Locally Defined Principal Curves and Surfaces

Umut Ozertem
B.S., Middle East Technical University, 2003
M.S., Oregon Health and Science University, 2006

Presented to the Division of Biomedical Computer Science within
The Department of Science & Engineering
and the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering
September 2008

The dissertation "Locally Defined Principal Curves and Surfaces" by Umut Ozertem has been examined and approved by the following Examination Committee:

Deniz Erdogmus
Assistant Professor
O.H.S.U.
Thesis Research Adviser

Todd K. Leen
Professor
O.H.S.U.

Misha Pavel
Professor
O.H.S.U

Jose C. Principe
Distinguished Professor
University of Florida

Dedication

To the light in my heart, song in my head, and the glitter in my eye.

Acknowledgements

Writing a PhD thesis is not something that you can do over the weekend; it takes a long time. Therefore, the first lines of this dissertation will be the words of gratitude to all those who helped me over the past years and made this possible.

To begin with, I would like to express my gratitude to my supervisor Dr. Deniz Erdogmus for his continued encouragement and support. Most important of all, Deniz always had confidence in me when I doubted myself, and helped me to gain my self-confidence. It's a real pleasure to watch him help and find a unifying view to generalize across seemingly disparate areas. I could not imagine a better advisor. Deniz has been a friend and mentor, as he always will be.

I also wish to thank members of my thesis committee, Dr. Todd K. Leen, Dr. Misha Pavel, and Dr. Jose C. Principe for their time and help for improving the quality of this dissertation with their valuable comments.

As well as students and faculty of AdSyL, I have been collaborating with Miguel A. Carreira-Perpinan, Robert Jenssen, Ismail Uysal, Engin Tola, Emre Ozkan, Ignacio Santamaria, Erhan Bas, Orhan Arikan, Kenneth E. Hild, Jayashree Kalpathy-Cramer and many members of CNEL at University of Florida. I thank everyone that supported me with insightful comments and helped me structure my ideas.

I've been very fortunate with my supervisors, and I'm deeply indebted to my former supervisor Dr. A. Aydin Alatan for guiding me to build the foundations of my knowledge in this field. Without his encouragement and guidance, I may not even have pursued a PhD degree.

I would like to thank my friends and my family for their support in all its forms, when it was most needed, but especially to my brother Arda for distracting me joyously and making me laugh as often and as loudly as he has.

Last, but not least, I thank to Cagri as she proved me that unconditional love is the key to get someone to listen all your complaints and frustrations, double check the proofs and equations, collect the required bibtex files from all over the internet, experience the dissertation writing endeavor with me and to proofread hundreds of pages - *including this one*.

Contents

List of Tables

List of Figures

Abstract

Locally Defined Principal Curves and Surfaces

Umut Ozertem, M.S.

Doctor of Philosophy,
Division of Biomedical Computer Science within
The Department of Science & Engineering
and the Oregon Health & Science University
School of Medicine

September 2008
Thesis Advisor: Deniz Erdogmus

Defined as self consistent *smooth* curves passing through the *middle* of the data, principal curves are used in many applications of machine learning as a generalization, dimensionality reduction and a feature extraction tool. The amount of smoothness and the middle of the data are not well-defined and the ill-posed definition of principal curves leads to practical difficulties in designing principal curve fitting algorithms, main causes of which are the desire to use global statistics such as conditional expectations to build a self consistent definition, and not decoupling the definition of principal curve from the data samples.

We take a novel approach by redefining the concept of principal curves, surfaces and manifolds with a particular intrinsic dimensionality, which we characterize in terms of the gradient and the Hessian of the probability density estimate. The theory lays a geometric understanding of the principal curves and surfaces, and a unifying view for clustering, principal curve fitting and manifold learning by regarding those as principal manifolds of different intrinsic dimensionalities.

Given the probability density of the data, the principal manifold of any intrinsic dimensionality and projections of any point in the feature space onto the principal manifold are uniquely defined. In real life, however, probability densities are never known, and should be estimated from the data samples. At this point, our definition of principal curves and surfaces does not impose any particular density estimation method, and we will provide results for kernel density estimation and Gaussian mixture models based density estimates. We will emphasize natural connections between challenges in the principal curve fitting and known results in kernel density estimation, and develop several practical algorithms to find principal curves and surfaces from data.

To present practical aspects of our theoretical contribution, we apply our principal curve algorithms to a diverse set of problems including image segmentation, time warping, piecewise smooth signal denoising, manifold unwrapping, optical character skeletonization, sharpening of time-frequency distributions, multiple-input multiple-output channel equalization and neighborhood graph construction.

All in all, this dissertation presents a theoretical contribution that brings a novel understanding of principal curves and surfaces, practical algorithms as general purpose machine learning tools, and applications of these algorithms to practical problems in several research areas.

Chapter 1

Introduction

Principal curves and surfaces fall into the general area of manifold learning and dimensionality reduction. As well as reviewing related approaches in dimensionality reduction literature, this chapter briefly presents in which category principal curve techniques fall into. Although it presents approaches related to principal curves, the aim of this chapter is not to provide a complete survey of existing dimensionality reduction techniques.

We leave the deeper review of existing approaches in principal curve literature to the introduction of Chapter 2, where we will provide our definition of principal curves and surfaces. Similarly, a review of current applications of principal curves is given in the beginning of chapter 4, where we focus on applications of principal curves.

1.1 Historical Background

The aim of the dimensionality reduction problem is to generate a *useful* low dimensional representation of the data, where close data points in the original space are mapped to close points in the projected space, and faraway points in the original space are mapped to faraway points in the projected space. As any unsupervised learning problem, the objective of dimensionality reduction is ill-defined and the

vagueness of *useful* in the definition above is intensional, since what useful is depends on the particular application. As well as exploratory data analysis and visual inspections, dimensionality reduction has been used for discovering patterns in the data.

Early methods in the dimensionality reduction literature are linear methods. If the data in a high dimensional space is mainly confined in a low dimensional subspace, linear methods can efficiently be used. However, even if the intrinsic dimensionality of the data is low, the data generally lies on a nonlinear manifold in the high dimensional space, and linear methods are bound to fail.

Principal components analysis (PCA) -also known as Karhunen-Loeve Transform in signal processing- is perhaps the most commonly used dimensionality reduction method [65, 64], which is defined using the linear projection that maximizes the variance in the projected space [62]. For a dataset, principal axes are the set of orthogonal vectors onto which the variance of the projected data points remains maximal. PCA has been widely used in many applications of multivariate analysis such as dimensionality reduction, feature extraction, noise reduction, enhancement, and coding, primarily because of its conceptual simplicity and its analytical properties.

Another closely related property of PCA, which in fact is discussed earlier in the literature [98], is that of all linear projections, the projection defined by the first principal component, namely the principal line, minimizes the mean square error between the data and its projection. Hence, the principal line provides a one-dimensional approximation for the data distribution, and gives the best linear summary of the data in the mean square reconstruction error sense. This property holds for principal subspaces of any dimension. In addition, for Gaussian distributions, the principal line is also self-consistent; that is, any point on the principal line is the conditional expectation of the data on the orthogonal hyperplane. Other

most widely used linear subspace projection methods include independent components analysis (ICA) [28, 63], linear discriminant analysis (LDA) [47], and factor analysis [53]. Linear methods are attractive due to their simplicity; however, they become insufficient in modelling nonlinear dependencies in the data.

Due to the insufficiency of the linear methods, many nonlinear projection approaches are proposed and studied. A common approach for building nonlinear projection schemes is to use a mixture of linear models [9]. Mixture models are attractive, since they are simple and analyzable as linear methods, however assuming the *suitable* model order, they are able to provide much more powerful tools compared to linear methods. Model order selection is a tough discrete optimization problem, and mixture methods suffer from the problems introduced by improper selection of model order. Techniques based on local PCA are good examples for mixture models [51, 87, 69, 70]. Another common way of developing nonlinear projections is to use generalized linear models [86, 43], which is based on the idea of constructing the nonlinear projection as a linear combination of nonlinear basis functions.

All reproducing kernel Hilbert space techniques such as kernel PCA [112] and kernel LDA [5] could be considered as nonparametric generalized linear models. These methods are based on mapping the data into a high dimensional space and performing the original linear method in this space, where the dot products are computed via a kernel function using the so-called *kernel trick*. More recent methods in this category use similarity matrices stemming from a weighted neighborhood graph rather than a kernel function, where the underlying implicit kernel functions are referred to as data-dependent or context-dependent kernel functions. These methods are referred to as graph-based kernel methods [113, 54].

Another track in dimensionality reduction is manifold learning algorithms based on generating the locality information of data samples using a neighborhood graph, which are also known as graph-based methods. In the neighborhood graph, the

nodes are the data points and the edges represent the neighborhood information. Generally these methods use the eigendecomposition of the data similarity matrix, generated using the neighborhood graph, and fall into the category of spectral methods in dimensionality reduction literature. Most well known spectral methods include isomap, local linear embedding, Laplacian eigenmaps, and maximum variance unfolding [120, 107, 6, 129]. The drawback of these methods is that the efficiency of these approaches depends on the quality of the neighborhood graph, and generally K-nearest neighbor or ϵ-ball graphs used for this purpose. Since they are based on data neighborhood graphs, these methods do not provide information on out-of-sample points in the feature space, and for every novel data the neighborhood graph needs to be reconstructed.

Hastie's proposition of self consistent principal curves [58] point out a whole different track in dimensionality reduction. Defining the self-consistency over the local conditional data expectations, they generalize self-consistency property of principal components and introduce the concept of principal curves. The main difference between the regression line and the principal line is that the regression line minimizes the squared error only in the response variable and it is not symmetric, whereas the principal line is symmetric and minimizes the squared error in all variables. Similarly, the regression curve is also not symmetric, and the idea of principal curves stems from generalizing the symmetry property of the principal line to a nonlinear regressor. Principal curves are defined as *smooth curves passing through the middle of the data* to provide a nonlinear summary of the data [58]. There are many principal curve algorithms in the literature, and it is probably safe to say that most of the algorithms in the literature are some variants of the Hastie's original approach. We will review these in section 2.1 in detail.

1.2 Brief Overview of Contributions of this Thesis to Current State-of-the-Art

This section briefly describes our main contributions in this dissertation. We start with a novel definition of principal curves and surfaces. Formal derivations in Chapter 2 are independent of the density estimation method used. Defining principal manifolds in terms of the probability density of the data allows us to link open ended problems of principal curve fitting literature, like optimal regularization constraints and outlier robustness to well established principles of density estimation literature, and yields practical algorithmic approaches as we will present in Chapter 3.

One significant problem with most principal curve algorithms in the literature is that, by definition, they are seeking for a *smooth curve*. In general, data may have loops, self intersections, and bifurcation points, in which case there does not exist a *smooth curve passing through the data* that can represent the data sufficiently. In the presence of such irregularities, our definition yields a principal graph, a collection of smooth curves, and the algorithms we present in Chapter 3 require no additional parameter to handle these irregularities.

The general definition of the d-dimensional principal manifold that we give in Chapter 2 converges to the modes of the data pdf for $d = 0$, which is a widely accepted clustering solution. The algorithms that we present in Chapter 3 simply show that principal curve and surface fitting can be done very similarly to clustering. Overall, both theoretically and practically, we provide a unifying framework for clustering, principal curve fitting, and manifold learning by regarding these as principal manifolds of different intrinsic dimensionality. This yields algorithm designs that can naturally be extended to higher order manifolds, as we will present in Chapter 3.

We also present solutions to very practical problems using principal curves. We

show that our principal curve definition provides a novel perspective for many applications, in which principal curve based approaches have never been used before.

Overall, we will present a theoretical contribution into the field of manifold learning, practical algorithms as general purpose machine learning tools, and novel applications of principal curves/surfaces in Chapters 2, 3, and 4, respectively.

1.3 Outline of the Thesis

In Chapter 2, we review the existing definitions and algorithms in the principal curve literature, and provide our definition of principal curves and surfaces. To underline the fact that the definition does not impose any particular density estimation method, we will present principal curve solutions for different density estimation methods using numerical integration methods that directly result from the definition.

Although numerical integration methods prove the validity of the definition, they cannot provide practical machine learning tools. Chapter 3 is on practical algorithms to find principal curves and surfaces that are based on the kernel density estimate of the data. After discussing why kernel density estimation is the density estimation method of our choice, we will develop practical algorithms based on KDE and present results in notional and real datasets. At the end of Chapter 3, we discuss the limitations due to finite sample sizes and the *curse of dimensionality*.

Chapter 4 is on several applications of our principal curve definition. We will present results on image segmentation, optical character recognition, signal denoising, manifold unfolding and blind source separation, and proximity graph construction, in all of which we directly use the algorithms that we developed in Chapter 3. Chapter 5 concludes the dissertation with our final comments and projected future work in both theoretical and practical aspects.

Chapter 2

Principal Curves and Surfaces

This chapter starts with a review of existing definitions in the principal curves literature. After the review, we present our definition of principal curves and surfaces. We start with a simple illustrative example, and then provide the formal definition of principal curves and surfaces. The existence conditions and limitations of our principal curve is also discussed.

Leaving real-data examples to the next chapter, the results we present in this chapter are mostly illustrative examples and the experiments are performed on notional datasets. Since the definition is not coupled with a particular density estimation method, we believe that the definition itself should be presented independent of the practical algorithms we derive in the next chapter. Therefore, all principal curve fitting results presented in this chapter are obtained using numerical integration methods that directly come from the definition, and we will present results using KDE and Gaussian mixture models.

2.1 Principal Curves and Closely Related Topics

Trevor Hastie and Werner Stuetzle's proposition of self consistent principal curves [57, 58] point out a whole different track for nonlinear dimensionality reduction. Defining the self-consistency over the *local* conditional data expectations, Hastie generalizes self-consistency property of principal components and introduces the

concept of principal curves. Hastie and Stuetzle define the principal curve as *an infinitely differentiable finite length curve that passes through the middle of the data.* Self-consistency means that every point on the curve is the expected value of the data points projecting onto this point.

Hastie and Stuetzle's major theoretical contributions are the following: (i) they show that if a straight line is self-consistent, it is a principal component (ii) based on the MSE criterion, self-consistent principal curves are critical points under some conditions. They use this second property to develop an algorithm that starts from the principal line and iteratively finds the principal curve by minimizing the average squared distance of the data points and the curve [57, 58]. Although they cannot prove the convergence of their algorithm, Hastie and Stuetzle claim that principal curves are by definition the fixed point of their algorithm, and if the projection step of their algorithm is replaced with least squares line fitting, the algorithm converges to the principal line.

There is no proof of convergence for Hastie-Stuetzle principal curve algorithm, which makes the theoretical analysis impossible. Therefore, existence of principal curves could only be proven for special cases such as elliptical distributions or distributions concentrated around a smooth curve, until Duchamp and Stuetzle's work on principal curves on the plane [39, 38].

Duchamp and Stuetzle also analytically calculate the principal curves of several planar distributions based on Hastie-Stuetzle definition, and they show that (i) principal curves in general are not unique (ii) if a density has several principal curves, they have to cross (iii) all principal curves are saddle points of MSE between the projection and the original data samples. The second property is somewhat analogous to the orthogonality of principal components, and the third property explains why cross-validation is not a suitable method for choosing the complexity of principal curve estimates. Third property is also interesting that it shows that it is problematic to design a principal curve algorithm just by minimizing projected

error. Hastie-Stuetzle principal curve algorithm is based on the idea of building a nonlinear counterpart of the principal line. In PCA, the principal line minimizes the mean squared projection error, and the smallest principal component maximizes it and all the remaining are saddle points. However, the same is not true for principal curves, and all principal curves are saddle points of the MSE distance function [39, 38].

Jeffrey Banfield and Adrian Raftery extend the Hastie-Stuetzle principal curve algorithm to closed curves and reduces the estimation bias[4]. Tibshirani introduces a semi-parametric model for principal curves. By approaching the problem from a mixture models point of view, they provide an algorithm that uses expectation maximization [121]. Pedro Delicado's proposition uses another property of the principal line rather than self-consistency [33]. Delicado's method is based on the total variance and conditional means and finds the principal curve of *oriented points* of the dataset. Stanford and Raftery propose another approach that improves on the outlier robustness capabilities of principal curves [119]. Probabilistic principal curves approach, which uses a cubic spline over a mixture of Gaussians to estimate the principal curves/surfaces [17], is known to be among the most successful methods to overcome the a common problem of the principal curve algorithms; the bias introduced in the regions of high curvature. Verbeek and coworkers use local principal lines to construct principal curves [125], and a soft version of the algorithm is also available [126], known as K-segments and soft K-segments methods. Mulier and Cherkassky point out the similarities between self-organizing maps [79] and principal curves [88].

Many principal curve approaches in the literature, including the original Hastie-Stuetzle algorithm, are based on the idea of minimizing mean projected error. An obvious problem with such approach is overfitting, and there are different methods in the literature to provide regularization. Sandilya and Kulkarni provide a regularized version of Hastie's definition by constraining bounds on the turns of the principal

curve to avoid overfitting [111]. They also show that principal curves with bounded turn always exist if the data distribution has finite second moments.

Kegl and colleagues define the regularization in another way by bounding the total length of the principal curve [75]. Similar to Sandilya's principal curve definition of bounded curvature, Kegl and colleagues show that principal curves of bounded length always exist, if the data distribution has finite second moments. Polygonal line algorithm, based on the bounded length principal curve definition provides better results than previous methods and merits special attention here.

Kegl's Polygonal Line algorithm starts with the principal line, and in each iteration, it adds a vertex to the polygonal line. After adding the vertex, the position of each vertex is updated by the projection and optimization steps. Projection step simply clusters the data with respect to the segment they project onto, and optimization step finds the new position of each vertex, based on minimizing the mean squared projection error and a curvature penalty term.

Kegl's Polygonal Line algorithm starts with the principal line, and in each iteration, it adds a vertex to the polygonal line. After adding the vertex, the position of each vertex is updated by the projection and optimization steps. Projection step simply clusters the data with respect to the segment they project onto, and optimization step finds the new position of each vertex, based on minimizing the mean squared projection error and a curvature penalty term. The most important advantages of this approach is that it is more robust than Hastie-Stuetzle algorithm, it eliminates the estimation bias, and it is faster than Hastie-Stuetzle algorithm, which has $O(N^2)$ computational complexity.

Later, Kegl also applies this algorithm to skeletonization of handwritten digits by extending it into the Principal Graph algorithm [74]. At this point, note that Hastie-Stuetzle definition of the principal curve requires the the principal curve not to intersect itself, which is quite restrictive, and perhaps, Kegl's Principal Graph algorithm is the only approach in the literature that can handle self-intersecting

data.

The disadvantage of bounded length principal curve approach is that the adaptation of the smoothing term employs heuristics, and the effective values of the required parameters cannot be obtained from the data directly. Unfortunately, the fact that principal curves exist as long as the data distribution has finite second moments and the algorithm converges to the principal curve of given length requires knowledge of the *suitable* length (or curvature), which, in general, is not known.

Works on extending principal curves into higher dimensions to seek principal manifolds mostly include approaches that use a linear combination of nonlinear multidimensional base functions and variants of self-organizing maps. LeBlanc and Tibshirani use multivariate adaptive regression lines [82]. Smola and colleagues use different regularization operators and obtain corresponding learning rates [117], [118]. Der and colleagues use self-organizing maps to construct principal manifolds [34].

Overall, the original principal curve definition by Hastie forms a strong basis for many, possibly all, principal curve algorithms one can find in the literature. Hastie and Stuetzle's algorithm, and all of its variants in the literature use the conditional data statistics to obtain the principal curve. Namely, they use either conditional expectation or squared projection error on the hyperplane orthogonal to the principal curve. Since it employs hyperplanes that extend to infinity, this approach uses global statistics of the data, which creates difficulties, where the nonlinear structure of the data makes turns such that the orthogonal hyperplane intersects the data at two distinct regions. Another difficulty arises when the hyperplanes that correspond to two different points on the principal curve intersect. This makes it ambiguous to which point the intersection is should be projected to. All these problems actually can be traced back to Hastie's original definition.

Many existing principal curve and principal manifold algorithms use least squares

regression or minimum squared projection error properties of linear principal component analysis to build a non-linear counterpart. Most probably, the underlying symmetric unimodal density distribution assumption implied by conditional expectation or least squares regression based approaches is also why many illustrations of current algorithms provide the most satisfactory results on synthetic datasets that are radially perturbed by a uniform or Gaussian distribution around a pre-constructed principal curve.

2.2 Definition of Principal Curves and Surfaces

2.2.1 An Illustration

Before we go into the details of the formal definition, we will present a simple illustration of our goals. For a point $\mathbf{x} \in \Re^n$, our principal manifold definition will generalize condition (i) into (ii): (i) A point is a local maximum of the pdf iff the gradient of the pdf is zero and the eigenvalues of the Hessian of the pdf are negative at $\mathbf{x}$. (ii) A point $\mathbf{x}$ is in the d dimensional principal manifold iff the gradient of the pdf is orthogonal to at least $(n\text{-}d)$ eigenvectors of the Hessian of the pdf at $\mathbf{x}$, and the eigenvalues corresponding to these orthogonal eigenvectors are negative. For one-dimensional principal manifolds, namely principal curves, this further simplifies to the following statement: *A point is on the principal curve iff the gradient of the pdf is an eigenvector of the Hessian of the pdf at this point and the remaining eigenvectors of the Hessian have negative eigenvalues.*

Figure 2.1 presents a simple comparison of the trajectories that follow eigenvectors of the local covariance, which is the proposed principal curve projection (will be defined later), and the trajectories of the local gradient directions, which essentially coincide with mean-shift projections used for clustering [21, 27]. These trajectories are shown for a two-dimensional Gaussian mixture probability density with five components. For different initial points in the feature space, following the trajectory

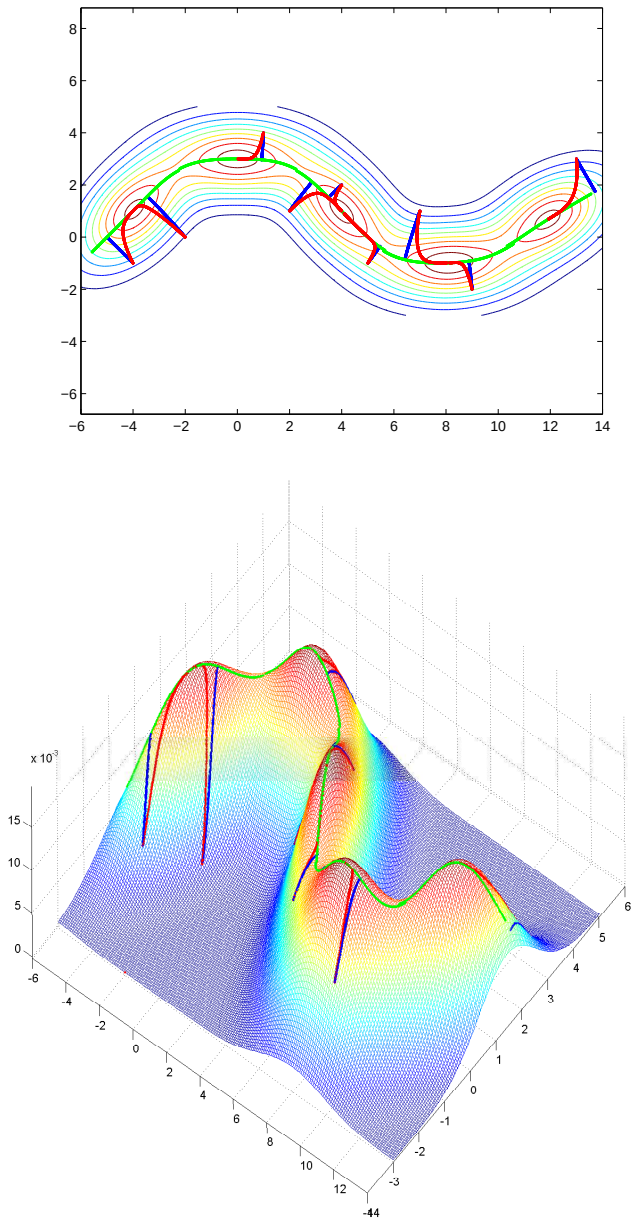


Figure 2.1: An illustration on a Gaussian mixture

.

of the local gradient (red) leads to the closest local maximum, whereas following the eigenvectors of the local covariance (blue) leads to the projection onto the principal curve. The principal curve of the probability density is also given (green).

Essentially, our proposition is to define the principal curves as the *ridge* of the probability density function. This can be regarded as likelihood maximization in a constrained space, where every point on the principal curve is a local maximum of the probability density in the orthogonal subspace of the curve.

2.2.2 Formal Definition of Principal Curves and Surfaces

We believe that a mathematically precise definition of principal curves should be given in terms of the data probability density, not in terms of data points or sample statistics. Therefore, in the rest of this chapter while proposing the definition and deriving the properties of principal curves and surfaces, we will assume that a probability density function expression that describes the data is available; it is either known or can be estimated parametrically or nonparametrically from the data. We will present results on theoretical distributions, and particulary for Gaussian distributions, we will show that our principal curve definition coincides with PCA.

Given a random vector $\mathbf{x} \in \Re^n$, let $p(\mathbf{x})$ be its pdf, $\mathbf{g}(\mathbf{x})$ be the transpose of the local gradient, and $\mathbf{H}(\mathbf{x})$ be the local Hessian of the probability density function. To avoid mathematical complications, we assume that the data distribution $p(\mathbf{x}) > 0$ for all $\mathbf{x}$, and is at least twice differentiable. Also let $\{(\lambda_1(\mathbf{x}), q_1(\mathbf{x})), \ldots, (\lambda_n(\mathbf{x}), q_n(\mathbf{x}))\}$ be the eigenvalue-eigenvector pairs of $\mathbf{H}(\mathbf{x})$, where the eigenvalues are sorted such that $\lambda_1(\mathbf{x}) \geq \lambda_2(\mathbf{x}) \geq \ldots \geq \lambda_n(\mathbf{x})$.

Definition 2.1. A point $\mathbf{x}$ is an element of the d-dimensional critical set, denoted by C^d iff $\mathbf{g}(\mathbf{x})$ is orthogonal to at least (n-d) eigenvectors of $\mathbf{H}(\mathbf{x})$, where orthogonality of two vectors means null Euclidean inner product.

Lemma 2.1. Critical points of $p(\mathbf{x})$ constitute C^0.

Proof. A point $\mathbf{x}$ is a critical point of $p(\mathbf{x})$ iff $\mathbf{g}(\mathbf{x})$=$\mathbf{0}$. Consequently, for $i =$

$1, \ldots, n, \langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$. Thus, $\mathbf{g}(\mathbf{x}) \perp \mathbf{q}_i(\mathbf{x})$. $\square$

This lemma illustrates that for $d = 0$ the critical set of zero intrinsic dimensionality converges to the critical points of the probability density; the local minima, local maxima, and saddle points. After partitioning the C^d into principal, minor and saddle sets, we will return to this point to use this connection between critical points and higher order critical sets to provide a unifying framework for clustering, principal curve fitting, and manifold learning.

Lemma 2.2. $C^d \subset C^{d+1}$.

Proof. Let $\mathbf{x} \in C^d$. By definition $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ for $i \in I$ where I is some index set, a subset of $\{1, \ldots, n\}$, with cardinality $|I| \geq (n - d)$. Therefore, there exist $(n - d)$ subsets of I with $(n - d - 1)$ elements. Let $I_{\subset}$ denote one of these subsets. Then we have $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ for all $j \in I_{\subset}$. Consequently, $\mathbf{x} \in C^{d+1}$. $\square$

In words, this lemma states that, by definition, lower dimensional critical sets are subsets of any higher dimensional critical set. For example, for any twice differentiable probability density, one can say that the principal curves must pass from the local maxima of the pdf. Similarly, two dimensional principal surfaces contain principal curves, and so on. From the aspect of implementing practical algorithms to find principal curves and surfaces, this is a very important property. Due to this hierarchical nature, same as in the sequential PCA algorithms [80, 130, 60], a deflation or inflation procedure could be employed to obtain the principle set of desired dimensionality in a sequential way.

Definition 2.2. A point $\mathbf{x} \in C^d$ but $\mathbf{x} \notin C^{d-1}$ is called a regular point of C^d. A point $\mathbf{x} \in C^{d-1}$ is called an irregular point C^d.

Lemma 2.3. If $\mathbf{x}$ is a regular point of C^d, then there exists an index set $I \subset \{1, \ldots, n\}$ with cardinality $|I| = (n - d)$ such that $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ iff $i \in I$. If $\mathbf{x}$ is an irregular point of C^d, then $|I| > (n - d)$.

Proof. Let $\mathbf{x}$ be a regular point of C^d. By definition $\mathbf{x} \in C^d$ but $\mathbf{x} \notin C^{d-1}$, then there exist some index set $I \subset 1, \ldots, n$ with cardinality $|I| \geq (n - d)$ but

not $|I| \geq (n-d+1)$, such that $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ for all $i \in I$. This implies that $|I| = (n-d)$. Let $\mathbf{x}$ be an irregular point of C^d. By definition, $\mathbf{x} \in C^d$ and $\mathbf{x} \in C^{d-1}$. Therefore, there exists some I with cardinality $|I| \geq (n-d)$ and $|I| \geq (n-d+1)$. Hence, $|I| > (n-d)$. $\square$

Lemma 2.4. Let $\mathbf{x}$ be a regular point of C^d and I be an index set with cardinality $|I| = (n-d)$ and such that $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ iff $i \in I$. The tangent subspace of C^d at $\mathbf{x}$ is $C^d_{\parallel}(\mathbf{x}) = span\{\mathbf{q}_i(\mathbf{x}) | i \notin I\}$ and the normal subspace of C^d at $\mathbf{x}$ is $C^d_{\perp}(\mathbf{x}) = \Re^n - C^d_{\parallel}(\mathbf{x}) = span\{\mathbf{q}_i(\mathbf{x}) | i \in I\}$.

Proof. Consider the following Taylor approximation in the vicinity of $\mathbf{x}$: $p(\mathbf{x}+\boldsymbol{\delta}) \approx p(\mathbf{x}) + \mathbf{g}^T(\mathbf{x})\boldsymbol{\delta} + \boldsymbol{\delta}^T\mathbf{H}(\mathbf{x})\boldsymbol{\delta}/2 + O(\|\boldsymbol{\delta}\|^3)$. Define *parallel* and *orthogonal* components of the local Hessian as follows (dropping the argument $\mathbf{x}$ whenever it is clear from context):

$$\mathbf{H}_{\parallel} = \sum_{i \notin I} \lambda_i \mathbf{q}_i \mathbf{q}_i^T \quad \mathbf{H}_{\perp} = \sum_{i \in I} \lambda_i \mathbf{q}_i \mathbf{q}_i^T \tag{2.1}$$

Since by hypothesis $\mathbf{g}^T\mathbf{q}_i = 0$ for $i \in I$, we can express the local gradient as a linear combination of the remaining eigenvectors: $\mathbf{g} = \sum_{i \notin I} \alpha_i \mathbf{q}_i$. For an arbitrary vector $\boldsymbol{\delta} = \boldsymbol{\delta}_{\parallel} + \boldsymbol{\delta}_{\perp} = \sum_{i \notin I} \beta_i \mathbf{q}_i + \sum_{i \in I} \beta \mathbf{q}_i$, clearly, $\mathbf{g}^T\boldsymbol{\delta} = \mathbf{g}^T\boldsymbol{\delta}_{\parallel}$. Also similarly $\boldsymbol{\delta}^T\mathbf{H}\boldsymbol{\delta} = \boldsymbol{\delta}_{\parallel}^T\mathbf{H}_{\parallel}\boldsymbol{\delta}_{\parallel} + \boldsymbol{\delta}_{\perp}^T\mathbf{H}_{\perp}\boldsymbol{\delta}_{\perp}$. Substituting these two expressions in the Taylor series approximation:

$$p(\mathbf{x}+\boldsymbol{\delta}) \approx p(\mathbf{x}) + \mathbf{g}^T\boldsymbol{\delta}_{\parallel} + (\boldsymbol{\delta}_{\parallel}^T\mathbf{H}_{\parallel}\boldsymbol{\delta}_{\parallel} + \boldsymbol{\delta}_{\perp}^T\mathbf{H}_{\perp}\boldsymbol{\delta}_{\perp})/2 \tag{2.2}$$

Consequently, at $\mathbf{x}$, a perturbation $\boldsymbol{\delta}_{\parallel} \in S_{\parallel}(\mathbf{x})$ yields $p(\mathbf{x}+\boldsymbol{\delta}_{\parallel}) \approx p(\mathbf{x}) + \mathbf{g}^T\boldsymbol{\delta}_{\parallel} + \boldsymbol{\delta}_{\parallel}^T\mathbf{H}\boldsymbol{\delta}_{\parallel}/2$. If infinitesimally small, this perturbation yields a point $\mathbf{x}+\boldsymbol{\delta}_{\parallel}$, where $\mathbf{g}(\mathbf{x}+\boldsymbol{\delta}_{\parallel}) \approx \mathbf{g}(\mathbf{x}) + \mathbf{H}_{\parallel}(\mathbf{x})\boldsymbol{\delta}_{\parallel}$. Noting that $\mathbf{g}^T(\mathbf{x}+\boldsymbol{\delta}_{\parallel})\mathbf{q}_i(\mathbf{x}) = \mathbf{g}^T(\mathbf{x})\mathbf{q}_i(\mathbf{x}) + \boldsymbol{\delta}_{\parallel}^T H_{\parallel}(\mathbf{x})\mathbf{q}_i(\mathbf{x}) = \alpha_i + \beta_i\lambda_i \neq 0$ for $i \notin I$. Hence, $\mathbf{x} + \boldsymbol{\delta}_{\parallel} \in C^d$; therefore, $C^d_{\parallel}(\mathbf{x})$ is locally tangent to C^d at $\mathbf{x}$. Similarly for $\boldsymbol{\delta}_{\perp} \in C^d_{\perp}(\mathbf{x})$, $g(x+\boldsymbol{\delta}_{\perp}) \approx \mathbf{g}(\mathbf{x}) + \mathbf{H}_{\perp}(\mathbf{x})\boldsymbol{\delta}_{\perp}$. Then $\mathbf{g}^T(\mathbf{x}+\boldsymbol{\delta}_{\parallel})\mathbf{q}_i(\mathbf{x}) = 0$ for $i \notin I$. Hence, $\mathbf{x} + \boldsymbol{\delta}_{\perp} \notin C^d$; therefore $C^d_{\perp}(\mathbf{x})$ is locally orthogonal to C^d at $\mathbf{x}$. $\square$

So far, we have defined the critical sets. Critical set of dimension d partitions the space into two parts, the d-dimensional tangent space and the $(n-d)$-dimensional normal space, and we have characterized these tangent and normal Euclidean subspaces to these submanifolds at every point. However, we have not characterized the critical manifolds as being locally maximum, minimum, or saddle. This characterization has to utilize the sign of the eigenvalues of the local Hessian corresponding to the eigenvectors that are in the orthogonal space, and will lead to the definition of locally maximal principal sets as the canonical solution for dimensionality reduction in a maximum likelihood manner.

Theorem 2.1.(Subspace Stationarity) Let $\mathbf{x}$ be a regular point of C^d and I be an index set with cardinality $|I| = (n-d)$ such that $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ iff $i \in I$. The following statements hold:

1. $\mathbf{x}$ is a local maximum in $C^d_\perp(\mathbf{x})$ iff $\lambda_i(\mathbf{x}) < 0 \ \forall i \in I$.

2. $\mathbf{x}$ is a local minimum in $C^d_\perp(\mathbf{x})$ iff $\lambda_i(\mathbf{x}) > 0 \ \forall i \in I$.

3. $\mathbf{x}$ is a saddle point in $C^d_\perp(\mathbf{x})$ iff $\exists \lambda_i(\mathbf{x}) < 0$ and $\exists \lambda_j(\mathbf{x}) > 0$ for $i,\ j \in I$.

Proof. Recall the definitions used in the proof of Lemma 2.4. Consider a perturbation $\boldsymbol{\delta}_\perp \in C^d_\perp(\mathbf{x})$ at $\mathbf{x}$ -a regular point of C^d-where $\|\boldsymbol{\delta}_\perp\| > 0$: $p(\mathbf{x}+\boldsymbol{\delta}_\perp) \approx p(\mathbf{x}) + \mathbf{g}^T\boldsymbol{\delta}_\perp + \boldsymbol{\delta}_\perp \mathbf{H}_\perp \boldsymbol{\delta}_\perp/2 = p(\mathbf{x}) + (1/2)\sum_{i\in I} \lambda_i \beta_i^2$. Note that (1) $p(\mathbf{x}+\boldsymbol{\delta}_\perp) < p(\mathbf{x}) \ \forall \ \delta_\perp$ iff $\lambda_i(\mathbf{x}) < 0 \ \forall i \in I$; (2) $p(\mathbf{x}+\boldsymbol{\delta}_\perp) > p(\mathbf{x}) \ \forall \ \delta_\perp$ iff $\lambda_i(\mathbf{x}) > 0 \ \forall i \in I$; (3) $p(\mathbf{x}+\boldsymbol{\delta}_\perp) \lesseqgtr p(\mathbf{x})$ for various $\boldsymbol{\delta}_\perp$ iff $\lambda_i(\mathbf{x}) \lesseqgtr 0 \ \forall i \in I$ for various $i \in I$. If some eigenvalues are zero at $\mathbf{x}$, then perturbations in the direction of corresponding eigenvectors will have the same quality as $\mathbf{x}$ as described by the theorem, thus plateaus will form, making $\mathbf{x}$ a platitude point (e.g., as in the case of a ring with uniform angular distribution and a unimodal radial perturbation). Furthermore, in the case of repeated eigenvalues uncertainties similar to those in PCA may occur.

Relaxing the inequalities to include zero-eigenvalues would result in platitude

points as usual. However, for the purposes of the following definitions, inequalities are kept strict for local maximum and minimum to simplify the discussion.

Definition 2.3. A point $\mathbf{x}$ is an element of the: (1) principal set P^d iff $\mathbf{x}$ is a regular local maximum point in $C^d_\perp(\mathbf{x})$; (2) minor set M^d iff $\mathbf{x}$ is a regular local minimum point in $C^d_\perp(\mathbf{x})$; (3) Saddle set S^d iff $\mathbf{x}$ is a regular saddle point in $C^d_\perp(\mathbf{x})$.

Lemma 2.5. (P^d, M^d, S^d) is a partition of C^d.

Proof. This conclusion is trivially obtained since the conditions for a point being an element of P^d, M^d, or S^d requires them to be in C^d besides the additional conditions on the sign of eigenvalues set by Theorem 2.1. This implies that (P^d, M^d, S^d) are all subsets of C^d, they do not intersect, and their union is C^d. □

Lemma 2.6. (1) $\mathbf{x} \in P^0$ iff $\mathbf{x}$ is a local maximum of $p(\mathbf{x})$. $\mathbf{x} \in M^0$ iff $\mathbf{x}$ is a local minimum of $p(\mathbf{x})$. (3) $\mathbf{x} \in S^0$ iff $\mathbf{x}$ is a saddle point of $p(\mathbf{x})$.

Proof. Let $\mathbf{x}$ be a critical point (local max, local min, or saddle) of $p(\mathbf{x})$. Then $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, therefore $\langle \mathbf{g}(\mathbf{x})\mathbf{q}_i(\mathbf{x}) \rangle = 0$ for $i \in I = \{1, \ldots, n\}$. Let $C^0_\perp(\mathbf{x})$ be $\Re^n$. Furthermore, the eigenvalues of $\mathbf{H}(\mathbf{x})$ have appropriate signs (all negative, all positive, mixed). This implies $\mathbf{x}$ is a regular critical point (local maximum, local minimum, saddle) of $C^0_\perp(\mathbf{x}) = \Re^n$, and by definition $\mathbf{x} \in (P^0, M^0, S^0)$. The converse is true with the same reasoning in reverse order. □

This lemma states that the modes of a probability density function (now called principal points), form the 0-dimensional principal set. Note that for a given pdf, modes provide a natural clustering solution. In fact, the widely used mean-shift algorithm [21, 27] utilizes this property to provide the clustering solution by mapping the data points to the modes of the density estimate obtained by KDE. Similarly, according to constrained likelihood maximization criteria based principal curve projections that we propose, the modes -the principal points- of the data probability density provides the best 0-dimensional summary of the data, obtained by mapping all the data points to the corresponding mode of each *attraction basin*[1] of the pdf.

[1]Attraction basin is defined as the set of points in the feature space such that initial conditions

Lemma 2.7. $P^d \subset P^{d+1}$. $M^d \subset M^{d+1}$.

Proof. This fact can be concluded in a manner similar to Lemma 2.2 by considering the appropriate eigenvalue sign combinations defined in Theorem 2.1. The same is not true for S^d since some $\mathbf{x} \in S^d$ may also satisfy $\mathbf{x} \in P^{d+1}$ or $\mathbf{x} \in M^{d+1}$ (e.g., a saddle point might be in the principal or minor curve). $\square$

In words, the principal and minor sets are confined in higher dimensional principal and minor sets. However, note that the same is not true for saddle sets. Going back to Figure 2.1, one can observe this inclusion property, as the principal curve (green) presented in this example passes all the local maxima points P^0 of the Gaussian mixture pdf.

For algorithmic purposes, the importance of this property is that the principal and minor sets can be, in principal, constructed by a procedure similar to deflation (inflation). One can determine the peaks P^0 and the pits M^0 of a pdf $p(\mathbf{x})$ and then trace out P^1 and M^1 by following the eigenvectors of the local covariance via a suitable differential equation with P^0 and M^0 initial conditions. The same could be done for each element of P^1 and M^1 as initial conditions to suitable differential equations to determine P^2 and M^2, etc. Similarly, projecting the data to a lower dimensional principal manifolds in a sequential manner, which is the opposite of what is explained above, is also possible.

If directly applied from the definition, the procedure outlined above, in general, requires numerical integration of complex nonlinear differential equations to identify the lines of curvature, which provide a natural local coordinate frame on the manifold. We have experimented with the numerical integration approach, employing Runge-Kutta order-4 as the integration technique, to determine P^1. The details on this will be presented in the next section along with examples on notional datasets.

---

chosen in this set evolve to a particular attractor. In the context of mean-shift the underlying cost function is the KDE of the data. In this case, attraction basins are regions bounded by minor curves, and the attractors are the modes of the pdf.

Definition 2.4. A point $\mathbf{x} \in (P^d, M^d)$ but $\mathbf{x} \notin (P^{d-1}, M^{d-1})$ is called a regular point of (P^d, M^d). A point $\mathbf{x} \in (P^d, M^d)$ and $\mathbf{x} \in (P^{d-1}, M^{d-1})$ is called an irregular point of (P^d, M^d).

Lemma 2.8. Let $\mathbf{x}$ be regular point of (P^d, M^d) and I be an index set with cardinality $|I| = (n-d)$ such that $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \rangle = 0$ iff $i \in I$. The tangent subspace to (P^d, M^d) at $\mathbf{x}$ is $(P^d_{\parallel}(\mathbf{x}), M^d_{\parallel}(\mathbf{x})) = span\{\mathbf{q}_i | i \notin I\}$ and the normal subspace of (P^d, M^d) at x is $(P^d_{\perp}(\mathbf{x}), M^d_{\perp}(\mathbf{x})) = span\{\mathbf{q}_i | i \in I\}$.

Proof. The argument is similar to Lemma 2.3. □

So far we have achieved the following: (1) a self-consistent local definition of critical, principal, minor, and saddle sets of a pdf is presented and the relationships between them are established, (2) the concept of critical nets is generalized to encompass manifolds with dimensionality higher than one, (3) a unifying framework between maximum likelihood clustering, curve and surface fitting, and manifold learning using deflation has been established. Theorem 2.1 establishes generalized conditions for a point being in a critical (stationary) submanifold utilizing local gradient and Hessian spectral information, of which the usual stationary point conditions remain as special cases. The definitions demonstrate that in general a globally *smooth* and maximally likely dimensionality reduction manifold that *passes through the data* may not be feasible. As will be illustrated with examples below, the principal sets for optimal dimensionality reduction might form irregular, self-intersecting manifolds in the global scheme, although their local existence and uniqueness is guaranteed.

2.2.3 Special Case of Gaussian Distributions, Connections to PCA

For a jointly Gaussian pdf $p(\mathbf{x})$ the definitions above reduce to the usual hyperplanes spanned by the eigenvectors of the covariance matrix. Consider a Gaussian density

with $\mathbf{\Sigma}^{-1} = \mathbf{V}\mathbf{\Gamma}^{-1}\mathbf{V}^T = \mathbf{V}_\parallel\mathbf{\Gamma}_\parallel^{-1}\mathbf{V}_\parallel + \mathbf{V}_\perp\mathbf{\Gamma}_\perp^{-1}\mathbf{V}_\perp^T$ and

$$\begin{aligned} p(\mathbf{x}) &= C_{\mathbf{\Sigma}} e^{-\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}/2}, \\ \mathbf{g}(\mathbf{x}) &= -p(\mathbf{x})\mathbf{\Sigma}^{-1}\mathbf{x}, \\ \mathbf{H}(\mathbf{x}) &= p(\mathbf{x})[\mathbf{\Sigma}^{-1}\mathbf{x}\mathbf{x}^T\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}] \end{aligned} \tag{2.3}$$

where $\mathbf{V} = [\mathbf{V}_\parallel, \mathbf{V}_\perp]$ and $\mathbf{\Gamma} = blockdiagonal(\mathbf{\Gamma}_\parallel, \mathbf{\Gamma}_\perp)$ are appropriately defined for some d-dimensional critical set. Consider a point $\mathbf{x} = \mathbf{V}_\parallel\, \boldsymbol{\alpha} \in C^d$. One can calculate easily

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= -p(\mathbf{x})\mathbf{V}_\parallel\mathbf{\Gamma}_\parallel^{-1}\boldsymbol{\alpha} = \mathbf{V}_\parallel\boldsymbol{\beta} \\ \mathbf{H}(\mathbf{x}) &= \mathbf{V}_\parallel[p(\mathbf{x})^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^T - p(\mathbf{x})\mathbf{\Gamma}_\parallel^{-1}]\mathbf{V}_\parallel^T + \mathbf{V}_\perp[-\mathbf{\Gamma}_\perp^{-1}]\mathbf{V}_\perp^T = \mathbf{Q}_\parallel\mathbf{\Lambda}_\parallel\mathbf{Q}_\parallel^T + \mathbf{Q}_\perp\mathbf{\Lambda}_\perp\mathbf{Q}_\perp^T \end{aligned} \tag{2.4}$$

where $[p(\mathbf{x})^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^T - p(\mathbf{x})\mathbf{\Gamma}_\parallel^{-1}] = \mathbf{B}_\parallel\mathbf{\Lambda}_\parallel\mathbf{B}_\parallel^T$ and $\mathbf{Q}_\parallel = \mathbf{V}_\parallel\mathbf{B}_\parallel$. Also, naturally we observe that $\mathbf{Q}_\perp = \mathbf{V}_\perp$ and $\mathbf{\Lambda}_\perp = -\mathbf{\Gamma}_\perp^{-1}$. The latter identities indicate that at $\mathbf{x}\in C^d$, the normal-subspace eigenvectors of the local Hessian coincide with some of the eigenvectors of the data covariance $\mathbf{\Sigma}$. Thus, linear PCA emerges naturally from the proposed principal curve definition for Gaussian distributions.

In the case of an arbitrary pdf $p(\mathbf{x})$, consider the following quadratic Taylor approximation:

$$log\ p(\mathbf{x}+\boldsymbol{\delta}) \approx log\ p(\mathbf{x}) + p^{-1}(\mathbf{x})\mathbf{g}^T(\mathbf{x})\boldsymbol{\delta} + (1/2)\boldsymbol{\delta}^T[p^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}) - p^{-2}(\mathbf{x})\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T]\boldsymbol{\delta} \tag{2.5}$$

This approximation corresponds to approximating the pdf locally with a Gaussian, thus establishes a connection with local PCA, nonlinear PCA, and local linear manifold learning techniques investigated earlier in the machine learning literature. Since the usual linear PCA induced hyperplanes remain as special case of the proposed definition corresponding to a joint Gaussian pdf, the following definition for ranking local eigendirections as principal-minor components is motivated.

Definition 2.5. Let $\mathbf{x}$ be a regular point in P^d (M^d). Let $\langle \mathbf{g}(\mathbf{x}), \mathbf{q}_i(\mathbf{x})\rangle = 0$ for $i \in I = \{1, \ldots, n\} - I^c$, where I is some index set with cardinality d. Define

the local covariance matrix of $p(\mathbf{x})$ to be $\Sigma^{-1}(\mathbf{x}) = -p^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}) + p^{-2}\mathbf{g}(\mathbf{x})^T\mathbf{g}(\mathbf{x})]$ and assume that its eigendecomposition is $\{\gamma_i(\mathbf{x}), \mathbf{v}_i(\mathbf{x})\}$ for $i \in \{1, \ldots, n\}$. Assume that the eigenvalues with index $i \in I^C$ of the local covariance matrix satisfy the following: $\gamma_1 > \ldots > \gamma_m > 0 > \gamma_{(m+1)} > \ldots > \gamma_d$. Then, the local ranking of principal directions at $\mathbf{x}$ from minor to principal follow the same ranking of indices.

For the special case of Gaussian distributions, the local covariances defined above becomes constant over the data space and equal to the data covariance. Thus, the local principal directions are aligned with the global principal directions and following these directions starting from any point, takes one to the corresponding subsurface of C^{d-1}. The process of solving a partial differential equations starting from any data point and following the direction of a particular eigenvector is the ideal nonlinear data projection method, which preserves the coordinates of the data in their projections.

An important point to note is that our principal curve definition preserves the ambiguities of PCA. Similar to PCA, the principal curve of a spherically symmetric distribution is not well-defined; however, local ranking can be achieved for any d-dimensional principal manifold under the assumption that the Hessian matrix has distinct eigenvalues.

2.2.4 Ranking of principal curves/surfaces in a principal set

In principal components analysis, the ordering of the principal component directions are naturally given by sorting the corresponding eigenvalues of the covariance matrix in an descending order. In this case, the first d components of the ordered principal component directions give the best d-dimensional summary of the data according to the mean squared projection error criteria.

As mentioned before, similar conditional expectation or mean squared projection error based definitions has driven the principal curves research, but in general, the definition is limited to the nonlinear counterpart of the first principal component.

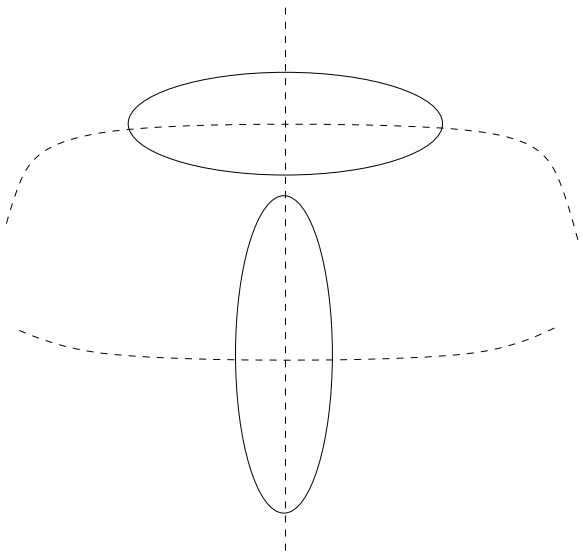


Figure 2.2: A T-shaped Gaussian mixture

.

In fact, there is no definition of *second, third, etc. principal curve* in the literature. Similarly, according to our definition, it is ambiguous to designate a portion of a principal set into the first, second, third, etc. component of that principal set.

Considering the above connection to PCA, one can see that our principal curve definition is not limited to the nonlinear counterpart of the first principal component, in fact, it covers all principal components in the one-dimensional principal set. At this point, one may think that our definition may allow to define an orthogonal set of principal curves, in other words, partition the one-dimensional principal set into *ordered* principal curves that are locally orthogonal to each other.

To define the ordering, one can employ the ordering of the eigenvalues of the local covariance at every point. It is easy to see that this approach will give the correct ordering in the trivial Gaussian example, where the principal set analysis converges to linear PCA. This idea, in fact, can be generalized into all radially symmetric probability densities, and second, third, etc. principal curves can be defined for

these densities using the eigenvalues of the Hessian matrix.

However, in general, data densities may take complex forms and counterintuitive scenarios may arise. Hence, generally, local information may not always indicate the global rank, and a global ordering in a principal set of given dimensionality may not be possible. To illustrate this fact, we will study a T-shaped Gaussian mixture consisting of two components, where principal curves form a graph, where the modes are the nodes. Note that both branches of this principal graph correspond to the leading eigenvector of the local covariance at different portions of the feature space and a global ranking is not possible. For this reason, we will stick to the local ranking at all times. This easily yields the informative portions of each branch in the principal set, to collection of which we will call the principal graph.

2.2.5 Existence of Principal Curves and Surfaces

The assumptions under which we define principal curves and surfaces already gives the existence conditions directly. Still, it is worthwhile to mention these briefly, before we move on to examples.

Considering Hastie's principal curve definition, the existence proof of principal curves is limited to some special cases, such as elliptical or spherical distributions concentrated around a smooth curve. It should also be noted that this definition of the principal curve requires the the principal curve not to intersect itself, which is quite restrictive.

The principal curve definition of Kegl and coworkers is theoretically more appealing in this context, since by their definition, the principal curve always exists if the distribution has finite second moments. Moreover, their definition also allows self-intersections, and Kegl presents results with self-intersecting data as well [74].

In our perspective, the principal curve exists locally as long as the data probability density is twice differentiable, such that the Hessian is nonzero. There is no

restriction of finite moments, which is an improvement on existing methods. However, by our definition the principal curve does not exist for uniform distributions. In practice, however, since we build our algorithms based on KDE with Gaussian kernel functions, even if the true underlying distribution is uniform, KDE guarantees that the gradient and Hessian exist for uniformly distributed datasets. In general, for densities with piecewise constant gradients, smoothing by convolution with a twice differentiable unimodal circularly symmetric pdf is possible. This corresponds to adding some independent noise.

2.3 Numerical Integrations to Find Principal Curves and Surfaces

2.3.1 Finding the Principal Set

Finding the principal set can be achieved by solving the partial differential equation to follow the eigenvectors of the local covariance matrix along the ridge of the pdf. Since, by Lemma 2.7, we know that the principal curve is passing from the modes of the pdf, one can find the modes of a pdf and *shoot* a trajectory along the direction of each eigenvector at the mode of interest to combine these modes. Finding the modes of a pdf is an easy problem. In practice, one can start with a number of initial points and use a gradient ascent or fixed point algorithm to find the corresponding modes. In Gaussian mixture densities or kernel density estimates, such a fixed point algorithm might be the well-known mean shift algorithm [21, 27]. To find the principal curve portion that connects the modes, one can reach to the other modes by following the corresponding eigenvector of the local covariance. We used forth order Runge-Kutta numerical integration algorithm (RK4) to follow the eigenvectors [81].

With a *suitably small* step size, we have obtained good results for principal curves

of various Gaussian mixture densities. The green curves in the illustration presented in Figure 2.1 is obtained using this method. *Given the data probability density, the principal curve is uniquely defined.* However, error accumulation, especially at high curvature points of P^1, prevent accurate determination of P^1 by brute force numerical integration using a fixed integration step size and this method is not preferred for practical purposes in learning theory and applications.

2.3.2 Finding the Projections onto the Principal Set

Similar to above, one can employ the numerical integrations to follow the eigenvectors of the local covariance until the stopping criterion is satisfied, that is the gradient is parallel with an eigenvector of the Hessian.

The projection trajectories (blue) presented in Figure 2 and 3 are obtained by using the RK4 numerical integrations initialized to the data samples. *Given the data probability and the initial point, the locally-optimum projection onto the principal curve using lines of curvature is uniquely defined.*

At this point, note that this procedure offers an alternative method to identify the principal curve by building the principal curve by combining the principal curve projections of data samples. This not only gives where on the curve the data points are being projected onto, but also overcomes the problems with numeric error accumulation at the high curvature points.

2.4 Examples of Principal Curves and Surfaces by Numerical Integrations

Here we present the results of the direct implementation of our definition. For visualization purposes, all examples here are selected two or three dimensional datasets, some of which has been used in earlier papers in principal curve literature. As well as these datasets, the results of polygonal line algorithm [75] and Hastie-Stuetzle

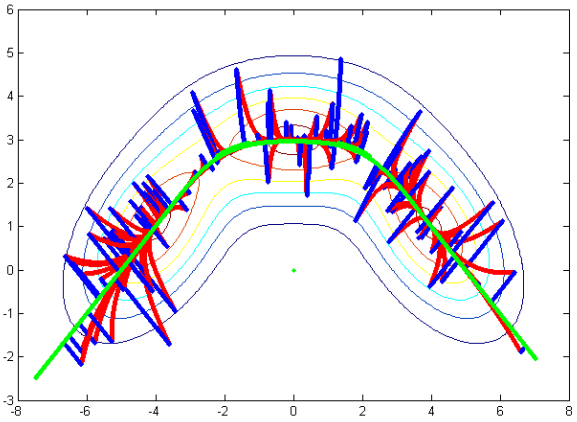


Figure 2.3: Gradient (red) and principal curve projection trajectories (blue) for a Gaussian mixture pdf.
.

algorithm [58] is courtesy of Balazs Kegl.

Again for the visualization purposes, sought principal manifolds in this section are either one or two dimensional. Experiments include theoretical known distributions, KDE and Gaussian mixture models, and RK4 numerical integration method is used to implement the principal curve projection.

2.4.1 Projection Trajectories on a Gaussian Mixture

Similar to the earlier illustration, here we present the trajectories of the gradient and the principal curve projection on Figure 2.3 - with more samples than in Figure 2.1. Trajectories of the gradient (red) maps the data points to the closest mode of the dataset and yields a clustering result. The trajectories of the principal curve projection (blue) and the principal curve being projected onto are given (green).

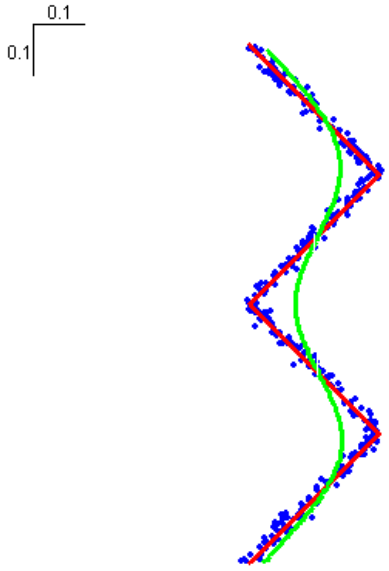


Figure 2.4: Zig-zag dataset (blue), generating curve (red), and Hastie-Stuetzle principal curve (green)
.

2.4.2 Zig-Zag Dataset

Zig-Zag dataset has been used in an earlier principal curve paper by [75], where a piece-wise linear principal curve is proposed. Figure 2.4 shows the data samples (blue) the true underlying generating curve (red) and result of Hastie's algorithm (green). Figure 2.5 presents the results of Kegl's polygonal line algorithm for different penalty coefficients. The length penalty coefficient is equal to 0.1 (top left), 0.3 (top right), 0.5 (bottom left), and 0.7 (bottom right), respectively.

We are going to present our principal curve fitting results for two different density

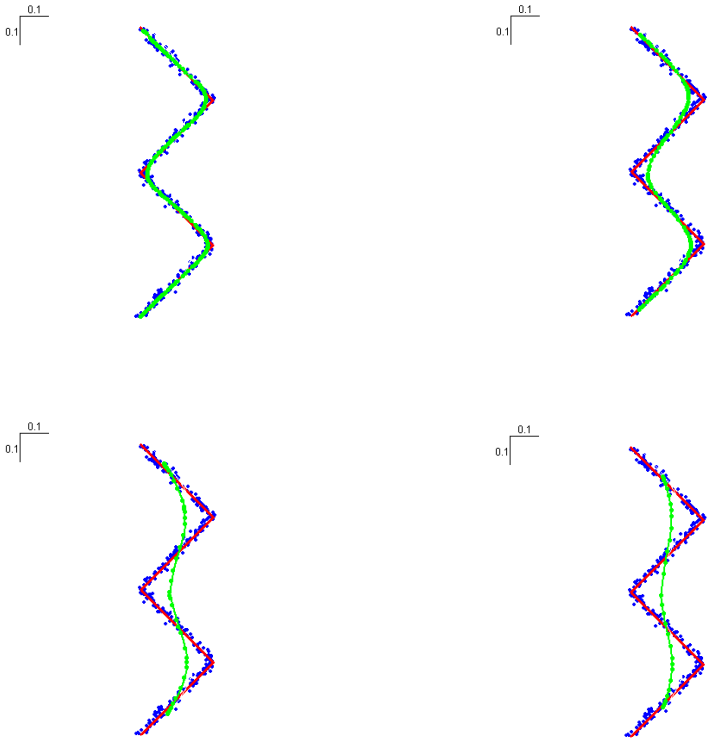


Figure 2.5: Zig-zag dataset (blue), generating curve (red), and result of the Polygonal Line Algorithm (green)

.

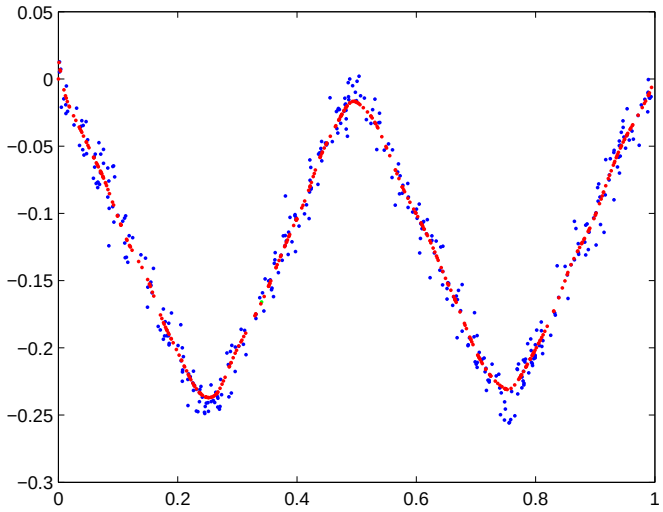


Figure 2.6: Zig-zag dataset, and KDE-based principal curve projetions

.

estimates. For this dataset $\mathbf{x}_i \in \Re^n$ for $i \in \{1, \ldots, N\}$, one can write the KDE as

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} K(\mathbf{x} - \mathbf{x}_i) \tag{2.6}$$

where the kernel function is selected as a Gaussian and bandwidth of the Gaussian is optimized using leave-one-out maximum likelihood approach[2]. Since ML kernel bandwidth selection is a data-driven procedure, for this case the resulting density estimate is purely nonparametric. Figure 2.6 shows our KDE-based principal curve results for the same dataset, where the data samples (blue) and their principal curve projections (red) are shown.

At this point, we will change the density estimation method and repeat the experiment for the same dataset using a Gaussian mixture model density. Here we employ a GMM with four components and use expectation-maximization (the details

[2]The details of leave-one-out maximum likelihood kernel bandwidth selection will be detailed in next chapter.

of which can be found in [40]). Results of GMM-based principal curve projections is given in Figure 2.7.

Comparing KDE-based results in Figure 2.6 and GMM-based results in Figure 2.7, one may argue that obtaining a *good* density estimate with GMM would require less data samples for this piecewise linear generating curve. As long as the parametric model is a suitable fit for the dataset and given the *suitable* model order, this is true. In the next chapter, we will provide a much detailed comparison of KDE and GMM based principal curves that includes quantitative results, and discusses well-known topics in density estimation like model order selection and curse of dimensionality.

2.4.3 Spiral Dataset

Since many principal curve algorithms are based on the idea of starting with the principal line and adding complexity to the structure (for example adding a vertex to piecewise linear curve) to minimize mean projected error, a dataset that folds into itself may lead to counterintuitive results, and spiral dataset is a benchmark dataset, that has been used in manifold learning and principal curve algorithm literature [75, 127] (again, this dataset is provided by Kegl).

Similar to above, we start with the results of Hastie-Stuetzle algorithm and Kegl's polygonal line algorithm. Figure 2.8 shows the data samples (blue) the true underlying generating curve (red) and result of Hastie's algorithm (green). Figure 2.9 presents the results of Kegl's polygonal line algorithm for different penalty coefficients. The length penalty coefficient is equal to 0.1 (top left), 0.2 (top right), 0.4 (bottom left), and 0.5 (bottom right), respectively.

As in the previous example, we use KDE and use the leave-one-out ML kernel bandwidth for this dataset as well. Figure 2.10 shows the same spiral dataset (blue) along with the principal curve projections (red). Comparing Figure 2.9 and Figure 2.10, one can see that both Polygonal Line algorithm (with the suitable parameter

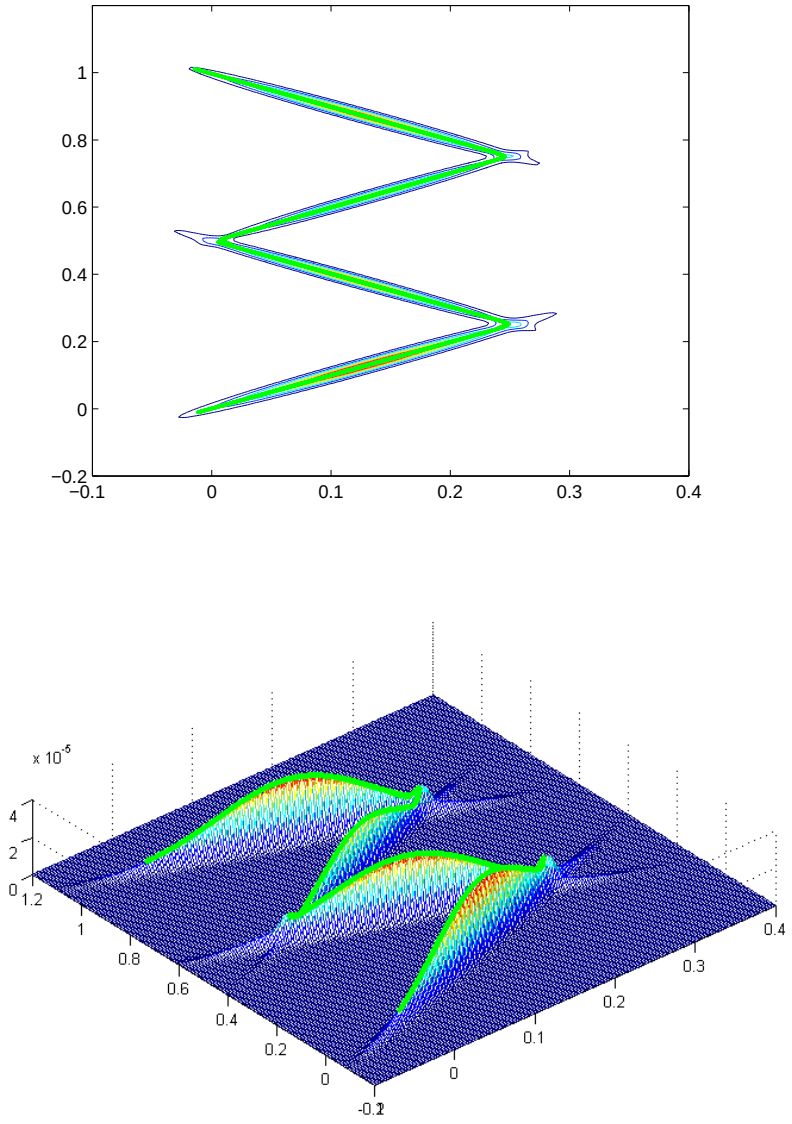


Figure 2.7: Zig-zag dataset, and GMM-based principal curve projetions (green)

.

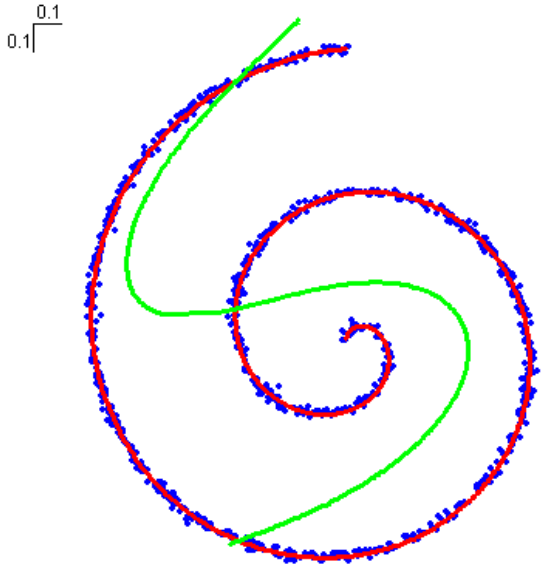


Figure 2.8: Spiral dataset (blue), generating curve (red), and Hastie-Stuetzle principal curve (green)
.

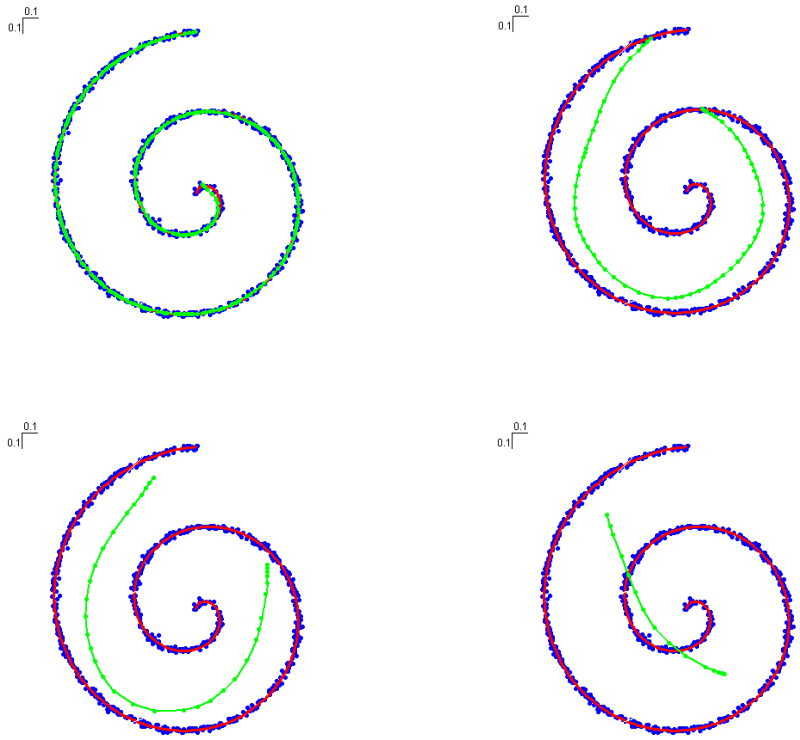


Figure 2.9: Spiral dataset (blue), generating curve (red), and result of the Polygonal Line Algorithm (green)

.

set) and our locally defined principal curve can achieve satisfactory results. Therefore, we create a more challenging scenario, where the spiral has more noise and fewer samples. Figure 2.11 shows our results (again using KDE with ML kernel bandwidth), and Figure 2.12 shows results of Polygonal Line algorithm for different penalty coefficients; 0.05 (top left), 0.1 (top right), 0.2 (bottom left), 0.3 (bottom right).

Figure 2.11 shows that our local definition is able to find the underlying spiral shape in this more challenging noisy case as well. An important question at this point is the following: Is there a theoretical upper limit for the noise level that our

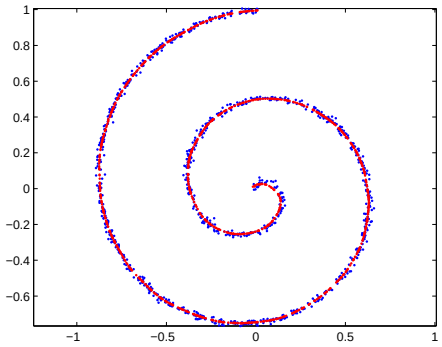


Figure 2.10: Spiral dataset (blue), and KDE-based principal curve projections (red)
.

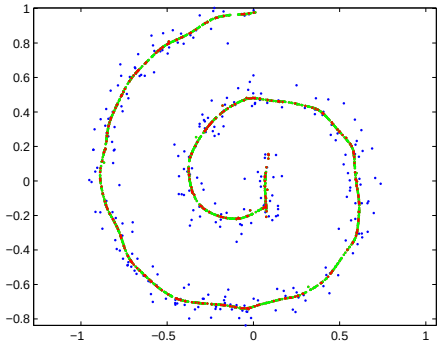


Figure 2.11: Spiral dataset (blue), and KDE-based principal curve (green), and projections of data samples onto principal curve (red)
.

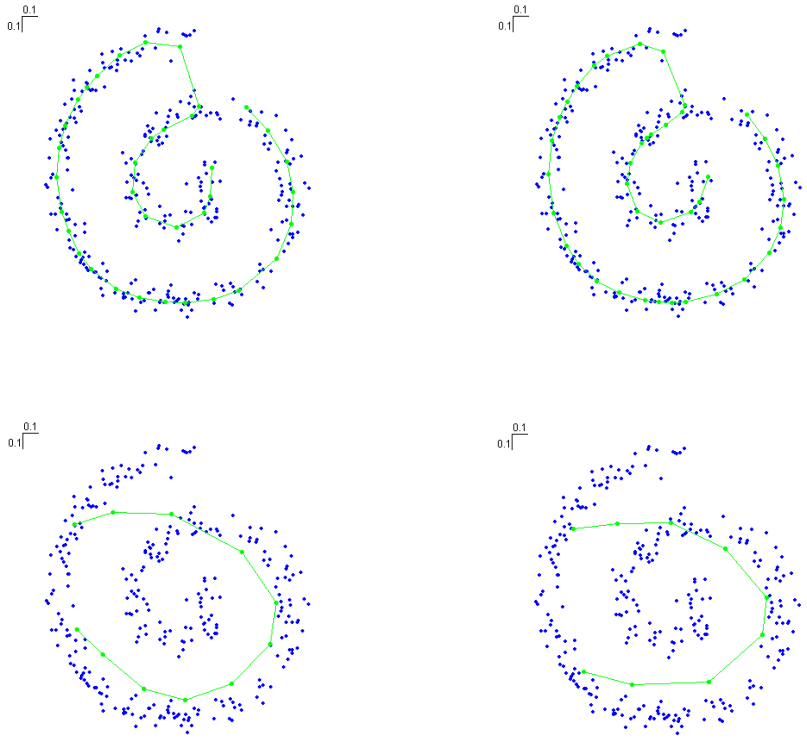


Figure 2.12: Spiral dataset (blue), generating curve (red), and result of the Polygonal Line Algorithm (green)

.

principal curve approach can handle? In practice, the principal curve projection can identify these two *branches*, as long as there is a minor curve in between; so that in the orthogonal space of the principal curve, there is a local minimum of the pdf in between these two branches. Generating a theoretical upper bound on the noise level, however, requires the knowledge of the true generating curve (to know the spacing in between these two branches) and the noise model. Below derivation provides the upper limit for the common case of additive Gaussian noise model.

Assume the orthogonal space of the principal graph is given as a Gaussian mixture with two components with the same standard deviation, centered at 0 and μ,

$$\begin{aligned} p(x) &= G(0,\sigma) + G(\mu,\sigma) \\ &= \tfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} + \tfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \end{aligned} \tag{2.7}$$

For this distribution, the gradient and the Hessian are

$$\begin{aligned} \nabla p(x) &= \tfrac{1}{\sqrt{2\pi}\sigma} \left(-\left(\tfrac{x}{\sigma^2}\right)\ e^{-\frac{x^2}{2\sigma^2}} - \left(\tfrac{x-\mu}{\sigma^2}\right)\ e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right) \\ H(x) &= \tfrac{1}{\sqrt{2\pi}\sigma} \left(\left(\tfrac{x^2}{\sigma^4} - \tfrac{1}{\sigma^2}\right)\ e^{-\frac{x^2}{2\sigma^2}} + \left(\tfrac{(x-\mu)^2}{\sigma^4} - \tfrac{1}{\sigma^2}\right)\ e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right) \end{aligned} \tag{2.8}$$

Using (2.8) and solving for $H(x) = 0|_{x=\mu/2}$, one obtains $\mu = 2\sigma$. So, under the additive Gaussian noise model assumption, the standard deviation of the Gaussian noise has to be less than $\mu/2$ to identify separate branches of the principal curve.

2.4.4 Loops, Self-intersections, and Bifurcations

Since they are specifically designed to fit *smooth* curves to the data, traditional parametric and nonparametric principal curve identification approaches in the literature have difficulties if there are loops, bifurcations and self intersections in the data. Perhaps the most efficient algorithm in this context is Kegl's principal graph algorithm [74], where Kegl modifies his polygonal line algorithm [75] with a table of predefined rules to handle the loops, intersections and bifurcations.

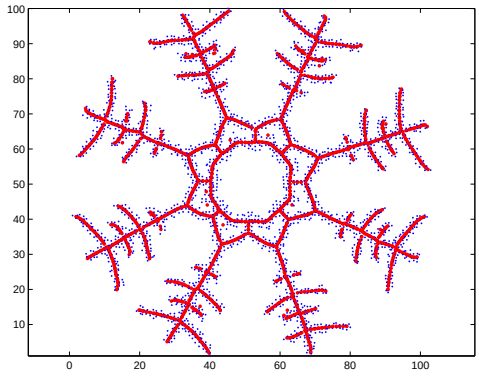


Figure 2.13: Snow crystal dataset (blue), and principal curve projections of the data (red)
.

Our definition can handle these irregularities with no additional effort. To underline this fact, in this experiment, we use a synthetically-created snow crystal dataset that has a number of loops, self intersections, and bifurcation points. To estimate the pdf of the data nonparametrically from the samples, here we employ KDE with ML kernel bandwidth again. Results are presented in Figure 2.13.

2.4.5 Extending the Definition to Higher Dimensional Manifolds

The generalization of principal curves to principal surfaces and manifolds is naturally achieved with the proposed definition and the framework as seen in previous discussions. Here we present the one and two dimensional principal manifold of the three-dimensional helix dataset. Figure 2.14 and Figure 2.15 show the results obtained for the principal curve and principal surface of this dataset from two different angles. In Figure 2.15, we present the surface built by the Delaunay triangulations [32] of the principal surface projections for better visualization.

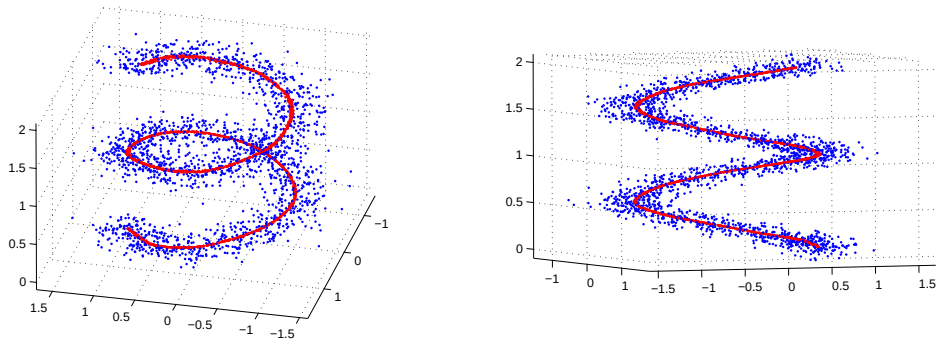


Figure 2.14: Helix dataset (blue), and principal curve projections of the data (red)
.

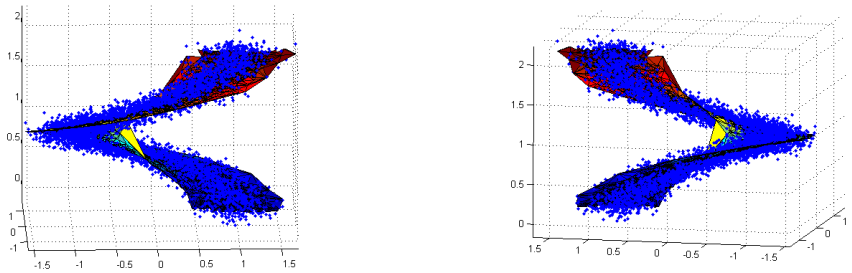


Figure 2.15: Helix dataset (blue), and principal surface of the data
.

Chapter 3

Practical Algorithms to Find Principal Curves and Surfaces

Developing practical machine learning tools are as important as the novelty of the theoretical contributions. Although the results that use numerical integration methods presented in the previous chapter verify the validity of the definition, they cannot be presented to the machine learning community as practical tools due to required computational load and error accumulation problems. In this chapter, we will implement several *fast* and *nonparametric* approaches stemming from subspace constrained likelihood maximization criterion that yield algorithms in the spirit of well-known mean shift algorithm [21, 27].

All algorithms presented in this section are based on the KDE of the data, and for this reason the chapter starts with the underlying reasons of why this is the density estimate of our choice. After developing some practical approaches and presenting comparisons with some quantitative results for different sample sizes and noise levels, we will analyze the effects of finite sample sizes and the extremal characteristics in high dimensional spaces.

Mainly, two types of algorithms are required. In applications like denoising, dimensionality reduction, and manifold unwrapping, the projections onto the principal curve are required for all data samples. On the other hand, for applications like OCR feature extraction, the principal curve itself -or a sufficient approximate-

is sufficient, and data projections are not required. The second type is generally much easier, and also yields faster implementations.

All algorithms we will present are deviants of the Eigenspace-constrained Mean Shift (ECMS) algorithm that is presented in Section 3.2. Eigenspace-constrained Mean Shift (Section 3.2) and Subspace-constrained Mean Shift (Section 3.3) have $O(N^2)$ computational complexity when using a KDE, where N is the number of samples. Both of these methods are nonparametric approaches and provide the data projections accurately. Parametric density estimate variations are trivial to derive.

Piecewise-linear and vector quantized versions presented in Section 3.4 and Section 3.5 have $O(Nm)$ computational complexity, where m is the number of points used in the principal curve approximation, and usually $m \ll N$. These approaches bring the advantage of lower computational complexity, however the accuracy of the principal curve approximations and data projections also decrease as m decreases.

3.1 Connections to Kernel Density Estimation

One can see many natural connections between the open ended problems in principal curve fitting, and studies in KDE literature. Translating these problems into known techniques in KDE literature not only yields practical algorithms, but also allows us to approach various problems in principal curve fitting literature from a principled point of view. Connections to density estimation bring the advantage of providing the regularization and outlier robustness in a nonparametric way, leading to purely data-driven nonparametric principal curve algorithms.

3.1.1 Seeking the principal curve in a region of interest

One of the challenges in principal curve literature is to define only a part of the principal curve in the region of interest without having to struggle with the data

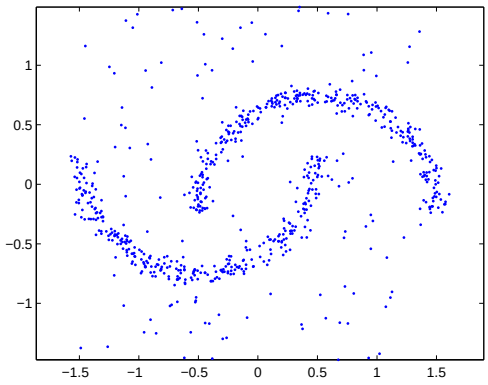
points that are far from the region of interest, especially for large number of samples. The original principal surface formulation itself is stated to be inefficient for large sample sizes [57, 58], and the reason behind all difficulties in this problem with the earlier definitions is the use of global statistics. In this framework, solution approaches to define the region of interest are limited to heuristic attempts like thresholding the distance and disregarding the rest of the data while employing global statistics.

In our KDE-based approach, one can easily see that the problem of region of interest translates to selecting a finite support kernel. To define the principal curve only in the region of interest, one only needs to estimate the pdf in this portion of the feature space, and the support of the kernel function clearly defines the boundaries of the feature space that need to be used in the estimation. Moreover, from a theoretical point of view, this approach makes theoretical analysis much easier as compared to above mentioned heuristics, since asymptotic properties of finite support KDE is well-known.

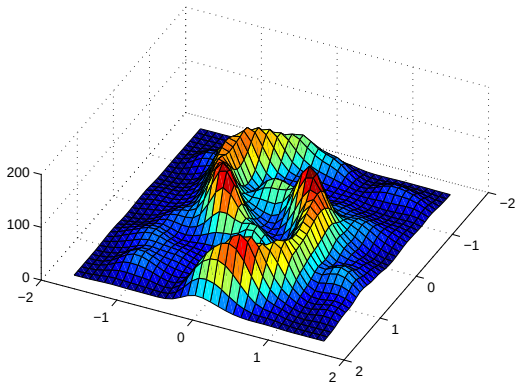
3.1.2 Outlier robustness of principal curves

Outlier robustness is another key issue in principal curve identification. Principal curve definitions that involve conditional sample expectations and mean squared projection error do not incorporate any data likelihood prior; hence, they treat each data sample equally. Such approaches are known to be sensitive to noise and in the presence of outlier data samples, of course, this will bias the results from the sought principal curve towards the outliers. Stanford and Raftery present results on outlier robustness along these lines by applying their principal curve clustering algorithm to reconnaissance images [119].

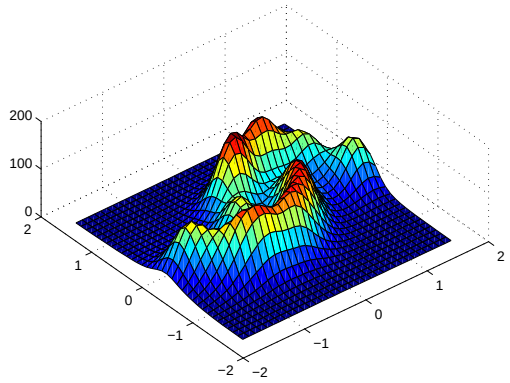
Outlier robustness is a well-known property of variable bandwidth KDE. In this approach, a data dependent kernel function is evaluated for each sample such



(a) The curve structures buried in noise



(b) Pdf of the data using KDE with a fixed kernel (selected according to Silverman's rule [116])



(c) Pdf of the data using KDE with a variable kernel bandwidth using the K-nearest neighbor distance with a global scale parameter optimized with maximum likelihood criterion

Figure 3.1: KDE with variable kernel width and outlier robustness

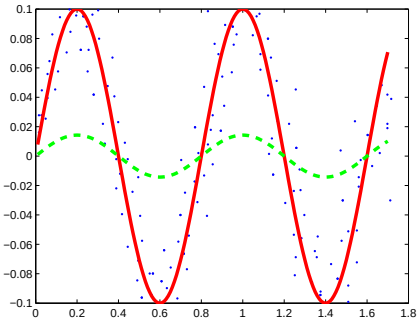
that the width of the kernel is directly proportional with the likelihood that sample is an outlier. This can be implemented in various ways, and the most commonly used methods are the K-nearest neighbor based approaches, namely: (i) the mean/median distance to the K-nearest neighbor data points, (ii) sum of the weights of K-nearest neighbor data points in a weighted KDE. Hence, the kernel bandwidth increases for the samples that are in a sparse neighborhood of data samples. At this point, note that to obtain an asymptotically unbiased and consistent density estimate the neighborhood parameter K should satisfy

$$\lim_{N\to\infty} K = \infty \ , \lim_{N\to\infty} \frac{K}{N} = 0 \tag{3.1}$$

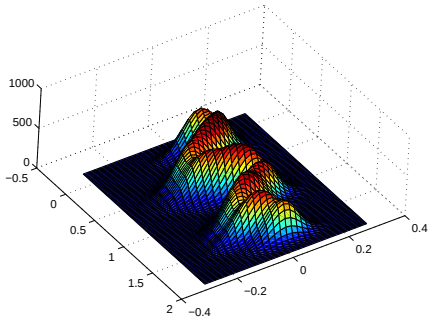
Typical choices for K are $K = \sqrt{N}$, or $K = \sqrt[4]{N}$. Depending on the application, this parameter can also be adjusted empirically.

Figure 3.1(a) presents a dataset consisting of two crescent-like clusters buried in noise. In fact, this dataset is similar to the illustration that Stanford and Raftery use as they propose their noise robust principal curve approach [119]. We present the fixed and variable bandwidth - using K-nearest neighbor method (i) mentioned above and selecting $K = N^{1/4}$- KDE of the dataset in Figure 3.1(b), and Figure 3.1(c), respectively.

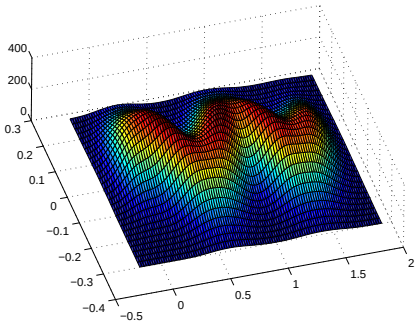
Note that in the resulting density estimate the variable size KDE eliminates the effects of the outliers without oversmoothing or distorting the pdf significantly in the support of the data. Selecting the kernel functions in a data dependent manner, by definition, makes our principal curve algorithms based on KDE robust to outliers in the data. Although the complexity of some of the above mentioned methods are negligible as compared to the principal curve projection step, depending on the method used, additional computational load of variable kernel bandwidth evaluations may increase the overall computational complexity.



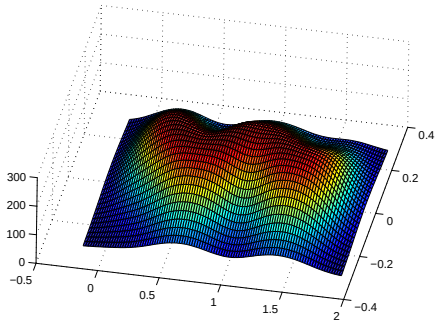
(a) The dataset and principal curves at different scales



(b) A narrow kernel that corresponds low projection error



(c) Maximum likelihood kernel bandwidth



(d) A wide kernel results in more generalization and more projection error

Figure 3.2: Mean projection error vs. overfitting as a kernel width selection

3.1.3 Mean projection error vs. overfitting

As in most unsupervised learning problems, principal curve algorithms have the problem of overfitting when they are defined over the sample expectations or minimization of the average projection error. However, note that overfitting is an issue that arises if the problem is defined in terms of a finite number of samples. On the other hand, by defining the principal curve fitting problem in terms of the data density, we assume that the required regularization constraints should be enforced by the preceding density estimation scheme directly.

Considering the trade-off between regularization and mean projection error, the original definition by Hastie-Stuetzle does not consider optimal regularization constraints. As mentioned before, there are many approaches in the principal curve literature to prevent overfitting, by constraining the length [75, 74] or curvature [111] of the principal curves. However, unfortunately it is not possible to obtain the optimal length or curvature from the data directly.

More explicitly, there are two main points here: (i) one may have to rerun the algorithm several times with different constraints and stopping conditions until the desired trade off between the projection error and overfitting is achieved (ii) for a particular result of the algorithm, it is not straightforward to decide about overfitting without visually observing the data. Both of the above two arguments become crucial while working in high-dimensional spaces. It is very hard to judge whether the desired trade-off between projection error and overfitting is reached, without observing the data, and this is why many principal curve applications in the literature are limited to two or three dimensional applications of the input domain.

Kegl brings up this question in his PhD dissertation. Considering the dataset and principal curve estimates in Figure 3.2(a), Kegl asks, which of the curves is the sought one. "Is the solid curve following the data too closely, or is the dashed curve generalizing too much?" [78] In general, of course, this is an open ended question that depends on the particular application. Still, with the connection to density

estimation, regularization methods that has been used in this field can provide nonparametric methods to define the regularization, where the related methods vary from heuristics to theoretically well-founded approaches like maximum likelihood. In other words, instead of trying for different length or curvature parameters, the connection to density estimation can provide purely data-driven approaches, where the regularization parameters are learned from the data directly.

Figure 3.2(b), Figure 3.2(c),and Figure 3.2(d) show regarding density estimates obtained using KDE for different kernel bandwidth selections for the dataset presented in Figure 3.2(a). Considering our KDE based algorithms, the trade-off between projection error and overfitting can be adjusted by setting the kernel width. One can select the kernel bandwidth manually by observing the data or exploiting the domain specific knowledge of the data. This is, of course, not much different than observing the data and selecting a suitable length or curvature constraint. However, the real advantage here is the rich literature on how to select the kernel function, and there are many theoretically well-founded ways of optimizing the kernel width according to maximum likelihood or similar criteria [116, 97, 26, 114, 66, 11, 102].

In KDE literature, perhaps the most common kernel optimization approach is to use the leave-one-out cross validation maximum likelihood procedure. Consider the random vector $\mathbf{x}$ with samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. The kernel density estimate that leaves the i^{th} sample out of density estimation is given by

$$p_i(\mathbf{x}) = \frac{1}{N} \sum_{j=1,\ j\neq i}^{N} K_{\Sigma_j}(\mathbf{x} - \mathbf{x}_j) \tag{3.2}$$

where $\Sigma_j = \sigma^2 \mathbf{I}$ and the objective of the kernel bandwidth optimization problem is defined by maximizing the log-likelihood function over all samples.

$$\max_{\sigma} \quad log \prod_{i=1}^{N} p_i(\mathbf{x}_i) \tag{3.3}$$

This optimization problem can be solved using a line search. Applying the maximum likelihood procedure to select data-oriented variable kernel bandwidths is also

straightforward. Let c_j^{KNN} denote the mean distance to K-nearest neighbor samples. One can select the variable bandwidth of the j^{th} sample as $\Sigma_j = \sigma c_j^{KNN}$, where σ is a global scale constant optimized using the ML procedure given above. An anisotropic counterpart of the isotropic selection method mentioned above can easily be obtained by defining $\mathbf{C}_i^{KNN}$ as the covariance of K-nearest neighbor data sampled instead of their mean distance, yielding $\Sigma_j = \sigma \mathbf{C}_i^{KNN}$.

Furthermore, anisotropic and/or variable size kernel functions naturally implement many types of constraints that cannot be defined by any length or bound of turn. By selecting anisotropic kernel functions, one can define the regularization constraint at different scales among different directions. Data-dependent variable bandwidth kernels will define varying constraints throughout the space, which can be estimated from the data directly, or manually adjusted according to specific needs depending on the application.

In summary, KDE not only connects the trade off between the projection error and generalization into well studied results of density estimation field, it also allows one to derive data-dependent constraints that vary throughout the space, which cannot be given by any length or curvature constraint whatsoever. Although this still cannot ultimately answer the open-ended question on the trade-off between the regularization and projection error, it provides a principled way to approach the problem in a nonparametric manner. We will provide results for fixed and variable maximum likelihood kernel bandwidth, which yields purely nonparametric principal curve algorithms, and maximum likelihood selection proves to be efficient in many practical applications. All arguments above also apply to parametric density models.

3.2 Eigenspace-constrained Mean Shift (ECMS)

A natural consequence of Theorem 2.1 is that a point is on the one dimensional critical set iff the local gradient is an eigenvector of the local Hessian, since the

gradient has to be orthogonal to the other $(n-1)$ eigenvectors. Furthermore, for this point to be on the principal curve, the corresponding $(n-1)$ eigenvalues must be negative.

By using KDE, a modification of the mean-shift algorithm by constraining the fixed-point iterations in the orthogonal space of corresponding $(n-1)$ eigenvector directions of the at the current point in the trajectory leads to an update that converges to the principal curves and not to the local maxima. For this case, the orthogonal space of corresponding $(n-1)$ eigenvector directions of the local covariance is the parallel space of the leading eigenvector of the local covariance. The algorithm could be modified to converge to the d-dimensional principal manifold P^d in a conceptually trivial manner, by selecting the constrained subspace as the subspace spanned by corresponding $(n-d)$ eigenvectors of the local covariance to constrain the mean-shift iterations into the subspace spanned by d leading eigenvectors of the local covariance.

Consider the data samples $\{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \Re^n$. The KDE of this data set (using Gaussian kernels) is given as

$$p(\mathbf{x}) = (1/N) \sum_{i=1}^{N} G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \tag{3.4}$$

where Σ_i is the kernel covariance for $\mathbf{x}_i$; $G_{\Sigma_i}(\mathbf{y}) = C_{\Sigma_i} e^{-\mathbf{y}^T \Sigma_i^{-1} \mathbf{y}/2}$. Note that for the derivation we use the general case of anisotropic variable kernel functions. For isotropic kernels one can use a scalar value instead of a full covariance, or for fixed kernel functions one can constrain the data dependency and drop the sample index i. Again for the general case, the gradient and the Hessian of the KDE are

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= -N^{-1} \textstyle\sum_{i=1}^{N} c_i \mathbf{u}_i \\ \mathbf{H}(\mathbf{x}) &= N^{-1} \textstyle\sum_{i=1}^{N} c_i (\mathbf{u}_i \mathbf{u}_i^T - \Sigma_i^{-1}) \\ \text{where} \quad \mathbf{u}_i &= \Sigma_i^{-1}(\mathbf{x} - \mathbf{x}_i) \quad \text{and} \quad c_i = G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \end{aligned} \tag{3.5}$$

Let $\{(\gamma_1(\mathbf{x}), \mathbf{v}_1(\mathbf{x})), \ldots, (\gamma_n(\mathbf{x}), \mathbf{v}_n(\mathbf{x}))\}$ be the eigenvalue-eigenvector pairs of $\mathbf{\Sigma^{-1}}(\mathbf{x})$

as defined in Definition 2.5 and the mean-shift update emerging from (3.5) is

$$\mathbf{x} \leftarrow \mathbf{m}(\mathbf{x}) = (\sum_{i=1}^{N} c_i \Sigma_i^{-1})^{-1} \sum_{i=1}^{N} c_i \Sigma_i^{-1} \mathbf{x}_i \tag{3.6}$$

At $\mathbf{x}$, the subspace mean-shift update is performed by projecting x into the constrained space $\tilde{\mathbf{x}}_k = (\mathbf{v}\mathbf{v}^T\mathbf{m}(\mathbf{x}))$. This nothing but the general approach of update and back-project that has been widely used in constrained optimization.

The stopping criterion can be selected according to the definition directly, which checks if the gradient is an eigenvector of the Hessian. Alternatively, the stopping criterion can be selected similar to mean-shift, where the magnitude of the update is used. This gives a computationally cheaper stopping criterion, since it does not use the Hessian, and the algorithm stops if the updates get smaller than a reasonable tolerance, $\|\mathbf{V}\mathbf{V}^T\mathbf{m}(\mathbf{x}(k)) - \mathbf{x}(k)\| < \epsilon$.

The iterations can be initialized any arbitrary location in the feature space, if the principal curve in a region of interest is sought. To find the principal curve projections of the data samples as well as the curve itself, the suitable way is to initialize the algorithm to the data samples themselves, as in mean-shift. The generalized version of ECMS algorithm that converges to the d-dimensional principal manifold is presented in Table 3.1, where the regarding principal curve algorithm can simply be obtained by setting $d = 1$.

‘

3.3 Subspace-constrained Mean Shift (SCMS)

The bottleneck of the computational load in ECMS is in the step 3 (in Table 3.1), where the Hessian and local covariance matrices, and the eigendecomposition of the local covariance matrix is evaluated. If the pdf is a flat ridge, the eigenvectors of the local covariance does not significantly change throughout the iterations. Exploiting this, one simplification over the ECMS algorithm is to perform this eigendecomposition at the initial point and keep the constrained space constant until convergence.

Table 3.1: ECMS Algorithm

1. Initialize the trajectories to a mesh or data points and $t = 0$.
2. For every trajectory evaluate $\mathbf{m}(\mathbf{x}(t))$ as in (3.6).
3. Evaluate the gradient, the Hessian, and perform the eigendecomposition of $\mathbf{\Sigma^{-1}}(\mathbf{x}(t)) = \mathbf{V\Gamma V}$.
4. Let $\mathbf{V}_{=}[\mathbf{v}_1 \dots \mathbf{v}_d]$ be the $(d-m)$ leading eigenvectors of Σ^{-1}
5. $\tilde{\mathbf{x}}(k) = \mathbf{V}\mathbf{V}^T\mathbf{m}(\mathbf{x})$
6. If $|\mathbf{g}^T\mathbf{H}\mathbf{g}|/\|\mathbf{g}\|\|\mathbf{H}\mathbf{g}\| > threshold$ then *stop*, else $\mathbf{x}(t+1) \leftarrow \tilde{\mathbf{x}}$.
7. Convergence is not achieved. Increment t and go to step 2.

Also the stopping criterion is different than ECMS, so that the Hessian evaluation is eliminated. The resulting Subspace-constrained Mean Shift algorithm is given in Table 3.2.

3.4 Piecewise Linear SCMS

Although SCMS gives the advantage of lower computational load as compared to ECMS with accurate results, the dependency on number of data samples is still quadratic. The way to get a lower data dependency is to select a representative subset of points for principal curve projection, and use these projections to interpolate the curve. Principal curve literature shows that in many cases piecewise linear structures can provide significantly accurate results; and therefore, in this section we will present a piecewise linear approach that lowers the computational load.

We propose a two step solution to find the representative points of the dataset to be used in principal curve projection: (i) find the modes (local maxima points) of the pdf (ii) combine the principal curve segment in between pairs of these modes. The modes of the pdf constitute the zero-dimensional principal set P^0, and we use

Table 3.2: SCMS Algorithm

1. Initialize the trajectories to a mesh or data points and $t = 0$.
2. Evaluate the gradient, the Hessian, and perform the eigendecomposition of $\mathbf{\Sigma^{-1}}(\mathbf{x(0)}) = \mathbf{V\Gamma V}$.
3. For every trajectory evaluate $\mathbf{m}(\mathbf{x}(t))$ as in (3.6).
4. Let $\mathbf{V}_{=}[\mathbf{v}_1 \dots \mathbf{v}_d]$ be the $(d-m)$ leading eigenvectors of Σ^{-1}
5. $\tilde{\mathbf{x}}(k) = \mathbf{V}\mathbf{V}^T\mathbf{m}(\mathbf{x})$
6. If $\|\mathbf{V}\mathbf{V}^T\mathbf{m}(\mathbf{x}(k)) - \mathbf{x}(k)\| < \epsilon$ then *stop*, else $\mathbf{x}(k+1) \leftarrow \tilde{\mathbf{x}}$.
7. Convergence is not achieved. Increment t and go to step 3.

the mean-shift iterations given in (3.6) to find the modes.

The next step is to construct P^1 by combining principal curve segments from these modes. For this, we use a subset of points for each principal curve segment. We select the subset of points to be projected in a sequential way, very similar to Kegl's polygonal line algorithm: for each pair of mode, we will project the midpoint of the line segment that connects two projected points in the piecewise linear approximation. This can be performed up to a *depth* of m, leading to $2^m + 1$ points in between every pair of mode. Table 3.3 summarizes the PL-SCMS algorithm.

For a simple illustration of the proposed piecewise linear approach for different depth values we use a semi-circle shaped dataset. This dataset has 500 samples, drawn from a uniform distribution along a circle with a Gaussian perturbation in the radial direction. Figure 3.3a shows the data samples (blue) along with the modes (green), that is the result of step 1 in Table 3.3, and leads to the principal curve approximation of depth 0. Figure 3.3b, 3.3c, and 3.3d presents the principal curve approximations of depth 1, 2, and 4, respectively. The piecewise linear principal curve approximations are shown with green.

The piecewise linear approximations provided by our approach is similar to the

Table 3.3: PL-SCMS Algorithm

1. Select the kernel size and run mean-shift iterations in (3.6), to find the modes
2. For all mode pairs project the midpoint of the line that connects these modes onto the principal curve by the following:
   - Evaluate the local covariance $\Sigma^{-1}(\mathbf{x})$ given in (6)
   - Find the greatest eigenvector of the local covariance $\mathbf{v}$, select the constrained subspace direction as either $\mathbf{v}$ or -$\mathbf{v}$, depending on which of these has a positive inner product with the gradient given in (6): $\mathbf{d} = \mathbf{v}\ sign(\mathbf{g}^T\mathbf{v})$
   - Until convergence: iterate (3.6), and project the update onto the constrained subspace direction $\mathbf{d}$.
3. Go back to step 2 if the required depth is not reached.

outcome of Kegl's piecewise linear apporach [75]; however, the difference is that once you estimate the probability density of the data, the principal curve estimate will not overfit the data no matter how dense you would like to populate the points on the principal curve. The irregularities like loops, self intersections, and bifurcation points do not require special attention and are handled naturally by the definition. Other than the kernel bandwidth which can be learned from the data, the resulting algorithm has only a single parameter, which is the depth that is used to control the trade off between accuracy and computational load.

In summary, the piecewise linear subspace constrained mean shift provides a computationally much cheaper accurate estimate of the principal graph that we define. In the cases where the projection of all data samples onto the principal curve is required like in denoising, compression, and dimensionality reduction, this approach might be insufficient. On the other hand, this approach provides accurate results for OCR feature extraction as we will present in the next chapter.

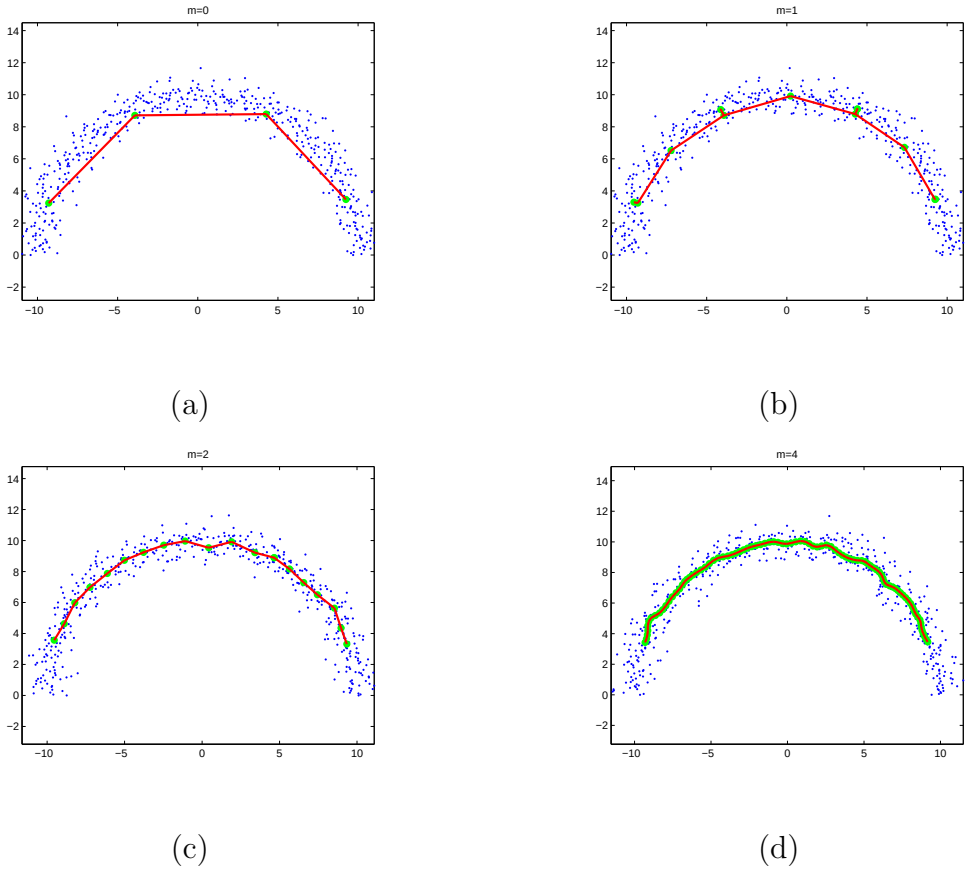


Figure 3.3: Principal curve approximations on the semicircle dataset with depths 0, 1, 2, and 4.

3.5 ECMS with Vector Quantization

We will present another cheap algorithm that uses the principal curve projections of a subset of points, similar to above piecewise linear approach. The construction of the principal curve is composed of three main parts: (*i*) selecting representative points from the data (*ii*) projecting these points onto the principal curve (*iii*) building the neighborhood graph over the pairwise distances of these principal curve projections.

There are two important points for the selection of the representative points from the dataset. First, this selection should be performed in a computationally

inexpensive manner, and second, the selected points are uniformly distributed in the support of the data. The cheapest way to achieve this is to use a subsampling of the data under the assumption that the data distribution is iid. On the other hand, although it brings some additional computational load, using a vector quantization or clustering algorithm relaxes the iid assumption. Many clustering or vector quantization algorithms can be used here for this purpose. We select K-means algorithm due to its simplicity and relatively low computational complexity [56].

Similar to piecewise linear SCMS in Section 3.4, we will use the SCMS algorithm to project the K-means of the dataset onto the principal curve. Note that in piecewise linear SCMS, since each curve segment is refined sequentially, the neighborhood of the points that are used in the principal curve approximation are given by definition. On the other hand, this information is not available in vector quantized SCMS approach, and a neighborhood graph over the projected points has to be constructed. Minimum spanning tree (MST) algorithm offers a robust way for constructing the neighborhood graph, and provides good results if the dataset does not have loops. If the data has loops, more elaborate approaches are required, and evaluating an ensemble of MST for slightly perturbed (by a Gaussian noise) versions of these points [14], or the postprocessing steps of Kegl's principal graph algorithm [74] can be used here.

As well as the graph construction method, the distance metric is also very important. Euclidean distance provides good results if the noise around the true underlying principal curve is low, and points that are used in the principal curve projection step are dense enough. On the other hand, density geodesics [94] provide a better alternative that provides accurate results in noisy data as well. Details of density geodesics is presented in the Appendix, and implementation of vector quantized SCMS is given in Table 3.4.

Table 3.4: VQ-SCMS Algorithm

1. From the dataset, select a subset of points x^* either by subsampling, clustering or vector quantization.
2. Project x^* onto the principal curve using SCMS to obtain $\widetilde{x}^*$
3. Build pairwise distance matrix of $\widetilde{x}^*$, and construct the MST of $\widetilde{x}^*$ to obtain the principal tree.

3.6 Comparisons and Experimental Results

Here we will show some experimental results of above algorithms, and provide comparisons between them and also with Kegl's popular Polygonal Line algorithm [75]. We already presented comparisons with this approach in the previous chapter. Here we take a step further where we provide quantitative results for different noise level and compare the computation times as well.

For illustrative purposes we use a two-dimensional dataset here. The noiseless data points are sampled from a spiral, and additive Gaussian noise is added to these points along the orthogonal (that is the radial) direction. Figure 3.4 shows the dataset for different noise levels. Results for ECMS, SCMS, and Polygonal line algorithm is presented in Table 3.5 along with corresponding running times. ECMS and SCMS use ML fixed isotropic ML kernel bandwidth, and presented running times include the ML training as well. For the Polygonal Line algorithm we performed a manual parameter tuning for each noise level and best results are presented.

Overall, as the noise level increases, the computation time of ECMS and SCMS both increase, presumably due to more iterations are required. Still, the computation time of both of these algorithms are much less as compared to Polygonal Line algorithm. In terms of MSE between the estimated and the true curve, ECMS and SCMS provide better performance as compared to Polygonal Line algorithm,

and for some noise levels the difference in performance is very small. However, the real advantage is being able to reproduce the similar/better results with no manual parameter tuning.

Table 3.5: Computation Time and MSE Performance Comparisons

	Computation time	Mean squared projection error	σ_{noise}
ECMS	3.237 sec.	0.00003184	0.005
SCMS	2.028 sec.	0.00003265	0.005
PL	18.422 sec.	0.00017677	0.005
ECMS	3.508 sec.	0.00011551	0.01
SCMS	2.143 sec.	0.00013057	0.01
PL	20.547 sec.	0.00024497	0.01
ECMS	3.986 sec.	0.00062832	0.02
SCMS	2.367 sec.	0.00064603	0.02
PL	22.671 sec.	0.00066665	0.02
ECMS	6.257 sec.	0.00194560	0.04
SCMS	4.198 sec.	0.00198021	0.04
PL	27.672 sec.	0.00269184	0.04
ECMS	7.198 sec.	0.00433269	0.06
SCMS	4.555 sec.	0.00441386	0.06
PL	19.093 sec.	0.00618819	0.06
ECMS	8.813 sec	0.00912748	0.08
SCMS	5.183 sec	0.01297617	0.08
PL	19.719 sec	0.01883287	0.08

3.7 Limitations, Finite Sample Effects, and the Curse of Dimensionality

In our perspective, density estimation is a prerequisite step for principal curve estimation. Since our principal curve definition assumes the pdf to be given, it depends on the reliability of the preceding density estimation step, which in general may not be an easy task. Stated by Bellman as *the curse of dimensionality* [7], it is a very well-known fact that density estimation becomes a much harder problem as the

dimensionality of the data increases. Therefore, before we move on to applications on real data, in this section we will present the performance of our principal curve fitting results for various density estimates with different number of samples and dimensions.

The first comparison is with the principal line estimation based on eigendecomposition of the data, where the true underlying probability distribution is Gaussian. The second comparison examines the model order estimation using a Gaussian mixture model, which in the limiting case, where the number of Gaussian mixtures is equal to the number of samples, converges to KDE. In all comparisons presented below principal curve projections are obtained by the SCMS algorithm given in Table 3.2, using the ML kernel bandwidth detailed in Section 3.1.3.

3.7.1 Comparison with eigenvector estimation

As mentioned before, the reason why we prefer to use KDE is its ability to adapt different complex shapes that data may take. Indeed, in the next chapter, KDE based principal curve estimation proves to be efficient in adapting many real-life data distributions of a diverse set of applications. However, one well-known disadvantage of KDE is the required number of samples as the dimensionality of the data increases. Here we discuss the case where the true underlying probability density is Gaussian; hence, the claim of *the requirement to adapt to complex shapes in the data* is an obvious overstatement. In this scenario, we will compare the principal line estimator based on PCA to the principal curve based on KDE, for different number of dimensions.

Consider the dataset $\{\mathbf{x}_{i=1}^{N}\}$ Gaussianly distributed in d-dimensional space, where $\mathbf{v}$ denotes the true principal line of this distribution, and $\mathbf{v}_*$ denotes the principal line obtained by sample PCA. What we are going to compare here is the following:

1. mean squared distance between the projection of the data samples onto the true first eigenvector and the estimated first principal component, $E\{\|\mathbf{v}^T\mathbf{x} -$

$\mathbf{v}_*^T\mathbf{x}\|\}$

2. mean squared distance between the projection of the data samples onto the true eigenvector and the principal curve projection $\widetilde{\mathbf{x}}$, $E\{\|\mathbf{v}^T\mathbf{x} - \widetilde{\mathbf{x}}\|\}$.

Figure 3.5 presents the MSE the principal line (red) and principal curve (blue) projections for 2, 3, 4, 5, 10, 20, 30,and 40 dimensions, and average MSE for 100 Monte Carlo simulations is shown. For all cases the MSE decreases for both methods as the number of samples increase. Principal line projection always results in better accuracy and the performance of principal curve projections drop exponentially for increasing dimensions.

3.7.2 Effects of the model order estimation

An important problem in parametric density estimation is model order selection. At this point, to develop general purpose algorithms and to avoid model order selection problems in the underlying density estimate, we work on the KDE of the data; however, note that all algorithms presented in this chapter can easily be modified to work over Gaussian mixtures, instead of KDE. Since we use a Gaussian kernel function in KDE-based implementation already, the required modification is as simple as replacing the data samples with the Gaussian mixture means and incorporating the individual weights and covariances. Using a parametric model has two advantages:

1. As opposed to $O(N^2)$ complexity of the KDE based SCMS and ECMS, the computational complexity of these approaches would decrease to $O(MN)$, where M is the number of mixtures in the Gaussian mixture and N is the number of samples, since typically $M \ll N$.

2. As also implied in the previous section, a parametric approach with a *suitable* model order and a *correct* fit, the algorithm would need less samples to achieve good principal curve estimates.

Here we will evaluate the stability of principal curve estimation for improper model order selections in the density estimation step, and compare the principal curve projection results for a Gaussian mixture with 3 components. For this purpose we use ECMS algorithm and modify it accordingly such that the underlying density estimate is a Gaussian mixture. Since the true underlying density is known to have 3 components, we measure the performance as of principal curve projection results for different number of components in the density estimate as the distance to the principal curve projections obtained with three components

$$J_d = E\{\|\widetilde{\mathbf{x}}_3(\mathbf{x}) - \widetilde{\mathbf{x}}_d(\mathbf{x})\|\}, d = 1, 2, 3, 4, 5, 6, 10, 15, 25, 50, 100, 200, 400 \qquad (3.7)$$

where the dataset $\mathbf{x}$ has 400 samples in 2-dimensional space. Figure 3.6 shows a realization of the Gaussian mixture, and Figure 3.7 presents the performance of the principal curve projections for different number of components in the Gaussian mixture estimation, and results of 50 Monte Carlo simulations is shown. Note that for increasing model orders, if the GMM has more number of components than the true underlying distribution, the generalization performance of the principal curve does not change significantly.

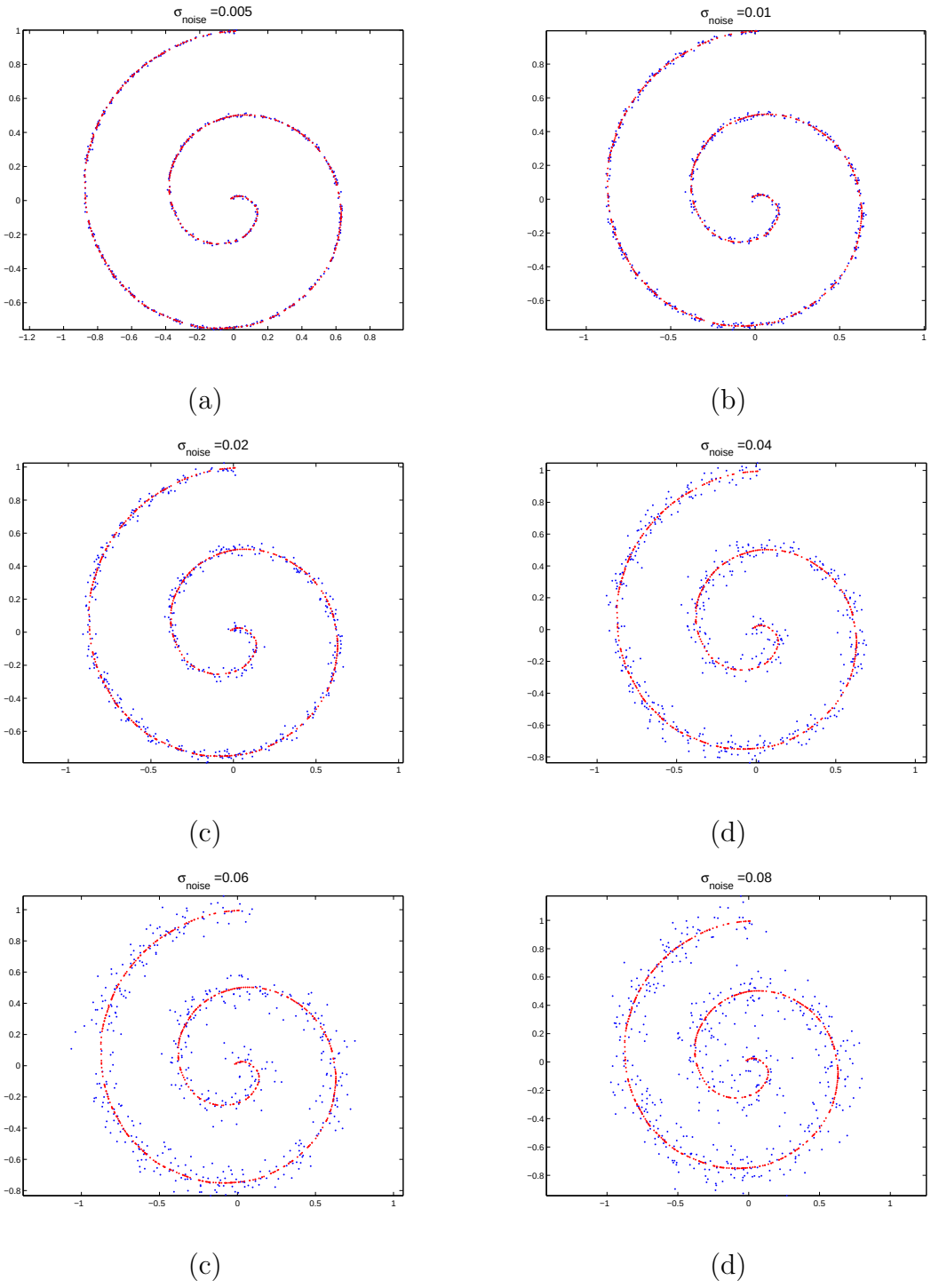


Figure 3.4: Sprial dataset (red) with its noisy versions (blue) for different noise levels.

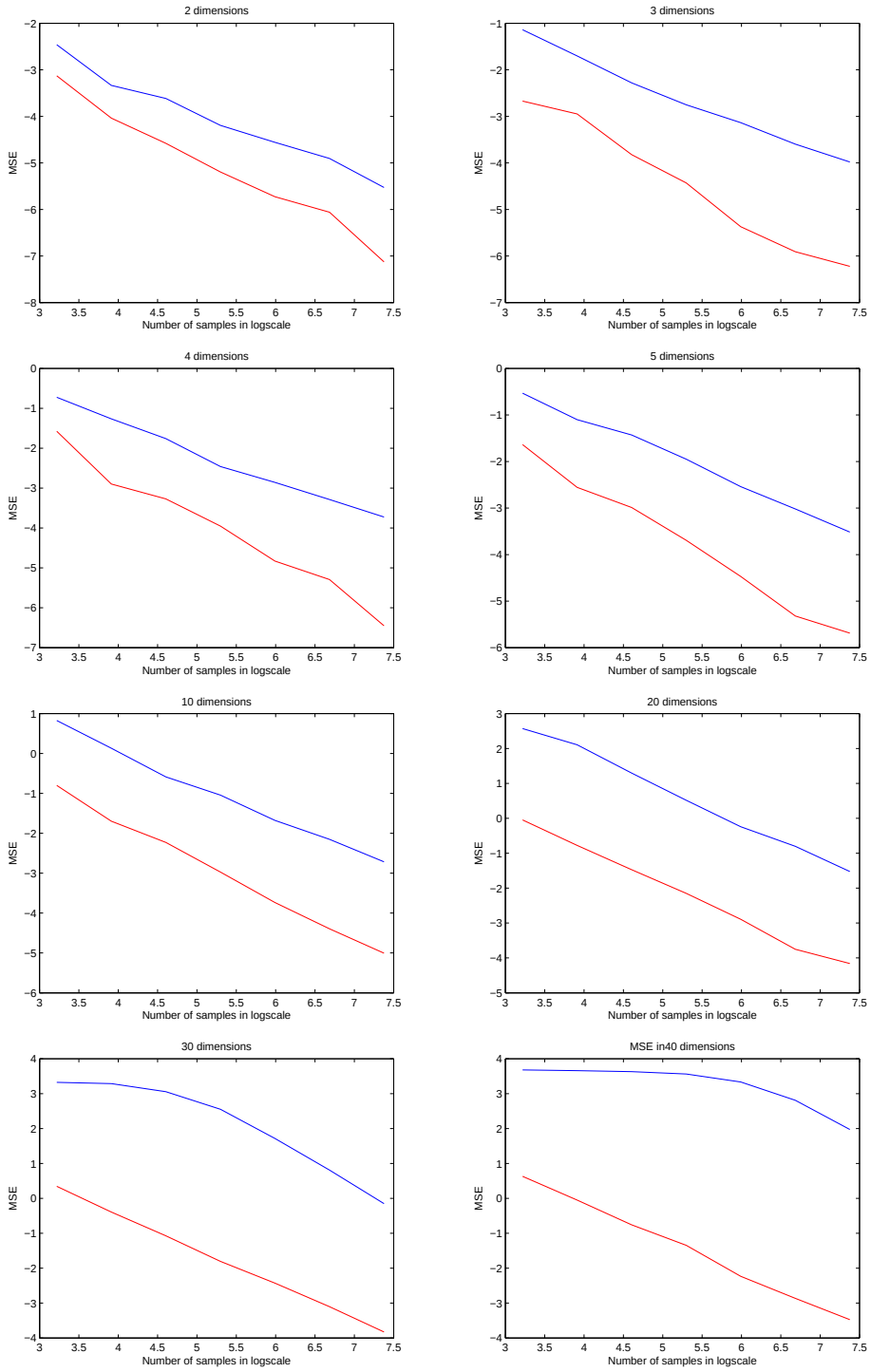


Figure 3.5: Mean projection error in log scale for principal line (red) and principal curve (blue). Average of 100 Monte Carlo simulations is shown.

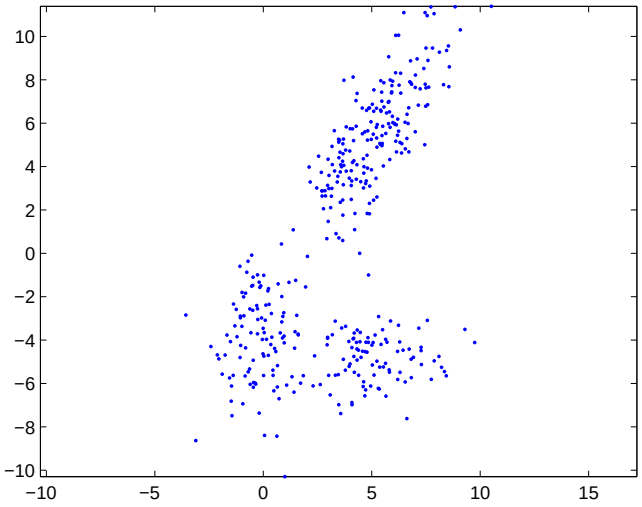


Figure 3.6: One realization of the 3-component Gaussian mixture data used in performance evaluations

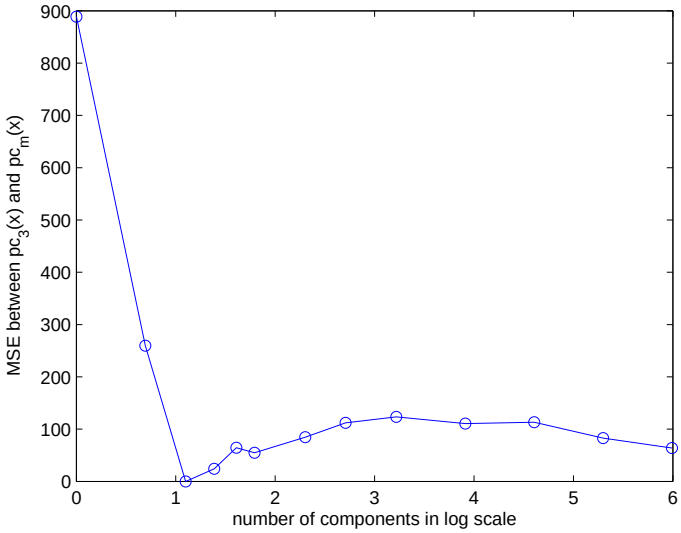


Figure 3.7: Principal curve projections for different number of components in the density estimate. Specifically, $d = 1, 2, 3, 4, 5, 6, 10, 15, 25, 50, 100, 200, 400$.

Chapter 4

Applications of Principal Curves

In this chapter, we will present applications of principal curves on a diverse set of problems, including image segmentation, optical character recognition, signal denoising, neighborhood graph construction, manifold unfolding, time-frequency reassignment and multiple-input multiple-output channel equalization. Note that the applications presented here use the algorithms presented in Chapter 3 directly, and they do not use any domain specific information. Our aim here is to show the applicability of the principal curve concept and the effectiveness of the general purpose algorithms presented in Chapter 3 into practical applications, rather than developing techniques specifically designed for a particular purpose. If necessary, preprocessing and postprocessing methods that exploit the domain specific information -like the differences between the polygonal line algorithm [75] and the principal graph algorithm [74]- can always be added to improve the performance.

This chapter starts with a literature review of existing principal curve applications in the literature. As well as existing applications of principal curves like optical character skeletonization, we will also present new application areas that principal curve based approaches have never been used before. For each application, a brief literature review will be presented. However, note that these are intended to provide a general understanding of current state-of-the-art, rather than providing a complete survey of these areas.

4.1 Principal Curve Applications in the Literature

The Hastie-Stuetzle principal curve algorithm was designed to be used in Stanford Linear Collider project [57]. The aim of the project was to discover new subatomic particles by colliding two intense and focused particle beams. There are two colliders of about 3 km length, and hundreds of magnets that guide the positron and electron beams. The principal curve approach is used to align misplaced magnets and reduce the total magnet movement required.

Banfield and Raftery used principal curves for identification of ice floes and their outlines in satellite images [4]. This identification is important to investigate the physical processes in the polar regions, and also for transportation in these areas. Banfield and Raftery model ice floes as closed principal curves and provide a robust method for estimating closed principal curves.

Stanford and Raftery apply their noise robust principal curve algorithm to detection of curvilinear seismic regions in reconnaissance images [119]. For this they use a two step approach: a hierarchical and agglomerative clustering method, and an EM algorithm that refines the principal curve clustering results and handles the background noise.

Kegl and Krzyzak provide handwritten character skeletonization results using the Principal Graph algorithm, which is an extension of the Polygonal Line algorithm [74]. The skeletons can be used for two purposes; feature extraction for optical character recognition or compression. Since skeleton is supposed to capture the overall structure of the character, it can be used as a feature set for automatic character recognition. Using the skeleton for compression is based on the idea that the handwritten characters are not observed pixel by pixel, but as a whole. Hence, the principal graph of the data can be used for this purpose as well.

Reinhard and Niranjan use the Hastie-Stuetzle principal curve algorithm to

model short term frequency spectrum of speech signals, and they compared the principal curve based approach with Hidden Markov Models and Generative Topographic Mapping [103, 104]. Their approach used principal curves to obtain a parametric modelling of the trajectory in the time-frequency space, which they used to generate a distance metric for diphone classification.

Glenn De'ath used the Hastie-Stuetzle Algorithm for indirect gradient analysis of multispecies abundance data for estimating the species response curves [31]. De'ath reports that the principal curve based technique outperforms correspondence analysis and multidimensional scaling, and provides results with and without noise.

Hermann, Meinicke, and Ritter used principal curves for sonification [61]. Sonification is the represention of high-dimensional data as a time-varying sound signal. As well as capturing the structure in the high-dimensional space, the principal curve is also used to define a time order for these high-dimensional data points.

Dong and McAvoy combined the principal curve method with an autoassociative neural networks-based algorithm to build a model that can be used for prediction [36]. They applied their algorithm to process control. Wilson and colleagues used radial basis functions to build self-consistent principal manifolds, and they applied their method to process monitoring and fault detection.

Chen and colleagues used principal curves to analyze freeway traffic streams [19]. Most of the prior work in this field is based on functional forms of traffic stream variables, whereas the proposed principal curve based method offers a nonparametric alternative. Comparisons show that the principal curve based model is superior to earlier modelling methods.

4.1.1 Image Segmentation and Active Contours

Image segmentation is one of the fundamental problems in image processing. The most common image segmentation applications include feature extraction from images, filtering of noisy images, object recognition, and object based video or image

coding. Definition of the image segmentation problem is as follows: partition the image into distinct regions in a way that these regions are homogenous, and none of the unions of these distinct regions form a homogenous structure. The homogeneity can be defined in any feature space; hence, it does not strictly require homogeneity in the color or the intensity space.

Existing image segmentation methods can be categorized into few main groups. The earliest methods are the *edge-based approaches* [13]. Edge-based approaches first detect the edges of the image, which are subsequently connected and combined to build object contours. However, these methods are efficient only if the pixel intensity itself is a suitable feature for segmentation; the main disadvantage of the edge-based methods is the fact that they are applicable only when the edges can be computed, which is not always possible. There are *region-based approaches*, like region growing, that have the advantage of low computational cost [50]. However, region-based methods are parametric, and results of these methods suffer from being very sensitive to the parameter values. Another category is *split and merge approach* that partitions the image into primitive regions and merges them to provide the final results. A similarity measure is defined to compare similarities between the pairs of neighboring regions to merge them until a stopping criteria is ensured [95]. The problem of setting good thresholds and selecting an effective stopping criterion is mostly overcome by utilization of clustering methods for image segmentation problems. *Clustering-based methods* bring the strength of generalization properties of unsupervised learning problems into the image segmentation area [27, 115]. These approaches are generally nonparametric or have a few parameters, whose efficient values can usually be estimated from the data.

Another approach that has evolved from edge based approaches is the active contours, so called *snakes* [72, 84, 131]. Snakes are based on the utilization of the shape priors with the gradient of the edge map of the image. Active contours move with the effect of internal and external forces. Internal forces, such as elasticity forces

or bending forces, depend on the contour itself; the external forces are independent of the shape of the contour and are only evaluated from the image intensities. These forces are determined in a way that they will make the snake move towards the object boundaries. The external energy function is defined such that it takes its optimal (generally minimum, depending on the definition of the optimization problem) values at the object boundaries, while the internal energy function preserves the shape of the snake. Generally, in parametric and geometric active contours, edge maps or some derivatives of edge maps are used to define the external energy function. A different track in active contours is the *region-based active* contours [105], which are based on the idea of making the contour move towards a boundary, such that both the interior and the exterior of the contour form homogenous regions.

There are several practical difficulties in using snakes, with the original definition, most of which are later addressed by the *gradient vector flow* (GVF) snake [131]. The most important problem is that the initialization of the active contour has to be in a narrow neighborhood of the object boundary. The range of initializations that leads to the desired segmentation is known as the *capture range* of the corresponding external force field. To overcome the low capture range problem in real life scenarios, snakes are usually initialized using the output of another segmentation algorithm, in which case, snakes can be regarded as a fine tuning step for another segmentation algorithm. To increase the capture range, several methods have been proposed [25, 10, 24]. Determining the way the snake should move through varying resolution levels remains unsolved. Another approach to solve the same problem is *distance potentials method* proposed by Cohen and Cohen [24], which introduces an external force model to guide the contour towards the object boundary. This approach significantly increases the capture range and overcomes the initialization difficulty in most of the cases.

Another well-known problem is that the snakes have difficulties in progressing into concavities along the boundary. Directional attractions [1], control points [30],

pressure forces [25] are among the methods proposed to solve this problem. However, the most satisfactory results are obtained with the GVF formulation by Xu and Prince [131], which also solves the capture range issue very effectively. *GVF snakes* provide a principled and effective way of defining the external field, providing an insensitivity to initialization and an ability to progress into boundary concavities. However, the method of defining the internal energy function with the optimal selection of its parameters remains to be determined. As with most parametric methods, the usual way of seeking the desired result is to run the algorithm several times for a set of different parameter values until a satisfactory performance is obtained.

In Nonparametric Snake formulation, we approached the problem of defining the energy function in a nonparametric way. The approach is based on the density estimate of the edge image. Here, we mainly focused on kernel density estimation (KDE) [97], and derived an algorithm that exploits the underlying kernel density estimate of the edge image.

4.1.2 Nonparametric Snakes

In this section, we will briefly review our Nonparametric Snake formulation. In this particular realization of the concept, we use kernel density estimation, Euclidean inner product, and fixed point iterations. A discussion of how to address typical problems encountered by active contours in the proposed nonparametric framework is very similar to the discussion in 3.1 [93] and is omitted here.

Consider an image $I(x, y)$. For each pixel, the vector $\mathbf{s} = [x, y]^T$ denotes its location. Let $E(\mathbf{s})$ denote the edge map of this image as a function of space. The edge maps can be obtained using any suitable edge detector of choice and the edge values can be binary or continuous.

In order to illustrate the edge distribution estimates obtained using KDE, consider the more intuitive case of a binary edge map $E(\mathbf{s})$ such that

$$E(\mathbf{s}) = \begin{cases} E(\mathbf{s}_i) = 1 & : \quad \mathbf{s}_i \textit{ is an edge pixel} \\ E(\mathbf{s}_i) = 0 & : \quad \textit{otherwise} \end{cases} \tag{4.1}$$

A KDE of the edge map can be constructed using variable-anisotropic-bandwidth kernels as follows:

$$p_{edge}(\mathbf{s}) = \frac{1}{N_{edge}} \sum_{i=1}^{N} E(\mathbf{s}_i) K_{\Sigma_i}(\mathbf{s} - \mathbf{s}_i^{edge}) \tag{4.2}$$

where N is the number of pixels and

$$N_{edge} = \sum_{i=1}^{N} E(\mathbf{s}_i); \tag{4.3}$$

Now suppose we have a continuous edge map $E_{cont}(\mathbf{s}) \in [0, \infty)$. The most common choices for a continuous edge map include: (i) a binary edge map convolved with a Gaussian, (ii) the magnitude-square of the gradient field, (iii) the magnitude-square of the gradient field convolved with a Gaussian. Explicitly, these choices are

$$E_{cont}(\mathbf{s}) = \begin{cases} (i) & : \quad E_{binary}(\mathbf{s}) * G(\mathbf{s}) \\ (ii) & : \quad \|\nabla I(\mathbf{s})\|^2 \\ (iii) & : \quad (ii) * G(\mathbf{s}) \end{cases} \tag{4.4}$$

For continuous edge maps, the density can be expressed as a variable-weight variable-bandwidth KDE as

$$p_{edge}(\mathbf{s}) = \sum_{i=1}^{N} w_i K_{\Sigma_i}(\mathbf{s} - \mathbf{s}_i) \tag{4.5}$$

where the weights w_i are obtained for each pixel as its normalized edge map value:

$$w_i = \frac{E(\mathbf{s}_i)}{\sum_{j=1}^{N} E(\mathbf{s}_j)} \tag{4.6}$$

Note that the binary edge map density is a special case of the latter, obtained by thresholding the weights appropriately to obtain binary weights for each pixel.

Given the active contour samples $\{\mathbf{s}_j^{snake}\}_{j=1}^{N_{snake}}$ and the density of the edge map $p_{edge}(\mathbf{s})$, our aim is to find a contour that *captures the structure* of the edge field $E(\mathbf{s})$ around the object to be segmented. We formulate this idea as maximizing the inner product between the probability density function of the snake $p_{snake}(\mathbf{s})$ and the probability density of the edge map $p_{edge}(\mathbf{s})$:

$$\max_{\{\mathbf{s}^{snake}\}} J(\{\mathbf{s}^{snake}\}) = \max \int p_{edge}(\mathbf{s})\ p_{snake}(\mathbf{s})\ d\mathbf{s} \tag{4.7}$$

where the probability density of the snake $p_{snake}(\mathbf{s})$ is also evaluated as a KDE, using the samples of the snake:

$$p_{snake}(\mathbf{s}) = \frac{1}{N_{snake}} \sum_{j=1}^{N_{snake}} K_{\Gamma_j}(\mathbf{s} - \mathbf{s}_j^{snake}) \tag{4.8}$$

where N_{snake} is the number of points on the snake. Substituting (4.8) and (4.5) into (4.7), the plug-in KDE-based estimate of the objective function is obtained:

$$J(\mathbf{s}^{snake}) = \sum_{j=1}^{N_{snake}} \sum_{i=1}^{N} \frac{w_i}{N_{snake}}\ K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{snake}) \tag{4.9}$$

Note that this cost function is additive in terms of the samples of the snake, therefore near the optimal point along the ridge of the edge density, higher and lower sampling rates of the snake would lead to an accordingly denser or sparser evaluation of the optimality criterion along the ridge.

To derive a fixed-point iteration for the samples of the snake, we equate the gradient of the optimization criterion by using the fact that for any fixed point of the density inner product cost function, the gradient of the inner product with respect to $\mathbf{s}^{snake}$ should be equal to zero. This yields the following:

$$\frac{\partial J(\mathbf{s})}{\partial \mathbf{s}_j^{snake}} = \sum_{i=1}^{N} \frac{w_i(\Sigma_i + \Gamma_j)^{-1}}{N^{snake}} (\mathbf{s}_j^{snake} - \mathbf{s}_i) K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{snake}) = \mathbf{0} \tag{4.10}$$

Reorganizing the terms and solving for $\mathbf{s}_j^{snake}$, the fixed point update rule can be

written as [1]

$$\mathbf{s}_j^{snake} \leftarrow \frac{\sum_{i=1}^{N} \mathbf{s}_i \; w_i(\Sigma_i + \Gamma_j)^{-1} \; K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{snake})}{\sum_{i=1}^{N} w_i(\Sigma_i + \Gamma_j)^{-1} \; K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{snake})} \tag{4.11}$$

This iteration is similar to that of mean-shift and has been shown to be an EM-update [15, 21], thus is convergent. Due to the elimination of spurious edge maxima, the nonparametric snake does not suffer from poor capture range.

The fixed point iterations above introduce a drawback: depending on the initial conditions, the original set of snake samples might be unable to progress into boundary concavities. We overcome this problem and also densely populate the boundary by using the fixed-point iterations recursively. A step by step summary of the Nonparametric Snake algorithm is given in Table 4.1.

4.1.3 Interpolations Using Principal Curves

Note that all active contour methods in the literature leave the smooth interpolation of the final samples to general purpose interpolation methods like nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, and spline interpolation. Similarly, nonparametric Snake also leaves the interpolation step to these selections.

Using the underlying density estimate of the edge image, suitably initialized principal curve projections can be used for interpolating the missing edges throughout the active contour. A suitable interpolation metric should bend the original feature space in a way that, where the probability density of the edge is high, the curve should be allowed to bend to follow the underlying edge density; conversely, where the underlying edge probability density is low, the interpolating curve should be smooth.

Figure 4.1 shows the edge density and the principal curve. Here, we zoom into the boundary and present the interpolating curve for several pairs of samples.

[1] In (4.11) the division denotes left multiplication with the inverse of the matrix given in the denominator. Specifically, for a spherical Gaussian the matrix in the denominator will become a scalar.

Table 4.1: Nonparametric Snake with principal curve interpolations

1. Generate the edge image, $E(\mathbf{s})$.
2. Select the kernel size and estimate the probability density of the edge image $p_{edge}(\mathbf{s})$.
3. Select the initialization of the snake and estimate the probability density of the snake $p_{snake}(\mathbf{s})$
4. Use the iteration scheme given in (4.11) to iterate the points on the snake.
5. Select a neighborhood threshold $th_{neighbor}$, which will define the resolution level of the snake - default value should be unity.
6. For the points on the snake, build a graph of pairwise distances and find the points that have no neighbors for the given threshold.
7. Select a perturbation step size μ and perturb these points towards M randomly selected directions. Typical values are $\mu = th_{neighbor}/5$ and $M = 5$. Optionally, if it's known that the interior or the exterior of the boundary provides a smoother cost function, these M randomly selected directions could be selected accordingly. This can be implemented by utilizing the inner product of the selected directions and the vector that connects the particular point to be perturbed to its location in the previous iteration.
8. Use SCMS to map these perturbed points to their projection on the contour. After convergence, add these points to the snake and include them in the neighborhood threshold calculation.
9. Repeat until the predefined neighborhood threshold is satisfied at all points.

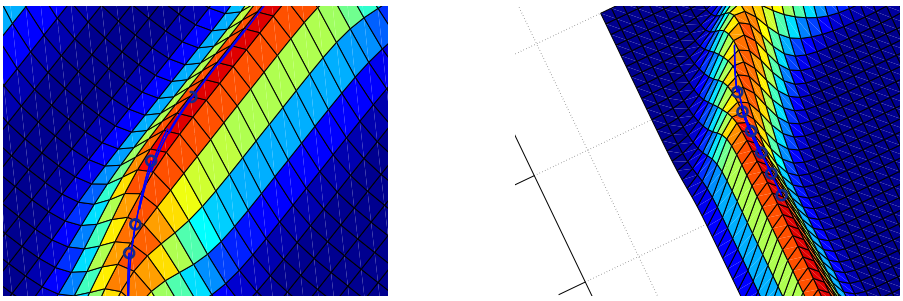


Figure 4.1: Two examples for the optimal interpolating curve with the edge probability density, where the big circles represent the points on the snake and interpolating curve combine these points

Pulling the interpolation curve towards the edge probability density, presented interpolation technique automatically detects boundary concavities. If the boundary has many concavities and the snake is initialized with insufficient number of points, the snake will presumably have difficulties in progressing into concavities. However, if the proposed ridge detection idea is used to interpolate the points on the snake, all concavities in the edge map will automatically be detected.

We used the approach presented in Table 4.1 for automatic lung and lung tumor segmentation for 3D-CT applications in radiation oncology [68]. Figure 4.2 shows the results of nonparametric snake with and without principal curve interpolations, where blue shows the edge pixels (of a binary edge image), red shows the initialization and white contour is the final result of the nonparametric snake. The nonparametric snake, being essentially a mean-shift approach using weighted KDE, can cause the snake points to cluster in regions of high density if the edges are non-uniformly sampled along the ridge. Principal curve interpolations solve this problem -again, nonparametrically- as presented with the light blue contour in Figure 4.2(b).

Aside from interpolations, seeking for the ridges in an image is a known image segmentation tool, which finds use in medical image processing [29, 85]. In this approach, the image is mapped into a three dimensional space consisting of the

Figure 4.2: Two examples for the optimal interpolating curve with the edge probability density, where the big circles represent the points on the snake and interpolating curve combine these points

pixel coordinates, x, y; and the corresponding intensity, $I(x, y)$. The segments of the image are considered to be relatively constant, and the segmentation is achieved by using the ridges of the image in this space. Principal curves would be directly applicable in this context, and one can use SCMS in the feature space of x, y and $I(x, y)$, starting from all pixels.

4.2 Optical Character Skeletonization

Principal curves have been used for skeletonization of optical characters [74]. Optical character skeletonization can be used for two purposes: feature extraction for optical character recognitionf and compression. One significant problem for applying principal curve algorithms to skeletonization of optical characters is that, by definition, algorithms are seeking for a *smooth curve*. In general, data may have loops, self intersections, and bifurcation points, which is obviously the case for the optical characters.

Kegl's principal graph algorithm is perhaps the only method in the literature that can successfully handle such irregularities [74]. In this approach, Kegl reshapes his polygonal line algorithm [75] to handle loops, and self intersections by modifying it with a table of rules and adding preprocessing and postprocessing steps. The

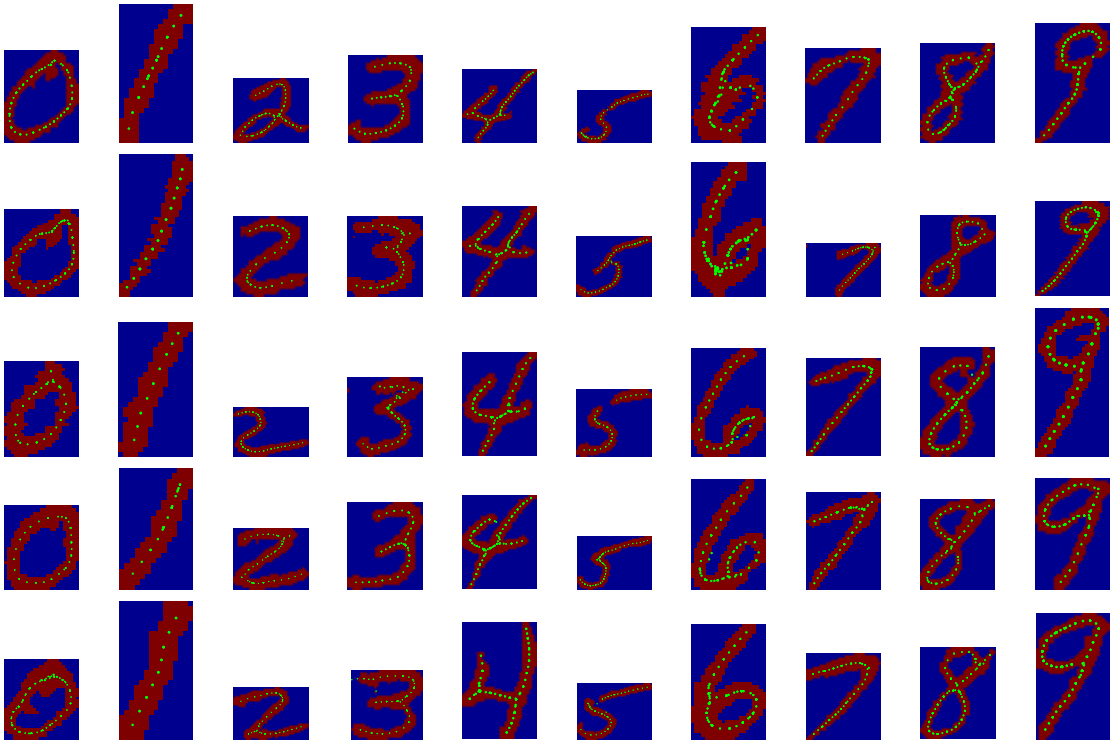


Figure 4.3: Principal graph results in optical characters

polygonal line algorithm, which is the basis of the principal graph algorithm, is based on the idea of bounding the length of the principal curve to prevent overfitting. At this point, modifying this algorithm with a table of predefined rules that include parameters and thresholds is nothing but further parameterizing the problem.

Since the back projection of the data samples is not required and the principal graph itself is sufficient, we use piecewise linear SCMS algorithm to show how our principal curve definition performs in this application. The dataset we use in this experiment consists of handwritten digits, and this dataset is provided by Kegl. We use the ML kernel bandwidth given in Section 3.1.3, and the depth parameter of PL-SCMS is selected as $m = 2$. Results are presented in Figure 4.3.

4.3 Principal Curve Time Warping

Time series analysis is an important field in adaptive signal processing with numerous applications in econometric studies in stock market, various biomedical and speech processing applications, or intelligent transportation systems analysis. A common problem in all these applications is to derive a suitable distance measure between time series signals. One common case in time series applications is that although some pairs of signals demonstrate similar characteristics (for unsupervised scenarios) or belong to the same class (for supervised scenarios), the predominant structures of the signals do not align in the time axis.

Dynamic time warping (DTW) is a technique to solve this alignment problem. DTW has evolved from the speech processing field and later applied to many different areas [109]. In general, DTW is a method that finds an optimal match between two sequences of feature vectors with certain restrictions - monotonicity, continuity, and boundary conditions. The problem is modelled as finding the minimum distance through the matrix of pairwise distances of the data samples, and as the name implies, to achieve the solution, DTW uses dynamic programming techniques.

DTW algorithm is later referred to as *the DTW model*, as it was reinterpreted to be a parametric model [110]. Although it has found widespread use, DTW is very susceptible to noise, and time warpings over low SNR signals may end up with many *singularities*. Singularities are defined as "unintuitive alignments where a single point on one time series maps onto a large subsection of another time series" [76]. Literature on DTW is rich on modifications to increase noise robustness; however, most of these are heuristic attempts. These approaches are not guaranteed to remove all singularities, and even worse, they may prevent the optimal solution from being found. Techniques include using moving average filters on the time series signals to reduce the high frequency content, assuming that most of the high frequency content is the noise [110, 8, 89]. This approach can be thought of as constraining the search

space of *allowable* warpings. Although constraining the search space of allowable warpings may prevent the optimal solution from being found, this approach is also reported to increase the classification performance [101].

It is probably safe to claim that almost all time warping algorithms in the literature are derivatives of the original DTW algorithm. Most recent approaches include derivative dynamic time warping (DDTW) [76] that uses the derivative of the signals rather than the original values, enhanced dynamic time warping (EDTW) [132] that brings a unifying view to DTW and hidden Markov models, and context dependent dynamic time warping (CDDTW) [100] that exploits application specific contextual characteristics of the signals to improve performance. The idea of DDTW is based on reducing the singularities of the final solution and provides an improvement over the noise robustness of the original DTW algorithm, whereas EDTW introduces the HMM based modelling of the time series to provide robustness.

We propose to use principal curves to improve the noise sensitivity of the time warping problem, and our approach allows one also to evaluate the time warping function in low SNR cases. This improvement not only brings increased robustness to current time warping applications, but also may trigger new application areas where the signals that need to be compared are buried in noise. Principal curve projections can be performed either in the original signal space or any feature set derived from the data. Therefore, this nonparametric denoising step can be coupled with the original DTW approach, or DDTW, EDTW, CDDTW or any other time warping approach in the literature.

In template matching or hypothesis testing algorithms, the test signals are compared with the noiseless template signal. Here, we consider the more realistic case of the time warping between two noise corrupted signals, assuming a noiseless template signal may not be available -or even necessary- in all applications. For the signals

to be compared, one can write

$$\begin{aligned} r_1(t) &= s(f(t)) + n_1(t), \quad t = t_1 < \ldots < t_N \\ r_2(t) &= s(t) + n_2(t), \qquad\quad t = t_1 < \ldots < t_N \end{aligned} \tag{4.12}$$

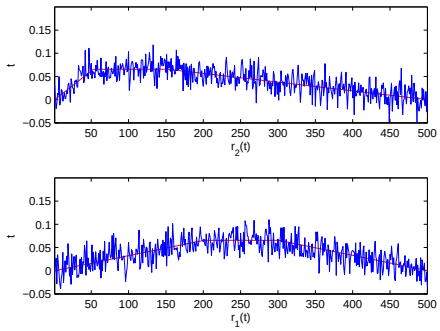
where $f(t)$ is the time warping function and $n_1(t)$ and $n_2(t)$ are unimodal additive noise, and here we assume that the signals $r_1(t)$ and $r_2(t)$ have the same length. We build the principal curve feature space as

$$\mathbf{r}_i = \begin{bmatrix} r_1(t_i) \\ r_2(t_i) \\ t_i \end{bmatrix}, \qquad t = t_1 < \ldots < t_N \tag{4.13}$$

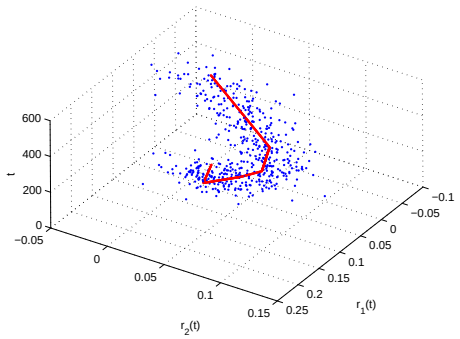
Figure 4.4 depicts some realizations of $\mathbf{r}$ along with the corresponding noisy signal pairs $r_1(t)$ and $r_2(t)$, and noiseless signal pairs $s(t)$ and $s(f(t))$. Figure 4.4(a) presents piecewise linear signals $s(t)$ and $s(f(t))$ (red), and their noisy versions $r_1(t)$ and $r_2(t)$ (blue). Note that, the data structure in $\mathbf{r}$ that is given in Figure 4.4(b) demonstrates the pairwise signal characteristics, as a perturbation around a predominant shape in time. As the noise level increases, the amount of perturbation around the predominant shape increases. Obviously, the noiseless signals are not available in real applications, and we propose to use the principal curve projections of the data samples to approximate the noiseless signal characteristics in $\mathbf{r}$ domain. *Projecting the signal samples onto the principal curve can be regarded as a nonparametric nonlinear filtering.* This projection provides a generalization of the whole shape of the data in feature space and effectively implements a data driven nonparametric noise reduction filter. This yields

$$\widetilde{\mathbf{r}}_i = \begin{bmatrix} \widetilde{s}(f(t_i)) \\ \widetilde{s}(t_i) \\ t_i \end{bmatrix}, \qquad t = t_1 < \ldots < t_N \tag{4.14}$$

where $\widetilde{\mathbf{r}}_i$ is the projection of $\mathbf{r}_i$ onto principal curve.



(a) Noisy signals r_1 and r_2 (blue) and the noiseless signals $s(t)$ and $s(f(t))$ (red) are shown.(SNR = 3dB)



(b) The data structure in the **r** space is shown for noisy signal pairs r_1 and r_2 (blue), and noiseless signal pairs $s(t)$ and $s(f(t))$ (red).

Figure 4.4: Some examples for pairs of -noisy and noiseless- signals and their structure in **s** space.

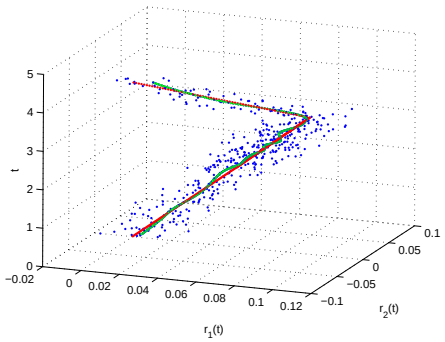
Particularly for the time warping application, we have a much easier scenario as compared to the general principal curve fitting problem due to the following:

1. Only the samples of the principal curve at time indices $t = t_1 < \ldots < t_N$ are sufficient. Higher time resolution or seeking for the portion of the principal curve that lies outside the given time interval are unnecessary.

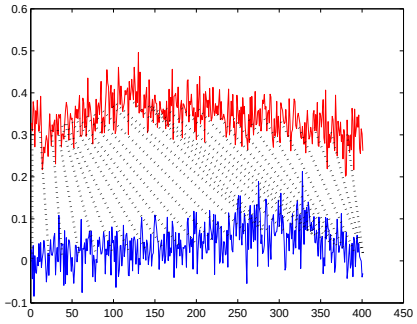
2. Under the unimodal additive noise assumption, there will be no *branching* in the principal curve, since s(t) and s(f(t)) are functions of time. Thus, for every subspace of **r** defined by $t = t_i$ $t_i \in [0, 1]$ there is a single point in the principal curve.

3. Unlike the general case of random vectors, the 3^{rd} dimension of **r** is deterministic; in the case of uniform sampling, we can model this density as being uniform for theoretical analysis.

One can select the initialization of the algorithm and the constrained space of the



(a) The noisy (blue), the noiseless data (red) characteristics in $\mathbf{r}$, along with the principal curve estimated from the noisy data (green)



(b) Noisy signals $r_1(t)$ and $r_2(t)$ and the corresponding alignments after principal curve denoising

Figure 4.5: A pair of noisy signals and the principal curve in the $\mathbf{r}$ domain, and alignment of this signal pair

projection using above simplifications. At this point, starting from the data samples themselves, and selecting the constrained space as $t = t_i$ for each data sample is our choice for the following reasons:

1. Selecting constrained space orthogonal to time index guarantees that there is only one denoised signal value at all time indices.

2. One important observation here is that the peak of the pdf in each constrained space $t = t_i$ is very close to principal curve.

The second observation here is based on the additive unimodal noise assumption. Although it is close, the constrained space $t = t_i$ does not exactly lead to the principal curve. Therefore, after convergence in $t = t_i$ subspace, we will use SCMS to ensure the projection onto the principal curve.

Table 4.2 presents a step by step implementation of the proposed approach. Overall, the proposed denoising scheme has a complexity of $O(N)$ per sample per

Table 4.2: Principal Curve Time Warping

1. Build the feature space $\mathbf{x}$, select the kernel bandwidth σ of the Gaussian kernel.
2. Evaluate the mean shift update using (3.6).
3. Project the mean shift update to the constrained space $t = t_i$.
4. If convergence is not achieved, go to step 2, if convergence is achieved go to the next step.
5. For every trajectory evaluate the mean shift update in (3.6).
6. Evaluate the gradient, the Hessian, and perform the eigendecomposition of $\mathbf{\Sigma^{-1}}(\mathbf{x}(k)) = \mathbf{V\Gamma V}$.
7. Let $\mathbf{v}$ be the leading eigenvector of $\mathbf{\Sigma^{-1}}$.
8. $\tilde{\mathbf{x}} = \mathbf{v}\mathbf{v}^T\mathbf{m}(\mathbf{x}(k))$
9. If $|\mathbf{g}^T\mathbf{H}\mathbf{g}|/\|\mathbf{g}\|\|\mathbf{H}\mathbf{g}\| > threshold$ then *stop*, else $\mathbf{x}(k+1) \leftarrow \tilde{\mathbf{x}}$.
10. If convergence is not achieved, increment k and go to step 6.

iteration, and projections onto principal curve approximates the noiseless signal samples.

$$\widetilde{\mathbf{r}}_i \simeq \begin{bmatrix} \widetilde{s}(f_{12}(t_i)) \\ \widetilde{s}(t_i) \\ t_i \end{bmatrix}, \qquad t = t_1 < \ldots < t_N \tag{4.15}$$

Figure 4.5(a) shows the found principal curve (green) for the signal pairs presented in Figure 4.5(b). Figure 4.5(b) also shows the time alignments of this signal pair. We will provide a more detailed analysis on the accuracy of the approximation given in (4.15) in the experimental results section.

As discussed in the previous chapter, a significant practical consideration for the implementation of the algorithm is the selection of the bandwidth of the Gaussian kernel function. In many kernel machine applications, spherically symmetric Gaussian kernels suffice. However, this cannot be the case for the $\mathbf{r}$ space. The variance

around the principal curve in the first two dimensions depends on the noise power of the two compared signals, whereas time index in the third dimension may have any arbitrary scale depending on the sampling rate. Moreover, in some applications, the bandwidth of the Gaussian kernel can be adjusted by exploiting the actual physical meaning of data feature space.

In many signal processing applications, the noise power can be estimated very reliably. In such cases, the estimate of the noise distribution can be used as the kernel function. Assuming n_1 and n_2 are independent Gaussian noise, one can write the covariance of the Gaussian kernel function as

$$\Sigma = \begin{bmatrix} \sigma_{n_1} & 0 & 0 \\ 0 & \sigma_{n_2} & 0 \\ 0 & 0 & N \end{bmatrix} \tag{4.16}$$

where N controls the amount of smoothness to be introduced along the time axis. Generally, this choice is not optimal; still, it eliminates tedious kernel optimization efforts and yields satisfactory results.

4.3.1 Experiments

To test the accuracy of the noiseless signal approximation provided by the principal curve, we present the error between the noiseless signal structure and the principal curve for different noise levels. Afterwards, we will present signal pairs and corresponding time warping functions for synthetic and real data examples.

Noiseless signal vs principal curve: As discussed before, an important problem in time warping applications is the increasing noise level. Therefore, here we present results for the approximation we make in (4.15), and give the integrated error between the noiseless signal structure and the principal curve structure for different noise levels. To be able to repeat the experiment for different noise levels,

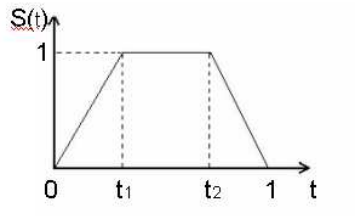


Figure 4.6: The noiseless signal given in (4.17).

in this experiment we will use the following synthetic signal

$$r_{synthetic}(t) = \begin{cases} \frac{t}{t_1} & : 0 \leq t \leq t_1 \\ 1 & : t_1 \leq t \leq t_2 \\ \frac{t-1}{t_2-1} & : t_1 \leq t \leq 1 \end{cases} \tag{4.17}$$

where t_1 is uniformly distributed between 0.05 and 0.45, and t_2 is uniformly distributed between 0.55 and 0.95. The shape of the signal is given in Figure 4.6. We generate realizations of this random signal $r_{synthetic}(t)$ of length $N = 200$ for each noise level, and add white Gaussian noise to obtain the samples of the noisy signal $s_{synthetic}(t)$ for 10dB, 5dB, 3dB, and 2dB. Hence, the noisy signal is

$$s_{synthetic}(t) = r_{synthetic}(t) + n_{Gaussian}(t) \tag{4.18}$$

For 100 pairs of random realizations of $s_{synthetic}(t)$, we use the iterative scheme given in Table 4.2 to project the data onto the principal curve. We evaluate the integrated error between the noiseless structure using pairs of $r_{synthetic}(t)$ and the approximation provided by principal curve for different noise levels.

Figure 4.7 shows the mean and the variance –the error bar demonstrates ± 2 standard deviation from the mean– of the integrated error of these 100 Monte Carlo

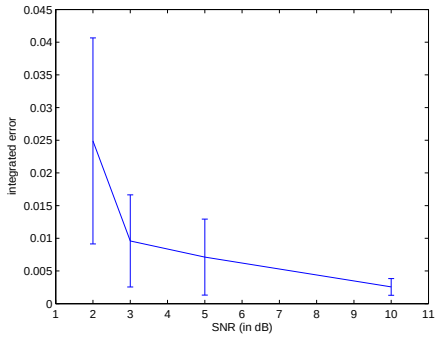
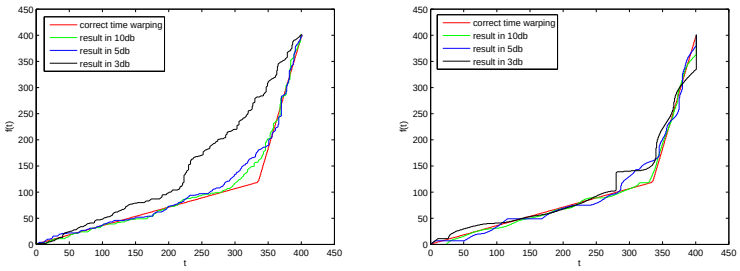


Figure 4.7: SNR vs. integrated error between the noiseless signal structure and the approximation given by the principal curve

simulations at noise levels 10dB, 5dB, 3dB and 2dB. The accuracy of the principal curve approximation decreases with increasing noise level. Still, the method is able to provide reliable approximations for noise levels as high as 2dB. The principal curve denoising results presented in this subsection exclude the solution of the time warping function to give the reader the chance to evaluate the principal curve denoising step independently.

Time Warping Functions in Different Noise Levels: In this section, we present the solutions of PCTW on synthetic and real data for different noise levels to show how the denoising actually affects the final results. For this experiment, we select the same signal structure used in the previous experiment for simplicity, and we compare results of DTW algorithm using (i) the original noisy signals (ii) the principal curve projections.

Figure 4.8 presents the solution of the time warping function for different noise levels. Also note that for this simple piecewise linear case, the correct time warping function is known, and presented along with the solutions. As the noise level increases, the principal curve denoising still leads to reasonably good results, whereas for the original noisy signal, the performance drops significantly.



(a) Solutions using the original data

(b) Solutions using the principal curve projections

Figure 4.8: The correct time warping function and solutions in different noise levels using the original data (left) and the principal curve projections (right).

Time Series Clustering: In this section, we will compare time series classification results for a synthetic dataset using DTW, with and without the principal curve denoising. To show how much the principal curve denoising changes the final classification performance, we will provide the pairwise distances for two clusters.

In this experiment, we use the following two-cluster dataset

$$r_{cluster1}(t) = \begin{cases} 1 & : 0 \leq t \leq t_1 \\ 0 & : t_1 \leq t \leq 1 \end{cases} \quad t_1 \in U(0.3, 0.4)$$

$$r_{cluster2}(t) = \begin{cases} 0 & : 0 \leq t \leq t_2 \\ 1 & : t_2 \leq t \leq 1 \end{cases} \quad t_2 \in U(0.5, 0.6) \tag{4.19}$$

where t_1 and t_2 are uniformly distributed as shown above. Figure 4.9 shows a collection of 20 signals, 10 signals per each cluster. Each realization of the signal has 50 samples.

Figure shows the pairwise distances between these signals, evaluated using DTW, with (red) and without (blue) the principal curve denoising for different noise levels. Here, we present the sorted in-cluster and inter-cluster distances, a total of 400 pairwise distances. The first 200 pairwise distances are between sample pairs of

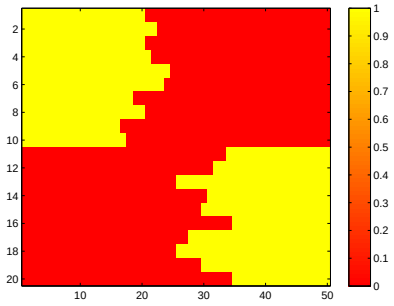


Figure 4.9: Time series clustering dataset

same cluster, and the second 200 represent the distances between sample pairs of different cluster. Principal curve denoising preserves the gap between in- and inter-cluster distances for noisy cases as well, which is essentially what is required for good clustering results.

Time Series Classification: Very similar to previous section, we will present time series classification results. In this experiment we use a publicly available process control dataset [99, 77] for different noise levels. This dataset has 300 training and 300 test samples.

We present the ratio of samples from the correct class among the K-nearest neighbors using Euclidean distance, DTW distance with (red) and without (blue) the principal curve denoising. We add Gaussian noise to both training and testing data and repeat the experiment for different noise levels and report results for different values of K.

Figure 4.3.1 presents the times series clustering results using Euclidean distance (blue) and DTW distance with (green), and without (red) the proposed principal curve denoising. One can see that for the original dataset in Figure 4.3.1(a), principal curve denoising does not change the results significantly. We repeat the same experiment by adding Gaussian noise to the data, and for the noisy cases, using

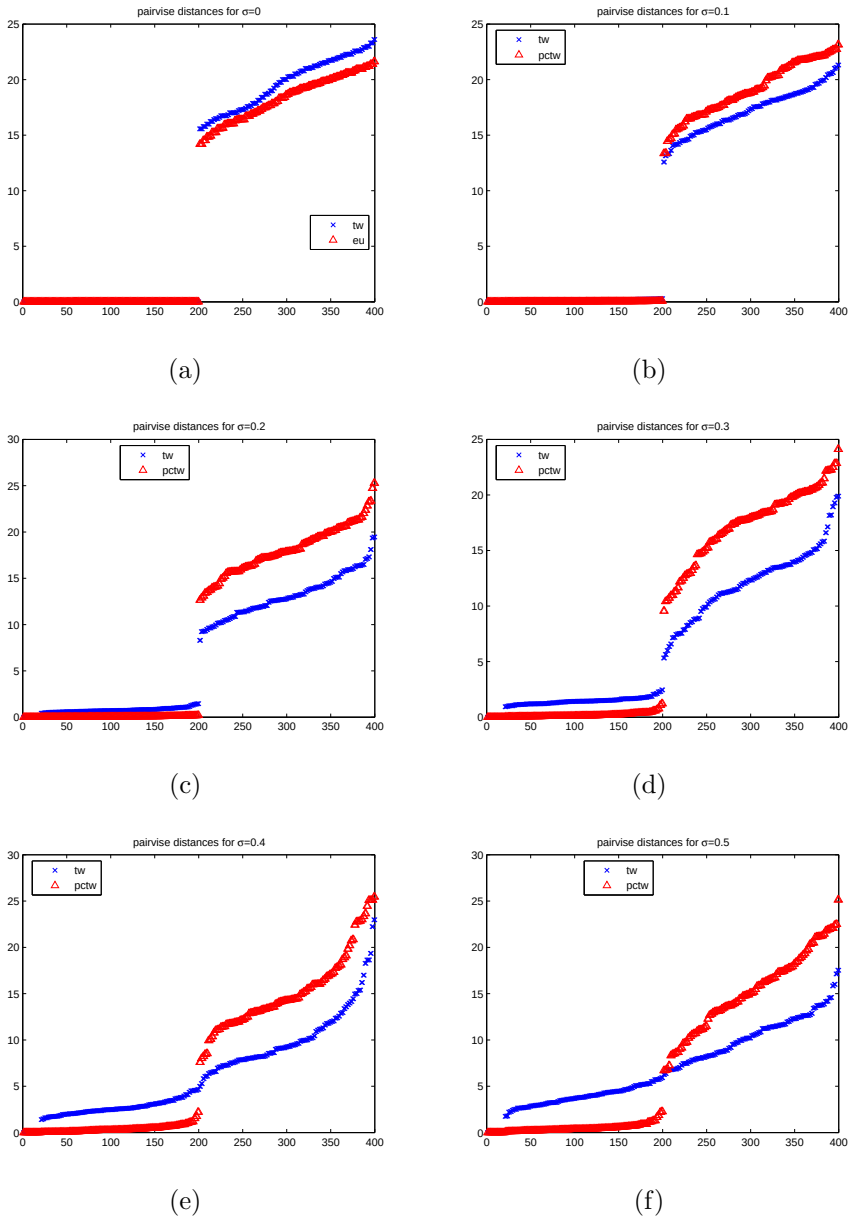
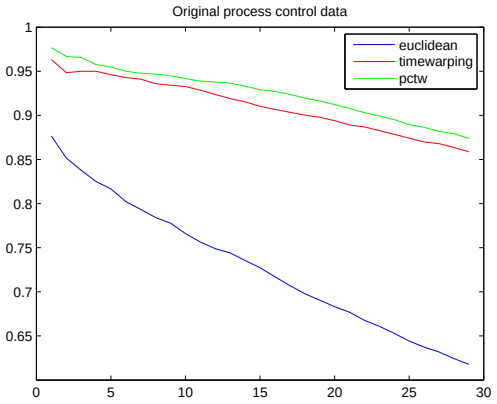
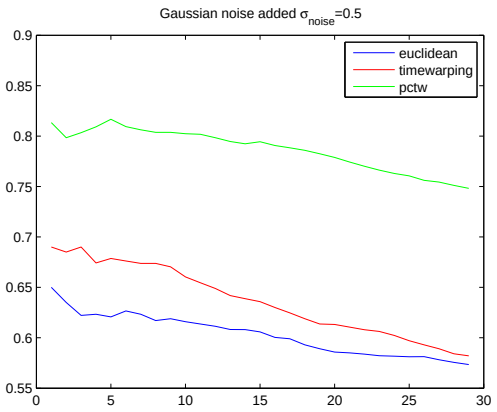


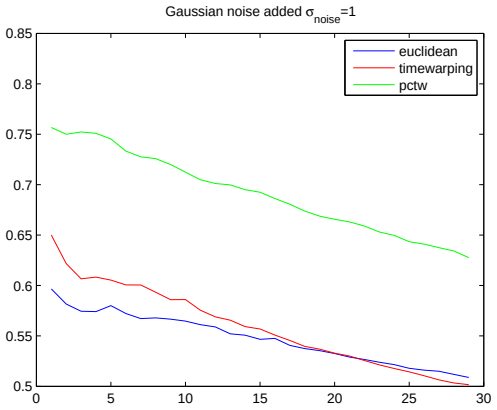
Figure 4.10: Time series clustering results. Pairwise DTW distances for original (blue) and principal curve denoised data (red) are presented.



(a)



(b)



(c)

Figure 4.11: Time series classification results.

principal curve denoising at the preprocessing stage improves the performance significantly.

Considering the results presented in time series classification context, note that our aim is not to improve upon the existing time warping applications. For comparison, here we use the original DTW algorithm and the signals themselves; however, principal curve denoising can be coupled with any time warping algorithm in the literature and can be used in any feature space. Different time warping algorithms or feature sets can always be introduced for particular applications.

4.3.2 Discussions

Reinterpreting the problem in the feature space of pairwise signals, one can employ principal curves to increase the noise robustness of time warping algorithms. In this context, the principal curve projections implement a nonlinear nonparametric noise reduction filter.

We derive the principal curve based denoising under unimodal additive noise assumption. Since the proposed principal curve projection converges to the maxima of the pdf in the constrained space, unimodal noise assumption is required for theoretical analysis to have a single maxima point -hence a single denoised signal value- in the pdf for all time indices. If the noise is multimodal, the effect of the multimodal noise can be removed by increasing the bandwidth of the Gaussian kernel in the subspace constrained mean shift. However, this may potentially lead to oversmoothing of the signal and should not be preferred if the signal has high frequency content. Overall, PCTW lays a robust and effective time warping technique.

4.4 Piecewise Smooth Signal Denoising

One common problem in signal denoising is that if the signal has a blocky, in other words, a piecewise-smooth structure, the denoised signal may suffer from over-smoothed discontinuities. Under the known noise model assumption, optimal filtering in frequency domain is a well established topic [59]. Frequency domain filtering has a number of attractive properties; it is conceptually simple, easily analyzable and computationally inexpensive. However, the main drawback of this approach is obvious if the signal and the noise are not separable in the frequency domain. Practically, in frequency domain filtering, one should compromise the discontinuities in the signal since they will be smoothed along with the noise.

As well as traditional frequency domain filtering techniques, current signal denoising techniques include wavelet transform and total variation based noise removal algorithms. These techniques stem from the idea of preserving the high frequency components, namely discontinuities of the signal while suppressing the noise. Since piecewise smooth signal structure is typical in images, total variation based noise removal is first introduced in image denoising [108, 35]. Later, Vogel proposed a fixed-point iterative scheme to solve the total variation penalized least squares problem [128].

Another way to achieve discontinuity preserving denoising is to utilize discrete wavelet transform as an orthogonal basis decomposition. This method is based on decomposing the signal into orthogonal wavelet basis, perform either soft or hard thresholding to wavelet coefficients, and transform the data to the time/spatial domain by inverse discrete wavelet transform. Note that any orthogonal decomposition of the signal can be used similarly for denoising. In such techniques, the performance of the denoising is typically measured by the rate of decay of the basis coefficients sorted in the magnitude. In the context of piecewise smooth signal denoising, wavelet decomposition is known to be superior to other widely used choices like Fourier and

cosine decompositions [23]. Earlier techniques based on hard thresholding of the wavelet coefficients suffer from Gibbs effect like artifacts around the discontinuities. This drawback is handled by soft thresholding of these wavelet coefficients [37], and later by applying previously mentioned total variation based approaches in wavelet domain analysis [41].

Techniques based on total variation and wavelet decomposition are parametric methods, and we propose to use principal curve projections for artifact-free piecewise continuous signal denoising to implement a data-driven nonlinear filter in a nonparametric way. To utilize principal curve projections, one should start with translating the signal into a suitable feature space. The most intuitive selection is to use the samples of the noisy signal itself along with the associated time indices. Assuming that the observed signal $x(t)$ is the original signal buried in additive noise that is, $x(t) = s(t) + n(t)$; this yields

$$\mathbf{x}_i = \begin{bmatrix} x(t_i) \\ t_i \end{bmatrix}, \qquad t_1 < \ldots < t_N \tag{4.20}$$

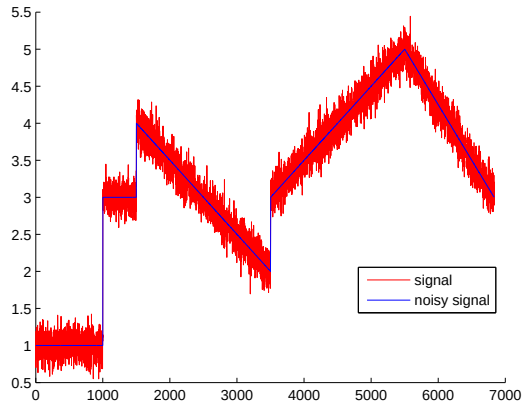
where t_i denotes the sampling times. At this point, we assume that the original signal $s(t)$ is sampled at least at the Nyquist frequency. Sampling and reconstruction are well established topics in signal processing, the details of which are out of the scope here.

Figure 4.12 depicts an illustrative piecewise smooth signal with its noisy version as well as the kernel density estimate of the noisy signal. Now consider the kernel density estimate of the noisy signal which is given by

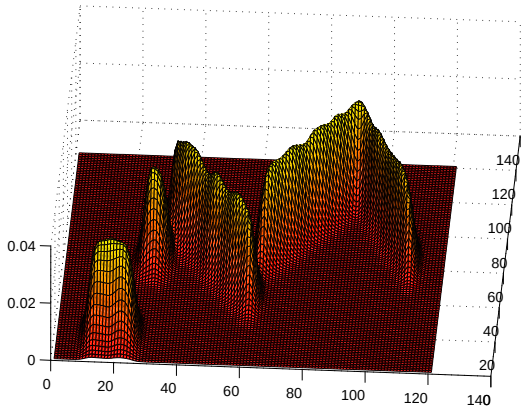
$$p(\mathbf{x}) = \sum_{i=1}^{N} K_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \tag{4.21}$$

where Σ_i denotes the covariance of the kernel function for the i^{th} sample[2]. Note that, the density estimate of $\mathbf{x}$ in the vicinity of discontinuities is not effected much

[2]This density estimate is given for the most general case of data-dependent variable bandwidth KDE. For fixed bandwidth version, one can just drop this data dependency.



(a) noisy and noiseless signals in time domain



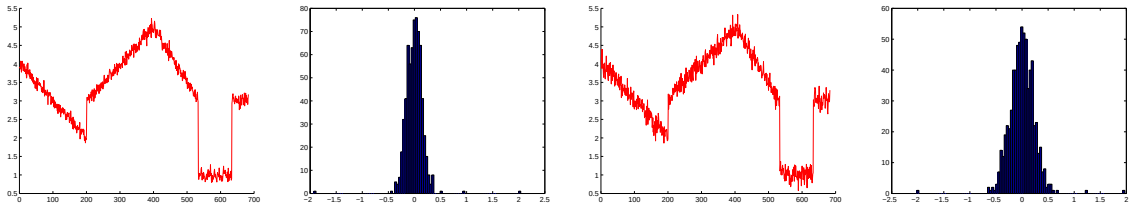
(b) KDE of the noisy signal in $\mathbf{x}$ space

Figure 4.12: The underlying noiseless signal and its noisy counterpart

by the samples on the opposite side of the discontinuity. Although these are close -or maybe even subsequent- samples along the time axis, the discontinuity in signal value makes them sufficiently distant in the $\mathbf{x}$ space. For unimodal additive noise distributions (as in the case of additive Gaussian noise) the noiseless signal will most likely be a *smoother* signal *passing through the middle* of the observed noisy samples. Hence, principal curve naturally fits into the context of artifact free signal denoising of piecewise smooth signals.

As in the time warping application, selecting the constrained space orthogonal

to time index guarantees that there is only one denoised signal value at all time indices, and the mode of the pdf in this constrained space is close to the principal curve. At this point, starting from the data samples themselves, and selecting the constrained space as $t = t_i$ for each data sample is our choice. Again, SCMS will be used as a postprocessing to ensure the projection onto the principal curve. A step by step implementation of the algorithm is given in Table 4.3.



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

Figure 4.13: Two realizations of the same piecewise smooth signal for two different SNR levels are shown in (a) and (b), along with their time difference histograms in (c) and (d).

Particularly for this problem, one can select the kernel bandwidth by considering the actual physical meaning of the data. If the amount of noise and the amount of fluctuations at the discontinuities are at different scales, kernel bandwidth selection can be achieved by observing probability density of the time difference of the signal. Figure 4.13 presents two realizations of a piecewise smooth signal at different noise levels. Although this may not be the general case for all noise levels, note that the noise distribution and discontinuities are clearly identifiable in this difference histogram for the presented signals, where the difference values at the discontinuities seem like outliers of the Gaussian noise distribution. Thus, one can select the bandwidth of the Gaussian kernel by observing the histogram of the one step difference of the signal.

Table 4.3: Principal curve signal denoising

1. Build the feature space $\mathbf{x}$, select the kernel bandwidth σ of the Gaussian kernel.
2. Evaluate the mean shift update using (3.6).
3. Project the mean shift update to $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, so that the mean shift procedure remains in the constrained space.
4. If convergence not achieved, go to step 2, if convergence is achieved go to the next step.
5. For every trajectory evaluate the mean shift update in (3.6).
6. Evaluate the gradient, the Hessian, and perform the eigendecomposition of $\mathbf{\Sigma^{-1}}(\mathbf{x}(k)) = \mathbf{V\Gamma V}$.
7. Let $\mathbf{v}$ be the leading eigenvector of $\mathbf{\Sigma^{-1}}$.
8. $\tilde{\mathbf{x}} = \mathbf{v}\mathbf{v}^T\mathbf{m}(\mathbf{x}(k))$
9. If $|\mathbf{g}^T\mathbf{H}\mathbf{g}|/\|\mathbf{g}\|\|\mathbf{H}\mathbf{g}\| > threshold$ then *stop*, else $\mathbf{x}(k+1) \leftarrow \tilde{\mathbf{x}}$.
10. If convergence is not achieved, increment k and go to step 6.

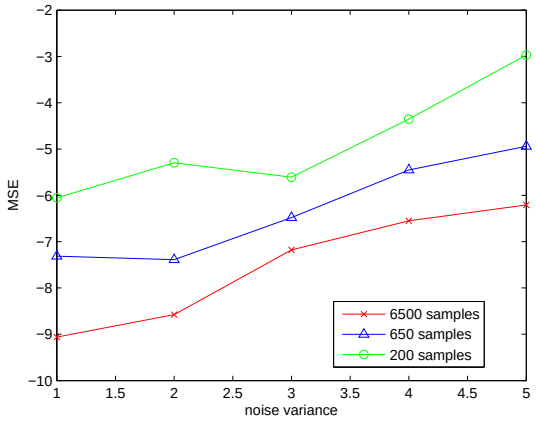


Figure 4.14: MSE (shown in logscale) between the noiseless signal and output of the denoising filter for different SNR levels

4.4.1 Experiments

This section presents the performance of the proposed denoising algorithm. To be able to control the SNR level in the experiments, synthetic examples are used, and denoising performance is reported for different SNR values. Another important point is, of course, the performance around the discontinuities. As well as quantitative results, we present the actual signal plots to provide a visual comparison of the noisy and denoised signals at different SNR levels. At the discontinuities, principal curve denoising does not yield any pseudo-Gibbs artifact effects like discrete wavelet denoising techniques, or any oversmoothing like frequency domain filtering. As one can observe from the denoising on the ramp-signal components, principal curve denoising also does not yield any staircase type effects like total variation based methods.

Figure 4.14 presents the denoising results for the noiseless signal presented in Figure 4.12 (blue) for different SNR levels. We measure the performance MSE between the output of principal curve denoising and the underlying noiseless signal. Here we also compare the number of samples used in the principal curve estimation. As presented in Chapter 3, the accuracy of the principal curve projection depends

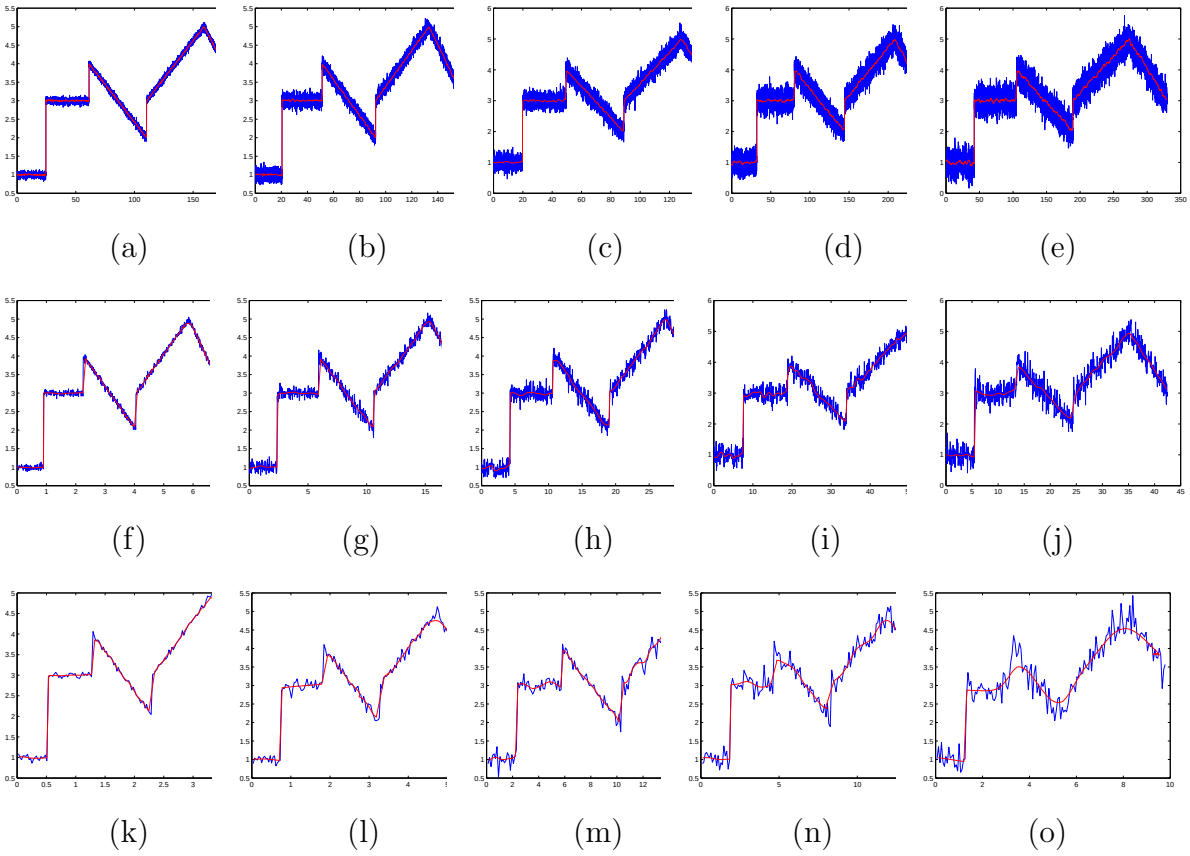


Figure 4.15: Two realizations of the same piecewise smooth signal for two different SNR levels are shown in (a) and (b), along with their time difference histograms in (c) and (d).

on the accuracy of the underlying density estimate. As the statistical variance of the density estimate increases in low sampling rates, the performance of the principal curve denoising decreases.

For the same signals of different SNR and sampling rates, which are compared in Figure 4.14, Figure 4.15 presents the noisy and denoised time signal pairs. For all above experiments, we use the maximum likelihood kernel size for each signal. So, we learn the kernel bandwidth from the signal directly, and use a different kernel bandwidth for each realization of the signal. Maximum likelihood training automatically adapts the kernel bandwidth according to the noise level of the data.

As a final remark, note that we are not presenting any comparisons with the other principal curve approaches in the literature. Since, by definition, they are looking for smooth curves, earlier principal curve algorithms are not suitable for piecewise smooth signals, and oversmoothing on the discontinuities would be unavoidable.

4.4.2 Discussion

Principal curve based denoising filter can successfully preserve the sharp discontinuities in the data under the additive unimodal noise assumption, without introducing any smoothing, high frequency Gibbs effect like artifacts or staircase type effects.

Further, this approach allows one to translate common problems in piecewise smooth signal denoising into selection of the kernel function. For example, as opposed to earlier methods in the piecewise-smooth signal denoising that are based on total variation or discrete wavelet transform, principal curve based denoising is much easier to build an online counterpart just by selecting a finite support kernel function. For a KDE built using a kernel function of finite support, the subset of signal samples that affect the pdf on a particular point of interest are clearly given, yielding an online version of the algorithm directly. Adapting the principal curve denoising for nonstationary or data-dependent noise distributions is also straightforward.

With the current implementation, principal curve denoising is computationally more expensive than earlier methods in the literature. At this point, KDE with finite support kernel will save a lot of computational effort, which can bring the computational cost up to linear complexity.

4.5 Principal Curve Proximity Graph

There are many algorithms in the machine learning literature that employ data proximity (neighborhood) graphs, which have the data samples at the nodes and

the pairwise distances at the vertices. Recently, there has been a growing interest in these graph based methods on dimensionality reduction, manifold learning, and data clustering; on the other hand, there is surprisingly less work on how to construct the graphs themselves, and mostly K-nearest neighbor or ϵ-ball graphs have been used for this purpose, which simply connect each data point to K nearest data points, or every point within a distance of ϵ. As a more robust alternative, we propose using principal curve projections that yields a proximity graph based on the probability density of the data.

The main focus in dimensionality reduction and manifold learning methods based on the data proximity graph are using the idea of enhancing the metric definition of the feature space by seeking for shortest paths over a neighborhood graph. Most famous methods that fall into this category include ISOMAP [120], Local Linear Embedding [107], and Laplacian Eigenmaps [6]. The idea of using the data proximity graph for clustering can simply be defined as ways of how to *cut* the graph into sub-graphs [115, 44], and perhaps the normalized cut is most well-known method that uses this idea [115], which is widely used in many applications in machine learning, mostly in image segmentation. In fact, the whole research on spectral clustering is nothing but variants of these graph based clustering methods. Overall, regardless of the purpose that it is going to be used, the aim of the graph is to provide a good summary of the data manifold. Simply, "*the graph should avoid shortcuts that travel outside the manifold, avoid gaps that erroneously disconnect the manifold, and be dense within the manifold and clusters*" [14], and a desired property of data graphs is robustness to its parameters and noise in the observed data.

In general, the optimal values of ϵ or K are never known, and since small changes in these parameters may lead to important changes in the neighborhood graph -in terms of shortcuts and gaps- these methods are problematic in terms of parameter sensitivity. Thus, in most cases these parameters require a manual tuning, which is achieved by running the graph-based algorithm for different parameters until

satisfactory results are obtained.

Another problem with these graphs is the noise in the observed data. In real life the observed data is rarely noisefree; it is buried in noise. Noise robustness is a very desired property of data proximity graphs, which can be considered twofold: (*i*) robustness to small perturbations in the data, (ii) robustness to background noise and outliers. Firstly, the graph should not change significantly if the whole dataset is perturbed by a *small* random noise. Secondly, outliers of the data distribution have to be handled in a way that they will not yield shortcuts outside the data manifold.

Unfortunately, neither the K-nearest neighbor nor the ϵ-ball graphs have these properties. Carreira-Perpiñán proposes to use an ensemble of minimum spanning trees by perturbing the dataset with a random noise for each realization of the ensemble [14]. Although it is not using any density estimation directly, this approach, in our opinion, is a stochastic approach for a kernel density estimate, where the amount of random perturbation controls the kernel width, and the statistical variance of the underlying density estimate drops with the number of realizations of perturbed data ensemble. Overall, this approach is effective in terms of perturbations in the data samples. However, it does not address the problem of outliers, which may yield shortcuts outside the data manifold.

We propose to use the principal curve - or a computationally cheap approximate of it- to construct the data proximity graph. The proposed graph construction method is composed of three main parts: (*i*) selecting representative points from the data (*ii*) projecting these points onto the principal curve (*iii*) building the neighborhood graph over the pairwise distances of these principal curve projections.

Particularly for the proximity graph construction problem, instead of using the principal curve projections of the whole dataset we propose to use a representative subset (or vector quantization) of the data to save computation. This approach not only decreases the overall computational requirement of the graph construction,

but also enables one to update the graph efficiently for novel samples, providing a suitable framework for online learning.

To build the principal curve, we propose to use VQ-SCMS algorithm. In the corresponding neighborhood graph, the data points are connected to the nearest node on the principal curve, and the connected nodes on the principal curve are defined by the curve itself. Particularly for manifold learning approaches, to learn the local structure of the points that are connected to the same node on the principal curve may also be connected to each other.

Since shortest paths are defined only over the connection of the nodes on the principal curve and $M \ll N$, seeking for shortest paths is much cheaper than K-nearest neighbor or ϵ-ball graphs. Moreover, for a novel sample, all pairwise distances are given with $O(M)$, since all that is required is to find the nearest node on the principal curve. Using the principal curve projections bring robustness to the outliers in the data as well as small perturbations in the dataset, and the shortcuts outside the data manifold as well as gaps in the data manifold are eliminated effectively.

Figure 4.16 presents the results on a semicircle dataset with 300 samples for $M = \sqrt{N}$ ($\simeq 17$) and $M = \sqrt[3]{N}$ ($\simeq 7$); the data samples (blue), the nodes on the principal curve (red) and the principal curve constructed using MST (green) are shown. Real distances of the data samples along the manifold and the distances evaluated over the principal curve proximity graph are presented in Figure 4.17. Note that the block-wise structure becomes less apparent as M increases.

Figure 4.18 and Figure 4.19 present the same, for a much harder scenario, using the spiral dataset with background noise. Note that the principal curve proximity graph is able to approximate the data manifold in the presence of background noise for different values of M effectively. In the presence of such background noise, shortcuts outside the manifold would be unavoidable with K-nearest neighbor and ϵ-ball graphs.

4.6 Future Prospects and Preliminary Results

As presented throughout this chapter, general purpose principal curve algorithms that we derived in Chapter 3 directly apply to many practical problems. Before we continue on our conclusions, we will briefly discuss our future research directions by presenting potential applications of principal curves and our preliminary results.

4.6.1 Manifold Unfolding and Nonlinear Blind Source Separation

Nonlinear blind source separation (NBSS) is an ill-posed problem that requires various sources of a priori knowledge regarding the joint source distribution and topology of the mixing function. In a landmark paper it has been shown that the square nonlinear independent component analysis (NICA) problem is ill-posed - that, without additional constraints it does not have a unique solution accepting the usual generalized scale and permutation ambiguities of ICA - and various conditions that would force the problem to have a unique solution have been proposed [63]. Traditionally several methods, primarily based on parametric model fitting - especially neural networks of various forms - have been proposed in the literature, while minimizing the usual mutual information inspired independence or separation measures [2, 96, 67]. The general form of the NBSS problem is

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) \tag{4.22}$$

where $\mathbf{f} : \Re^m \rightarrow \Re^n$ maps the source vector $\mathbf{s} \in \Re^m$ to the measurement vector $\mathbf{x} \in \Re^n$. In general, $\mathbf{f}$ is assumed to be a smooth function and $m = n$, since even in this case per existence theorems, there are infinitely many solutions to this ill-posed inverse estimation problem. Limiting the search to the space of invertible $\mathbf{f}$ makes both theoretical and practical sense, especially since without additional domain information about the mechanisms that generate $\mathbf{x}$, a noninvertible $\mathbf{f}$ would cause information loss that cannot be recovered.

An extensively studied invertible nonlinear mapping topology is the so-called post-nonlinear mixture model of the form $\mathbf{x} = \mathbf{g}(\mathbf{As})$, where $\mathbf{A}$ introduces an invertible (or full column rank if $n > m$) linear mixture and $\mathbf{g}$ is an invertible nonlinear map that acts elementwise (only monotonic scaling, not mixing further) on the mixed intermediate vector [67]. This problem is well defined, and relatively straightforward extensions of traditional linear ICA techniques are able to cope with the specific mixture topology.

An interesting contribution that focused on the overdetermined post-nonlinear mixture problem by exploiting the geodesic distances on nonlinear manifolds was made by Lee et al. [83]. Noticing that the monotonic nonlinearity simply acts as a local metric distortion independently on each coordinate axis, ideas from isometric dimensionality reduction and manifold learning were employed to unwrap the nonlinear mixture data manifold into the original source distribution space. Harmeling et al also followed a similar manifold unfolding approach using kernel techniques [55]. The idea of using principal curve projections for NBSS could be considered as an extension of these works focusing on square mixture scenarios. The method would be immediately applicable to overdetermined situations where the data lies on a low dimensional manifold embedded in a higher dimensional Euclidean space - the principal coordinates proposed in the following would simply reduce to zeros for any additional dimension outside the data manifold automatically.

We presented our preliminary results on NBSS via nonlinear manifold unfolding with case studies with periodic and random sources [42]. In general, nonlinear coordinate unfolding can be achieved by measuring the distances along and orthogonal to the principal curve.

Orthogonal distance to the curve: The easiest way to approximate the orthogonal distance is to use the distance between the data sample and its principal curve projection. This assumes that the projection path is a straight line, which in general, is not true but it still provides a good approximate in many cases. Ideally,

the distance can be measured using the numerical integration method with a small step size and accumulating the distance at every step until convergence.

Measuring distances along the curve: Measuring the distance along the curve can be achieved either parametrically and nonparametrically. One can fit a parametric model, presumably a spline, to the curve to measure distances according to this model. Alternatively, one can build a neighborhood graph over the principal curve projections of the data and employ Dijkstra's shortest path algorithm to measure distances along the curve. We will use this approach in the below illustration.

Consider the following mixture where the source distribution along the curve is uniform and the source distribution orthogonal to the curve is Gaussian presented in Figure 4.20(a). The data samples (blue) are shown with their principal curve projections (red). Figure 4.20(b) shows the scatter plot of the source distribution along and orthogonal to the curve, where the histograms of these sources are presented in Figure 4.20(c) and Figure 4.20(d), respectively.

4.6.2 Time-Frequency Reassignment

Spectrogram reassignment is a natural application area for our principal curve definition. As any other bilinear energy distribution, the spectrogram is faced with an unavoidable trade-off between the reduction of misleading interference terms and a sharp localization of the signal components. To reduce the smoothing effects introduced by the window function in short-term Fourier transform, reassignment methods are used to sharpen the time-frequency representation by using the rate of change of phase of the signal, which finds numerous applications in speech signal processing and signal roughness analysis [52, 48]. Parameter envelopes of spectral components are obtained by *following ridges on the smooth time-frequency surface*, using the reassignment method [3] to improve the time and frequency estimates for the envelope breakpoints.

Current methods use the rate of change of the phase to reassign the spectrogram to find the ridges of the time-frequency surface. This approach is known to be problematic, where rotation invariance is a desired property of the reassigned spectrogram. Principal curve projections, on the other hand, would yield rotation invariant reassignment results in 2D. Although most of the literature focuses on cross interference terms, presence of the *auto-cross terms* yields a much more challenging scenario [91, 92]. The auto-cross terms are also known as inner interference terms and they appear if the signal has multiple signal components in the time-frequency domain, and in the presence of the auto-cross terms, the rotation invariance property of the reassignment method is more critical.

Overall, time-frequency reassignment is a natural application for our principal curve definition, and developing a principal curve based spectrogram reassignment method will be in our research agenda. Figure 4.21 shows our preliminary results for a synthetic dataset. Figure 4.21(a) shows the ideal time-frequency representation. Wigner-Ville distribution of the signal, and the smoothed Wigner-Ville distribution, where the cross-terms in the original spectogram are eliminated are shown in Figure 4.21(c). Figure 4.21(d) also shows the principal curve of this time-frequency surface.

4.6.3 Clustering for MIMO Channel Equalization

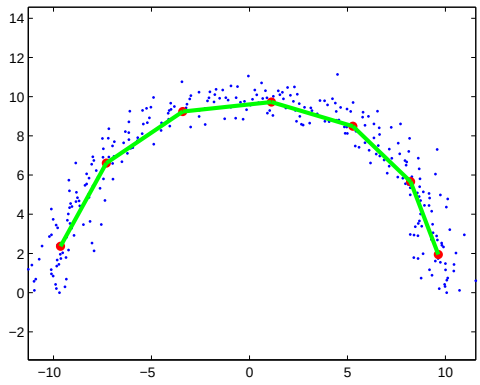
Recently, multiple-input multiple-output wireless communication drawn considerable attention, and there are reliable and computationally inexpensive symbol detection algorithms in the literature [49]. On the other hand, applications in time-varying environments poses a harder problem to the changing channel state, and some supervised algorithms have been proposed to tackle this issue, where an initialization phase is used in the beginning for training purpose [106, 71, 22].

Blind channel equalization approaches in the literature are based on clustering [20]. However, these approaches mostly focus on time-invariant single-input single-output channels. Recently, a spectral clustering technique is proposed that extends

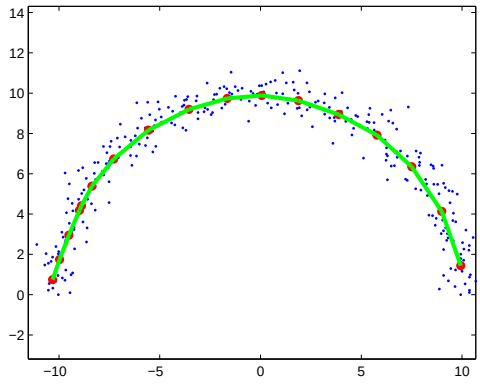
the applications into time-varying multiple-input multiple-output channels as well [124, 123]; Van Vaerenbergh and Santamaria introduce the time scaling into the feature space before employing the clustering algorithm to *untangle* the clusters. A similar idea proves to be effective in Post-Nonlinear Blind Source Separation as well [122]. Consider the clustering problem in four dimensions presented in Figure 4.22(a). The advantage in this application is that the actual number of clusters is known, still the fast time-varying nature of the channel poses a very difficult clustering problem.

Now consider the five dimensional case, where the time index of every sample is added into the feature space. Figure 4.22(b) shows this case by plotting two three dimensional subspace projections of the data, where the first four dimensions of the data is given with respect to time index. By using this five-dimensional feature space, Van Vaerenbergh and Santamaria employ a spectral clustering algorithm to solve the channel equalization problem with no supervision. Still, noise robustness and increasing symbols that leads to increasing number of clusters in the data, remain a significant problem. At this point, observing the intertwined structure of the clusters, one obvious question is the following: Can one improve the clustering performance by using the geometry of the data, or more explicitly, exploiting the fact that the clusters are curves in the feature space.

Our perspective to approach this question is to work on the pairwise clustering of the principal curve projections instead of the data samples. Observing Figure 4.23 that shows the results of the principal curve projection of the data obtained using SCMS, one can see that performing the pairwise clustering over the principal curve projections instead of the samples is very promising.

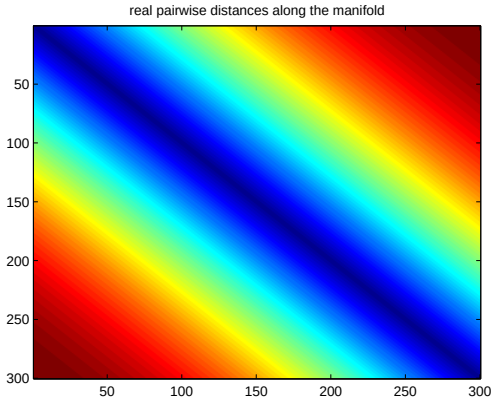


(a) Result for $M = 7$

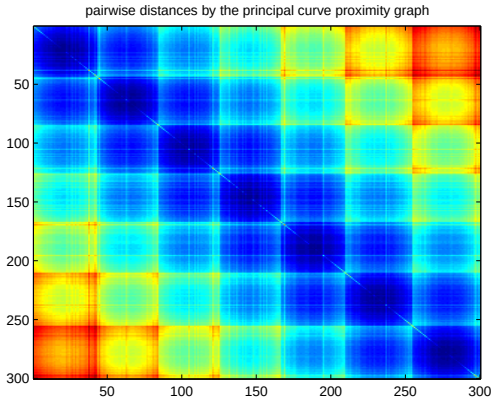


(b) Result for $M = 17$

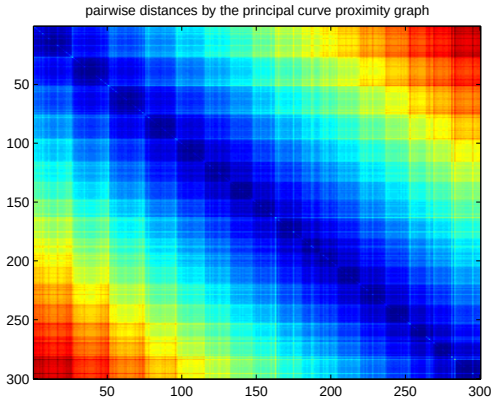
Figure 4.16: Semicircle dataset and results of VQ-SCMS algorithm for different M values.



(a) Real distances along the data manifold

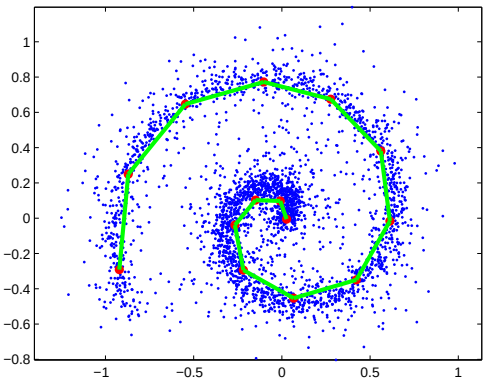


(b) Distances of PCPG for $M = 7$

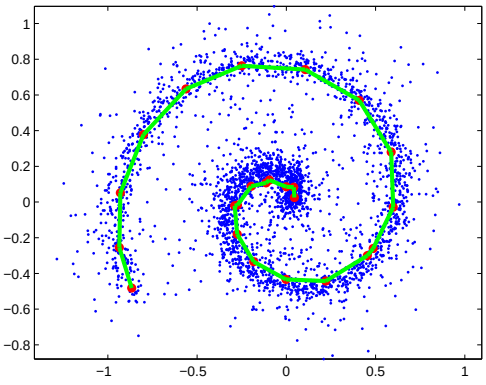


(c) Distances of PCPG for $M = 17$

Figure 4.17: Semicircle dataset: Real pairwise distances, and distances given by PCPG for different M values.

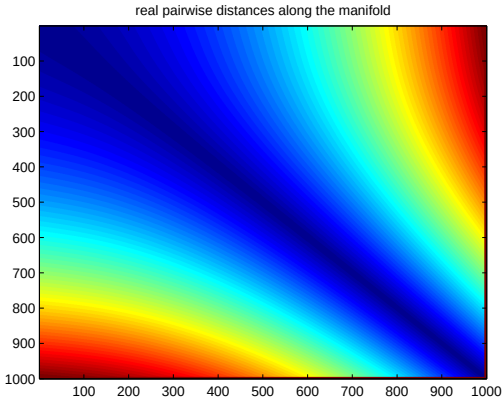


(a) Result for $M = 14$

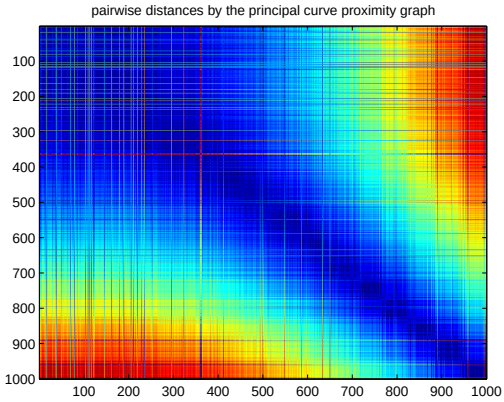


(b) Result for $M = 24$

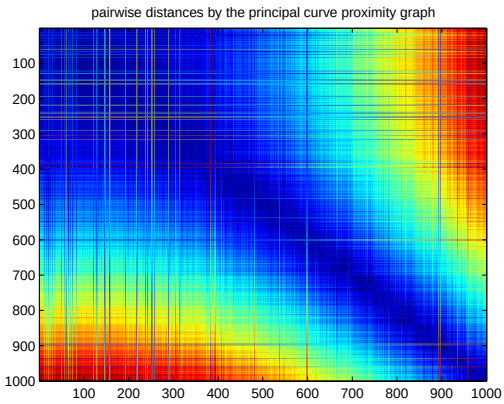
Figure 4.18: Spiral dataset and results of VQ-SCMS algorithm for different M values.



(a) Real distances along the data manifold

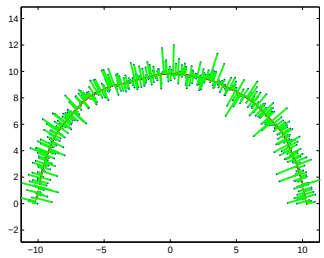


(b) Distances of PCPG for $M = 14$

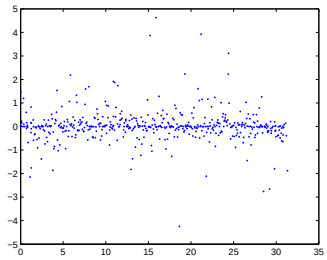


(c) Distances of PCPG for $M = 24$

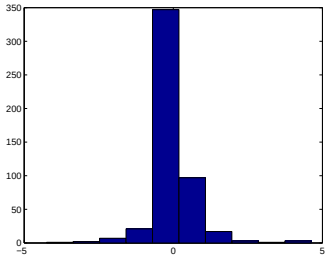
Figure 4.19: Spiral dataset: Real pairwise distances, and distances given by PCPG for different M values.



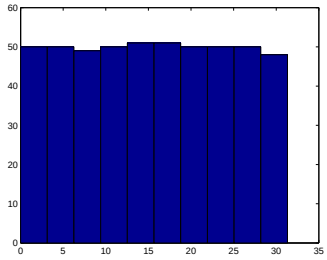
(a) Scatter plot of the mixture, along with the principal curve and orthogonal distances to the curve



(b) Scatter plot of the sources

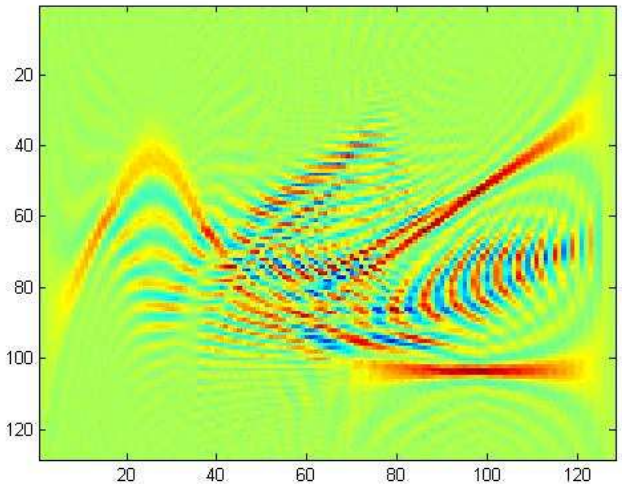


(c) Histogram of the source orthogonal to the curve

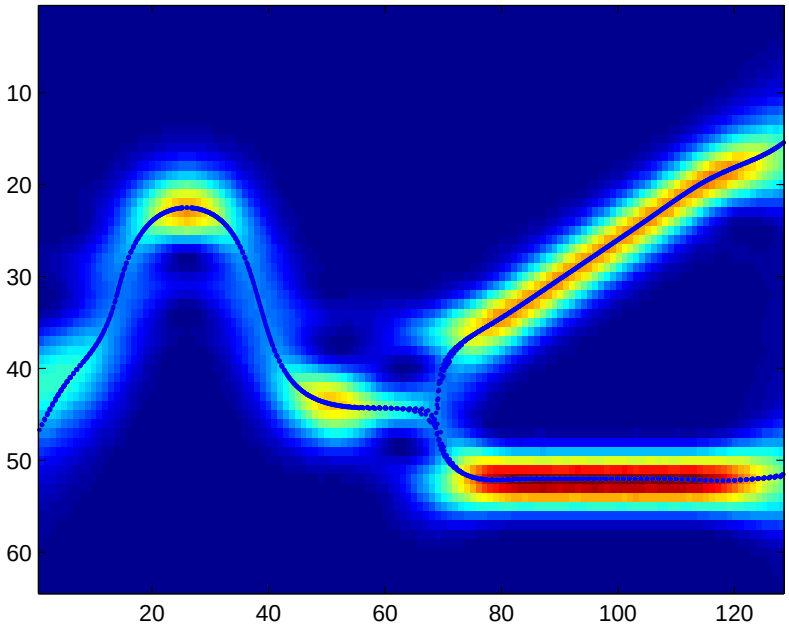


(d) Histogram of the source along the curve

Figure 4.20: Spiral dataset: Real pairwise distances, and distances given by PCPG for different M values.

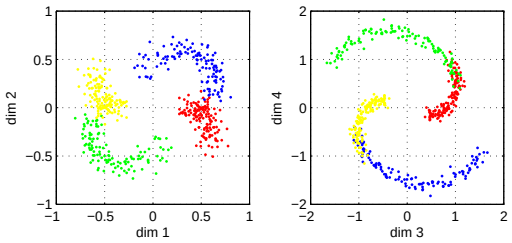


(a) The Wigner-Ville distribution

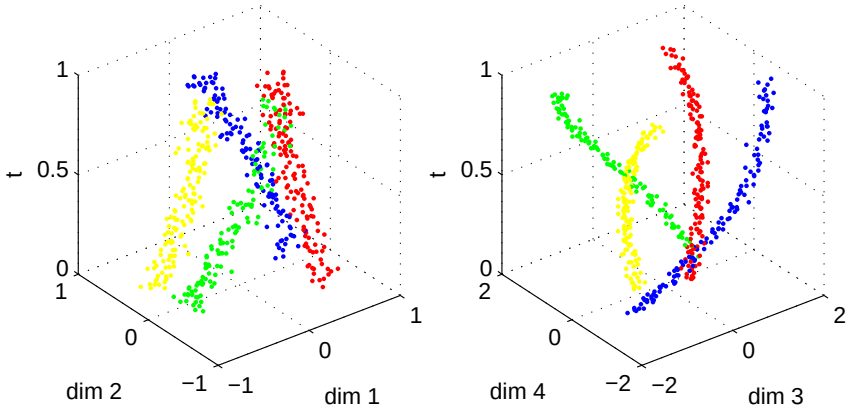


(b) Smoothed Wigner-Ville distribution and its principal curve

Figure 4.21: Wigner-Ville distribution, its smoothed version and principal curve of the smoothed distribution



(a) Four dimensional data, coming from four symbols



(b) Four dimensional data with time embedding

Figure 4.22: Symbol clustering problem for a MIMO channel

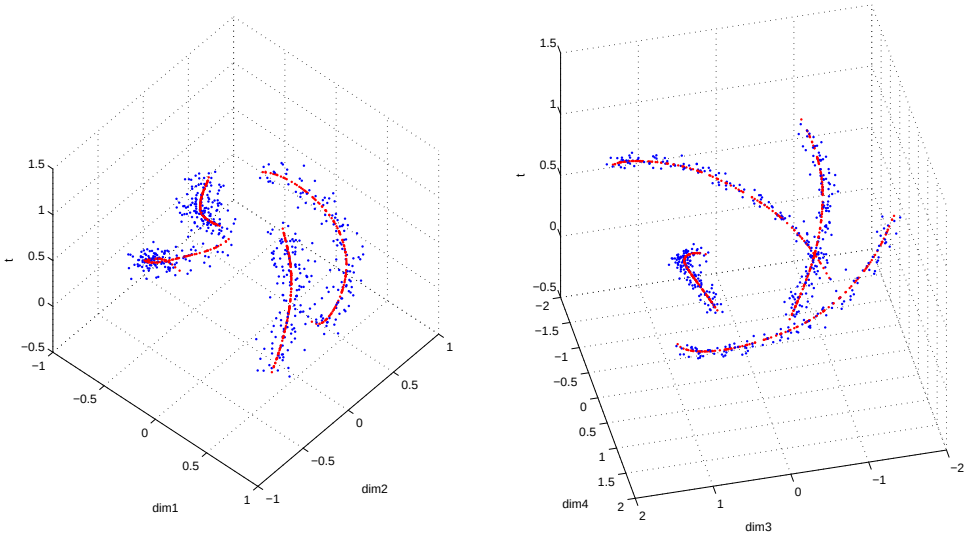


Figure 4.23: Samples (blue) and their principal curve projections (red)

Chapter 5

Conclusions and Future Work

Principal curves and surfaces have numerous applications in statistical machine learning as well as other disciplines. As opposed to traditional methods that use least square error type criteria or employ a parametric structure, we propose a novel definition for principal curves and surfaces by generalizing the likelihood maximization reasoning of earlier definitions, and characterizing the principal curves and surfaces in terms of the gradient and the Hessian of the density estimate.

We provide a mathematically rigorous definition for principal curves and surfaces as a function of the data probability density, in which we exclude any smoothness, regularization, and overfitting constraints. Overfitting is an issue that arises in finite sample based learning with unknown model topology. Characterizing principal curves and surfaces in terms of the probability density rather than the data samples, allows one to borrow the required smoothness constraints and outlier robustness properties from the density estimation literature directly. Using KDE and learning the bandwidth by likelihood maximization is a more theoretically well-founded approach as compared to defining an additional parameter to control the amount of smoothness, which in general requires manual tuning. Besides, since the ridges of the pdf can intersect each other, handling self-intersecting structures requires no additional effort.

Unlike traditional machine learning papers on manifold learning, which tend to focus on practical aspects such as reconstruction error, we focus on the definition

of the underlying manifold from a more theoretical point of view, and our proposition yields a unified framework for clustering, principal curve fitting and manifold learning. A nice connection between principal curves/surfaces and data clustering is given by the principal set of intrinsic dimensionality zero, which yields the modes of the probability density. Hence, projecting the data onto zero dimensional principal set coincides with a widely accepted clustering solution.

Algorithmically, principal curves can be found using subspace constrained likelihood maximization idea, very similar to well known mean shift algorithm, and Subspace Constrained Mean-Shift implementation strengthens the theoretical connection to clustering from a practical point of view. Our principal curve implementation does not require significantly more time or memory storage as compared to mean shift, which already has been utilized in many practical application domains. Moreover, approaches that have been proposed to accelerate mean shift can directly be applied to algorithms that we provide.

Overall, the elegant approach finds use in many applications including signal denoising, neighborhood graph construction, image segmentation and many other. To demonstrate the usefulness of the algorithms presented in Chapter 3, results that are presented in Chapter 4 do not use any domain specific information, which can always be incorporated to increase performance in any particular application.

Some of our future goals are presented at the end of the Chapter 4 along with preliminary results. From the theoretical point of view, we will work on the following problems:

The relationship between the curvature of the curve and the number of samples: Consider the case where the noise distribution is fixed, and the true underlying curve is a sinusoid. As the frequency of the sinusoid increases, the number of samples to estimate this true curve with a given statistical variance increases. In general, one should investigate the relationship between the curvature of the true underlying curve and the number of samples required to estimate the principal curve.

A practical algorithm based on the principal curve projections of the training set: To provide another practical algorithm construct convex regions from the training set and project these data samples onto the principal curve. For the testing set, evaluate the principal curve projections only using the projections of the training set. Traditional interpolation techniques will not work here, since the direct interpolation of projections of the training set may or may not be on the principal curve. However, on average, as the volume of the convex region decreases, the projection error should decrease.

The connection between the principal curve and density geodesics: The path that minimizes the objective function of density geodesics can be evaluated in a similar way to PL-SCMS. For a pair of points that are the on the principal curve, the path is directly given by the principal curve segment in between these points.

Bibliography

[1] Abrantes, A., and Marques, J. A class of constrained clustering algorithms for object boundary extraction. *IEEE Transactions on Image Processing 5*, 11 (Nov 1996), 1507–1521.

[2] Almeida, L. B. Misep - linear and nonlinear ica based on mutual information. *J. Mach. Learn. Res. 4*, 7-8 (2004), 1297–1318.

[3] Auger, F., and Flandrin, P. Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Transactions on Signal Processing 43*, 5 (May 1995), 1068–1089.

[4] Banfield, J. D., and Raftery, A. E. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association 87*, 417 (1992), 7–16.

[5] Baudat, G., and Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural Computation 12*, 10 (2000), 2385–2404.

[6] Belkin, M., and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 6 (June 2003), 1373–1396.

[7] Bellman, R. E. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

[8] Berndt, D. J., and Clifford, J. Using dynamic time warping to find patterns in time series. In *KDD Workshop* (1994), pp. 359–370.

[9] Bishop, C. M. *Neural Networks for Pattern Recognition, 1st Ed.* Clarendon Press, Oxford, 1997.

[10] Bresson, X., Vandergheynst, P., and Thiran, J.-P. Multiscale active contours. *Int. J. Comput. Vision 70*, 3 (2006), 197–211.

[11] Brewer, M. A model-based approach for variable bandwidth selection in kernel density estimation, 2000.

[12] Brodowski, J. Dijkstra shortest path algorithm. WWW: http://www.animal.ahrgr.de/Anims/en/dijkstra.aml.

[13] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis Machine Intelligence 8*, 6 (1986), 679–698.

[14] Carreira-Perpiñán, M. A., and Zemel, R. S. Proximity graphs for clustering and manifold learning. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, Cambridge, MA, 2005, pp. 225–232.

[15] Carreira-Perpinan, M. A. Gaussian mean-shift is an em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 5 (2007), 767–776.

[16] Chang, H., and Yeung, D.-Y. Robust path-based spectral clustering with application to image segmentation. In *Proc. 10th Int. Conf. Computer Vision (ICCV'05)* (Beijing, China, Oct. 15–21 2005), pp. 278–285.

[17] Chang, K., and Grosh, J. A unified model for probabilistic principal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 59–74.

[18] Chapelle, O., and Zien, A. Semi-supervised classification by low density separation. In *Proc. of the Tenth Int. Workshop on Artificial Intelligence and Statistics (AISTATS 2005)* (Barbados, Jan. 6–8 2005), R. G. Cowell and Z. Ghahramani, Eds., pp. 57–64.

[19] Chen, D., Zhang, J., Tang, S., and Wang, J. Freeway traffic stream modeling based on principal curves and its analysis. *IEEE Transactions on Intelligent Transportation Systems 5*, 4 (2004), 246–258.

[20] Chen, S., Mulgrew, B., and Grant, P. A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on Neural Networks 4*, 4 (Jul 1993), 570–590.

[21] Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 8 (1995), 790–799.

[22] Choi, J., Yu, H., and Lee, Y. Adaptive mimo decision feedback equalization for receivers with time-varying channels. *IEEE Transactions on Signal Processing 53*, 11 (Nov. 2005), 4295–4303.

[23] Cohen, A., and D'Ales, J.-P. Nonlinear approximation of random functions. *SIAM Journal on Applied Mathematics 57*, 2 (1997), 518–540.

[24] Cohen, L., and Cohen, I. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 11 (November 1993), 1131–1147.

[25] Cohen, L. D. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding 53*, 2 (1991), 211–218.

[26] Comaniciu, D. An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis Machine Intelligence 25*, 2 (2003), 281–288.

[27] Comaniciu, D., and Meer, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619.

[28] Comon, P. Independent component analysis – a new concept? *Signal Processing 36*, 3 (1994), 287–314.

[29] Crowley, J., and Parker, A. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 2 (March 1984), 156–169.

[30] Davatzikos, C., and Prince, J. An active contour model for mapping the cortex. *IEEE Transactions on Medical Imaging 14*, 1 (Mar 1995), 65–80.

[31] De'ath, G. Principal curves: a new technique for indirect and direct gradient analysis. *Ecology 80*, 7 (1999), 2237–2253.

[32] Delaunay, B. N. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 6 (1934), 793–800.

[33] Delicado, P. Principal curves and principal oriented points, 1998.

[34] Der, R., Steinmetz, U., Balzuweit, G., and Schüürmann, G. Nonlinear principal component analysis. Tech. rep., Institut für Informatik, Universität Leipzig, 1998.

[35] Dobson, D. C., and Santosa, F. Recovery of blocky images from noisy and blurred data. *SIAM Journal on Applied Mathematics 56*, 4 (1996), 1181–1198.

[36] Dong, D., and McAvoy, T. Nonlinear principal component analysis-based on principal curves and neural networks. *American Control Conference, 1994 2* (29 June-1 July 1994), 1284–1288 vol.2.

[37] Donoho, D. L. De-noising by soft-thresholding. *IEEE Transactions on Information Theory 41*, 3 (1995), 613–627.

[38] Duchamp, T., and Stuetzle, W. Extremal properties of principal curves in the plane. *The Annals of Statistics 24*, 4 (1996), 1511–1520.

[39] Duchamp, T., and Stuetzle, W. Geometric properties of principal curves in the plane. *Robust Statistics, Data Analysis, and Computer Intensive Methods: In Honor of Peter Huber's 60th Birthday 109* (1996), 135–152.

[40] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification.* Wiley-Interscience Publication, 2000.

[41] Durand, S., and Froment, J. Reconstruction of wavelet coefficients using total variation minimization. *SIAM J. Sci. Comput. 24*, 5 (2002), 1754–1767.

[42] Erdogmus, D., and Ozertem, U. Nonlinear coordinate unfolding via principal curve projections with application to bss. In *14th International Conference on Neural Information Processing* (2007).

[43] Fahrmeir, L., and Tutz, G. *Multivariate Statistical Modelling Based on Generalized Linear Models.* Springer-Verlag, New York, 1994.

[44] Felzenszwalb, P. F., and Huttenlocher, D. P. Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 2 (2004), 167–181.

[45] Fischer, B., and Buhmann, J. M. Bagging for path-based clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence 25*, 11 (Nov. 2003), 1411–1415.

[46] Fischer, B., and Buhmann, J. M. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence 25*, 4 (Apr. 2003), 513–518.

[47] Fisher, R. The case of multiple measurements in taxonomic problems. *Annals of Eugenics 7* (1936), 179–188.

[48] Fitz, K., Haken, L., and Christensen, P. Transient preservation under transformation in an additive sound model, 2000.

[49] Foschini, G., Golden, G., Valenzuela, R., and Wolniansky, P. Simplified processing for high spectral efficiency wireless communication employing multi-element arrays. *IEEE Journal on Selected Areas in Communications 17*, 11 (Nov 1999), 1841–1852.

[50] Fu, K. S., and Mui, J. K. A survey on image segmentation. *Pattern Recognition 13*, 1 (1981), 3–16.

[51] Fukunaga, K., and Olsen, D. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers 20*, 2 (1971), 176–183.

[52] Fulop, S. A., and Fitz, K. Separation of components from impulses in reassigned spectrograms. *Acoustical Society of America Journal 121* (2007), 1510–+.

[53] Gorsuch, R. L. *Factor Analysis*. Lawrence Erlbaum, Hillsdale, NJ, USA, 1983.

[54] Ham, J., Lee, D., Mika, S., and Scholkopf, B. A kernel view of the dimensionality reduction of manifolds, 2004.

[55] Harmeling, S., Ziehe, A., Kawanabe, M., and Müller, K.-R. Kernel-based nonlinear blind source separation. *Neural Computation 15*, 5 (2003), 1089–1124.

[56] Hartigan, J. A., and Wong, M. A. A K-means clustering algorithm. *Applied Statistics 28* (1979), 100–108.

[57] Hastie, T. *Principal Curves and Surfaces*. PhD thesis, Stanford University, 1984.

[58] Hastie, T., and Stuetzle, W. Principal curves. *Jour. Am. Statistical Assoc. 84* (1989), 502–516.

[59] Haykin, S. *Adaptive filter theory (3rd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[60] Hegde, A., Principe, J. C., Erdogmus, D., Ozertem, U., Rao, Y. N., and Peddaneni, H. Perturbation-based eigenvector updates for on-line principal components analysis and canonical correlation analysis. *Journal of VLSI Signal Processing Systems 45*, 1-2 (2006), 85–95.

[61] Herman, T., Meinicke, P., and Ritter, H. Principal curve sonification, 2000.

[62] Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol. 24* (1933).

[63] Hyvärinen, A., and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks 12*, 3 (1999), 429–439.

[64] Jackson, J. E. *A User's Guide to Principal Components*. John Wiley and Sons, New York, 1991.

[65] Jolliffe, I. T. *Principal Components Analysis*. Springer-Verlag, Berlin, 1986.

[66] Jones, M. C., Marron, J. S., and Sheather, S. J. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association 91*, 433 (1996), 401–407.

[67] Jutten, C., and Karhunen, J. Advances in blind source separation (bss) and independent component analysis (ica) for nonlinear mixtures. *Int. J. Neural Syst. 14*, 5 (October 2004), 267–292.

[68] Kalpathy-Cramer, J., Ozertem, U., Hersh, W., Fuss, M., and Erdogmus, D. Semi-supervised segmentation using non-parametric snakes for 3d-ct applications in radiation oncology. In *accepted to Machine Learning for Signal Processing* (2008).

[69] Kambhatla, N., and Leen, T. K. Fast non-linear dimension reduction. pp. 152–159.

[70] Kambhatla, N., and Leen, T. K. Dimension reduction by local principal component analysis. *Neural Computation 9*, 7 (1997), 1493–1516.

[71] Karami, E., and Shiva, M. Decision-directed recursive least squares mimo channels tracking. *EURASIP Journal of Wireless Communication Networks 2006*, 2 (2006), 7–7.

[72] Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1987), 321–331.

[73] Kaufman, L., and Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[74] Kegl, B., and Kryzak, A. Piecewise linear skeletonization using principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 59–74.

[75] Kegl, B., Kryzak, A., Linder, T., and Zeger, K. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 3 (2000), 281–297.

[76] Keogh, E., and Pazzani, M. Derivative dynamic time warping, 2001.

[77] Keogh, E., Xi, X., Wei, L., and Ratanamahatana, C. A. The ucr time series classification-clustering homepage, 2006.

[78] Kgl, B. *Principal Curves: Learning, Design, And Applications*. PhD thesis, Concordia University, Montreal, Canada, 1999.

[79] Kohonen, T., Ed. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[80] Kung, S., Diamantaras, K., and Taur, J. Adaptive principal component extraction (apex) and applications. *IEEE Transactions on Signal Processing 42*, 5 (May 1994), 1202–1217.

[81] Lambert, J. D., and Lambert, D. *Numerical Methods for Ordinart Differential Systems: The Initial Value Problem*. Wiley, New York, 1991.

[82] LeBlanc, M., and Tibshirani, R. Adaptive principal surfaces. *Journal of the American Statistical Association 89*, 425 (1994), 53–64.

[83] Lee, J. A., Jutten, C., and Verleysen, M. Non-linear ica by using isometric dimensionality reduction. In *Independent Component Analysis and Blind Signal Separation* (2004), A. P. e. C.G. Puntonet, Ed., Lecture Notes in Computer Science (LNCS), Springer-Verlag, pp. 710–717.

[84] Leymarie, F., and Levine, M. D. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis Machine Intelligence 15*, 6 (1993), 617–634.

[85] Lifshitz, L. M., and Pizer, S. M. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis Machine Intelligence 12*, 6 (1990), 529–540.

[86] McCullagh, P., and Nelder, J. A. *Generalized Linear Models*. Chapman & Hall/CRC, London, 1989.

[87] Meinicke, P., and Ritter, H. Local pca learning with resolution-dependent mixtures of gaussians, 1999.

[88] Mulier, F., and Cherkassky, V. Self-organization as an iterative kernel smoothing process. *Neural Computation 7*, 6 (1995), 1165–1177.

[89] Myers, C., Rabiner, L., and Rosenberg, A. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Signal Processing 28*, 6 (Dec 1980), 623–635.

[90] Omer, I., and Werman, M. The bottleneck geodesic: Computing pixel affinity. In *Proc. of the 2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'06)* (New York, NY, June 17–22 2006), C. Schmid, S. Soatto, and C. Tomasi, Eds., pp. 1901–1907.

[91] Ozdemir, A., and Arikan, O. A high resolution time frequency representation with significantly reduced cross-terms. *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing 2* (2000), II693–II696 vol.2.

[92] Ozdemir, A., Durak, L., and Arikan, O. High resolution time-frequency analysis by fractional domain warping. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing 6* (2001), 3553–3556 vol.6.

[93] Ozertem, U., and Erdogmus, D. Nonparametric snakes. *IEEE Transactions on Image Processing 16*, 9 (2007), 2361–2368.

[94] Ozertem, U., Erdogmus, D., and Carreira-Perpinan, M. A. Density geodesics for similarity clustering. *IEEE International Conference on Acoustics, Speech and Signal Processing* (March 31 2008-April 4 2008), 1977–1980.

[95] Pal, N., and Pal, S. A review on image segmentation techniques. *Pattern Recognition 26(9)* (1993), 1277–1294.

[96] Parra, L., Deco, G., and Miesbach, S. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation 8*, 2 (1996), 260–269.

[97] Parzen, E. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics 33*, 3 (1962), 1065–1076.

[98] Pearson, K. On lines and planes of closest fit to points in space. *Philos. Mag. 2* (1901), 559–572.

[99] Pham, D. T., and Chan, A. B. Control chart pattern recognition using a new type of self-organising neural network. *Proceedings of Institution of Mechanical Engineers 212* (1998), 115–127.

[100] Pikrakis, A., Theodoridis, S., and Kamarotos, D. Recognition of isolated musical patterns using context dependent dynamic time warping. *IEEE Transactions on Speech and Audio Processing 11*, 3 (May 2003), 175–183.

[101] Ratanamahatana, C. A., and Keogh, E. Everything you know about dynamic time warping is wrong. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004).

[102] Raykar, V. C., and Duraiswami, R. Fast optimal bandwidth selection for kernel density estimation. In *Proceedings of the sixth SIAM International Conference on Data Mining* (2006), J. Ghosh, D. Lambert, D. Skillicorn, and J. Srivastava, Eds., pp. 524–528.

[103] Reinhard, K., and Niranjan, M. Subspace models for speech transitions using principal curves, 1998.

[104] Reinhard, K., and Niranjan, M. Parametric subspace modeling of speech transitions. *Speech Commun. 27*, 1 (1999), 19–42.

[105] Ronfard, R. Region-based strategies for active contour models. *International Journal of Computer Vision 13*, 2 (1994), 229–251.

[106] Rontogiannis, A., Kekatos, V., and Berberidis, K. A square-root adaptive v-blast algorithm for fast time-varying mimo channels. *IEEE Signal Processing Letters 13*, 5 (May 2006), 265–268.

[107] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (December 2000), 2323–2326.

[108] Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D 60*, 1-4 (1992), 259–268.

[109] Sakoe, H., and Chiba, S. A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics, Budapest* (Budapest, 1971), vol. 3, Akadémiai Kiadó, pp. 65–69.

[110] Sambur, M., and Rabiner, L. A statical decision approach to the recognition of connected digits. *IEEE Transactions on Signal Processing 24*, 6 (Dec 1976), 550–558.

[111] Sandilya, S., and Kulkarni, S. R. Principal curves with bounded turn. *IEEE Transactions on Information Theory 48*, 10 (2002), 2789–2793.

[112] Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*, 5 (1998), 1299–1319.

[113] Shawe-Taylor, J., and Singer, Y., Eds. *Regularization and Semi-supervised Learning on Large Graphs* (2004), vol. 3120 of *Lecture Notes in Computer Science*, Springer.

[114] Sheather, S. J., and Jones, M. C. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological) 53*, 3 (1991), 683–690.

[115] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888–905.

[116] Silverman, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986.

[117] Smola, A., Williamson, R., and Scholkopf, B. Generalization bounds and learning rates for regularized principal manifolds, 1998.

[118] Smola, A. J., Mika, S., and Schölkopf, B. Quantization functionals and regularized principal manifolds. Tech. Rep. NC-TR-98-028, University of London, UK, 1998.

[119] Stanford, D. C., and Raftery, A. E. Finding curvilinear features in spatial point patterns: Principal curve clustering with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 6 (2000), 601–609.

[120] Tenenbaum, J. B., de Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (December 2000), 2319–2323.

[121] Tibshirani, R. Principal curves revisited. *Statistics and Computation 2* (1992), 183–190.

[122] Vaerenbergh, S. V., and Santamaría, I. A spectral clustering approach to underdetermined post-nonlinear blind source separation of sparse sources. *IEEE Transactions on Neural Networks 17*, 3 (May 2006), 811–814.

[123] Vaerenbergh, S. V., and Santamaria, I. A spectral clustering approach for blind decoding of mimo transmissions over time-correlated fading channels. In *Intelligent Systems: Techniques and Applications*, E. Hines and M. Martinez, Eds. Shaker Publishing, 2008.

[124] Van Vaerenbergh, S., Estébanez, E., and Santamaría, I. A spectral clustering algorithm for decoding fast time-varying BPSK MIMO channels. In *15th European Signal Processing Conference (EUSIPCO 2007)* (Poznan, Poland, Sept. 2007).

[125] Verbeek, J. J., Vlassis, N., and Kröse, B. A k-segments algorithm for finding principal curves. *Pattern Recognition Letters 23*, 8 (2002), 1009–1017.

[126] Verbeek, J. J., Vlassis, N. A., and Kröse, B. J. A. A soft k-segments algorithm for principal curves. In *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks* (London, UK, 2001), Springer-Verlag, pp. 450–456.

[127] Vincent, P., and Bengio, Y. Manifold parzen windows, 2003.

[128] Vogel, C. R., and Oman, M. E. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing 17*, 1 (1996), 227–238.

[129] Weinberger, K. Q., and Saul, L. K. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision 70*, 1 (2006), 77–90.

[130] Wong, A. S.-Y., Wong, K.-W., and Wong, C.-S. A practical sequential method for principal component analysis. *Neural Processing Letters 11*, 2 (2000), 107–112.

[131] Xu, C., and Prince, J. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing 7*, 3 (Mar 1998), 359–369.

[132] Yaniv, R., and Burshtein, D. An enhanced dynamic time warping model for improved estimation of dtw parameters. *IEEE Transactions on Speech and Audio Processing 11*, 3 (May 2003), 216–228.

Appendix A

Density Geodesics

We consider the problem of clustering a dataset using pairwise-affinity methods, which are based on an affinity or similarity function $w(\mathbf{x}, \mathbf{y})$ that defines how close two data points $\mathbf{x}$, $\mathbf{y}$ are, as opposed to feature-based methods such as k–means, which work directly with the feature vectors $\mathbf{x}$ and $\mathbf{y}$. Examples of pairwise-affinity methods are hierarchical (agglomerative and divisive) clustering [73] and spectral clustering [115], among others. They have the advantage of dealing more easily with clusters of complex shapes, since they do not impose a model (e.g. Gaussian-shaped clusters) on the data. While different methods use the affinities in different ways (e.g. sequentially merging data points in agglomerative clustering vs projecting the data on the eigenspace of an affinity matrix in spectral clustering), the definition of affinity is of paramount importance in all them, and it is our main focus here.

Most work uses an affinity function $w(\mathbf{x}, \mathbf{y})$ that depends only on the feature vectors for the points $\mathbf{x}$ and $\mathbf{y}$, and more specifically on the distance between $\mathbf{x}$ and $\mathbf{y}$. For example, the popular Gaussian affinity $w_\sigma(\mathbf{x}, \mathbf{y}) = \exp\left(-\left\|\mathbf{x} - \mathbf{y}\right\| / \sigma^2\right)$, which also introduces a scale parameter σ that offers more flexibility but has to be tuned to each particular dataset, often by trial and error, for the method to succeed. We call this a *context-independent* affinity function, in that it does not depend on the rest of the data set (other than $\mathbf{x}$ and $\mathbf{y}$). However, this is a very limited representation of the metric structure of the feature space, which in our view is determined by the dataset. Refer to fig. A.1. On the left, we have 3 points which

are equidistant from each other, and so $w_\sigma(A,B) = w_\sigma(A,C) = w_\sigma(B,C)$. However, in the context of the dataset shown on the right (represented as a density rather than a finite dataset) and from a clustering point of view, it makes intuitively little sense for the points to be equidistant. In order to go from A to B we have to traverse a low-density region, while from A to C there are paths (not necessarily straight) traversing high-density regions. Our objective is to define *context-dependent* affinity functions where the context is given by the entire dataset (a global context). While this might be achieved by working directly with the dataset points (e.g. by defining a particular type of graph with appropriate edge weights), we aim at a more general framework informed by differential geometry concepts where we define a density-dependent metric on the feature space. By basing our metric on a probability density function $p(\mathbf{x})$, applicable to any point $\mathbf{x}$ in the space, we are also able to define affinities between any pair of points in the space, not just between the points in the dataset. In practice, the density is estimated from the data either nonparametrically (e.g. with a kernel density estimate) or parametrically (e.g. with a Gaussian mixture trained by EM). A second component of our affinity definition is that of *paths*, more specifically *geodesics*—paths that extremize a cost functional. This captures the idea of fig. A.1(right) where, of the many paths joining A and C, we care about the one that is as short as possible in Euclidean distance while traversing high-density regions; this will be made specific in the next section. In summary, we propose a context-dependent affinity function $w_p(\mathbf{x},\mathbf{y})$, defined as the largest path affinity of all paths joining $\mathbf{x}$ and $\mathbf{y}$, where the path affinity depends on the path and the density p, and is essentially the line integral along the path of a function of p. Thus, we call this idea *affinity by density geodesics*.

The reason why we expect density geodesics to work with clustering is that the corresponding affinities should make the cluster structure obvious. From fig. A.1(right), the affinities between every pair of points in the left cluster (e.g. A, C) will be high (and similar in magnitude), while the affinities between points in different clusters

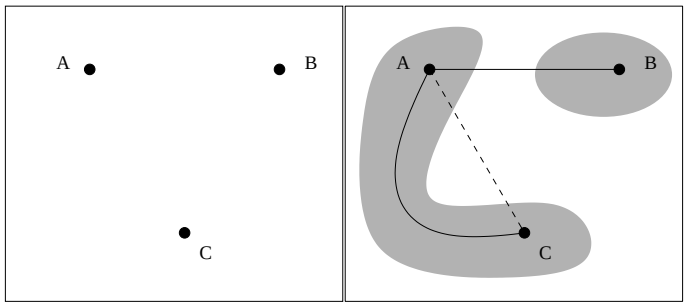


Figure A.1: Points A, B, C are equidistant in Euclidean distance (left), but data distribution (right) suggests A and C are very close to each other and very far from B.

will all be low. Thus, the affinity matrix will have a blocky aspect with near-zero across-cluster affinities and high within-cluster affinities (and so will its eigenvectors). A spectral algorithm will map the dataset to a structure consisting of widely separated, point-like clusters, which is trivial to cluster. Note, however, that this idea does not work for dimensionality reduction with pairwise-affinity methods such as Isomap [120] or Laplacian eigenmaps [6], because the internal metric structure of a cluster is almost completely erased.

Although our general formulation of clustering based on a metric induced by the data density is new to our knowledge, previous work has considered (separately) the ideas of paths between data points and of using the data density for clustering. In path-based clustering [46], one assumes a neighborhood graph (e.g. k–nearest-neighbors) on the dataset, and defines a distance (inverse affinity) as a minimax distance

$$\overline{d}_{nm} = \min_{P \in \mathcal{P}_{nm}} \max_{l=1,\ldots,|P|} d(P_l, P_{l+1})$$

where $\mathcal{P}_{nm}$ = "all paths joining data points $\mathbf{x}_n$ and $\mathbf{x}_m$". Thus, the distance is the smallest bottleneck (= longest link) over $\mathcal{P}_{nm}$. Its motivation, as in fig. A.1(right), is to obtain low across-cluster affinities—the bottleneck being the distance between the two clusters, which dominates other intermediate links. However, the minimax

distance has the undesirable property that it does not change if we change the link distances (other than the bottleneck), and is sensitive to outliers. Also, for overlapping clusters, a bottleneck may not exist (particularly for large datasets), even if the density across clusters is much lower than within clusters. Besides, the minimax distance only applies to points in the dataset, unlike our affinity, which is defined on the entire space. Several modifications have tried to address the sensitivity to outliers, by using bagging [45] and by normalizing link distances by the node degree [16]. In the method of [18], the minimax distance $\overline{d}_n m$ is modulated by a Gaussian, $\exp\left(-\overline{d}_{nm}/\sigma^2\right)$, and further the "max" is softened via another user parameter to interpolate smoothly between the two extreme cases of the maximum (= path bottleneck) and the mean (= path length). However, they applied this not to clustering but to transductive Support Vector Machines. Finally, the bottleneck geodesics [90] seek paths that are both short (geodesics) and dense (no bottlenecks). They define a local density estimate at each data point $\mathbf{x}_n$ as $\sum_{n\sim m} \|\mathbf{x}_n - \mathbf{x}_m\|^{-2}$ (where $n \sim m$ is a link in a neighborhood graph). They then combine this with the path length in various ways, e.g. by normalizing the link length by the density estimate, or by normalizing the path length by the path bottleneck. Like the other methods, this yields affinities only for the dataset points.

An alternative approach to introducing context in the definition of pairwise affinities is by constructing better neighborhood graphs. The usual graphs (k–nearest-neighbors, ϵ–ball, fixed grid for image segmentation) enforce global parameters for each data point no matter its context, so every point must have k neighbors or connect to all points at distance ϵ or less. In the perturbation approach of [14], multiple graphs are built on jittered duplicates of the dataset and then combined into an average graph that adapts locally to the data and is by construction more robust to noise. However, in this method, point pairs such as (A, C) in fig. A.1 are still assigned a low affinity, and the path in the graph counts distances but not density.

A.1 Density Geodesics

A mathematically rigorous distance definition should be given as a function of the data probability density function. The data pdf $p : \mathbb{R}^n \rightarrow \mathbb{R}^+$ should warp the metric space $\mathbb{R}^n$ as a metric imposing mass, such that the distance metric is inversely proportional with the data pdf - lower the pdf higher the distance and visa versa. In this context, under the presence of a nonuniform pdf, distances between data pairs based on Euclidean distance are not accurate and the Euclidean distance metric, which is uniform over the entire space $\mathbb{R}^n$, is only optimal if the data pdf is uniform. Using this fact, we postulate the Euclidean metric as the canonical metric under uniform density and generalize this to an arbitrary pdf, defining density geodesics. We achieve this using the transformation that maps the uniform function to an arbitrary pdf.

A suitable distance metric that considers the data pdf as a mass imposing function has to have the following two properties: (i) the metric should emphasize the valleys in the data density, should promote large density and penalize low density regions, (ii) the metric definition should properly take into account local and directional stretching effects imposed on the data as well - very similar to the stretching effects of the Mahalanobis distance. In a differential geometric framework, the infinitesimal length of a curve segment depends on its direction (tangent vector), the local stretching (Riemann metric of smooth pdf manifolds) to make the metric locally invariant to stretching.

Let $f : \mathbb{R}^n \rightarrow [0,1]^n$ be an invertible map that transforms the pdf p to a uniform one in the unit hypercube. Due to change of variables, the Euclidean length of a curve in the unit-cube under a uniform density has to be modified to utilize a metric $\mathbf{M}(\mathbf{x}) = (\nabla^T \mathbf{f}(\mathbf{x}) \nabla \mathbf{f}(\mathbf{x}))^{-1}$. Assuming a density penalizing/promoting measure h, we define the length of a curve $\mathbf{c}$ in $\mathbb{R}^n$ under the pdf p as

$$l_{\mathbf{c}} \doteq \int_0^1 h(p(\mathbf{c}(t)))(\dot{\mathbf{c}}^T(t)\mathbf{M}(\mathbf{c}(t))\dot{\mathbf{c}}(t))^{1/2} dt \tag{A.1}$$

Under this definition of curve length, the distance between two points $\mathbf{x}$ and $\mathbf{y}$ is the length of the shortest curve (geodesic) connecting them: $d(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{c}} l_{\mathbf{c}}$ such that $\mathbf{c}(0) = \mathbf{x}$ and $\mathbf{c}(1) = \mathbf{y}$. The minimization of the functional in (A.1) requires calculus of variations and is computationally expensive. For a given iid data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with sufficiently large N drawn from p, a simple yet sufficient approximation to the geodesic could be determined by restricting the search to the paths over a fully- or partially-connected neighborhood graph. We utilize the ϵ-ball neighborhood graph where an edge is included in the graph if the length of the straight line connecting $\mathbf{x}_i$ to $\mathbf{x}_j$ as measured by (A.1), $l_{ij} < \epsilon$ and employ a shortest path search algorithm [12].

In pairwise affinity clustering, for sharp block-diagonal affinity matrices one needs to penalize geodesics that pass through relatively low density regions and promote geodesics that remain in high density regions. The bottleneck clustering principle [46] is also aimed at achieving this, however, the length of a curve is measured not as an integral over the curve but as an extremum of local length, which occurs at such valleys. By selecting monotonically decreasing $h : \mathbb{R} \to \mathbb{R}$ such that $\lim_{p \to 0} h(p) = \infty$ and $\lim_{p \to \infty} h(p) = 0$, this could be achieved. Alternatively, one could simply select the neutral measure of $h(p) = 1$.

A.1.1 One Dimensional Case Studies

To illustrate the proposed metric independently from the geodesic search step, we will present three examples on one dimensional data densities: (i) distances under a uniform density; (ii) piecewise uniform clusters; and (iii) distances under an arbitrary pdf p.

Uniform Density: Consider the distance between two points a/ε and b/ε, where $\{a, b\} \in [0, 1]$, under a uniform density in the interval $[0, 1/\varepsilon]$ as shown in Figure A.2. The function that maps this density to a uniform density in $[0, 1]$ has a constant slope of ε; thus, the metric is $M(x) = 1/\varepsilon^2, x \in [0, 1]$. For the contour c

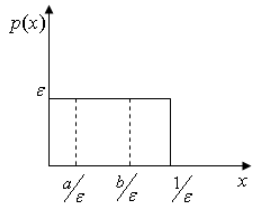


Figure A.2: Uniform density

.

connecting a to b, we have $\dot{c} = |b-a|/\varepsilon$. Assuming $h(p) = 1$ and substituting all expressions in (A.1), we obtain $l_{[a/\varepsilon, b/\varepsilon]} = |b-a|/\varepsilon^2$. The inverse square dependency of this metric on ε is intuitively interpreted as one factor coming from the scaling of the actual Euclidean distance and another from the local stretching of the density.

Piecewise Uniform Clusters: Consider a piecewise uniform density shown in Figure A.3, and its local metric

$$p(x) = \begin{cases} 1/2 - \varepsilon & \text{if} \quad x \in [0,1] \cup [2,3] \\ 2\varepsilon & \text{if} \quad x \in [1,2] \\ 0 & \text{otherwise} \end{cases}$$
$$M(x) = \begin{cases} 1/(1-2\varepsilon)^2 & \text{if} \quad x \in [0,1] \cup [2,3] \\ 1/(2\varepsilon)^2 & \text{if} \quad x \in [1,2] \\ \infty & \text{otherwise} \end{cases} \tag{A.2}$$

For $\{a, b\} \in [0, 1]$ and $c \in [2, 3]$, we are interested in the distances $d(a, b)$ and $d(a, c)$ assuming $h(p) = 1$. For $d(a, b)$, we have $\dot{c} = |b-a|$ and the integral can be computed over the segment $[0, 1]$ as $l_{[a,b]} = 2|b-a|/(1-2\varepsilon)$. For $d(a, c)$, the integral needs to be computed in three segments. Letting $c(t_1) = 1$ and $c(t_2) = 2$ for the curve connecting a to c, we can determine that $l_{[a,c]} = |b-a|\{2t_1/(1-2\varepsilon) + (t_2-t_1)/(2\varepsilon) + 2(1-t_2)/(1-2\varepsilon)\}$. Note that $\lim_{\varepsilon \to 0} l_{[a,b]} = 2|b-a|$, while $\lim_{\varepsilon \to 0} l_{[a,c]} = \infty$.

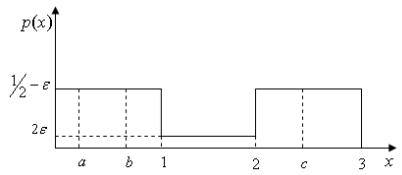


Figure A.3: Piecewise Uniform Clusters

.

Arbitrary PDF: For a univariate pdf $p(x)$, the invertible function that maps the random variable to a uniform density in $[0, 1]$ is its cumulative density function (cdf). Consequently, its derivative is the pdf, thus the metric becomes $M(x) = p^{-2}(x)$. For an arbitrary $h(p)$ measure, by substituting $M(x)$ and $\dot{c}(t) = (b-a)$ in (A.1), we can determine that $d(a,b) = |b-a| \int_0^1 h(p(a+(b-a)t))p^{-1}(a+(b-a)t)dt$. This shows that appropriate selection of $h(p)$ and consideration of local directional metrics will modify the Euclidean distance in the desirable manner.

A.2 KDE-based Implementation of Density Geodesics

In this section, we will present our particular implementation of density geodesics. Note that the distance definition is based on the data probability density, and any density estimation technique can be used in this framework.

Data densities may take complex forms, and in general, it might be very hard to find a suitable parametric family. After selecting the parametric form (for example the number of components for a Gaussian mixture density estimate), model fitting is achieved by a likelihood maximization over the data samples. At this point, KDE offers a nonparametric alternative with a flexibility of adaptation to different data structures. Mainly for this adaptation property, we will use KDE for our implementation here. However, note that some density estimation schemes might

be more advantageous in some specific scenarios, and for these cases, the density estimation step can easily be replaced.

As mentioned earlier, a significant consideration in KDE is the selection of the kernel bandwidth. As in the principal curve implementations, we again use leave-one-out maximum likelihood kernel bandwidth here [1].

Another point is the selection of the penalty function $h(p)$. To promote high density regions and penalize low density regions, this function should be selected such that $\lim_{p\to 0} h(p) = \infty$ and $\lim_{p\to\infty} h(p) = 0$, and a suitable family of penalty functions could be $h(p) = p^{-\alpha}$. In our experiments, we used $h(p) = p^{-1}$ and obtained satisfactory results.

An important observation is that the data pairs that are far from each other require much more computation. At this point, some simplifications to decrease the computational complexity can be implemented into the distance evaluation. One obvious simplification is to employ a neighborhood graph and evaluate the edges on this graph only, assuming that the edges in shortest path that will be sought in the final step are all in this graph. This leads the intercluster affinities to turn out to be infinity. Further computational savings can be achieved by setting a maximum value to pairwise distances, assuming that this edge will not be in the shortest path or the difference between the preset maximum and the actual value is negligible. Table A.1 summarizes our KDE based implementation, including these optional steps.

A.3 Experimental Results

In our experiments we concentrate on spectral clustering, but our definition is equally applicable to hierarchical clustering and other affinity methods. We compare the Euclidean distance (yielding the widely used usual Gaussian affinity), low

[1] For a more detailed discussion of using maximum likelihood method to evaluate fixed or data dependent kernel bandwidths please refer to Section 3.1.3

Table A.1: Summary of the KDE-based Density Geodesics Implementation

1. Select the kernel function $K(\cdot)$.

   Use the ML procedure given in Section 3.1.3 or use any other method.

   Anisotropic and data dependent variable bandwidth kernels lead to better pdf estimates, but they are computationally more expensive.

2. (**optional**) Determine the pairs of data points that the direct path will be evaluated.

   Keep in mind that this step is optional. A neighborhood graph is not required, but can be used to save computation.

   One can use any neighborhood graph here. Where K-nearest neighbor graph and ϵ-ball are the most commonly used choices, more elaborate options are also can be used [14].

3. For all the pairs in the dataset (or connected pairs in the neighborhood graph) integrate $p(\mathbf{x})^{-1}$ along the line that connects these data points. Build the matrix $\mathbf{D}$, where $\mathbf{D}_{ij}$ is $\int_{\mathbf{x}_i}^{\mathbf{x}_j} p(\mathbf{x})^{-1} d\mathbf{x}$ and the integration path is the line that connects these data samples. To evaluate the numerical integration we use the adaptive Simpson quadrature method (*quad* function in MATLAB).

   We estimate the data probability density using $p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} K_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i)$ This KDE equation is given for the most general case of data dependent anisotropic kernels, where Σ_i is the kernel function covariance for the i^{th} sample. For isotropic kernels, simply use a distance, or for fixed bandwidth kernel function drop the dependency on the sample index i.

4. Input $\mathbf{D}$ into Dijkstra's algorithm and evaluate shortest distances over the pairwise $p(\mathbf{x})^{-1}$ distances.

density separation (LDS) algorithm [18], and density geodesics. All three examples have two natural clusters, and we use the eigengap between the second and third eigenvalues of the affinity matrix to quantify the performance. As the difference between these two eigenvalues approaches to -1, the desired two cluster solution is more healthy. In Figure A.4, we present result for an easy clustering problem with two Gaussian clusters with 100 samples each. As all the clusters are well separated, all three methods return blockwise distance matrices for this example. The peak performance of LDS is greater than Euclidean distances, yet still lower then density geodesics. The interval of σ values that gives the correct clustering result is much wider for the density geodesics. The dataset (top left), and the eigengap (bottom left) are shown along with the distances (top row) and the corresponding similarity values (bottom row) at their best performance for Euclidean distance, LDS, and density geodesics (from left to right).

As some inter-cluster Euclidean distances are less than some in-cluster Euclidean distances, ring dataset lays a harder clustering problem, and has been widely used in clustering papers. In this example, density geodesics also demonstrate superior performance as compared to LDS and Euclidean distance, with a bigger eigengap for a wider interval of σ. Results are given in Figure A.5 similarly to the previous example.

One problem with bottleneck methods is that due to noise or heavytailed distributions, a bottleneck may not exist. Here we show an example where two Gaussian clusters are next to each other. The pdf drops significantly at the cluster boundary, yet there is no strong bottleneck due to the samples around the boundary. As shown in Figure A.6, density geodesics provide a better eigengap for a wider range of σ in this example as well.

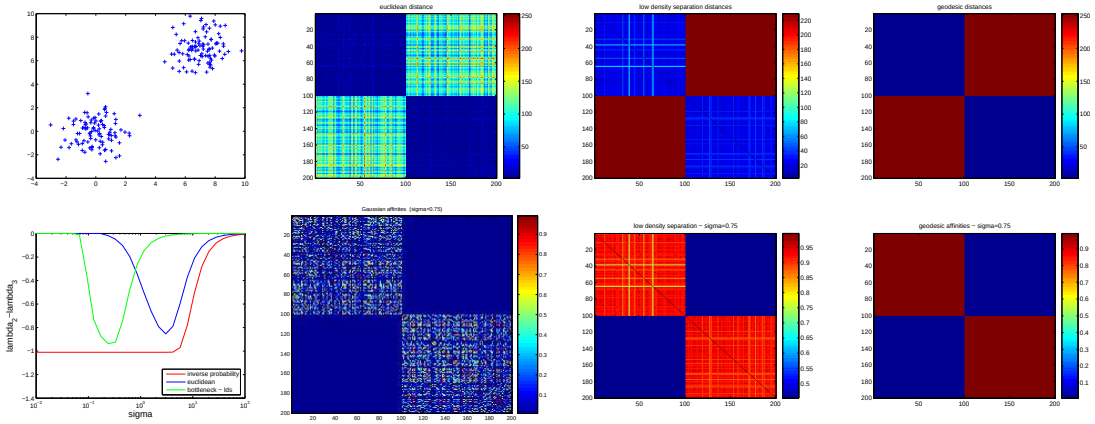


Figure A.4: Gaussians dataset

.

A.4 Discussion

The problem of affinity measure selection is at the core of similarity based clustering techniques. Various propositions include the usual algebraic distance measures as well as path-bottleneck geodesics along certain neighborhood graphs. Intuitively, the distance (inverse affinity) between two data points cannot be assessed without any regard to the context set forth by the distribution of other data samples. In principle, distances between pairs should be influenced strongly by the probability distribution of data. Stating from the postulate that Euclidean distances are canonical for data points drawn from unit-uniform densities, we develop a principled path length measure rooted in Riemannian geometry. It has been demonstrated theoretically and through analytical calculations for specific one-dimensional case studies that the proposed distance metrics satisfy desirable invariance properties. For high dimensional cases, in order to reduce the computational complexity, the metric is simplified.

Density geodesics provide a novel distance definition that improves on the path

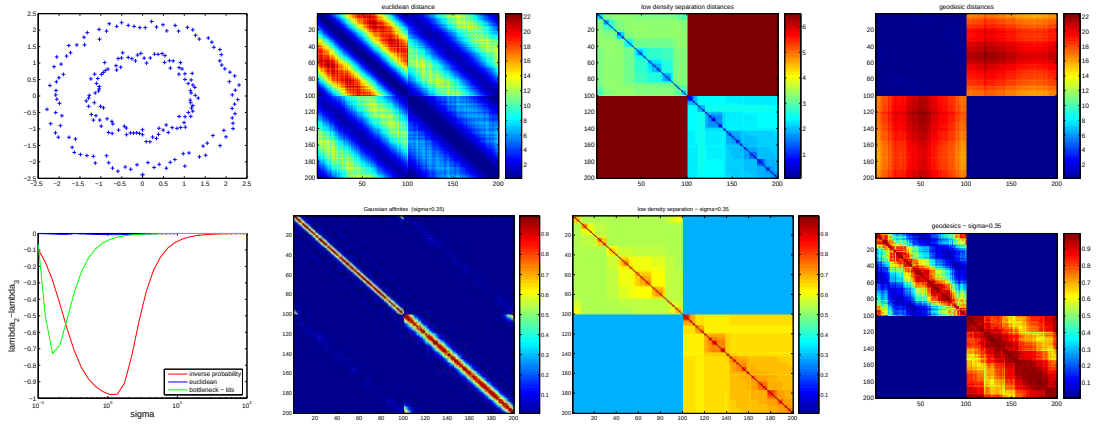


Figure A.5: Rings dataset

.

bottleneck approaches and defined as a function of data probability density. Bottleneck distance does not change if one changes the distances that are not the bottleneck (so graph-bottleneck completely erases the in-cluster structure), is sensitive to noise and requires a good neighborhood graph. Density geodesics, on the other hand, provides noise robustness and does not require a neighborhood graph. In the implementation, the fully connected graph is replaced with a neighborhood graph only to save computations, not to find the bottlenecks by using the graph. Experimental results demonstrate that the proposed distance metric yields blocky affinity matrices, which will yield to clear clustering solutions.

A promising prospect is to develop a semisupervised learning method based on density geodesics. Almost all semisupervised learning algorithms assume that there is a valley in the probability density around the class boundary, and density geodesics find a natural use here. Another future prospect is to employ density geodesics in a supervised scenario, where the geodesics are evaluated over the class conditional probability density functions.

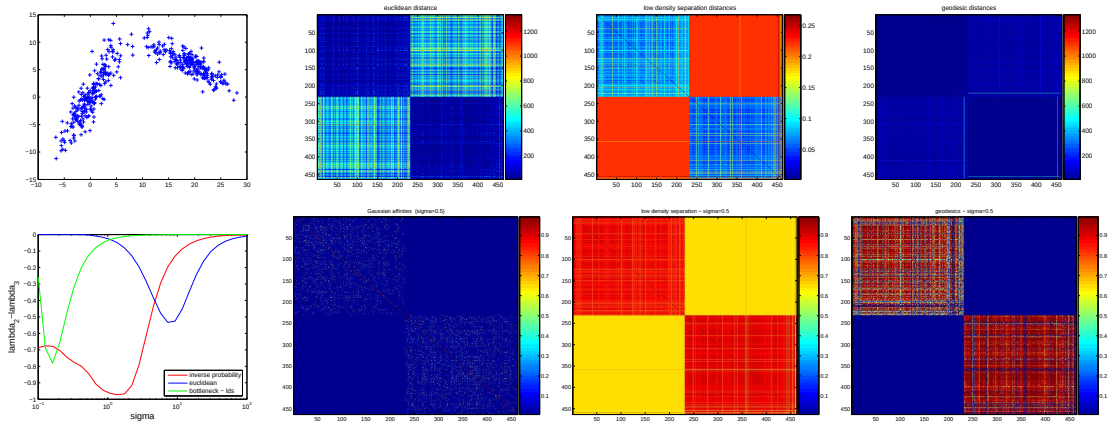


Figure A.6: Overlapping Gaussians dataset

.

Biographical Note

Umut Ozertem was born in Izmir, Turkey, on November 29^{th}, 1980. He received his B.S. in Electrical and Electronics Engineering in 2003 from the Middle East Technical University, Ankara, Turkey. Between June 2003 and August 2004, he worked as a researcher at the Multimedia Research Group in TUBITAK (Technical Research Council of Turkey) under the supervision of Prof. A. Aydin Alatan. Since 2004, he has been working towards his Ph.D. in the Computer Science and Electrical Engineering Department at Oregon Health and Science University under the supervision of Prof. Deniz Erdogmus. He received his M.S. in Electrical Engineering in 2006 from the Oregon Health and Science University. His current research interests include adaptive and statistical signal processing, information theoretic approaches in machine learning algorithms, and nonparametric methods.