore believe that developing principled algorithms for investigating morphology of the organs, along with thorough data collection and

Linear classification	
Training accuracy (%)	100
Cross-validation accuracy (%)	65.8 ± 15.0
VC dimension	1766.35
VC bound	1.45
RBF classification	
Training accuracy (%)	100
Cross-validation accuracy (%)	73.7 ± 13.9
VC dimension	39.24
VC bound	1.35

Table 6.2: Performance estimates for the corpus callosum in the affective disorder study. The cross-validation confidence intervals and the VC bound were computed for 95% confidence level ($\eta = 0.05$). The training accuracy is reported for the parameter setting that yielded the best cross-validation results.

analysis, can have a significant impact on the field.

In the next two sections, we demonstrate our method on two studies of corpus callosum for which our technique identified shape differences between the group of patients and the normal controls. Our findings in both studies indicate differences between the groups, with a similar concern about the size of the training data set as for the hippocampus study.

6.5 Corpus Callosum in Affective Disorder

Corpus callosum is a bundle of white matter fibers connecting the two hemispheres of the brain. The two-dimensional cross-section of the bundle is actually what studied in the medical research. Example images of corpus callosum and its segmentation were shown in Chapter 3. To ensure consistency, all the scans in the study have been aligned manually by the trained physicians so that the cross-section is indeed perpendicular to the bundle. Further details on the data collection and scan alignment can be found in [24].

In this study, we compared 18 affective disorder patients with 20 normal controls. Table 6.2 summarizes the performance estimates for the linear and the Gaussian RBF classification on this data set. The gap between performance estimates for the linear

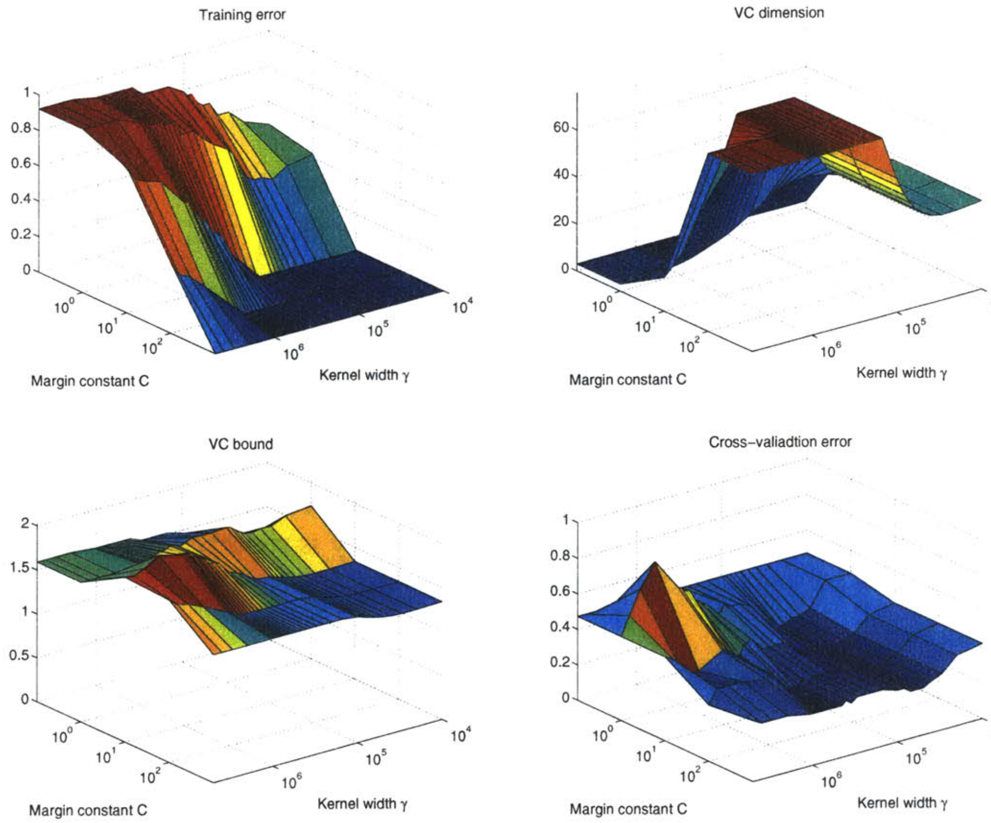
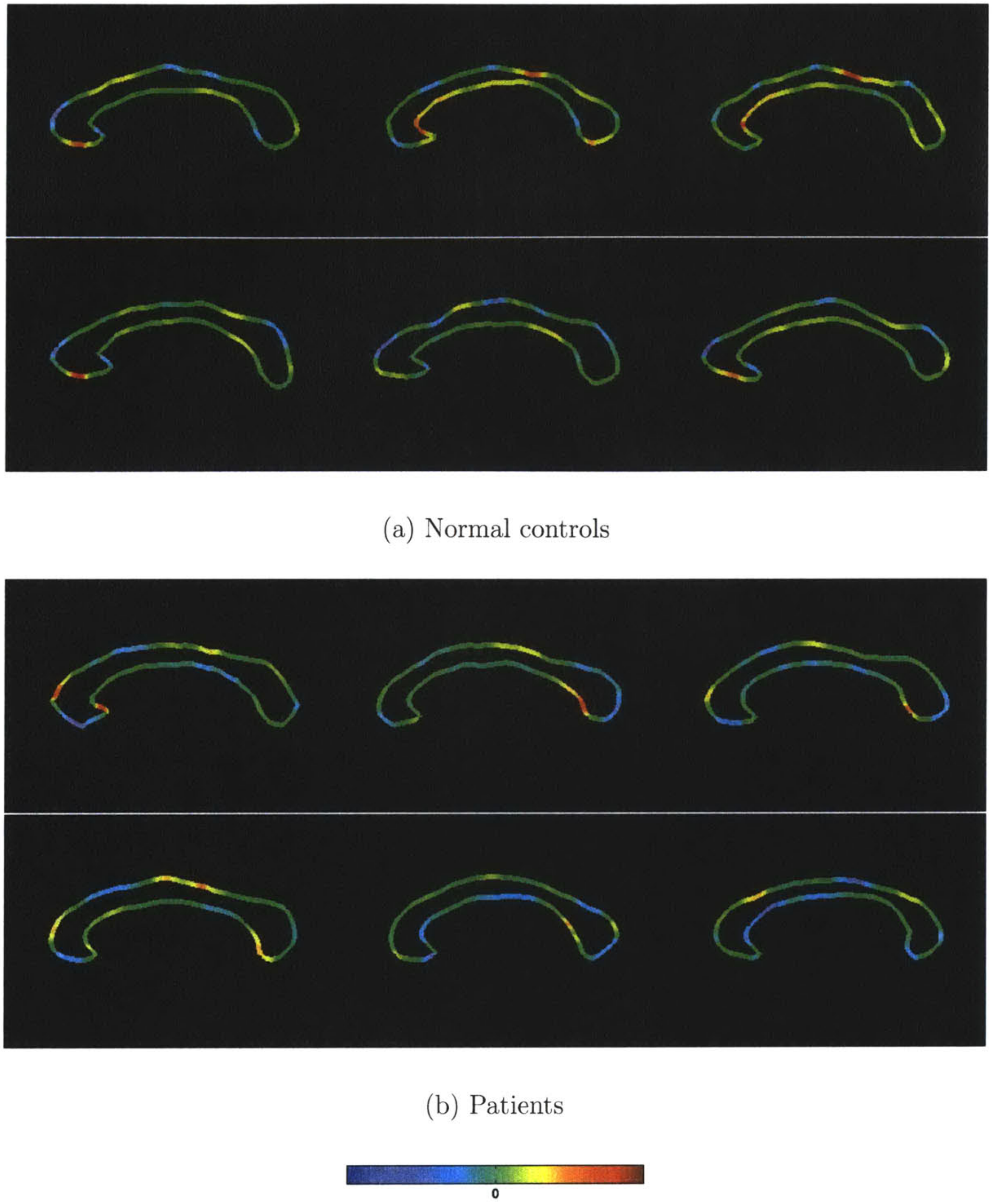


Figure 6-17: Training results for corpus callosum in the affective disorder study. The graphs demonstrate the training error R_{emp}, the upper bound on the VC dimension (4.35), the VC bound (4.32) and the cross-validation error $\hat{R}$.

and the non-linear classification is not as wide as in the hippocampus study, but it is still substantial. The 95% confidence interval for the linear classification touches the 50% baseline, while the confidence interval for the best RBF classifier is well above it. Figure 6-17 shows the details of the training results for the non-linear classification. The cross-validation error graph is noisier for this case than the hippocampus study, achieving its minimum at several places in the valley. We use the VC bound to choose among the parameter settings that correspond to the minimum of the cross-validation error.

Figure 6-18 shows the detected shape differences as deformations of the first 6 support vectors from each group. Similarly to the hippocampus study, there is a lot of redundancy in deformation represented by the support vectors. The most prominent difference captured by the classifier is the deformation in the anterior part of the



(a) Normal controls

(b) Patients

Figure 6-18: Discriminative direction for corpus callosum in the affective disorder study shown as deformations of 6 support vectors from each group. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

structure (the left end of corpus callosum in the images). We observe a significant amount of horizontal expansion and vertical contraction in the anterior part in the patient group and the deformation of the opposite sign in the normal control group. The amount of deformation varies across the shapes, but it is present in almost all of them. This deformation corresponds to the anterior of corpus callosum being "squashed" horizontally and elongated vertically in the affective disorder patients compared to the normal control group.

Vectors #2, #3 in the normal control group and vectors #5, #6 in the patient group indicate that for some of the cases, the width of the middle part of the corpus callosum is wider in the affective disorder patient: there is a consistent contraction associated with the patient examples and expansion associated with the normal controls.

Vectors #2 and #3 in the patient group indicate some amount of deformation in the posterior part of the structure. Interestingly, this deformation is not represented at all in the normal control group. We noted earlier that many support vectors have a matching counterpart from the other class that represents a deformation of opposite sign, but of very similar nature. This is an example when there seems to be a "gap" in the boundary support on the normal control side. This is an interesting phenomenon that needs further investigation. Potentially, one might be able to construct "virtual" support vectors by artificially reflecting the existing support vectors across the boundary. This is closely related to one of the open questions we mentioned in the previous chapter, namely, a global search for a trajectory in the feature space for deforming an example shape towards and across the separating boundary.

To summarize, there is a consistent deformation of the anterior part of the structure that corresponds to horizontal narrowing and vertical extension of the anterior part of the corpus callosum in the affective disorder patients. In addition to that, the middle part is widened in some of the patients compared to normal controls. Similarly to the hippocampus study, more data will have to be collected for validation of these results.

Linear classification	
Training accuracy (%)	100
Cross-validation accuracy (%)	63.8 ± 15.6
VC dimension	1540.33
VC bound	1.45
RBF classification	
Training accuracy (%)	100
Cross-validation accuracy (%)	69.4 ± 15.0
VC dimension	72.13
VC bound	1.45

Table 6.3: Performance estimates for the corpus callosum in the schizophrenia study. The cross-validation confidence intervals and the VC bound were computed for 95% confidence level ($\eta = 0.05$). The training accuracy is reported for the parameter setting that yielded the best cross-validation results.

6.6 Corpus Callosum in Schizophrenia

Similarly to the hippocampus, corpus callosum is hypothesized to be affected by schizophrenia. In this study, we compared 16 schizophrenia patients and 20 normal controls. Table 6.3 and Figure 6-19 summarize the training results for this experiment. The performance estimates for this study are lower than those for the affective disorder study. Furthermore, the cross-validation estimates for different settings of the Gaussian kernel parameters do not form a smooth surface with a distinctive minimum anymore. The cross-validation estimates also disagree with the VC bound on the optimal settings of the parameters. Based on these indicators, we conclude that the statistical results do not predict the performance of the resulting classifier reliably. Therefore, without more data, we cannot conclusively state that the differences in the training set captured by the classifier will also be found in a larger population.

However, we can still compute the deformation that represents the differences in the training data set. Although we cannot generalize the results for the whole population, they can help us understand certain properties of the technique. Interestingly, we discover that all support vectors represent a very similar deformation in this experiment (Figure 6-20). The middle "bridge" of the corpus callosum seems to be more bent in the patient group than in the normal control group: the deformation

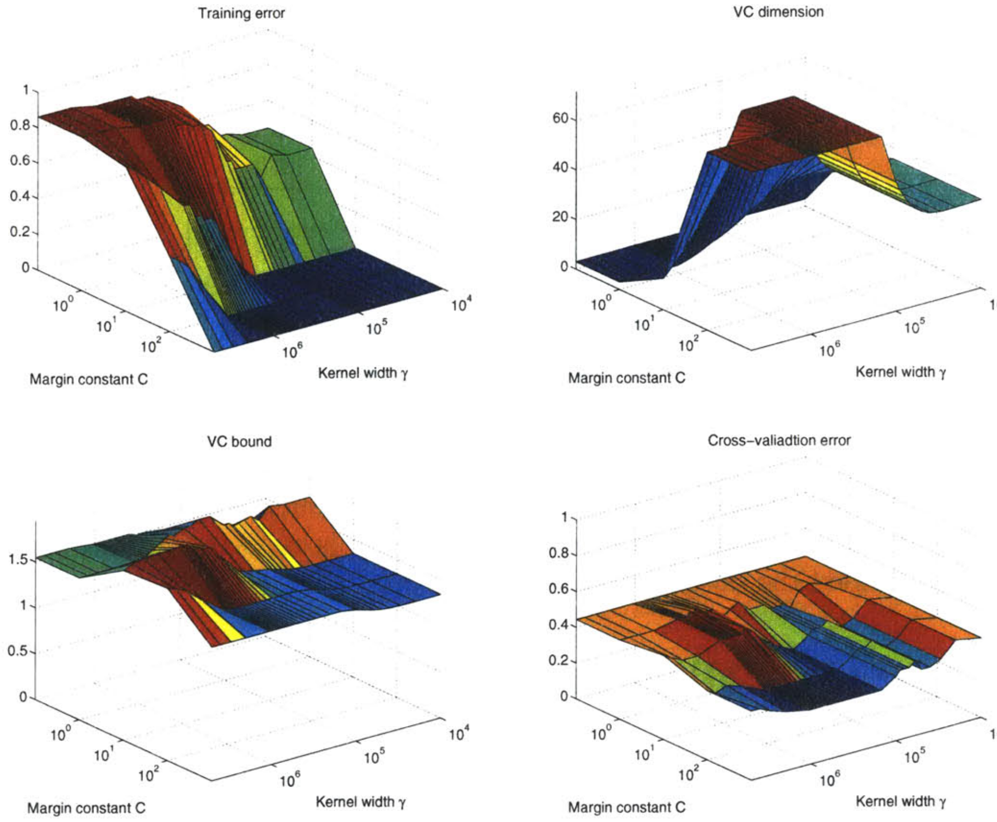
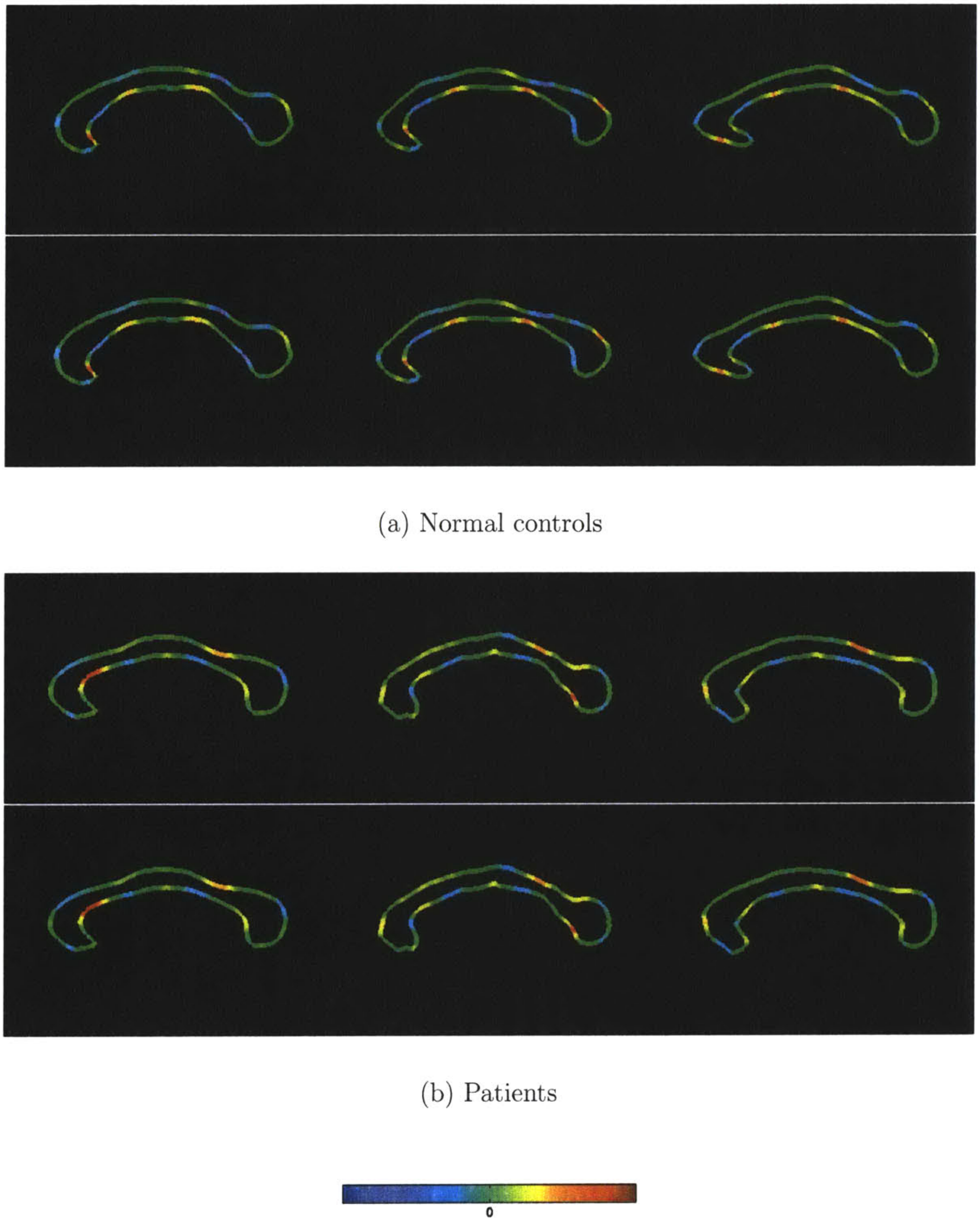


Figure 6-19: Training results for corpus callosum in the schizophrenia study. The graphs demonstrate the training error R_{emp}, the upper bound on the VC dimension (4.35), the VC bound (4.32) and the cross-validation error $\hat{R}$.

identified by the classifier is almost symmetric around the middle of the structure and corresponds to the normal control examples "bending" around the middle and the patient examples "unbending" in a very similar fashion.

We found it fascinating that the deformation was so similar for all support vectors. It suggests that the separating boundary between the classes, or more precisely, the discriminative direction, varies very slowly as we move from one support vector to another in the feature space. This is similar to the linear case, where the discriminative direction does not change spatially at all. We can confirm this fact by examining the width of the Gaussian kernel used by the resulting classifier. The range of the kernel width values for the RBF classifiers that achieve the highest cross-validation accuracy is between $2{\cdot}10^5$ and 10^6. Figure 6-20 show the deformations computed for the classifier with the kernel width $\gamma = 6{\cdot}10^5$. Comparing this with the distances



(a) Normal controls

(b) Patients

Figure 6-20: Discriminative direction for corpus callosum in the schizophrenia study shown as deformations of 6 support vectors from each group. The color coding is used to indicate the direction and the magnitude of the deformation, changing from blue (inwards) to green (no deformation) to red (outwards).

between the training examples in the feature space (the median squared distance is $2.5{\cdot}10^5$, the mean is $3.3{\cdot}10^5$, the maximal squared distance is $1.8{\cdot}10^6$), we conclude that the Gaussian kernels placed at the support vectors will combine into a smooth, slowly varying surface. For comparison, the width of the kernel for the best classifier in the affective disorder case was $5{\cdot}10^4$, resulting in much more variation in the deformation associated with different support vectors.

To conclude, we have identified shape differences in this training set, but the statistical estimators cannot predict reliably the generalization of the findings for the population. Collecting additional data for analysis will allow to demonstrate whether the identified differences are specific to this data set or they can be found in the general population with high probability.

6.7 Lessons Learned

In this section, we reflect on our experience with the technique, unexpected problems that arose in the experiments and the insights they provided into the nature of the statistical shape analysis.

Successful studies. As demonstrated in the previous sections, the visualized shape differences contain detailed morphological information which can be correlated with functional information on the organ of interest, hopefully leading to a better understanding of the disease and its development. The representation of morphological differences as deformations of the original input shapes is significantly more detailed, localized and informative than the volume and the area measurements traditionally used in the medical research. In addition to establishing the fact of statistical differences between the two populations, we can employ the analysis techniques presented in this work to start explaining the source of such differences.

The experiments highlighted the importance of statistical testing of the resulting hypothesis. While the training algorithm will produce a classifier which can be analyzed for discriminative direction for any two sets of examples, the shape differences found in the training set are useful for understanding the phenomenon in question only if they accurately reflect the morphological differences in the entire population. Therefore, estimating the expected performance of the resulting classifier and the statistical significance of the morphological variability it represents is a crucial component of the analysis. It effectively estimates to what extent we should trust the training data set to represent the population.

Failed studies. While our method identified statistically significant differences between the groups in the experiments reported in the previous sections, it failed to do so in several other studies. However, the experiments that failed to produce statistically significant differences between the two classes can still help us to better understand the problem and potentially improve the technique. In such experiments, different statistical indicators, such as the VC-bound and the cross-validation accuracy, disagree significantly on the optimal settings of the training parameters, often predicting close to 50% baseline classification accuracy on new examples. Furthermore, when the predicted performance is higher than the baseline, the confidence intervals might be too wide to allow us to make any claims on how well the differences detected in the training set represent the situation in the whole population. We mentioned these concerns when discussing the study of corpus callosum in schizophrenia in Section 6.6.

We observed all of the warning signs listed above in a study of gender-related differences in the corpus callosum. Differences in the brain morphology between males and females have been a topic of medical research for many years. In the corpus callosum, the evidence is conflicting: several studies report size and shape differences [19, 43], while others claim to have not found statistically significant variability when the data were normalized for social status, education level and other factors that might affect the size of the brain [4]. To avoid the controversy associated with gender differences in the corpus callosum, we used only male subjects in the studies of affective disorder and schizophrenia reported in the previous sections. Separately from these studies, we applied the algorithm to a set of corpus callosum images of 18 male and 13 female subjects provided by Dr. G. Gerig, University of North Carolina[4]. In this particular experiment, the cross-validation accuracy at the 95% confidence level was 64 ± 17%, which includes the 50% baseline. While the discriminative direction could be extracted from the resulting classifier, the statistical indicators imply that we cannot trust the results of the analysis to represent the differences in the population.

Such failed experiments bring up an interesting general question of when one should stop the search for shape differences. Any morphological study starts with a hypothesis of shape differences which is to be confirmed by the empirical evidence from the collected images. We could get statistically unsatisfactory results described above either because our representation and analysis cannot capture the shape differences present in the population, or because such differences do not exist. In the former case, we can improve the technique for shape representation and statistical analysis to include more complex models of morphology and its variability and collect more data to reduce the confidence intervals. But the fundamental question remains, when should we abandon the search for better analysis techniques and more training data and declare that there are no differences between the two populations? This problem is common in many fields of research, as the current theoretical framework provides us only with tools for establishing the fact of existence of a particular phenomenon. It is nearly impossible, at least with our current system of reasoning, to prove the absence of the hypothesized effect.

Combining data sets. Obtaining segmented images for morphological studies is a time- and effort-consuming process. Thus, one might consider combining data sets collected by different research groups to combat the problems of insufficient training data. Following this idea, we attempted to combine data from two similar studies of corpus callosum in schizophrenia. The first data set was created at Brigham and

[4] These images are a part of a corpus callosum study in schizophrenia patients. Segmentations were done by Rebecca Crow, University of Oxford in collaboration with Guido Gerig, UNC Chapel Hill, Tim Crow, University of Oxford, and the Image Analysis Group, ETH Zuerich. We will come back to this study later in the section.

Women's Hospital and is described in Section 6.6. Dr. G. Gerig provided us with the second data set which was mentioned earlier in this section in connection to the gender-related study of corpus callosum. We used male subjects from both studies to create a new, combined collection which included 35 schizophrenia patients and 38 normal controls. Upon comparing the pairwise distances between the feature vectors in the new data set, we discovered that the distances between cases from two different imaging centers were substantially greater than the distances between cases processed by the same research group. It turns out that the differences in the segmentation methodology cause the shape of corpus callosum segmentations to vary among research groups. And while these differences are small, they are consistent enough that we could build a linear classifier for discriminating between the two groups of the male normal controls from the different imaging centers with the cross-validation accuracy of 92%±8%! In Chapter 2, we mentioned that image segmentation must be considered as an integral part of the feature extraction algorithm, and this example demonstrates how inconsistencies in segmentation can affect the analysis.

While creating pools of training data for statistical analysis is a promising idea, a common segmentation methodology in the community must be devised in order to avoid problems we observed in this study. Otherwise, the increased complexity of the concept to be learned will undermine the benefits of the larger data set as the algorithm is presented with not just the shape differences between the two classes, but also the inter-class variability introduced by discrepancies in segmentation methodology.

6.8 Summary

This chapter reports the results of statistical shape studies for both artificial and real medical examples. The technique identified the true differences between the two groups in the artificial data sets. The validation of the method is more difficult in the real medical studies, as the ground truth is unknown. Furthermore, since the field of statistical shape analysis is relatively young, no extensive previous results are available for comparison. To the best of our knowledge, this is the first work in which the statistical differences between the groups have been obtained from the discriminative model, i.e., a classifier function for labeling new examples. More work is needed to understand the medical significance of the results presented here, and as more results on similar data become available in the field, we will be able to compare the shape differences detected by different techniques. The experiments raised several interesting research questions to be explored in the future, such as collecting more data to improve statistical confidence indicators, importance of consistency in segmentation protocols across the data sets, importance of statistical testing and performance prediction, as well as fundamental questions of hypothesis rejection in application to shape analysis.

Chapter 7

Conclusions

In this thesis, we study the problem of image-based statistical analysis of morphological differences between populations. The analysis consists of three main steps: feature extraction, training a classifier and the interpretation of the results in terms of the shape differences detected by the model. These differences can then be visualized for inspection and future analysis by medical researches. We discuss available shape descriptors and statistical analysis tools and justify our choice of the distance transforms for shape representation and the Support Vector Machines algorithm for learning a classifier. The original contributions of this thesis involve the final step of the analysis, namely, the interpretation of the resulting classifier in terms of the shape changes that distinguish between the two example classes. We present a novel technique for classifier analysis in terms of the input features in the general context of the statistical learning theory. Furthermore, we instantiate the technique for shape analysis by establishing a locally linear parameterization of the distance transform space by the space of deformations of the corresponding boundary surface. Such parameterization yields a representation of the shape differences captured by the classifier as deformations of the input shapes relative to the examples from the opposite class.

We demonstrate the method on both artificial examples that illustrate the approach and the real medical studies in which the resulting deformations describe shape changes due to diseases and can be helpful in advancing the medical research towards explaining the mechanisms by which the organs are affected. To the best of our knowledge, this is one of the first few attempts to automatically detect and interpret shape differences between populations, and the first work that takes advantage of the discriminative modeling. Since the discriminative models require significantly fewer data than the generative models to reliably estimate the differences between the classes, a discriminative approach has better chances of succeeding in small training sets. Anatomical studies have always been challenging exactly because the images are difficult to collect and process and therefore the available training sets are typically very small. Thus, we believe that the proposed framework can allow the medical researches to efficiently utilize the available data in order to study various diseases.

7.1 Future Directions of Research

Experimental studies suggested several directions of future work which we discussed in the previous chapter, from refining the analysis technique for interpretation of shape differences to collecting more data for strengthening the statistical confidence indicators. In this section, we would like to mention two promising directions of research in statistical shape analysis that are enabled by the approach presented in this dissertation.

First, we note that the training algorithm and the discriminative direction analysis can be used as a very effective tool for investigating the power of different shape descriptors for representing morphological variability. In Chapter 2, we listed several theoretical requirements that a shape descriptor must satisfy in order to be useful for statistical analysis. Our approach to classification and further interpretation of the results allows us to compare the descriptors empirically based on their performance in shape-based statistical tests. More often than not, the shape analysis methodology is presented in the literature as a monolithic structure where the shape description and the statistical analysis are inseparable. In reality, these two components can be improved independently of each other. Furthermore, we believe that shape representation does and should depend on the organ of interest (we would not expect the same descriptor work equally well for the hippocampus and for the cortical folds), while the statistical analysis can be easily adapted to work with a large family of descriptors. It can therefore be used as a test-bed for various shape representations.

The other interesting observation is concerned with the medical implications of the experimental results produced by our method. Once we start working with non-linear classifiers, we accept a possibility of the differences between the normal subjects and the patients varying over the shape space. For example, we noticed that the support vectors in the reported studies represented several different classes of deformations and could be grouped according to the type of the corresponding deformation. Thus, the results of the analysis could be used for partitioning the region of the feature space occupied by the examples of pathology into sub-regions characterized by the nature of the deformation that separates the pathology from the normal cases. This partition, especially if supported by other evidence, such as symptoms, functional data, etc., can be used as an initial indication that the disease in question is really a collection of several different disorders. As a specific example, schizophrenia is believed by many researches to include mental conditions that the medical science cannot separate out reliably using the current research techniques. Additional morphological information provided by our analysis can assist in establishing the existence of such sub-disorders. Naturally, the statistical shape comparison can be performed between the newly defined sub-classes of patients. Such studies would require significantly more example images than are currently available, and we hope that this work will enable detailed anatomical shape analysis studies which in turn will lead to concentrated large-scale data acquisition efforts.

To conclude, we proposed and demonstrated a principled framework for statistical shape analysis based on the existing shape representation and statistical learning methods and the novel analysis techniques for interpretation of the statistical model as shape deformations between the two groups of interest. This approach provides detailed information on shape differences between populations and facilitates the studies of the disorders through understanding of the induced anatomical changes.

Bibliography

[1] Y.S. Abu-Mostafa. Learning From Hints. *Journal of Complexity*, 10(1):165-178, 1994.

[2] S. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251-276, 1998.

[3] S. Amari and S. Wu. Improving Support Vector Machines by Modifying Kernel Functions. *Neural Networks*, 783-789, 1999.

[4] K. M. Bishop and D. Wahlsten. Sex Differences in The Human Corpus Callosum: Myth or Reality? *Neuroscience and Biobehavioral Reviews*, 21:581-601, 1997.

[5] H. Blum. A Transformation for Extracting New Descriptors of Shape. In *Models of the perception of Speech and Visual Form*, MIT Press, Cambridge, MA, 1967.

[6] F. L. Bookstein. The Line Skeleton. *CGIP: Computer Graphics and Image Processing*, 11:123-137, 1979.

[7] F. L. Bookstein. Landmark Methods for Forms Without Landmarks: Morphometrics of Group Differences In Outline Shape. *Medical Image Analysis*, 1(3):225-243, 1997.

[8] G. Borgefors. Distance Transformations in Digital Images. *CVGIP: Image Understanding*, 34:344–371, 1986.

[9] J. W. Brandt and V. R. Algazi. Continuous Skeleton Computation by Voronoi Diagram. *In Computer Vision, Graphics, and Image Processing. Image Understanding*, 55(3):329-338, 1992.

[10] C. Brechbühler, G. Gerig and O. Kübler. Parameterization of Closed Surfaces for 3-D Shape Description. *CVGIP: Image Understanding*, 61:154-170, 1995.

[11] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121-167, 1998.

[12] C. J. C. Burges. Geometry and Invariance in Kernel Based Methods. *In Advances in Kernel Methods: Support Vector Learning*, Eds. B. Schölkopf, C. J. C. Burges, A. J. Smola, MIT Press, 89-116, 1999.

[13] G. Christensen, R. D. Rabbitt, M. I. Miller. A Deformable Neuroanatomy Textbook Based on Viscous Fluid Mechanics. *In Proceedings of 27th Conference on Information Sciences and Systems*, Eds. J. Prince and T. Runolfsson, 211-216, 1993.

[14] D. L. Collins, W. Dai, T. M. Peters, and A. C. Evans. Model-Based Segmentation of Individual Brain Structures from MRI Data. *In Proceedings of Visualization in Biomedical Computing 1992*, SPIE 1808:10-23, 1992.

[15] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham. Training Models of Shape from Sets of Examples. *In Proceedings of British Machine Vision Conference*, 9-18, Springer-Verlag, 1992.

[16] T. F. Cootes, C. Beeston, G. J. Edwards and C. J. Taylor. A Unified Framework for Atlas Matching Using Active Appearance Models. *In Proceedings of IPMI'99: Information Processing in Medical Imaging*, LNCS 1613:322-333, 1999.

[17] J. G. Csernansky, S. Joshi, L. Wang, J. M. Haller, M. Gado, J. P. Miller, U. Grenander, and M. I. Miller. Hippocampal Morphometry in Schizophrenia

by High Dimensional Brain Mapping. *In Proceedings of National Academy of Science*, 95(19):11406-11411, 1998.

[18] P.-E. Danielsson. Euclidean Distance Mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[19] C. Davatzikos, M. Vaillant, S. Resnick, J. L. Prince, S. Letovsky, and R. N. Bryan. A Computerized Method for Morphological Analysis of the Corpus Callosum. *Journal of Computer Assisted Tomography*, 20:88-97, 1996.

[20] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. *John Wiley & Sons*, 1973.

[21] B. Efron. The Jacknife, The Bootstrap, and Other Resampling Plans. SIAM, Philadelphia, PA, 1982.

[22] T. Evgeniou, M. Pontil, and T. Poggio. Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13(1):1-50, 2000.

[23] D. S. Fritsch, S. M. Pizer, B. Morse, D. H. Eberly, and A. Liu The Multiscale Medial Axis and Its Applications in Image Registration. Patter Recognition Letters, 15:445-452 1994.

[24] M. Frumin, P. Golland, R. Kikinis, Y. Hirayasu, D. F. Salisbury, J. Hennen, C. C. Dickey, M. Anderson, F. A. Jolesz, W. E. L. Grimson, R. W. McCarley, M. E. Shenton. Shape Differences in the Corpus Callosum in First Psychotic Episode Schizophrenia and First Psychotic Episode Affective Disorder. *Submitted to American Journal of Psychiatry.*

[25] G. Gerig and M. Styner. Shape versus Size: Improved Understanding of the Morphology of Brain Structures. *To appear In Proceedings of MICCAI'01: Medical Image Computing and Computer-Assisted Intervention*, October, 2001.

[26] F. Girosi, M. Jones, and T. Poggio. Regularization Theory and Neural Networks Architectures. *Neural Computation*, 7(2):219-269, 1995.

[27] P. Golland, W. E. L. Grimson and R. Kikinis. Statistical Shape Analysis Using Fixed Topology Skeletons: Corpus Callosum Study. *In Proceedings of IPMI'99: Information Processing in Medical Imaging*, LNCS 1613:382-387, 1999.

[28] P. Golland and W. E. L. Grimson. Fixed Topology Skeletons. *In Proceedings of CVPR'2000: Computer Vision and Pattern Recognition*, 10-17, 2000.

[29] P. Golland, W. E. L. Grimson, M. E. Shenton and R. Kikinis. Small Sample Size Learning for Shape Analysis of Anatomical Structures. *In Proceedings of MICCAI'2000: Medical Image Computing and Computer-Assisted Intervention*, LNCS 1935:72-82, 2000.

[30] P. Golland, W. E. L. Grimson, M. E. Shenton and R. Kikinis. Deformation Analysis for Shaped Based Classification. *In Proceedings of IPMI'01: Information Processing in Medical Imaging*, LNCS 2082, 517-530, 2001.

[31] T. Graepel and R. Herbrich. From Margin To Sparsity. *Advances in Neural Information Processing Systems 13*, Eds. S. A. Solla, T. K. Leen, and K.-R. Muller, MIT Press, 2001.

[32] International Brain Mapping Consortium.
Web site *http://nessus.loni.ucla.edu/icbm.*

[33] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *In Proceedings of the European Conference on Machine Learning*, LNCS 1398:137-142, 1998.

[34] T. Joachims. Transductive Inference for Text Classification Using Support Vector Machines. *In Proceedings of 16th International Conference on Machine Learning*, 200-209, 1999.

[35] T. Kapur, W. E. L. Grimson, W. M. Wells and R. Kikinis. Enhanced Spatial Priors for Segmentation of Magnetic Resonance Imagery. *In Proceedings*

MICCAI'1998: Medical Image Computing and Computer-Assisted Intervention, 457-468, October 1998.

[36] A. Kelemen, G. Székely, and G. Gerig. Three-dimensional Model-Based Segmentation. *In Proceedings of IEEE International Workshop on Model Based 3D Image Analysis*, Bombay, India, 87–96, 1998.

[37] R. Kimmel, D. Shaked, N. Kiryati and A. M. Bruckstein. Skeletonization via Distance Maps and Level Sets. *CVIU: Computer Vision and Image Understanding*, 62(3):382-391, 1995.

[38] M. E. Leventon, W. E. L. Grimson and O. Faugeras. Statistical Shape Influence in Geodesic Active Contours. *In Proceedings of CVPR'2000: Computer Vision and Pattern Recognition*, 316-323, 2000.

[39] F. Leymarie and M. D. Levine. Faster Raster Scan Distance Propagation on the Discrete Rectangular Lattice. *CVGIP: Image Understanding*, 55(1):84–94, 1992.

[40] F. Leymarie and M. D. Levine. Simulating the Grassfire Transform Using an Active Contour Model. In *IEEE Transactions PAMI*, 14(1):56-75, 1992.

[41] T. L. Liu, D. Geiger and R. V. Kohn. Representation and Self-Similarity of Shapes. *In Proceedings of ICCV'99*, 1129-1135, 1999.

[42] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21:163–169, 1987.

[43] A. M. C. Machado and J. C. Gee. Atlas Warping for Brain Morphometry. *In Proceedings of SPIE Medical Imaging 1998: Image Processing*, SPIE 3338:642-651, 1998.

[44] J. Martin, A. Pentland, and R. Kikinis. Shape Analysis of Brain Structures Using Physical and Experimental Models. *In Proceedings of CVPR'94: Computer Vision and Pattern Recognition*, 752-755, 1994.

[45] U. Montanari. Continuous Skeletons from Digitized Images. *Journal of the ACM*, 16(4):534-549, 1969.

[46] R. Ogniewicz and M. Ilg. Voronoi Skeletons: Theory and Applications. *In Poceedings CVPR'92: Computer Vision and Pattern Recognition*, 63-69, 1992.

[47] A. V. Oppenheim, A. S. Wilsky *et al.* Signals and Systems. 2nd Edition, *Prentice Hall*, 1996.

[48] E. E. Osuna, R. Freund, F. Girosi. Training Support Vector Machines: An Application to Face Detection. *In Proceedings of CVPR'97: Computer Vision and Pattern Recognition*, 130-136, 1997.

[49] X. Pennec, N. Ayache, and J. P. Thirion. Landmark-based Registration Using Features Identified Through Differential Geometry. *In Handbook of Medical Imaging*, Ed. I. N. Bankman, Academic Press, 499-513, 2000.

[50] S. M. Pizer, D. S. Fritsch, P. Yushkevich, V. Johnson, and E. Chaney. Segmentation, Registration, and Measurement of Shape Variation via Image Object Shape. *IEEE Transactions on Medical Imaging*, 18(10): 851-865, 1996.

[51] J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465-471, 1978.

[52] J. Rissanen. Stochastic Complexity and Modeling. *Annals of Statistics*, 14:1080-1100, 1986.

[53] S. Romdhani, S. Gong and A. Psarrou. A Multi-View Nonlinear Active Shape Model Using Kernel PCA. *In Proceedings of BMVC'99*, 483-492, 1999.

[54] E. M. Santori and A. W. Toga. Superpositioning of 3-Dimensional Neuroanatomic Data Sets. *Journal of Neuroscience Methods*, 50:187-196, 1993.

[55] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *Annals of Statistics*, 26:1651-1686, 1998.

[56] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input Space vs. Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, 10(5):1000-1017, 1999.

[57] B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.). Advances in Kernel Methods – Support Vector Learning. MIT Press, 1999.

[58] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299-1319, 1998.

[59] M. E. Shenton, R. Kikinis, F. A. Jolesz, S. D. Pollak, M. Lemay, C. G. Wible, H. Hokama, J. Martin, D. Metcalf, M. Coleman, and R. W. McCarley. Abnormalities in the Left Temporal Lobe and Thought Disorder in Schizophrenia: A Quantitative Magnetic Resonance Imaging Study. *New England Journal of Medicine*, 327:604-612, 1992.

[60] A. J. Smola and B. Schölkopf. From Regularization Operators to Support Vector Kernels. *Advances in Neural Information Processing Systems 10*, 343-349, 1998.

[61] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. *NeuroCOLT2*, Technical Report NC2-TR-1998-030, Royal Holloway College, 1998.

[62] M. Spivak. A Comprehensive Introduction to Differential Geometry. Third edition. *Publish or Perish, Inc.,*, 1999.

[63] L. Staib and J. Duncan. Boundary Finding with Parametrically Deformable Models. *IEEE PAMI*, 14(11):1061-1075, 1992.

[64] G. Székely, A. Kelemen, C. Brechbüler and G. Gerig. Segmentation of 2D and 3D Objects from MRI Volume Data Using Constrained Elastic Deformations of Flexible Fourier Contour And Surface Models. *Medical Image Analysis*, 1(1):19-34, 1996.

[65] V. N. Vapnik. The Nature of Statistical Learning Theory. *Springer*, 1995.

[66] V. N. Vapnik. Statistical Learning Theory. *John Wiley & Sons*, 1998.

[67] N. Vayatis and R. Azencott. Distribution-Dependent Vapnik-Chervonenkis Bounds. *In Proceedings of EuroCOLT'99*, LNAI 1572, Computational Learning Theory, 230-240, 1999.

[68] W. M. Wells, R. Kikinis, W. E. L. Grimson and F. Jolesz. Adaptive Segmentation of MRI Data. *IEEE Transactions on Medical Imaging*, 15:429-442, 1996.