UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology

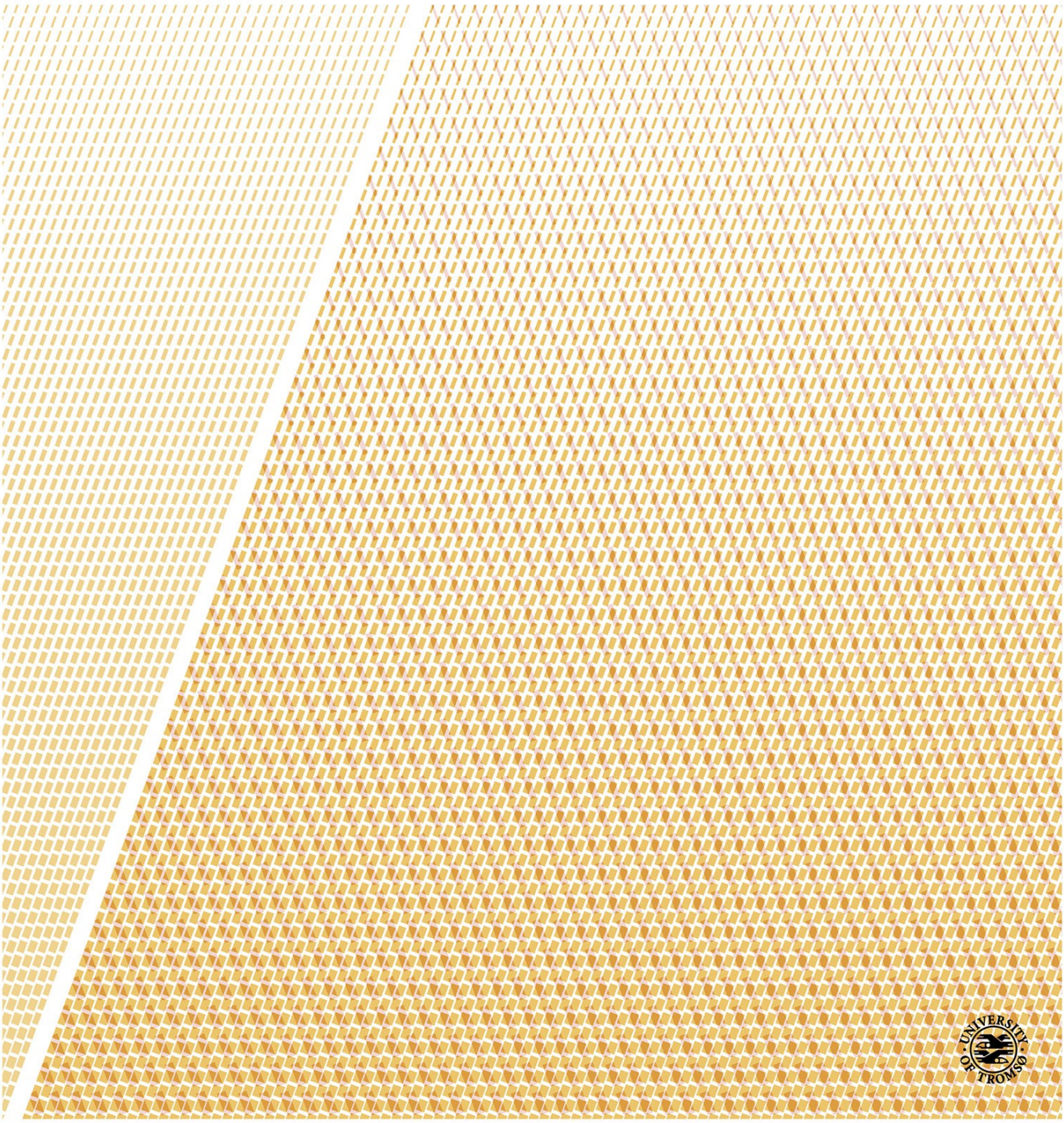
Department of Physics and Technology

Machine Learning using Principal Manifolds and Mode Seeking

—

Jonas Nordhaug Myhre

A dissertation for the degree of Philosophiae Doctor – August 2016



Abstract

A wide range of machine learning methods have taken advantage of density estimates and their derivatives, including methodology related to principal manifolds and mode seeking, finding use in a number of real applications.

However, research concerned with improving density derivative estimation and its practical use have received relatively limited attention. Also, the fact that the derivatives of a distribution over a point set can provide a statistical framework for manifold learning has not yet been used to its full potential.

The aim of this thesis is to help fill these gaps, and to provide novel machine learning algorithms and tools based on principal manifolds using density derivatives. We present three different lines of works aiming towards this goal.

The first work presents a fast and exact kernel density derivative estimator. The method takes advantage of the fact that the derivatives of a multivariate product kernel can be decomposed into a product of univariate differentiations. By cutting redundant multiplications we obtain significant speedup while retaining an exact estimator.

Next, we present a novel algorithm for manifold unwrapping based on tracing the gradient flow along a manifold estimated using density derivatives. This allows a direct and geometrically intuitive approach consistent with theory from differential geometry. Promising results are shown on both real and synthetic data sets.

Finally, we provide a novel framework for robust mode seeking. It is based on ensemble clustering and resampling techniques. This allows a clustering algorithm that is both robust with respect to parameter choices as well as being capable of handling data sets of very high dimension. Concretely, we build the ensemble by running multiple instances of a k nearest neighbor mode seeking algorithm. We show good results on benchmark tests, as well as a case study involving medical health records.

Acknowledgements

Robert Thank you for inviting me into the world of Machine Learning. It has been very fun, but also very challenging. A good combination indeed! Every time we have small meetings and chats I get very inspired and feel great support. This skill and combined with your great knowledge of Machine Learning and its community has been invaluable to me.

Deniz Thank you for letting me visit your lab and introducing me to the world of principal manifolds. The pace, rigor, efficiency and knowledge of you and your lab was truly inspiring.

Co-authors and colleagues First of all sorry for grouping you together in a big lump like this.. Matineh and Devrim, I really enjoyed our work on the principal manifold stuff! You are both strong mathematicians and I have my strengths elsewhere (...) so I think we made a good team. Even though the time difference is a problem I hope we can continue. Tromsø-colleagues: thank you for keeping up with my office antics. Seeing, and experiencing, the growth of the Machine Learning group has been very rewarding to me (this also goes to Robert). Hopefully we can continue to collaborate in the future. Regardless of what happens I will continue to draw on your desks when you are not there.

Kjærsti You deserve to be on top of this list. Unfortunately the unwritten rules of academia and my fear of breaking them as I write this has it otherwise. Your ability of being strict and pushing me forward, while at the same time being so kind and forgiving at the right moments I have not yet seen in another person. Together with Bertil, Petra and Sofus you keep me focused on a completely different level.

To the rest of my family: Thank you for standing by me and taking care of my kids and cats. Without you guys this would not have been possible.

I would also express my gratitude to the members of the committee, Mark and Aasa, for spending time and effort by reading this thesis.

To those I forgot to thank: There is either a reason that I forgot to thank you or there is not. In any case you get a big thank you from me, completely free of charge!

Cheers, Jonas.

Contents

Abstract i
Acknowledgements ii

1 Introduction **1**
1.1 Motivating examples 2
1.1.1 Nonlinear structure in data 3
1.1.2 Parameter sensitivity in unsupervised learning 5
1.2 Short summary of papers 7
1.3 Reading guide 9

I Methodology and context **10**

2 Probability density estimation **11**
2.1 Parametric models for density estimation 12
2.2 Non-parametric methods for density estimation 13
2.3 Kernel density estimation 13
2.3.1 Gradient and Hessian of the KDE 15
2.3.2 Selection of the bandwidth parameter 17
2.3.3 Product kernels and derivatives 18
2.4 k-nearest neighbor density estimation 19

3 Differential geometry in machine learning **20**
3.1 Manifolds and related concepts 21
3.2 Manifold learning 24
3.2.1 Linear methods for manifold learning 25
3.2.2 Nonlinear methods for manifold learning 27
3.2.3 Calculating geodesics 30

3.2.4 Intrinsic manifold learning methods 32

4 Principal manifolds 33
4.1 Principal curves 33
4.1.1 Extensions and alternative formulations 34
4.2 Principal manifolds as ridges of the probability density function 36
4.2.1 Subspace constrained gradient flow: projecting noisy points onto the ridge 38

5 Unsupervised learning – Clustering 41
5.1 Density based clustering 42
5.1.1 Clustering by level sets 44
5.1.2 Mode seeking – Clustering by following the gradient . . 45
5.2 Ensemble methods in clustering 46

II Summary of research 48

6 Paper I - Computationally Efficient Exact Calculation of Kernel Density Derivatives 49
6.1 Contributions by the author 50

7 Paper II - Manifold unwrapping using density ridges 52
7.1 Contributions by the author 53
7.2 Paper III - Invertible nonlinear cluster unwrapping 54
7.3 Contributions by the author 54

8 Paper IV - A robust clustering using a kNN mode seeking ensemble 55
8.1 Contributions by the author 56

9 Concluding remarks 57
9.1 Short discussion: Weaknesses and alternative approaches . . . 58
9.2 Future work 59

III Included papers 61

10 Paper I 62

11 Paper II **75**

12 Paper III **119**

13 Paper IV **126**

A Diffeomorphic projection model using landmark matching **143**
A.1 Learning the diffeomorphic Projection Model 143
A.2 Projecting Out-of-Sample Test Data 146

List of Figures

1.1 This sketch illustrates two concepts: (1) how nonlinear projections can be used to estimate an underlying nonlinear structure and (2) how a smooth underlying surface of lower intrinsic dimension can be a good approximation in the case of noisy non-linear data. 3

1.2 The top three principal components of the 1 digit in the MNIST data set (blue), the smooth underlying manifold estimated by principal manifolds (green) and a smooth interpolation along the manifold between two arbitrary digits (red). The points along the red curve are mapped back to the input space and shown in the top right corner. 4

1.3 Illustration of the data space partitioning induced by the gradient flow of a probability density estimate. The data set is a mixture of five Gaussians. 6

1.4 Non-linear toy data that cannot be clustered correctly by a mode seeking algorithm. 7

1.5 An overview of the context of the contributions of this thesis. The topic of each paper is marked with red. KDE is kernel density estimation and kNN is k-nearest neighborhood density estimation. 8

2.1 Example of kernel density estimation for a sample from two Gaussian distributions with unit variance and mean 0 and 10 in $\mathbb{R}$. The bandwidth parameter h determines the width of each of the kernel functions (blue) placed over each data point. 14

3.1 Concepts from differential geometry illustrated on the sphere embedded in $\mathbb{R}^3$. 22

3.2 A common way of defining the smoothness of manifolds: the maps $\phi \circ \psi^{-1}$ and $\psi \circ \phi^{-1}$ should be smooth. 23

4.1 Illustration of the self consistent property. The principal curve is the expected value of all points orthogonally projected onto the curve. 35
4.2 Example of the density ridge for a sinusoid sampled with $\mathcal{N}(0, 0.03I)$ additive noise. 37
4.3 Example of the SCMS algorithm. A noisy one-dimensional manifold is sampled with noise 4.3c, and we see that the ridges capture the underlying structure. In 4.3a and 4.3b, we see a comparison between zoomed in versions the mean shift and the SCMS trajectories. 39
4.4 Example of the SCMS for a two-dimensional manifold sampled with noise. 40

5.1 A taxonomy of clustering approaches. The contributions of this thesis are mostly in the areas marked with red. 43
5.2 The two main frameworks of non-parametric density based clustering. 44

6.1 The key features of the efficient kernel density derivative algorithm. 51

Chapter 1

Introduction

Important features of the probability density function such as critical points (modes), curvature, ridges and valleys, and to some extent cluster structure can be described using derivatives [151, 7, 158, 138, 93, 46, 51, 36].

Already in 1977[1] R. S. Singh pointed out important problems that can be solved by estimating a density and its derivatives [158]. Even before that, in 1975, Fukunaga and Hoestler [82] introduced the first version of the famous mean shift algorithm, widely used for clustering.

In the following years, a wide range of methods have taken advantage of density estimates and their derivatives. In astronomy, the filamentary geometry of the cosmic web has inspired many works [44, 86, 160, 45]. The geometry of roads in images has been analysed using density derivative based principal curves [40, 127, 135]. Similarly principal curves have also been used in medical- and neuroimaging [186, 13]. Other applications include estimating economic summary indexes [188], tracking and reconstruction in neutrino oscillation experiments [8] and ice floe detection [11]. The gradient field of the density derivatives has been used with great success. Examples include the Microsoft's Kinect® computer vision system [156], object tracking [52, 134, 182] and brain connectivity visualization [25].

More recent theoretical works include Kullback-Leibler divergence approximation [151], least squares log-density gradients used for clustering [150] and

[1] quite some time ago in the machine learning timeline

higher order kernels for density derivatives [106].

Topics related to principal manifolds and mode seeking have thus had a considerable impact on machine learning and related fields, as exemplified above.

However, the field of research concerned with improving density derivative estimation have received comparably limited attention. Also, the fact that the derivatives of a distribution over a point set can provide a statistical framework for manifold learning [135, 85] has not yet been used to its full potential.

The aim of this thesis is to help fill these gaps, and to provide novel machine learning algorithms and tools based on principal manifolds and modes using density derivatives.

Towards that end we present novel estimators and methods that takes advantage of probability density derivatives. We make use of the close connection between probability density derivatives and geometry – in the form of principal manifolds – to propose new algorithms for both manifold learning and unsupervised learning.

Specifically, our objectives in this thesis are:

- Faster methods and estimators involving probability density derivatives.
- Novel applucitions of principal manifolds inspired by differential geometry.
- More robust algorithms in the application of probability density derivatives.

In the following of this, we provide some illustrative examples.

1.1 Motivating examples

To illustrate our main ideas and the methodological setup considered in the course of this thesis, we have included two motivating examples. The first illustrates how density derivatives can be useful tools in machine learning,

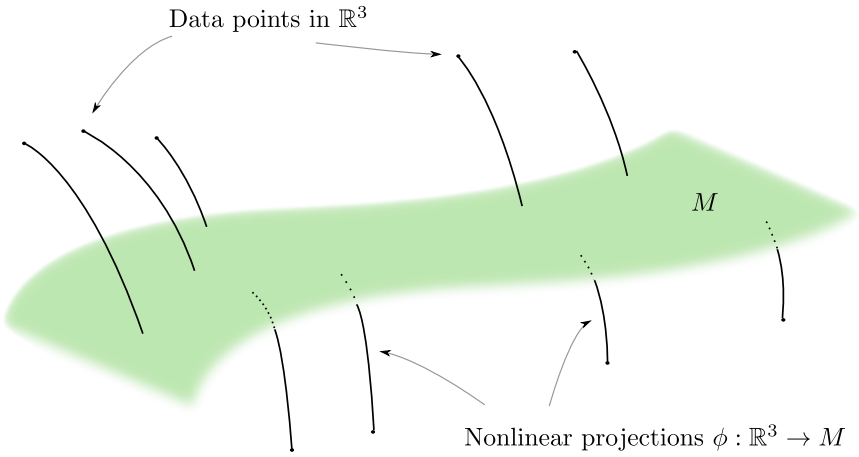


Figure 1.1: This sketch illustrates two concepts: (1) how nonlinear projections can be used to estimate an underlying nonlinear structure and (2) how a smooth underlying surface of lower intrinsic dimension can be a good approximation in the case of noisy non-linear data.

especially in cases where the data manifest nonlinear structure. The second illustrates problems that arise in evaluating density derivative estimation due to the use of non-parametric density estimates. This poses several difficulties, especially in unsupervised learning applications such as clustering, where no ground truth is available and methods such as cross-validation cannot be adopted to establish the best estimators.

1.1.1 Nonlinear structure in data

How do we deal with non-linear structure in data? In Figure 1.1 we see an illustrative example of how nonlinear structure in data can appear in practice. We consider a set of arbitrary measurements, which are drawn from a smooth non-linear hypersurface (a manifold M) and have been corrupted by noise. The geometry of the non-linear structure typically comes from the data-generating process, e.g. images that rotate, translations or body movements in medical applications or in principle any significant features that change smoothly over time or space [73, 180, 79].

Let us assume that we want to perform statistical inference, e.g regression,

along the smooth underlying manifold in the case depicted in Figure 1.1. A reasonable workflow would then be: (1) estimate or *learn* the structure of M, (2) project the data[2] onto the smooth structure (3) perform inference along M, (4) map the input back to the ambient space ($\mathbb{R}^3$ in the example). The last stage is optional, depending on context. For example in the case of images a map back to the input space is desirable to be able to visualize the results (this is known as the pre-image problem in kernel methods [117]).

In this thesis we have investigated how density estimate derivatives can be used to solve some of the steps in the general workflow presented above. In

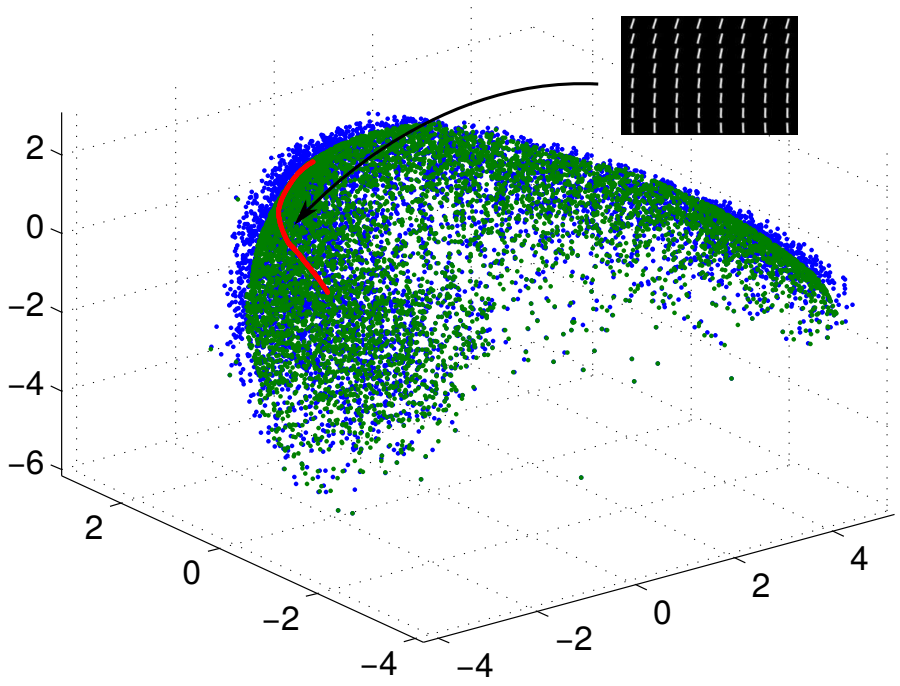


Figure 1.2: The top three principal components of the 1 digit in the MNIST data set (blue), the smooth underlying manifold estimated by principal manifolds (green) and a smooth interpolation along the manifold between two arbitrary digits (red). The points along the red curve are mapped back to the input space and shown in the top right corner.

Figure 1.2 the first three principal components of a set consisting of images of

[2]We assume there are many more data points than those illustrated in Figure 1.1.

handwritten digits of the number one – the MNIST data set [119] – are shown in blue. The green dots represents the smooth two-dimensional principal manifold estimated, a structure represented by the gradient and Hessian of a kernel density estimate. Once the smooth underlying surface (manifold) is estimated, inference can take place. In this example we perform smooth interpolation (non-linear regression) by calculating the shortest distance – called a geodesic – along the manifold between two arbitrarily chosen points (shown in red). An alternative approach would be to *unfold* the manifold and use a linear interpolation method in the unfolded space.

Finally, the interpolated points are mapped back to the input space using, for example, a two-layer neural network enabling visual inspection (this is is possible due to the extensive amount of training data available in this data set[3]). Indeed we see that the digits transform smoothly from the tilted number 1 (top left) to the more horizontal digit (bottom right).

1.1.2 Parameter sensitivity in unsupervised learning

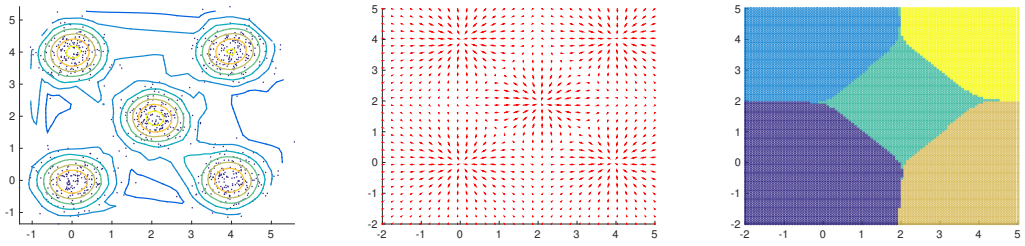
Another important class of density derivative applications is unsupervised learning, most often implemented by *mode seeking* methods [51, 46, 61, 125]. This is also called population clustering in some settings [34], and has the benefit that the definition of a cluster is defined directly by the probability density. It is based on the fact that the gradient flow of the probability density (with some additional technical constraints [34]) induces a partition of the input space.

An example of this is shown in Figure 1.3 where we see a mixture of five Gaussians and the estimated density, Figure 1.3a. The corresponding gradient flow field is shown in Figure 1.3b, and the induced partition of the input space in Figure 1.3c.

Mode seeking methods are almost exclusively based on non-parametric density estimates, which have several benefits, but also an inherent sensitivity to the critical bandwidth[4] of the estimate that is hard to overcome in clustering settings. For this reason, mode seeking methods have been most successful

[3]This constitutes parts of research related to this thesis that is not yet published.

[4]The bandwidth h in kernel density estimation and k in nearest neighbor methods [166]. See Chapter 2 for details.



(a) Data set and the contours of the estimated (kernel) density.

(b) Gradient flow of the estimated (kernel) density function.

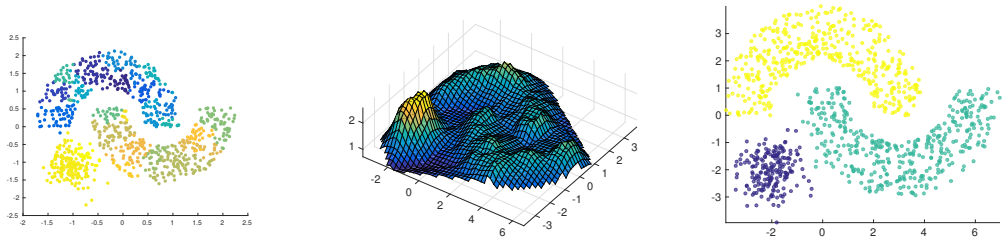
(c) Partition of input space induced by gradient flow.

Figure 1.3: Illustration of the data space partitioning induced by the gradient flow of a probability density estimate. The data set is a mixture of five Gaussians.

in applications where the clustering process is mostly a pre-processing stage such as object tracking [52, 175], pose estimation [156] or 3D reconstruction [171].

Part of these issues are due to the fact that a cluster has to be *unimodal* to be picked up by a mode seeking algorithm. This is obviously a strong limitation. In fact, results from topological analysis of multivariate Gaussian mixtures state that even in parametric mixture models the number of modes can be higher than the number of clusters [144].

An elegant way of avoiding parameter tuning and enabling simple algorithms to handle greater variation in both cluster structure and cluster separation is represented by *ensemble methods* [161, 75, 173]. Instead of a single clustering algorithm, an ensemble evaluates multiple partitions with different parameters, initializations or both, and measures agreement across all partitions. In Figure 1.4 we see a non-linear data set that cannot be correctly clustered by a mode seeking algorithm. However, when applying an ensemble of mode seeking algorithms with random parameter initialization, the true cluster structure is captured in Figure 1.4c.



(a) Clustering using a single mode seeking run.

(b) k-nearest neighborhood density with $k = \sqrt{n}$.

(c) Clustering result using a mode seeking *ensemble*.

Figure 1.4: Non-linear toy data that cannot be clustered correctly by a mode seeking algorithm.

1.2 Short summary of papers

The following papers are included in this thesis:

- M. Shaker, J. N. Myhre, D. Erdogmus, "**Computationally Efficient Exact Calculation of Kernel Density Derivatives**", published in *Journal of Signal Processing Systems*, December 2015, Volume 81, Issue 3, pp 321–332.
- J. N. Myhre, M. Shaker, M. D. Kaba, D. Erdogmus, "**Manifold unwrapping using density ridges**", unpublished manuscript[5].
- M. Shaker, J. N. Myhre, M. D. Kaba, D. Erdogmus, "**Invertible nonlinear cluster unwrapping**", published in the *Proceedings of the 2014 IEEE International Workshop on Machine Learning for Signal Processing*.
- J. N. Myhre, K. Ø. Mikalsen, S. Løkse and R. Jenssen, "**A robust clustering using a kNN mode seeking ensemble**", in review with *Pattern Recognition*.

Paper I: We suggested and implemented a new tree-based algorithm for removing redundant multiplications in kernel density derivative estimation. This leads to a fast *exact* estimator, which is not based on approximation.

[5]A pre-print was posted on arXiv.org in March 2016, http://arxiv.org/abs/1604.01602

Paper II and III: These two papers present novel ideas for manifold unwrapping using *density ridges*. Density ridges are manifold estimators based on the estimated gradient and Hessian of a probability density. In addition the algorithms were implemented in parallel using adaptive ODE solvers, such that a significant speedup was achieved.

Paper IV: The final paper proposes a novel algorithm for non-parametric density based clustering. We used concepts from *ensemble clustering*, resulting in an algorithm which is more robust towards parameter sensitivity and is capable of handling high dimensional data. Parameter sensitivity is one of the biggest problems in non-parametric density estimation.

Figure 1.5 shows how our contributions are related to the overall use of probability density estimation in machine learning.

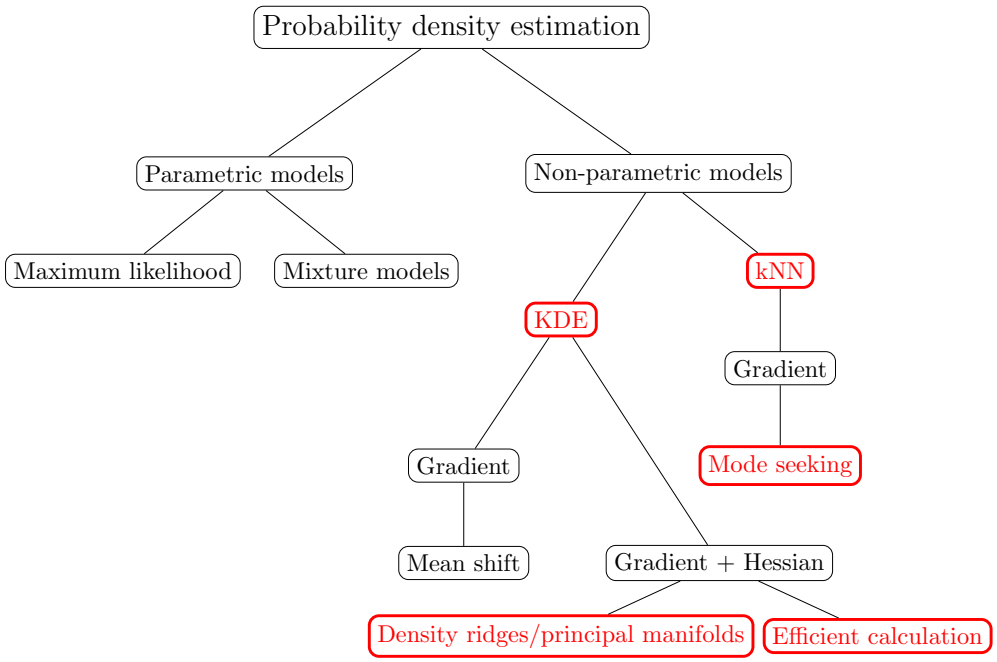


Figure 1.5: An overview of the context of the contributions of this thesis. The topic of each paper is marked with red. KDE is kernel density estimation and kNN is k-nearest neighborhood density estimation.

1.3 Reading guide

This remainder of this thesis consists of three parts, *methodology*, *summary of research* and *appended papers* and an appendix.

Methodology: This is a theoretical background that introduces the reader to the methodologies discussed in this thesis. The first three chapters review basic theory from mathematics, statistics and machine learning. In the last two chapters, we discuss the connection of the basic theory with the more recent theory used in our contributions.

Chapter 2 Presents the basics of probability density estimation and the derivatives of the kernel density estimator (relevant to Paper I).

Chapter 3 Presents a short summary of relevant concepts from differential geometry, as well as applications in machine learning (*manifold learning*).

Chapter 4 Presents principal manifolds, a general tool for estimating smooth surfaces from point clouds. The first part is general. The second part presents the special case of principal manifolds expressed through the gradient and Hessian of the probability density (relevant to Paper II and Paper III).

Chapter 5 Gives an introduction to unsupervised learning through density based clustering and explains how the derivatives of the probability density function can be used to perform cluster analysis (relevant to Paper VI).

All the chapters are written in an independent manner, and could be read separately for reference.

Research contributions: In this part we present a short overview of the scientific contribution represented by each paper included in this thesis. We also include concluding remarks and areas of future research in this part.

Included papers: This part contains the publications included in the thesis in their published or manuscript form.

Appendix: The appendix contains material that was used in Paper III, but not suitable for the Methodology part.

Part I

Methodology and context

Chapter 2

Probability density estimation

In this chapter we present the probability density function and some fundamental estimators and properties of estimators.

The probability density function (pdf) is one of the fundamental building blocks of machine learning and statistics [166, 101, 177, 91]. Methods such as naïve Bayes [166], mixture models [145], Gaussian processes [143] and information theoretic learning [141], based on an estimated pdf are all popular tools in the machine learning community.

Given a set of data points, $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x} \in \mathbb{R}^d$, the pdf $p(\mathbf{x})$ describes the relative probability that the data falls into a certain event, in this case an interval or subset in $\mathbb{R}^d$. $p(\mathbf{x})$ must integrate to one, $\int_{-\infty}^{\infty} p(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$ and the probability of an event B is given as $p(\mathbf{x} \in B) = \int_B p(\mathbf{x})\,\mathrm{d}\mathbf{x}$.

In practice, the true density function of observed data is often unknown and we have to rely on estimates. It is common to separate probability estimation into two branches: *parametric* and *non-parametric* estimation. We start by mentioning parametric models before we proceed to the non-parametric methods and their derivatives, which is the part most relevant to this thesis. Also, note that in this thesis we operate in a non-Bayesian setting[1].

[1] We assume no prior information of the distribution, and the parameters involved are considered deterministic in nature.

2.1 Parametric models for density estimation

Parametric density estimation assumes that a parametric model for the density is known in the form $p(\mathbf{x}|\theta)$ and one seeks to estimate the parameters θ, such as e.g. the mean $\boldsymbol{\mu}$ and variance Σ for a normal distribution [166].

Maximum likelihood estimator Given an iid[2] sample from $p(\mathbf{x}|\theta)$, $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x} \in \mathbb{R}^d$, we can form the *likelihood* function [145]

$$L(\theta|\mathbf{x}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\theta). \tag{2.1}$$

The choice of parameters $\hat{\theta} = \arg\max L(\theta|\mathbf{x})$ that maximizes Eq. (2.1) is called the maximum likelihood estimator [145]. An important property of the MLE is that it is asymptotically unbiased [166].

Mixture models Mixture models assumes in addition to a parametric model that a point $\mathbf{x}_i$ is sampled with probability π_j from a convex combination of k elementary distributions:

$$p(\mathbf{x}) = \pi_1 p_1(\mathbf{x}) + \pi_2 p_2(\mathbf{x}) + \cdots + \pi_k p_k(\mathbf{x}), \tag{2.2}$$

$\sum_j \pi_j = 1$. A mixture distribution is typically estimated with the expectation-maximization (EM) algorithm [145].

The principal manifold framework presented later in this thesis, which is based on non-parametric methods, can in fact also be estimated via a mixture of Gaussians [68, 135]. Due to practical issues related to mixture models (the number of components and computational efficiency among others), the experiments in the papers in this thesis used non-parametric density estimators.

[2] iid: Independent and identically distributed.

2.2 Non-parametric methods for density estimation

Non-parametric density estimation does not assume a parametric model of the density. Instead it focuses on estimating the density directly from data. We start with the naïve density estimator which is the most basic non-parametric estimator (if we exclude the *histogram*, which is mainly used as a visual tool).

The naïve estimator The probability of a univariate[3] observation being in a small region is

$$p(x) = \lim_{h \to 0} \frac{1}{2h} \mathrm{P}\left(x - h < X < x + h\right), \tag{2.3}$$

which can be modeled similar to a histogram by counting the number of observations within a bin of width $2h$ at x

$$\hat{p}(x) = \frac{1}{2hn} \#\left\{i | X_i \in (x - h, x + h]\right\}. \tag{2.4}$$

This results in a piece-wise constant function with height equal to the normalized number of points in each bin. By replacing the bins with smooth functions, a smooth counting function can instead be used, leading to the kernel density estimator.

2.3 Kernel density estimation

Given a data set $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x} \in \mathbb{R}^d$, the kernel density estimator, also known as Parzen window estimator, is a smoothed version of the naïve estimator given as follows [154]:

$$\hat{p}(\mathbf{x}|h) = \frac{1}{n} \sum_{i=1}^{n} K_h\left(\mathbf{x} - \mathbf{x}_i\right), \tag{2.5}$$

[3]The extension to multivariate data is trivial.

where $K_h(\mathbf{u}) = h^{-d}K\left(\frac{\mathbf{u}}{h}\right)$ is a *kernel function*, an integrable function that satisfies $\int K(\mathbf{x})\ \mathrm{d}\mathbf{x} = 1$. The parameter h is called the bandwidth of the kernel and determines the amount of smoothness in the density estimate. In the univariate case we have a single scalar bandwidth h, while in the multivariate case we can have a diagonal matrix $H = hI$, where I is a $d \times d$ identity matrix, or a fully parametrized bandwidth matrix H [37].

There are many kernel functions that satisfies these properties, but in this work unless otherwise noted we use the (multivariate) Gaussian kernel:

$$K_H\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{1}{(2\pi)^{d/2}|H|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)H^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right). \tag{2.6}$$

In Figure 2.1 an example of kernel density estimation, with $h = 1$, on samples drawn from a mixture of two Gaussians, $\mathcal{N}(0, 1)$ and $\mathcal{N}(10, 1)$.

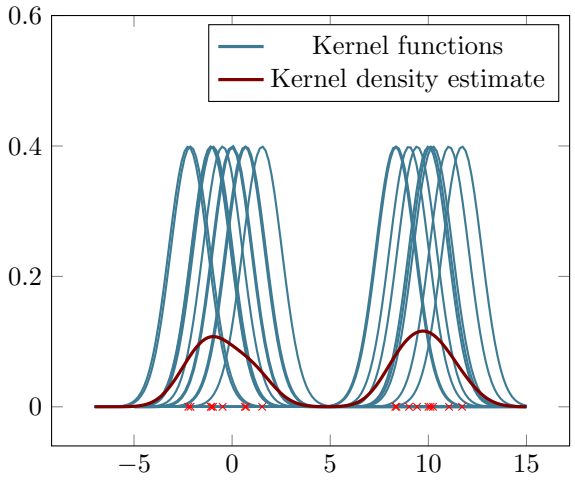


Figure 2.1: Example of kernel density estimation for a sample from two Gaussian distributions with unit variance and mean 0 and 10 in $\mathbb{R}$. The bandwidth parameter h determines the width of each of the kernel functions (blue) placed over each data point.

Other popular choices of kernel functions include the Epanechnikov kernel $K(u, v) = \frac{3}{4}(1 - (u - v)^2)$, $u \geq 1$ [177], the quartic kernel $K(u, v) = (1 - (u - v)^2)^2$, $u \geq 1$ [102] or the polynomial kernel $K(u, v) = (\alpha u^T v + c)^d$.

The kernel density estimator is under mild conditions asymptotically unbiased and consistent as the bandwidth h decreases and the sample size increases [138, 166, 83]. Still, in the finite sample setting we have to consider the ever present trade-off between bias and variance. The bias and variance can be found by Taylor series expansions of the true density function $p(\mathbf{x})$, see Wand and Jones [177] for further details:

$$\text{Bias: } \mathrm{E}\left[\hat{p}(\mathbf{x}|h)\right] - p(\mathbf{x}) = \frac{h^2\sigma_K^2 p''(\mathbf{x})}{2} + o(h^2) \tag{2.7}$$

$$\text{Variance: } \mathrm{Var}\left[\hat{p}(\mathbf{x}|h)\right] = \frac{p(\mathbf{x})}{hn}\mu_2(K) + o\left(\frac{1}{n}\right), \tag{2.8}$$

where $\sigma_K^2 = \int u^2 K(u)\mathrm{d}u$ and $\mu_2(K) = \int K^2(u)\mathrm{d}u$ and $p''(\cdot)$ denotes second order derivatives. Here we see that choosing a small h, gives a low bias, but high variance. In the opposite case with a large h we get reduced variance at the cost of increased bias.

2.3.1 Gradient and Hessian of the KDE

The gradient vector $\nabla^T\hat{p}(\mathbf{x})$ and Hessian matrix, $\hat{H}(\mathbf{x}) = \nabla\nabla^T\hat{p}(\mathbf{x})$, of the KDE, with scalar bandwidth h for ease of notation, is given by:

$$\hat{g}(\mathbf{x}) = \nabla^T\hat{p}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{x}-\mathbf{x}_i}{h^2}K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right). \tag{2.9}$$

$$\hat{H}(\mathbf{x}) = \nabla\nabla^T\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{u}_i\mathbf{u}_i^T - \frac{1}{h^2}I\right)K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right), \tag{2.10}$$

where $\mathbf{u}_i = \frac{\mathbf{x}-\mathbf{x}_i}{h^2}$ (notation adapted from [85]).

Several properties connected to the derivatives of the KDE are relevant to this thesis. We start with the induced flow from the gradient and the flow of the Hessian eigenvectors, known as the *subspace constrained* gradient flow [135, 85].

Gradient flow The gradient vector field of the probability density function induces a flow over the support of the probability density [5].

This can be visualized by inserting a test particle at some point in the gradient field and letting it flow along the gradient field with velocity given by the gradient vectors [86].

A kernel density estimate with a Gaussian kernel is positive definite, such that the gradient field will always point towards local maxima in the density estimate. Carrying out a gradient ascent scheme over the gradient flow field will thus give integral curves that converges to a *critical point*, $\hat{g}(\mathbf{x}) = 0$, of the density. More concretely, given a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ (e.g. a probability density function) the following gradient ascent scheme

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \alpha \nabla f(\mathbf{x}_{l-1}), \quad l \geq 1 \tag{2.11}$$

will approximate the integral curve from an initial point $\mathbf{x}_0$ that converges to a critical point [5]. A simple and practical form of this scheme is the *mean shift* [51, 46], which converges to the true gradient flow lines [5]:

$$\mathbf{x} \leftarrow \frac{\sum_{i=1}^{n} \mathbf{x}_i \, K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} - \mathbf{x} = \mathbf{m}(\mathbf{x}) - \mathbf{x}. \tag{2.12}$$

The vector $\mathbf{m}(\mathbf{x})$ is called the mean shift vector.

The gradient flow, represented either through Eq. (2.12) or Eq. (2.9), forms the foundation of a range of geometric tools for analysis. We will come back to this in Chapter 5.1.

Hessian eigenvector field The Hessian of the pdf assigns a symmetric matrix $H \in \mathbb{R}^{d \times d}$ to each point in the support of the density. Following directly, the eigenvectors of the Hessian forms an orthogonal frame bundle [2]. Thus, each set of eigenvectors is an appropriate basis for $\mathbb{R}^d$. If the probability density, p_M, is supported on a manifold, M, and sampled with additive Gaussian noise, p_{noise} such that[4] $p = p_M * p_{noise}$ [85]. Then (given low enough noise variance) the orthogonal basis provided by the Hessian can be split into a component approximately tangent to the manifold and a component representing the normal space of the manifold [43].

This basis decomposition forms the foundation of the *subspace constrained mean shift* algorithm presented in Section 4.2.1.

[4] $*$ denotes convolution.

2.3.2 Selection of the bandwidth parameter

There exists many methods for selecting the bandwidth parameter h, most of which are based on minimizing the *mean integrated square error* (MISE) [177]:

$$\mathrm{MISE}\{\hat{p}(\mathbf{x}|h)\} = \mathrm{E}\left[\int \left(\hat{p}(\mathbf{x}|h) - p(\mathbf{x})\right)^2 \mathrm{d}x\right], \tag{2.13}$$

and its asymptotic expansion found by Taylor expansion. These measures form the foundation of the *normal scale rule*, *smooth-* and *least squares*-cross validation methods and *plug-in* methods [177, 35, 37]. Assuming a Gaussian distribution and minimizing the asymptotic MISE gives the famous Silverman's rule of thumb [157], $h = 1.06\hat{\sigma}n^{1/5}$, where $\hat{\sigma}$ is the sample standard deviation [177]. A recent method proposes mixing cross-validation methods with plug-in estimators and has shown promising results [105].

Unfortunately, none of these are optimized for density derivative estimation. In a recent paper by Chacon and Duong [36], a unified framework for data-driven density derivatives was proposed. By combining matrix theory with the MISE (and integrated squared error), they established bandwidth selectors specifically created for density derivative estimation. Some of the standard methods were covered (smooth cross validation and plug in methods), and promising results were shown on both density estimation and mean shift (requires derivatives).

Comment: In the work leading up to Paper II and Paper III, we tested the methods of Chacon and Duong [36] as implemented in the ks package for the R statistical programming language [142, 62]. Most experiments in the papers of this thesis are aimed towards geometrical purposes, and even though the methods are tuned to derivative estimation they did not perform optimally in our experience. Instead we resorted to either manual tuning or the heuristic of selecting the kernel size as the average distance to the kth nearest neighbor, inspired by Shi et al. [155] and presented in Myhre and Jenssen [131],

Obviously this is not a sustainable choice, so the problem of estimating bandwidths for use in manifold estimation settings remains an open problem.

2.3.3 Product kernels and derivatives

The final topic we will present related to kernel density estimation is the concept of product kernels – as used in Paper I. A product kernel is a way of expressing a multivariate kernel as a product of univariate kernel functions:

$$\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left[\prod_{k=1}^{d} K_{h_k}\left(x^k - x_i^k\right)\right]. \tag{2.14}$$

K_{h_k} is a univariate kernel function with bandwith h_k for the kth dimension and $\mathbf{x} = \left[x^1, x^2, \ldots, x^d\right]^T \in \mathbb{R}^d$ as usual.

A nice property of product kernels is that the gradient and Hessian can be written in combinatorial form using only univariate differentiations. The first degree partial derivatives (elements of the gradient vector) are obtained using the operator $\nabla_c = \partial/\partial x_c$ with $c = 1, ..., d$:

$$\nabla_c K(\mathbf{x}) = K^{(1)}(x^c)\prod_{\substack{k=1\\k\neq c}}^{d} K(x^k), \tag{2.15}$$

where $x^c \in \mathbb{R}$ and $K^{(1)}(x^c)$ is the first derivative of the univariate kernel K. Thus $\nabla K(\mathbf{x}) = [\nabla_1 K(\mathbf{x}), \nabla_2 K(\mathbf{x}), \ldots, \nabla_d K(\mathbf{x})]^T$ represents the gradient vector. Note that we have omitted the bandwidth for notational simplicity, we recall that it is just a scaling of the width of the kernel.

The second order derivatives are obtained with $\nabla^2_{rc} = \partial^2/\partial x^r \partial x^c$ where $r, c = 1, ..., d$:

$$\begin{aligned}\nabla^2_{rc} K(\mathbf{x}) &= \delta_{rc}\, k^{(2)}(x^c)\prod_{\substack{k=1\\l\neq c}}^{d} k(x^k) \\ &+ (1-\delta_{rc})\, k^{(1)}(x^r)k^{(1)}(x^c)\prod_{\substack{k=1\\k\neq r\\k\neq c}}^{d} k(x^k).\end{aligned} \tag{2.16}$$

This can be arranged in the $d \times d$ Hessian matrix $H(\mathbf{x})$, $H_{ij} = \nabla^2_{ij} K(\mathbf{x})$.

2.4 k-nearest neighbor density estimation

The k nearest neighborhood (kNN) density estimator is based on the intuition that the probability density of a point is closely related to the number of points that are close to it. It it is given as:

$$\hat{p}(\mathbf{x}|k) = \frac{k}{n\text{vol}_k(\mathbf{x})}, \tag{2.17}$$

where n is the number of data points. $\text{vol}_k(\mathbf{x})$ is the volume of the d-dimensional hyper-sphere centered at $\mathbf{x}$ with radius equal to the distance to the kth neighbor:

$$\text{vol}_k(\mathbf{x}) = \frac{\pi^{d/2}}{\Gamma\left(d/(2+1)\right)}||\mathbf{x} - \mathbf{x}_k||^d. \tag{2.18}$$

$\mathbf{x}_k$ denotes the kth nearest neighbor of $\mathbf{x}$ and $\Gamma(i) = (i-1)!$ is the Gamma function. Often, to compensate for the poor scaling of the Gamma function, $\Gamma(i)$, in higher dimensions, a simplified version of the kNN density is used [174, 61]

$$\hat{p}(\mathbf{x}|k) = \frac{k}{n||\mathbf{x} - \mathbf{x}_k||^2}. \tag{2.19}$$

Due to the random nature of $||\mathbf{x}-\mathbf{x}_k||$, the kNN density is harder to estimate in terms of bias and variance compared to the KDE. This can be alleviated by conditioning on $||\mathbf{x} - \mathbf{x}_k||$. In that case the bias and variance turns out to be equal to the KDE bias and variance as in Eq. (2.8). Also, the tails of the kNN estimate will in fact be smoother than the KDE estimate [122], as a consequence of the varying nature of $||\mathbf{x} - \mathbf{x}_k||$.

Chapter 3

Differential geometry in machine learning

In this chapter we present some fundamentals of differential geometry. Many of these concepts form the foundation of the work done in Paper II and Paper III. We also include well known algorithms from the machine learning literature that exploit these fundamentals.

Differential geometry is the study of *mathematical sets with smooth geometry*[1] in arbitrary dimensions. These smooth surfaces are called *manifolds* and have been used in a wide range of problems in machine learning. The nature of such applications varies between explicit assumptions about the data and intrinsic assumptions on the definition concerning the problem itself. Thus, we split the applications into three categories:

- Problems where the data themselves lie on or close to a submanifold of the original space the data is sampled from [163, 165, 179]. This is often referred to as *the manifold assumption* [16].
- Problems where the solution of the cost function that is to be optimized lies on a manifold. For example optimization over real symmetric matrices [1] or the parameter space of probability distributions [3, 92].
- Metric learning where distance measures are adapted to conform with

[1]This formulation is borrowed from Nicolas Boumal et al. [26] at the front page of http://www.manopt.org/

the intrinsic geometry of the data or the problem [187, 104, 178].

In this thesis the focus is strictly on the first category – the data we are dealing with is assumed to lie on or close to a manifold of lower intrinsic dimension than the input dimensionality of the data.

The main idea comes from the observation that data sets or data structures seldom fill the vector space they are represented in. Even in low dimensional settings, e.g. $\mathbb{R}^3$, data sets often concentrate around clearly bounded sub regions that can be described by manifolds [27, 165, 139, 149, 88, 191].

3.1 Manifolds and related concepts

We start by defining a manifold and then proceed to present related topics that are relevant.

A clear definition of a manifold can be found in either books of Lee or Tu, [120, 168]:

Definition 1 *A (topological) manifold is a second countable, locally Euclidean, Hausdorff space.*

Local Euclidean structure is analogous to how humans perceive the surface of the earth. At smaller scales traversing a path along the surface will seem like a straight line, but on larger (non-human) scales paths along the surface of the earth are clearly curved. A Hausdorff space is a space where two separate points have disjoint neighboorhoods [12]. E.g. a surface embedded in $\mathbb{R}^3$ that intersects with itself will have points that shares neighborhoods and is thus not Hausdorff.

The perhaps most well known and intuitive example of a manifold is the sphere of radius r, $S^{d-1} = \left\{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||^2 = r\right\}$.

Given a manifold M of dimension d, at each point $p \in M$ the *tangent space*, T_pM, is the Euclidean space of dimension d which is tangent to M at p [120]. The term *tangent to*, can intuitively be interpreted as either the space of tangent vectors of all possible curves passing through p or the space spanned by the partial derivatives of the parametrization of M at p [120]. A disjoint union of all tangent spaces of M is called the *tangent bundle* of M.

Vectors in T_pM can be expressed by a local basis of differentials $E_i = \frac{\partial p}{\partial x^i}$. These are called the *normal coordinates* at p [120]. These normal coordinates parametrizes a Euclidean subspace of same dimension d as M. Figure 3.2 shows an illustration of the concepts presented above.

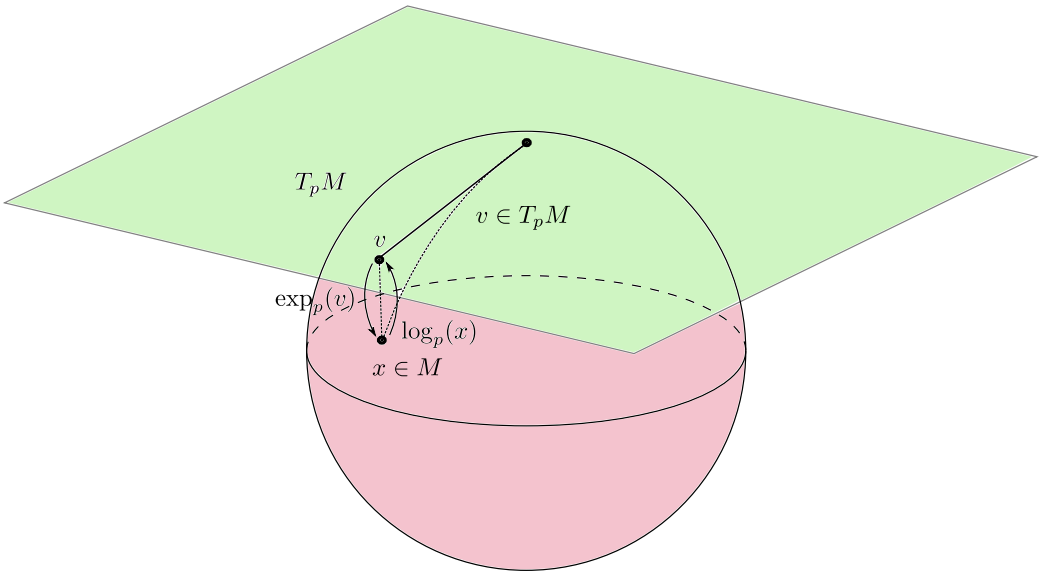


Figure 3.1: Concepts from differential geometry illustrated on the sphere embedded in $\mathbb{R}^3$.

Smoothness in a manifold can be defined either through smooth *coordinate chart transitions* or through the smooth change of the *metric tensor* of the manifold [120].

A coordinate chart is a homeomorphism[2] ϕ between an open subset $\mathcal{U}$ of the manifold M and a open subset of $\mathbb{R}^d$ [1]. Given different charts ϕ and ψ, the coordinate transformations $\phi \circ \psi^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ and $\psi \circ \phi^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ from chart to chart should be smooth, i.e. C^∞ (derivatives of all orders should exist)[1]. A set of (overlapping) charts covering M is called an *atlas* of M [139]. The *metric tensor* of a manifold M is a symmetric and positive definite function $G_M \in \mathbb{R}^{d \times d}$ that determines inner products on each tangent space T_pM [120]. This enables the calculation of length and angles locally at each point of the manifold. If the choice of metric varies smoothly, we

[2]A continuous function between topological spaces with a continuous inverse.

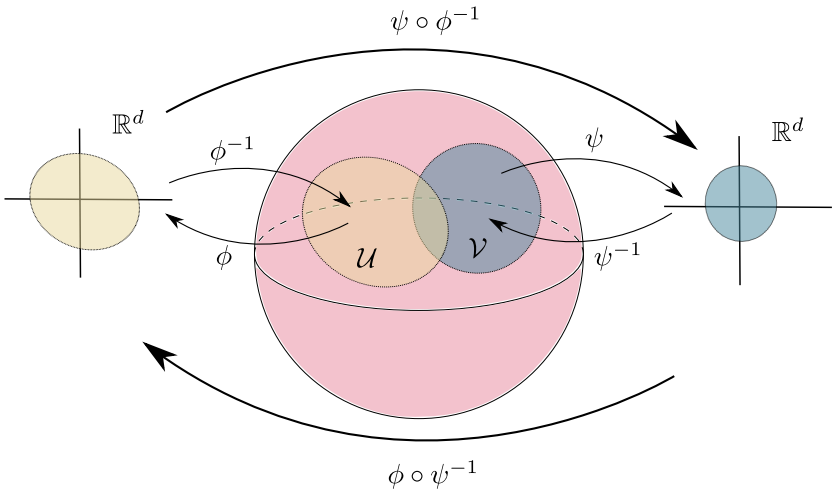


Figure 3.2: A common way of defining the smoothness of manifolds: the maps $\phi \circ \psi^{-1}$ and $\psi \circ \phi^{-1}$ should be smooth.

say that the manifold is a *Riemannian* manifold. In Euclidean space the metric is defined such that the distance between two points is described by a straight line. On Riemannian manifolds the idea of a straight line is replaced by a *geodesic*, which informally can be described as the shortest path along the manifold between two points [120][3].

A vector $\mathbf{v} \in T_pM$ can be mapped to M by following the geodesic starting at p for a time equal to the length of $\mathbf{v}$ as measured by the metric tensor (we assume that the geodesic is parametrized by a single variable, hence the term 'time' is used). We denote this mapping as the *exponential map*. The inverse mapping from M to T_PM is the *logarithmic map* [27].

Finally we introduce the concept of submanifolds and embeddings. A *submanifold* is a manifold that is *embedded* in another manifold (often called the ambient manifold). Surfaces – e.g. the sphere in $\mathbb{R}^3$ – are perhaps the simplest example of an embedded manifold. Formally, the mapping from the ambient manifold to the submanifold is an embedding if it is injective and homeomorphic [58].

Throughout this thesis (and in most machine learning applications) all man-

[3]The formal definition of the geodesic is dependent on the *connection* or *covariant derivative* of the manifold [120], which is beyond the scope of this text.

ifolds considered are submanifolds embedded in Euclidean space $\mathbb{R}^D$ [120]. This is common, but not often stated, in most applications where the manifold assumption is in place.

3.2 Manifold learning

The family of machine learning algorithms that assume that data lie on or close to an underlying manifold is called *manifold learning*. How to actually learn something about the underlying manifold is manifested in a multitude of approaches. Some methods aim to implicitly approximate the structure of the underlying manifolds, while other methods aim to 'unfold' the manifold such that the data approximately resides on a linear subspace. We begin by briefly describing a few algorithm that inspired our work. Non-isometric manifold learning [59] learns the tangent space of each point on the manifold estimated from a point set and suggests several practical algorithms for inference on the tangent space estimates. The Atlas approach of Pitelis et al. [139] learns overlapping local linear approximations and joins them together to form an atlas over the manifold. Maximum variance unfolding – later renamed semi-definite embedding – [178] learns a function that maximizes the pairwise distances between data points while at the same time constraining neighboring distances to stay fixed. This will in effect stretch out or *unfold* the manifold. Brun et al. [27] learns smooth geodesics along the manifold by spline interpolations of paths from Dikjstras algorithm [57]. This allows for approximate local coordinates on the manifold – an intrinsic unfolding of the manifold. Finally, we mention Ke Sun et al. [163] who proposed a data transformation that minimizes curvature and entropy, also in an attempt to unfold the underlying structure in the data.

The latter idea, trying to unfold or unwrap manifolds such that the data space in practice becomes close to linear (flat), has received a lot of attention the last decade [165, 149, 179, 15, 53, 190, 147, 164]. There are multiple benefits to this concept:

- Many fundamental machine learning algorithms – such as e.g. support vector machines, k nearest neighborhood classification or random forests [166, 53] – only requires pairwise distances as inputs. If the data resides on a manifold, pairwise distances comes in the form of geodesics,

which are notoriously hard to compute. A flattened manifold would in principle enable Euclidean distances.

- A manifold that has been flattened or unfolded will in addition allow the use of linear statistical methods. This is clearly a major benefit since linear methods are well established.
- Visualization: Data sets that exhibit high curvature, or are embedded in higher dimensions than three cannot be visualized properly. Unfolded structures are much easier to visualize [163, 169].

In practice, unfolding or unwrapping a manifold can intuitively be performed in two different ways. Either we use some function that stretches or flattens the manifold directly, or we can somehow estimate the structure of the manifold such that we can treat the manifold like a Euclidean space. Finally we acknowledge the fact that manifold learning and *dimensionality reduction* are most often part of the same algorithm, [28, 170, 166]. In practice this means that some methods tries to learn the structure of the manifold, some methods reduces the dimensionality of the data, and some methods combine both. Our contributions in this thesis, mostly represented by Paper II, are not directly concerned with dimensionality reduction, which therefore will not be discussed in further detail.

In the next section we will present a selection of algorithms from the manifold learning literature. These methods have either been used in this thesis or illustrates essential ideas within manifold learning and have all influenced the work in this thesis. Notable references for this section is the technical report of van der Maaten et al. [170] and the book of Theodoridis and Koutroumbas [166].

3.2.1 Linear methods for manifold learning

We begin by introducing the linear projection methods of principal component analysis (PCA) and multidimensional scaling (MDS). These are widely used in statistics and machine learning, and have also been applied in Paper II and Paper IV.

Let $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x} \in \mathbb{R}^D$ be a D-dimensional data set consisting of n data points. We then seek a representation $Y \in \mathbb{R}^{d \times n}$ where $d < D$.

Principal component analysis reduces the dimensionality of X by linear projection to the top d eigenvectors of the sample covariance matrix, Σ [109]. Eigendecomposition of Σ yields $\Sigma(X) = XX^T = U\Lambda U^T$, where U is the matrix containing eigenvectors as rows and $\Lambda_{ii} = \lambda_i$ is a diagonal matrix containing the eigenvalues. Y, the transformed data will be the first d columns of U, $U_{1,\dots,d}$. An out-of-sample point $\hat{\mathbf{x}}$ can be directly projected $\hat{\mathbf{y}} = U^T_{1,\dots,d}\hat{\mathbf{x}}$.

The construction of PCA can either be interpreted as minimizing mean squared error or total variance (since $\mathrm{Var}(X) = \sum_i \lambda_i$) [166].

Considering the geometry involved in the sample covariance eigendecomposition, PCA can be interpreted as fitting an ellipsoid to the data. The axes of the ellipsoid represents variance in the data and dimensionality reduction is performed by eliminating minor axes of the ellipsoid.

Multidimensional scaling is closely related to PCA, but takes a pair-wise dissimilarity matrix, D_X, between the data points as input. The objective is to reduce dimension while keeping the distances in the low dimensional space as close as possible to the original data space. The cost function is framed as:

$$\underset{y}{\text{minimize}} \sum_{ij} \left(d_{ij}^2 - \|\mathbf{y}_i - \mathbf{y}_j\|^2 \right), \tag{3.1}$$

where $\mathbf{y}_i \in \mathbb{R}^d$ is the low dimensional representation of X. The solution to (3.1), [166], is the top d eigenvectors of the doubly centered Gram matrix, $K = -\frac{1}{2}\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right) D_X \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$, multiplied with the square root of the eigenvalues such that $Y = \Lambda^T_{1,\dots,d} U_{1,\dots,d}$, where U and Λ is as defined for PCA.

Multidimensional scaling has been used as a component in several manifold learning algorithms, perhaps most notably in ISOMAP [165] and non-isometric manifold learning [59].

As a final note, these are linear methods such that they must be used with care; for a data set with high curvature, both PCA and MDS could result in unwanted flattening if the dimension is reduced too much.

3.2.2 Nonlinear methods for manifold learning

In this section we present three benchmark algorithms for non-linear manifold learning, isometric mapping (ISOMAP) [165], maximum variance unfolding (MVU) [180] and Laplacian eigenmaps [15]. We also present two lesser known algorithms, based on normal coordinates of the manifold, riemannian manifold learning and fast manifold learning based on Riemannian normal coordinates [124, 27]. The latter two are coordinate unfolding taking place in the tangent space of some reference point, which is similar to our work in Paper II and Paper III.

ISOMAP ISOMAP replaces the pair-wise distances of MDS with approximate geodesic distances calculated using Dijkstra's algorithm [57]. This will result in an isometric unfolding of the manifold. It builds on the assumption that the data we are given is sampled from a manifold without noise [165], and that this manifold can be sufficiently modeled by nearest neighborhood graph G.

Maximum variance unfolding MVU, is one of the first manifold learning methods to actually explicitly require an unfolding of the data. MVU is a cost function based approach, where the unfolding is formulated as stretching the dataset as much as possible by maximimizing the variance, while at the same time keeping nearest neighbor distances intact:

$$\begin{aligned} &\text{maximize} \quad \sum_{ij} \frac{1}{2n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ &\text{subject to} \quad \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{y}_i - \mathbf{y}_j\|^2, \ \forall i, j \in G, \end{aligned} \tag{3.2}$$

where G is the nearest neighborhood graph over the input space, $\mathbf{x}$ are input points and $\mathbf{y}$ are the desired (unfolded) outputs. The optimization is reformulated and solved as a semi-definite program [180, 118].

Laplacian eigenmaps Laplacian eigenmaps is a spectral manifold learning technique similar in construction to MVU. Given a sparse weighted graph

G_{ij} between points $\mathbf{x}_i$ and $\mathbf{x}_j$, Laplacian eigenmaps seeks an optimal embedding, ϕ, in a lower dimensional space:

$$\phi(Y) = \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij}, \tag{3.3}$$

where w_{ij} are weights (often generated from a Gaussian kernel, Eq. (2.6)), and $\mathbf{y}$ are the desired output points. This is rewritten and solved as an eigenvalue problem:

$$D^{-1}L\mathbf{y} = \lambda\mathbf{y}, \tag{3.4}$$

where $D_{ii} = \sum_j G_{ij}$ is the degree of G and $L = W - D$ [170]. The optimal embedding, that keeps neighborhoods close, is given by the top $d+1$ eigenvectors (The largest eigenvector is all ones and is omitted as it contributes nothing to the solution).

All of the three algorithms mentioned above have shown strong empirical results in many settings, but are limited by construction. Noise and shortcuts in the nearest neighborhood graph can destroy the topology of the embedding for both ISOMAP and MVU [173, 29]. This is due to poor approximations of geodesics in ISOMAP and erroneous constraints added in MVU. Moreover, there is an underlying assumption, in MVU and ISOMAP, that the manifold is isometric to $\mathbb{R}^d$, which limits the number of manifolds that can be represented, see e.g. [139]. This implies the (quite strong) assumption that a manifold can be represented by a single chart, and aims to find mapping from the manifold to the chart, $M \to \mathbb{R}^d$. In fact for a manifold to be described by a single chart it has to be a developable (intrinsically flat!) surface [148].

The embedding of the Laplacian eigenmaps is optimal, [15], in the sense described by the cost function. But exactly what that embedding encodes is very hard to interpret in practice. Also recent research shows that the Gaussian kernel is flawed when not used in Euclidean geometry [70].

Methods based on normal coordinates Fast manifold learning based on Riemannian normal coordinates (FastML) [27] and Riemannian manifold learning (RML) [124] aims to learn a flat representation of a manifold by calculating *Riemannian normal coordinates*.

Normal coordinates represent points on the manifold, M, through coordinates in the local tangent space T_pM [120]. A point in T_pM represents a

point on the manifold via the exponential map, see Section 3.1. Intuitively the normal coordinates represents a flat, or Euclidean, version of the manifold.

FastML [27] selects a single reference point and calculates the approximate geodesic distance to all points on the manifold. To obtain the approximate normal coordinates – represented by radial geodesics [120] – the directions of the geodesics are estimated via the finite difference gradient of the geodesic path. The dimension of the manifold is estimated through PCA on a neighborhood around the reference point.

RML [124] is very similar to FastML, except that several local charts are allowed (PCA on more than one reference point). Also, the geodesic approximation using Dijkstra is replaced by a more sophisticated algorithm, making use of smooth second-order polynomials. It also suggests an incremental learning approach, such that out-of-sample points can be directly included.

Both these methods are sensitive to noise due to the graph based approximations of geodesics. Moreover, in the FastML case the entire manifold will again be represented by a single chart, which we know is a limiting factor.

Learning a manifold as an Atlas Both ISOMAP and MVU are global methods that are sensitive to noise. All manifold learning methods mentioned so far do not take into account that the manifold can be sampled with noise. In addition, ISOMAP, MVU and FastML are *global* methods that try to represent the manifold with a single chart.

The Atlas algorithm [139] represents a mix between local and global strategy and can handle noisy samples from the manifold. The algorithm relies on creating overlapping charts by local linear approximations through local PCA. An optimization scheme alternating between assigning points to charts and estimating the parameters of the charts is employed. Moreover, a regularization term is also added to constrain the number of charts to give a compact representation of the manifold.

Atlas is somewhat different from the other methods mentioned. It does not require the manifold to be flat or to obey Euclidean geometry, instead it learns a representation of the manifold through the collection of charts. The

representation was shown to give positive results on kNN classification [139] and semi-supervised learning [140].

3.2.3 Calculating geodesics

Calculating geodesics is an important step in many manifold learning algorithms and for applications inspired by differential geometry such as e.g. image registration [14, 128, 31]. In this section we review three different frameworks for calculating geodesics:

- Graph based algorithms: Dijkstra's algorithm.
- Intuitive engineering approaches: Snakes and gradient descent.
- Calculus of variation: Euler-Lagrange.

Dijkstra's algorithm Dijkstra's algorithm is a solution to the single source shortest path problem in a connected graph with positive weights – it finds the shortest distance between two given vertices in a graph [57]. The algorithm has been extensively used in manifold learning approaches [165, 27, 124]. Pseudocode for the algorithm is given in Algorithm 1. Several extensions to Dijkstra's algorithm have been made, see e.g. [21, 78], and in Berstein et al. [20] it was proven that the algorithm will asymptotically approximate geodesics on the manifold (given some conditions).

A major problem when using Dijkstra's algorithm in manifold learning – or statistical settings in general – is the fact that the graph needed to calculate the shortest path is very sensitive to noise [139]. The most common approach to handle this problem in manifold learning is through regularization using splines or polynomials [27, 124].

Geodesics by gradient descent An alternative approach for calculating geodesics, inspired by so-called *snakes* used in computer vision [111], was proposed by Dollar et al. [59]. The idea is to minimize the length of a discretized path between two points by gradient descent.

$$\text{minimize} \sum_{i=2}^{m-1} ||\gamma_i - \gamma_{i-1}||^2, \tag{3.5}$$

Input: Graph $G(V, E)$ with nodes $v \in V$ and edges $e \in E$ and source node s
Output: Shortest path from s to all other nodes V
dist[s]=0;
for *all* $v \in V - \{s\}$ **do**
  dist[v]$\leftarrow \infty$;
end
$S \leftarrow \varnothing$;
$Q \leftarrow V$;
while $Q \neq \varnothing$ **do**
  $u \leftarrow$ mindistance$(Q, dist)$; `// Element in` Q `with minimum distance`
  $S \leftarrow S \bigcup \{u\}$;
  for $v \in$ neighbors[u] **do**
    if dist[v] $>$ dist[u] $+ e(u, v)$ **then**
      dist[v] $\leftarrow$ dist[u] $+ e(u, v)$
    end
  end
end

Algorithm 1: Dijkstra's algorithm

where γ_i is the path between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, often initialized with Dijkstra's algorithm. To avoid shrinking the path to a single point, the start and endpoints are held fixed, $\gamma_0 = \mathbf{x}_i$ and $\gamma_m = \mathbf{x}_j$. Also, some constraints need to be imposed to force the trajectory to stay on the manifold, either by projection (as in Paper II) or by de-noising [59]. The geodesic can thus be approximated by alternating between minimizing (3.5) and manifold projection/de-noising.

This method produces smooth geodesics, but needs a good initialization as well as some measure of whether the curve stays on the manifold or not. The latter is needed to formulate a stopping criteria for the algorithm in practice.

Geodesics by Euler-Lagrange In the case of a Riemannian manifold M with a given metric G_M, the distance between two points on the manifold is

given as

$$\text{length}(\gamma) = \int \sqrt{\gamma'(t)^T G_M(\gamma'(t))\gamma'(t)}\mathrm{d}t, \tag{3.6}$$

where $\gamma : [0, 1] \to M$ is the path between two data points $\mathbf{x}_i, \mathbf{x}_j \in M$ and $\gamma'(t) = \mathrm{d}\gamma(t)/\mathrm{d}t$ [104]. The particular γ that minimizes equation (3.6) such that $\gamma(0) = \mathbf{x}_i$ and $\gamma(1) = \mathbf{x}_j$ is a geodesic. Solving (3.6) requires the Euler-Lagrange equation to hold:

$$\frac{\partial L}{\partial \gamma} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \gamma'}, \tag{3.7}$$

where $L = \gamma'(t)^T G_M(\gamma'(t))\gamma'(t)$. Thus, assuming G_M is known, we can solve the equations using numerical solvers [104, 98]. In practice, the metric tensor G_M is rarely known or given, so it has to be estimated from the data or constructed through known quantities in the problem formulation. This is known as *metric learning* and is an entire branch of machine learning in and of itself, which will not be further discussed here. See the review papers by Kulis [116] or Bellet et al. [17].

3.2.4 Intrinsic manifold learning methods

As a final comment, we mention the area of research which is devoted to what we call *intrinsic manifold learning*. Here, contrary to the previously presented material, the manifold structure does not come from the geometric structure of the data, but instead from the geometric structure of the problem formulation. Examples include:

- Optimization problems formulated over symmetric matrices that restrict the solution space to a smooth manifold [1, 187, 47].
- Shape deformation in images that can be modeled by a specific manifold structure [72, 73].
- The fact that parameter spaces of statistical models exhibit manifold structure [92, 3].
- Optimization over the manifold of linear subspaces (the Grassmannian manifold) [103, 10].

Chapter 4

Principal manifolds

In this chapter we present principal manifolds and corresponding algorithms. These have been used extensively throughout this work, and connects the ideas from differential geometry with kernel density derivatives (Section 4.2).

Principal manifolds are in general considered to be curves or surfaces that somehow pass through the 'middle' of the data. Obviously the term 'middle' of the data is ambiguous, and many different definitions exist. We will present a selection of algorithms from the literature and end this section with the concept of principal manifolds defined as the ridges of the probability density function. This is the definition most relevant to this thesis.

4.1 Principal curves

The notion of a *principal curve* was originally introduced by Hastie and Stuetzle [100] as an extension from linear principal component analysis (PCA) to 'curves that pass through the middle of the data'. It was further developed to include *principal surfaces* and in general *principal manifolds* [191, 96].

A key point of principal curves and surfaces is that they allow a statistical framework to be formulated around the concept of estimating an underlying manifold [88].

We start with Hastie and Stuetzles definition of a principal curve or surface,

which is rooted in the ideas of *orthogonal subspaces* and *self consistency*.

Definition 2 (Orthogonal subspace) *Given a curve $\gamma : I \to \mathbb{R}^D$, the orthogonal subspace at a point $\gamma(t)$, $t \in \mathbb{R}$, is the space N such that all vectors $\mathbf{v} \in N$ satisfy $\mathbf{v}^T\gamma'(t) = 0$.*

$\gamma'(t) = 0$ is the tangent vector of γ at t. This definition extends naturally to a general surface S by replacing $\gamma'(t)$ with T_pS, the tangent space of the surface at a point p.

A *principal curve* is defined by the idea of self consistency.

Definition 3 (Self consistency) *A curve $\gamma(t)$, $t \in \mathbb{R}$, is said to be self consistent if all points along the curve is the average of all points projected to it, $\gamma(t) = E\left(\mathbf{x}|\hat{t}_\gamma(\mathbf{x}) = t\right)$,*

where $\hat{t}_\gamma(\mathbf{x})$ is called the projection index of $\mathbf{x}$, the closest point t in γ to $\mathbf{x}$.

Definition 4 (Principal curve) *A curve is a principal curve if it satisfies the self consistent property.*

In other words, the principal curve itself is the average value of points orthogonally projected onto the curve (the expected value of the orthogonal subspace). Figure 4.1 shows a conceptual illustration of the self consistency criteria on a sinusoidal curve sampled uniformly with small spherical Gaussian noise.

To end this introduction to principal curves we note that the extension to cover principal surfaces and principal manifolds in general is trivial. Definition 2 can be directly extended to any arbitrary manifold where the tangent space and normal space is well defined, and thus the principal surface/manifold viewed as the expected value of points in the orthogonal subspace follows trivially.

4.1.1 Extensions and alternative formulations

Whilst being an intuitive and elegant proposal for locating nonlinear curves or surfaces in point clouds, the self consistency is by construction flawed. In fact, the optimal solution is a space-filling curve [88] and all practical solutions requires some form of regularization. In this section we present some of the

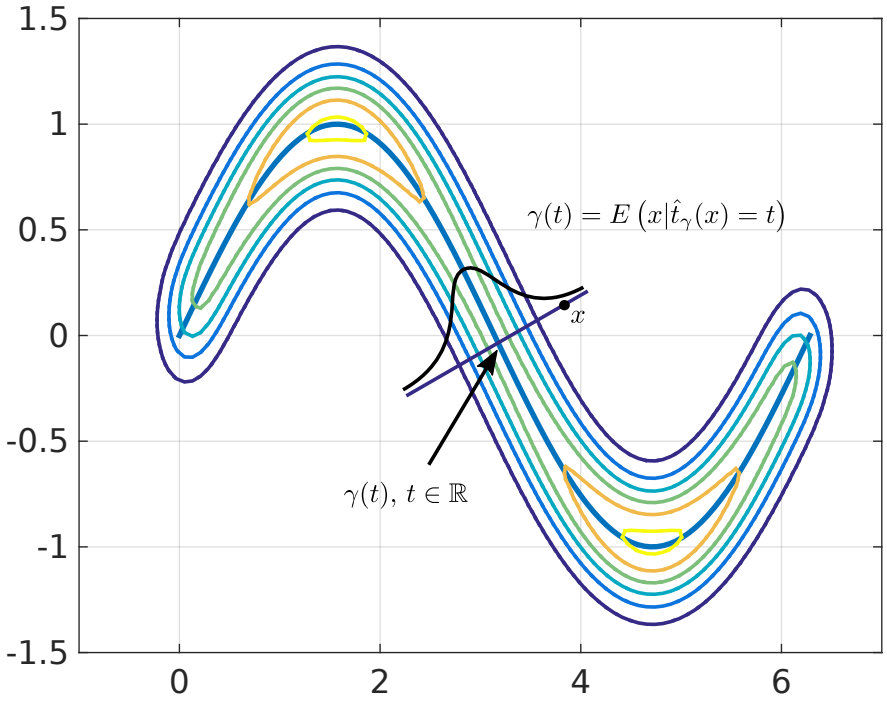


Figure 4.1: Illustration of the self consistent property. The principal curve is the expected value of all points orthogonally projected onto the curve.

most important extensions and alternative formulations of principal curves and surfaces that has seen practical use.

Kegl et al. [112] suggested a principal curve as a continuous curve of finite length that minimize the expected squared distance to the curve. This enabled them to prove that for any distribution with finite second order moments the principal curve always exist and develop two versions, one optimal and one efficient, of the polygonal line algorithm for practical implementation.

As noted by Einbeck et al. [65] and Delicado [55], there are several other flaws in the original construction of principal curves. Most notably the principal curve estimate is often strongly biased. They are also not able to handle

self intersecting or crossing curves[1]. Also, both the algorithms og Hastie and Stuetzle [100] and Kegl et al. [112] are dependent on initialization using the first linear principal component, which can in many cases lead to a poor estimate. Delicado [55] formulated principal curves as smooth curves passing through principal oriented points, a point-wise interpretation of self consistency. The local principal curve algorithm of Einbeck [65] presented a more practical version where the first principal component of the local centers of mass defines the principal curve. The local centers of mass are calculated through a kernel function weighted mean.

Related and with many successful applications, but different in construction and idea, is the *elastic principal graph* for estimating principal manifolds by Gorban and Zinovyev [95], where a grid approximation with varying amounts of elasticity is used to model principal manifolds.

There exists a plethora of different algorithms for estimating or identifying curves, surfaces or manifolds from point clouds, too many to mention here. Well known examples include self-organizing maps [114], nonlinear principal component analysis [60], principal geodesic analysis [73], curvilinear component analysis [56], local tangent space alignment [191], ridgeline manifolds [144], riemannian principal curves [102] and probabilistic principal surfaces [38].

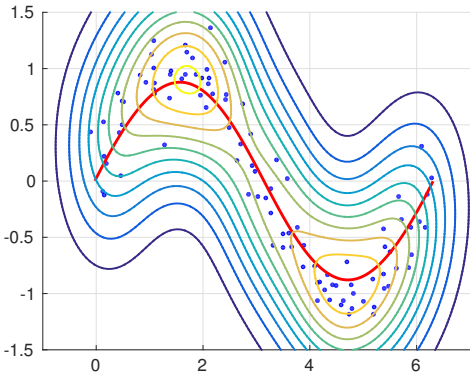
4.2 Principal manifolds as ridges of the probability density function

Ozertem and Erdogmus [135] presented a new view of principal curves and surfaces. Instead of formulating principal curves as self-consistent curves passing through the data, they proposed to define principal curves as the *ridges* – or more generally the *critical sets* – of the underlying probability density function of the data which the curve or surface passes through.

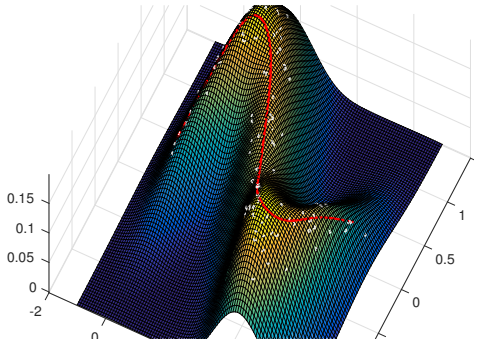
To enhance the intuition behind a principal curve as the ridge of a probability density function, we include an example shown in Figure 4.2. In Figure 4.2b

[1]We have to note here that a smooth manifold cannot have self-intersections, so we have to be careful to separate between algorithms to estimate manifolds and true manifolds.

we see a noisy sinusoid on the two horizontal axes (x and y) and the probability density on the vertical (z) axis. From this viewpoint the idea of a ridge is quite clear, completely analogous to the ridge of a mountain range.



(a) Density ridge (red), data points (blue), and kernel density estimate (contours).



(b) Density ridge (red), data points (white), and kernel density estimate (surface plot).

Figure 4.2: Example of the density ridge for a sinusoid sampled with $\mathcal{N}(0, 0.03I)$ additive noise.

The extension to ridges of higher dimension is not as picturesque as seen in the sinusoid example, but the formal definition is directly generalisable.

The ridge interpretation has two direct implications; first it allows the regularization of the principal manifold to be directly inherited from the smoothness of the underlying pdf – a well established subject in the case of kernel density estimation [177]. Second, it presents an alternative form of the self-consistency criteria, the density ridge is the *local maxima* of the orthogonal subspace, not the expected value.

Given a data set $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x} \in \mathbb{R}^d$, we define the modes[2] and ridges of the probability density as (notation adapted from [42]):

$$\mathcal{M} = \text{mode}(p) = \left\{\mathbf{x} \in \mathbb{R}^D : \nabla p(\mathbf{x}) = 0,\ \lambda_i(\mathbf{x}) < 0,\ \forall i\right\}$$
$$\mathcal{R} = \text{ridge}(p) = \left\{\mathbf{x} \in \mathbb{R}^D : V(\mathbf{x})V(\mathbf{x})^T \nabla p(\mathbf{x}) = 0,\ \lambda_{D-d}(\mathbf{x}) < 0\right\}.$$

Where $V(\mathbf{x})$ is a subset of the eigenvectors of the Hessian of the pdf at $\mathbf{x}$ and d represents the dimensionality of the ridge. The spectral decomposition

[2] We note that a mode of the pdf can be considered a zero dimensional ridge or manifold.

of the Hessian of the pdf is $H(\mathbf{x}) = Q(\mathbf{x})\Lambda(\mathbf{x})Q(\mathbf{x})^T$, where $Q(\mathbf{x})$ is the matrix of eigenvectors sorted according to the size of the eigenvalue and $\Lambda_{ii}(\mathbf{x}) = \lambda_i$, $\lambda_1(\mathbf{x}) > \lambda_2(\mathbf{x}) > \ldots$, is a diagonal matrix of sorted eigenvalues. Furthermore $Q(\mathbf{x})$ can be decomposed into $\left[Q_{\parallel}(\mathbf{x})\ Q_{\perp}(\mathbf{x})\right]$, where $Q_{\parallel}$ is the d first eigenvectors of $Q(\mathbf{x})$, and $Q_{\perp}$ are the $D-d$ smallest.

Definition 5 (Ozertem 2011) *A point* $\mathbf{x}$ *is on the d-dimensional ridge,* R *of its probability density function, when the gradient* $g(\mathbf{x})$ *is orthogonal to at least* $D-d$ *eigenvectors of* $H(\mathbf{x})$ *and the corresponding* $D-d$ *eigenvalues are all negative.*

The intuition behind the definition of a ridge is as follows: If a point lies on a ridge of the pdf the density value should ideally decay sharply in the direction that points away from the ridge resulting in high curvatures ($\lambda << 0$). Conversely, moving along the ridge should yield much less variation in density, resulting in low curvature ($\lambda \approx 0$).

Comparing to the differential geometry framework presented in Chapter 3, we can note a few observations:

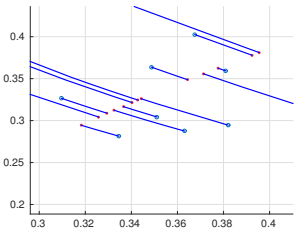
- For a point, $\mathbf{x}$, in R, $g(\mathbf{x}) \in \text{span}\left(Q_{\parallel}(\mathbf{x})\right)$. Thus (considering R as a manifold), $Q_{\parallel}(\mathbf{x})$ is a basis of $T_{\mathbf{x}}R$.
- Since $g(\mathbf{x})^T Q_{\perp}(\mathbf{x}) = 0$, the normal space of R is $\text{span}\left(Q_{\perp}(\mathbf{x})\right)$.

A direct consequence is that the collection of all parallel eigenvectors $Q_{\parallel}$ is a tangent bundle of R.

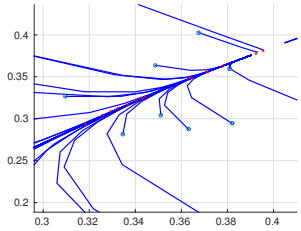
Finally we note that the word *ridge* is as seen earlier most intuitive in the one-dimensional case. A two-dimensional ridge amounts to a two dimensional surface or, as Genovese et al. coined it, a *wall* [84]. Nonetheless we keep the term ridge for any curve, surface or higher dimensional manifold that satisfies Definition 5.

4.2.1 Subspace constrained gradient flow: projecting noisy points onto the ridge

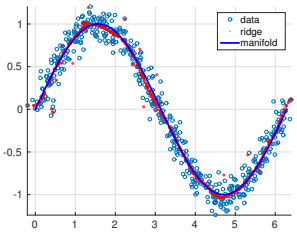
The observation that principal manifolds, modeled by the ridges of the pdf, are the maxima of the orthogonal subspaces allows for a simple algorithm to project noisy points onto the ridge [135].



(a) Zoomed in version of the subspace constrained gradient flow trajectories.



(b) Zoomed in version of the mean shift trajectories.



(c) Partition of input space induced by gradient flow.

Figure 4.3: Example of the SCMS algorithm. A noisy one-dimensional manifold is sampled with noise 4.3c, and we see that the ridges capture the underlying structure. In 4.3a and 4.3b, we see a comparison between zoomed in versions the mean shift and the SCMS trajectories.

Recall from Chapter 2 that the mean shift algorithm approximates the gradient flow of the pdf and converges to a local maximum of the pdf. It follows directly from Definition 5 that the modes are *a part of the ridge*, [85, 135]. Furthermore, a point that is close to the ridge should lie in the orthogonal subspace of some point on the ridge (represented by the last $D-d$ eigenvectors of the Hessian).

Thus, a point can be projected towards the density ridge by following the mean shift flow projected onto the local orthogonal subspace.

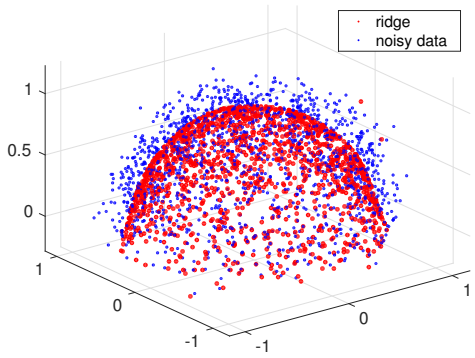
Given mean shift as defined in Eq. (2.12) and $V(\mathbf{x}) = Q_{\perp}(\mathbf{x})$, the (orthogonal) subspace constrained mean shift (SCMS) [135] is given as:

$$\mathbf{x} \to V(\mathbf{x})V(\mathbf{x})^T\mathbf{m}(\mathbf{x}) - \mathbf{x}. \tag{4.1}$$

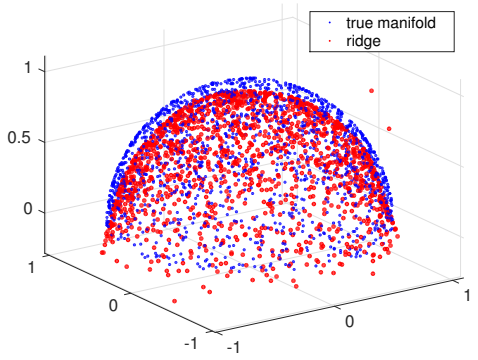
This allows noisy points to be projected onto the underlying smooth manifold represented by the ridge. Stopping criteria for the SCMS can either be when the steps are below a certain threshold or via checking if the gradient is orthogonal to the orthogonal subspace, $||V(\mathbf{x})^T g(\mathbf{x})||/||g(\mathbf{x})|| < \epsilon$.

In Figure 4.3 Figure 4.4 we see examples of the SCMS on one- and two-dimensional manifolds respectively.

The convergence of the SCMS algorithm has not yet been completely proved. Ghassabeh et al. [90] showed that it inherits the convergence properties of



(a) Two-dimensional ridge estimate from noisy data.



(b) The two-dimensional ridge estimate and true manifold. We see observe that there is inherent bias.

Figure 4.4: Example of the SCMS for a two-dimensional manifold sampled with noise.

mean shift and that the steps between the projected gradients $(V(\mathbf{x})V(\mathbf{x})^T\mathbf{m}(\mathbf{x}))$ goes to zero such that the algorithm is useful in practice. Furthermore, given the existence of a smooth underlying manifold sampled with small noise, Genovese et al. [85] showed that the ridges are consistent estimators of the underlying manifold under Hausdorff loss.

Finally, what often comes up when discussing estimation, is 'the curse of dimensionality' [18]. As dimension increases kernel density estimation becomes exponentially harder. This is due to the number of data points needed to fill the space increases exponentially. Since kernel density estimation dominates this thesis we simply have to be careful with dimensions that are too high. However, some results are promising, such as the work by Adler et al. [2], where the *reach*, a practical estimator for the curvature of the manifold, was shown to be independent of the ambient space.

In the next chapter we leave the world of manifolds and move on to introduce related concepts from unsupervised learning.

Chapter 5

Unsupervised learning – Clustering

This chapter introduces clustering, a slightly different concept compared to the previous chapters. We convey how mode seeking clustering is related to density derivatives and present general algorithms for clustering, both relevant to work presented in Paper IV.

Clustering algorithms aim to find natural subsets that reflects underlying structure in data without a priori information or human guidance – in some sense the purest form of unsupervised learning [99, 71, 108, 107]. In general, a clustering algorithm should assign data points into groups such that members within a cluster should be more similar to each other than to members from other clusters.

Clustering has been used in a wide range of applications, such as cosmic web reconstruction [44], pose estimation for the Microsoft Kinect® computer vision system [156], clustering of galaxies [4], grouping of biological sequences [81, 64, 115], air traffic analysis [80, 94], medical image segmentation [66, 159] among many others. It is also a growing field with many novel contributions in recent years, including robust multi-way clustering [30], sparse subspace clustering [67], clustering based on optimal transport [33], consistent methods for tree based clustering [39], ensemble clustering using matrix completion [185], methods from deep learning [183] and large scale graph methods [136].

Clustering is too big a field to be fully described in this text. The excellent survey papers in the following references are most recommended [176, 173, 71, 108, 107, 125].

We start with the definition of clustering by Jain [107].

Definition 6 (Jain 2010) *Given a representation of n objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low.*

Traditionally it has been common to separate clustering algorithms into *hierarchical* methods and *partitional* methods.

Hierarchical clustering results in a hierarchy of nested clusters. There are two common modes of hierarchical clustering. *Agglomerative* where each data point is initialized as a single cluster and then recursively joined to form a hierarchy of clusters ending in a single global cluster. Famous algorithms include single-, average- and complete-linkage [166, 97, 153]. The opposite direction with recursive splitting from a single cluster to each data point being a single cluster is called *divisive*, but is seldom used in practice.

Partitional clustering aims to find a single global clustering that partitions the data space into separate regions. There exist many practical considerations in the partitioning of the data space such as hard, fuzzy or probabilistic assignments or how to select the actual number of clusters wanted – a parameter that is needed as input in most cases. Well known algorithms such as mean shift [51], k-means [74] and mixture models by expectation maximization [145] are partitional clustering algorithms.

A taxonomy of clustering, including the most well known general algorithmic schemes, is shown in Figure 5.1.

5.1 Density based clustering

A particular sub-field of partitional clustering most relevant to this thesis is *density based clustering* [125]. In density based clustering, the clusters are assumed to be associated with properties of the underlying probability density function of the data. The most common assumption is that connected

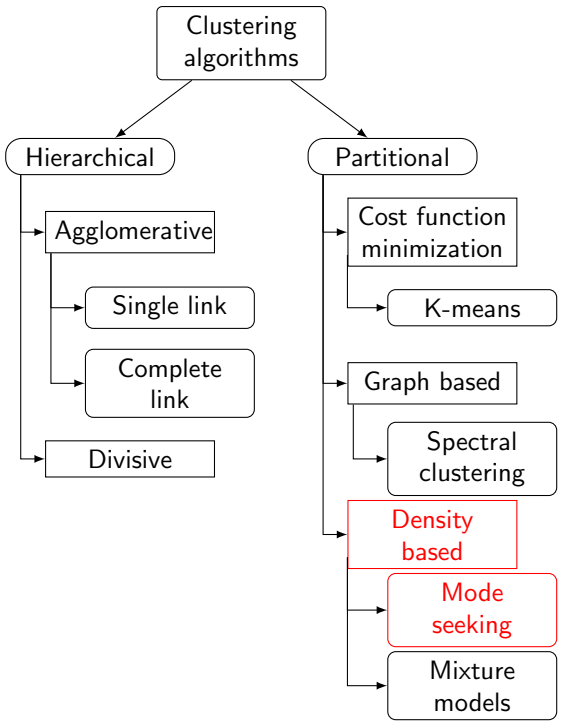
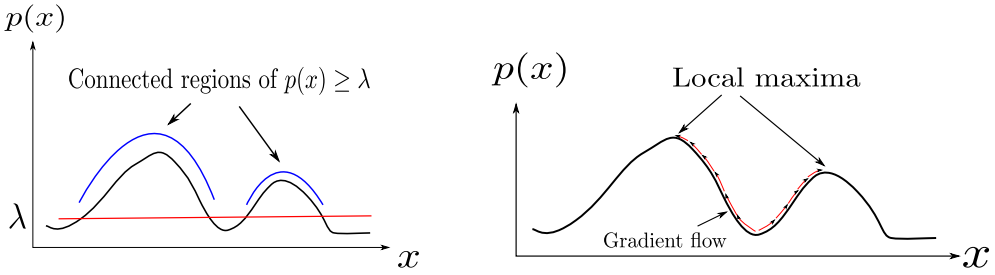


Figure 5.1: A taxonomy of clustering approaches. The contributions of this thesis are mostly in the areas marked with red.

regions of high density defines a cluster. This makes for an explicit and data driven definition of cluster structure [34].

Density based clustering (similarly to density estimation, see Section 2.1) is commonly divided into parametric and non-parametric methods. Here we focus exclusively on non-parametric methods.

Non-parametric methods can be further divided into *mode seeking* methods and *level set* methods [125]. Mode seeking methods identify local modes (maxima) of the data density function and takes them as cluster centers [61]. All points in a local basin of attraction connected to a local mode are considered to be in the same cluster. Conversely, level set methods, also called cluster tree methods [39, 54], are based on the idea that if one thresholds the pdf such that only areas of high density are left, then regions that are still connected should be clusters [34].



(a) The principle of level set clustering. The regions marked in blue represents connected components of the level set.

(b) The principle of mode seeking clustering. The gradient ascent towards local modes induces a partition of the support of the pdf.

Figure 5.2: The two main frameworks of non-parametric density based clustering.

5.1.1 Clustering by level sets

Level set clustering is based on the idea that connected regions of high density is a natural representation of a cluster [39, 162]. Depending on context the terms *population clustering* and *cluster trees* have also been used [54, 9]. Given a probability density p in $\mathbb{R}^d$ we define the level set:

$$L(\lambda) = \left\{\mathbf{x} \in \mathbb{R}^d : p(\mathbf{x}) \geq \lambda\right\}, \ \lambda \in (0, \max(p)). \tag{5.1}$$

Given an estimate of $p(\mathbf{x})$, most often in the form of a KDE, the goal is to identify connected components of $L(\lambda)$. Cuevas et al. [54][1] formulated it as finding "islands of high probability in a sea of low probability".

Finding connected components is usually formed as a graph problem [125]. Given a certain threshold λ_*, a graph G_{λ_*} with vertices consisting of all points in $L(\lambda_*)$ is formed. The problem is thus reduced to identifying connected components of G_{λ_*} [54].

The challenges of clustering in this context lies in both determining connected edges in G_λ and setting the proper threshold λ. A range of studies have used different values of λ and created hierarchical cluster trees [125, 9]. Li et al.[121] proposed a mix between modal clustering tree based level sets. In Figure 5.2a, the basic concept of level set clustering is illustrated.

[1]See also Figure 9 in Paper III.

5.1.2 Mode seeking – Clustering by following the gradient

A different, but related application of kernel density derivatives is clustering by mode seeking. As mentioned in Section 2.3.1 each point in the support of a pdf p will have a gradient flow that converges to a local maximum ($\nabla p = 0$). A simple illustration of this is presented in Figure 5.2b.

This will indirectly lead to a partition of the input space [89]. See Chacon [34] for a thorough and theoretical account of the idea of basin of attraction for clustering. This forms the foundation of clustering by *mode seeking*.

A number of different strategies have been proposed for finding local modes, or equivalent approximations.

Mean shift This algorithm is the most straight-forward exploitation of the gradient flow [51, 87, 46, 5]. For each input data point, or a grid of values over the input space, the mean shift trajectories are calculated:

$$\mathbf{x} \leftarrow \frac{\sum_{i=1}^{n} \mathbf{x}_i \, K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} - \mathbf{x} = \mathbf{m}(\mathbf{x}) - \mathbf{x}. \tag{5.2}$$

Each point that converges to the same local maximum (within a certain numerical threshold) are said to be in the same cluster.

Modal EM The modal EM algorithm reformulates the expectation maximization algorithm to consider kernel density estimation as a mixture of n Gaussians [121], and adapts it to find local maxima instead of maximizing the likelihood. This enables formulating mode seeking as an EM algorithm.

$$m_k = \frac{\pi_k p_k(\mathbf{x}_r)}{p(\mathbf{x}_r)}, \; k = 1, \ldots, n \tag{5.3}$$

$$\mathbf{x}_{r+1} = \operatorname*{argmax}_{\mathbf{x}} \sum_{k=1}^{n} m_k \log p_k(\mathbf{x}). \tag{5.4}$$

This has several benefits: Other models, even parametric, can be used to find modes, it is computationally efficient and proven to be an ascending algorithm [121].

Other algorithms: DBSCAN and clustering via search and find *Density based spatial clustering of applications with noise* (DBSCAN) is a clustering algorithm that captures modal regions [69]. It is thus not restricted to unimodal clusters, such as mean shift and related algorithms [34, 125]. The basic idea of DBSCAN is that for each point in a cluster the density of its corresponding neighborhood should be above some threshold. Clusters can be found using an iterative and sequential approach, and it is not sensitive to noise [69, 125]. However, it is dependent on two parameters, one for determining neighborhood size and one determining the minimum number of points in each cluster. Several extensions have been suggested [22, 113, 125].

Clustering by fast search and find of density peaks [146] is similar in construction to DBSCAN. It is a mode seeking algorithm, and is based on two assumptions. (1) cluster centers (modes) are surrounded by neighbors with lower local density. (2) cluster centers are (relatively) far from other points with higher local density. The algorithm calculates a decision graph based on the distances to points with higher density, which is thresholded to obtain a clustering.

Both of these methods represent successful mode seeking (or mode region seeking in the case of DBSCAN) based on a more heuristic approach than mean shift.

5.2 Ensemble methods in clustering

Inspired by the success of ensemble methods in classification, the field of *ensemble clustering* has emerged [173, 107]. The idea consists of combining a set of *weak* – fast and simple like e.g. k-means [166] – clusterings with either different initializations, different parameters or different algorithms altogether, to form a clustering ensemble such that consensus over the ensemble forms a more robust clustering than a single algorithm will give. Empirical results have clearly shown the potential of such methods [161, 77, 173, 75].

General approaches in ensemble clustering can be separated in two stages. The first stage builds the clustering ensemble and the second stage calculates the consensus over the repeated clusterings. In building the clustering ensemble there is naturally a lot of variation since in principle any cluster-

ing algorithm, parameter selection or initialization can be applied in virtually infinitely many combinations. The main difference within the family of ensemble clustering algorithms comes from how the consensus over the ensemble is established. Here we find two prominent directions, *median partition* based algorithms and *co-association*, or *consensus matrix*, based methods [123, 75, 173].

The median partition problem is based on optimizing a cost function that finds a partition P (clustering) that is as close as possible to all the different partitions in the ensemble: $\hat{P} = \arg\max \sum_j \Gamma(P, P_j)$, where Γ is a similarity measure between partitions. We will not go into further details, but refer to the survey paper by Vega-Pons ans Ruiz-Shulcloper [173].

The co-association strategy for calculating consensus can be considered a voting or counting process. In the counting process, a clustering algorithm is run repeatedly, and how many times a point is clustered in the same cluster, or how many times a point is clustered with another point, is counted.

Given an ensemble of M clustering trials, the elements of S, the counting- or co-association matrix [75, 76, 129], is then calculated by

$$s_{ij} = \frac{n_{ij}}{M}, \tag{5.5}$$

where n_{ij} is the number of times $\mathbf{x}_i$ and $\mathbf{x}_j$ has been assigned to the same cluster.

In the ideal case,

$$s_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same cluster,} \\ 0 & \text{otherwise.,} \end{cases} \tag{5.6}$$

leading to a matrix with a block structure if the data are sorted according to group labels (of course unknown to the algorithm). This can be considered a similarity matrix, and forms the foundation of further clustering. Fred and Jain [75, 77] have used k-means in combination with different hierarchical methods (most notably single- and average-linkage [166]). Monti et al [130] used the same scheme, but also added resampling. In Myhre et al. [132] a spectral clustering procedure, [176, 133] based on Cauchy-Schwarz divergence was used.

Part II

Summary of research

Chapter 6

Paper I - Computationally Efficient Exact Calculation of Kernel Density Derivatives

In this paper we present a fast *exact* estimator for kernel density derivatives. We have already seen a wide range of applications of density derivatives in the introduction to this thesis. With that in mind and the growing amount of data we are facing every day, the need for development of fast estimators of density derivatives is an important task. Exact estimators are especially suitable for large scale applications, as errors due to poor approximations can accumulate and grow out of proportion

In multivariate settings, the kernel function can be written as a product of univariate kernels. While this greatly simplifies the expressions for the gradient vector and the Hessian matrix, it also introduces a significant number of redundant multiplications.

We proposed and implemented a tree-based algorithm for performing faster exact kernel density derivative estimation. The computational complexity was also proven as a function of the order of the derivatives:

$$\sum_{l=1}^{d} \binom{l+r}{r}, \tag{6.1}$$

where d is the dimension and r is the derivative degree. Figure 6.1 illustrates

how the redundant multiplications are omitted.

The novelties in this contribution lies in the exact nature of the estimator. Other algorithms either try to reduce the number of pairwise points in the kernel matrix, or try to come up with some approximation of the kernel function.

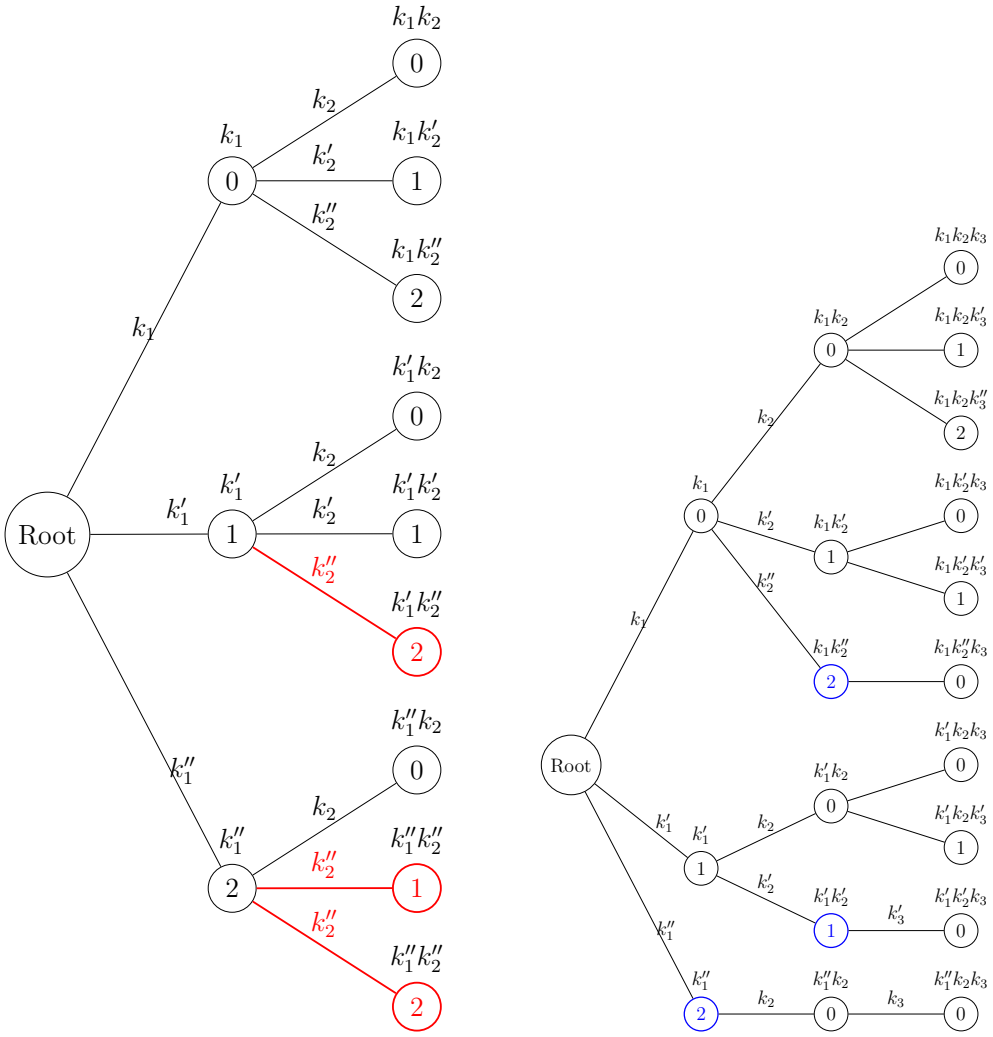
6.1 Contributions by the author

The idea was conceived at the Cognitive Systems Laboratory at Northeastern University by Dr. Deniz Erdogmus and associates.

My contributions:

- In collaboration with others I suggested the tree based structure which enabled the practical implementation.
- I implemented the MATLAB prototype that was used to carry out the experiments in the paper.

The paper was written as an equal collaboration between me and Matineh Shaker, a PhD student at the Cognitive Systems Laboratory under supervision of Dr. Erdogmus.



(a) Full computational tree for derivative degree 2 and dimension 3.

(b) Pruned computational tree for derivative degree 2 and dimension 3.

Figure 6.1: The key features of the efficient kernel density derivative algorithm.

Chapter 7

Paper II - Manifold unwrapping using density ridges

This paper presents novel ideas for unfolding manifolds modeled by density ridge estimators. An explicit unfolding based directly on the geometry of the data is introduced, contrary to other methods which are based on cost functions and or regularized models.

The original idea was rooted in the fact that a d-dimensional ridge estimate can be decomposed into d orthogonal one-dimensional ridges (principal curves). These orthogonal curves will enable a curvilinear coordinate system in each mode basin attraction of the underlying pdf. Tracing curve lengths along the one dimensional ridges will yield linear coordinates in the modes (similar to *normal coordinates* as presented in Section 3.1). Thus, a manifold estimated by a d-dimensional ridge can in principle be decomposed into separate curvilinear coordinate charts for each mode. Finally, these can be stitched together to form a global atlas for the manifold.

Promising results were obtained for one-dimensional manifolds and two-dimensional manifolds where the main variation could be described by a one-dimensional manifold. For manifolds of higher dimensions, however, the decomposition into one-dimensional orthogonal curves requires a significant amount of smoothing such that the d dimensional estimate will experience severe bias. To solve this we instead resort to linear approximation at each local attraction basin. We do this by projecting the points into the space

spanned by the local Hessian eigenvectors – equivalent to a first order Taylor approximation.

For future work the estimators of Chacon and Duong [36], should be further investigated as they imply different smoothing levels for the density, gradient and Hessian estimates.

The novelties in this work lies in using the density ridge manifold as the basis for manifold unwrapping. Previous research related to density ridge manifold estimators have dealt with its statistical properties such as uncertainty, convergence and existence [41, 85]. None have made direct use of the evident connections to manifold learning.

7.1 Contributions by the author

The original idea was based on observations made by Dr. Erdogmus in previous research on principal curves and surfaces in collaboration with Dr. Umut Ozertem [135, 68].

Together with Matineh Shaker and partially M. Devrim Kaba I implemented the framework in the one-dimensional and unimodal case.

My contributions:

- I implemented an adaptive Runge-Kutta scheme for projecting the data to the ridges, which enabled parallelization and significant speedup.
- I suggested and implemented the chart translation that enabled extension to multimodal cases.
- I suggested and implemented the multidimensional extension that inevitably led to the success of the unfolding algorithm.
- I wrote the manuscript draft for the final paper.

An extended abstract from this work was presented at the poster session of the Geometry in Machine Learning (GIMLI) workshop at ICML 2016.

7.2 Paper III - Invertible nonlinear cluster unwrapping

This paper was a predecessor to Paper II. It concerns manifold unwrapping in the one-dimensional and unimodal case.

Additionally, the paper contains an idea for out-of-sample projections based on concepts from image registration. This enables us to train a model that can project new points onto the ridge as well as calculate the inverse projection. This was also tested, at a later stage, on training a model that learned out-of-sample and inverse for the multidimensional unwrapping (from Paper III). Unfortunately, this projection model is very computationally expensive, so it was not included in the later work that lead to Paper II.

A more detailed description of the diffeomorphic projection model is presented in Appendix A.

The novelties in this paper are the same as the previous paper, as well as the introduction of a diffeomorphic model for out-of-sample and inverse projections.

7.3 Contributions by the author

- I implemented the methods in MATLAB and performed most of the experiments.
- I contributed to an early draft while staying as a visiting researcher in the Cognitive Systems Laboratory of Dr. Deniz Erdogmus.

Chapter 8

Paper IV - A robust clustering using a kNN mode seeking ensemble

In this paper we introduce concepts from *ensemble clustering* to improve mode seeking methods with respect to parameter sensitivity.

It marks a clear separation from the rest of the work, as a kNN density estimate is used instead of the kernel density estimate. This was chosen for several reasons. First of all, the kNN density estimator is much faster and more robust in higher dimensions. Second, a previously overlooked clustering algorithm – kNN mode seeking – was revived by Duin et al. [61] that showed considerable speedup compared to mean shift, at comparable accuracy. The latter two makes it ideal for use in a clustering ensemble, in addition to the fact that it is also capable of handling data sets of very high dimension.

Our proposed algorithm builds a cluster ensemble by repeated runs of the kNN mode seeking algorithm with random parameter initialization. Random subsampling of the data is also included in the ensemble to increase robustness. Finally clustering agreement over the ensemble is calculated using hierarchical clustering.

The novelties in this paper lies in introducing mode seeking algorithms into the ensemble clustering framework. In addition, the method provides progress towards mode seeking as a complete clustering tool without critical param-

eters that needs tuning.

8.1 Contributions by the author

The idea was conceived by myself and further developed in collaborations with Karl Øyvind Mikalsen and Sigurd Løkse, at the Machine Learning Group @ UiT – http://site.uit.no/ml/. The implementation and experiments were carried out by myself with the help of Karl Øyvind Mikalsen. I wrote the main draft of the manuscript.

Chapter 9

Concluding remarks

This thesis presents initial work in trying to establish principal manifolds, estimated by density ridges, as practical machine learning tools. We have provided geometrically intuitive algorithms and shown that the framework has potential. Moreover, we have presented ensemble methods as a direction for improving the otherwise sensitive mode seeking algorithms.

A novel method for unfolding a manifold, of any dimension, estimated by a density ridge has been presented. The unfolding algorithm showed good results on both real and synthetic data sets, and is intuitive in construction. Moreover, the algorithm has a basic foundation in differential geometry that allows unfolding any manifold estimate that is represented by a tangent bundle (we have to know the basis of the tangent space at each point).

Furthermore, a diffeomorphic projection model, inspired by techniques for smooth image registration, was implemented and tested. While showing promising results on low dimensional and synthetic data sets, it is as of now too computationally expensive to be considered practical.

In the final part of this work, we investigated how *ensemble* methods can act as a tool to increase robustness towards critical parameters of non-parametric density estimates. The particular application was in this case clustering by mode seeking. In terms of clustering performance and speed, the algorithm performed satisfactory and no parameter tuning was needed – a very promising result. However, due to the not so trivial connection between kernel density estimation and k nearest neighbor estimation, it is hard to comment

on how the robustness results would transfer to for example manifold estimation.

9.1 Short discussion: Weaknesses and alternative approaches

As all frameworks of machine learning and statistics have their strengths, they also have weaknesses that have to be acknowledged. Here we list a few of the most obvious issues that should be considered either as future work, or as guidelines for adapting our work into other frameworks.

Bayesian methodology This work have focused strictly on non-parametric estimation. In principle the idea is very enticing, no assumptions and no other parameters than the smoothness of the estimation. In practice however, this is often too limiting, and one has to resort to engineering solutions and heuristics. *Bayesian* methods allow the insertion of prior information into the problem and treats parameters as stochastic variables[145, 92, 166]. The setting can therefore still be non-parametric, but the bandwidth or smoothing parameter in the estimate is assumed stochastic. This allows for much more flexible models, both parametric and non-parametric.

Dimension As mentioned at the end of Section 4, whenever the kernel density estimator is used, the curse of dimensionality has to be considered. In this work we have in most cases overlooked this problem, as our main goal was to create a functional unwrapping algorithm given a density ridge estimate.

This will certainly pose constraints on the problems that our proposed algorithm can handle in practice. However, knowing or estimating the intrinsic dimensionality of the manifold can alleviate this problem, we recommend the review of intrinsic dimensionality estimators by Campadelli et al. [32].

Feature extraction A large area of modern machine learning research is related to feature extraction or learning representations of data (we recom-

mend the review of Bengio et al. [19]). Features or representations of data set are in practice considered good if they are fed into a linear classifier, typically a Support Vector Machine [166] and provide good classification results. This is the foundation of Deep Learning [152], which automatically learns non-linear features from data (given labels).

In this work we have only considered 'raw' data sets with no feature extraction and preprocessing, and one could argue that this is a too limited case for practical use.

9.2 Future work

Below we list some of the most important areas of future work related to each paper.

Paper I : The most important task left in this work is to provide an open source implementation of the algorithm and provide options for testing bandwidth selectors, such as Chacon and Duong [35, 36] or Botev et al. [24]. The review of Heidenrich et al. [105] provides a good overview of the state of the art as of 2013.

The algorithm is in its current state available in MATLAB®, which is not optimized for speed[1] and not free software. There is a plan to rewrite it to a c++ toolbox, but that is as of now not available yet.

Paper II and III : The future work related to the manifold unwrapping project can be divided into:

- Bandwidth selection/smoothness of estimate.
- Intrinsic properties of the manifold we want to estimate and unfold.
- High-dimensional behaviour.

As previously mentioned, we have in this work only used heuristic bandwidth selection. The same references to bandwidth estimator as for Paper I

[1] Somewhat ironic.

should be investigated here [35, 36, 24, 105]. Especially the work of Chacon and Duong [36] is appropriate here, since they proposed different amounts of smoothing depending on degree of derivative (gradient, Hessian etc.). Furthermore, not directly related to bandwidth, Sasaki et al. [150, 151] proposed a direct estimator[2] of the derivative of the probability density estimate.

Regarding intrinsic properties of the manifold, curvature and dimension are the two most important. Estimating these are not trivial, as their interplay in arbitrary dimension can be extremely varied. The previously mentioned paper by Campadelli [32] and the work of Adler et al. [2] for estimating curvature through *reach* should be considered.

Finally, the kernel density estimate breaks down when the dimension is too large. In manifold learning settings, the bounds on how big the difference between the intrinsic dimension of the manifold and the ambient/noisy space (the codimension) can be, given compact samples on the manifold, is an open and very interesting question.

Paper IV : The framework of ensemble clustering is very flexible, so there are virtually endless possibilities when it comes to combining algorithms and methods for future work. We therefore conclude with a short list of possible directions

- For large scale tasks sparse hierarchical clustering could be used [181, 189].
- The recent robust single linkage by von Luxburg et al. [39] could replace the hierarchical stage in this paper.
- Spectral clustering techniques could be used in the final step [176, 132, 184]
- Quick Shift or medioid shift could replace kNN mode seeking [172, 137].
- Different ensemble combination strategies should also be investigated [161, 126].

[2]In this thesis the derivative of the kernel density estimate, not the derivative of the true density, was used.

Part III

Included papers

Appendix A

Diffeomorphic projection model using landmark matching

Here we present further details of the diffeomorphic projection model used in Paper III, presented in Chapter 7.2. We generalize a diffeomorphic landmark matching methodology commonly used in image registration, which provides a diffeomorphism [49, 110, 6, 128].

A.1 Learning the diffeomorphic Projection Model

The purpose of this learning process is to identify a diffeomorphic model for approximate but fast transformations between two coordinate systems. Also, because the model is diffeomorphic, it allows inverse transformations.

The original techniques, found in image registration literature, develop a diffeomorphic large deformation model by cascading diffeomorphic models for small deformations, essentially by numerically approximating the solution of a differential equation with boundary and smoothness conditions imposed on the approximate solution [49], [48]. Small deformations parametrize a displacement field $\mathbf{u}$, which is added to the the initial point to find the transformation as $\phi(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$. Clearly, starting from $\mathbf{x}$ and adding a small perturbation has the advantage that the Jacobian of the overall mapping is identity plus a perturbation (hence nonsingular for all $\mathbf{x}$), leading to a

diffeomorphism.

These approximations fail in the case of large non-linear deformations, and is thus not suitable for manifold unwrapping. The large deformation frameworks cascades such small perturbations optimally in order to guarantee a one-to-one, smooth, and continuous mapping, with a nonsingular Jacobian implying a diffeomorphism [6]. A key element of a diffeomorphic mapping is that it preserves the topology (bot not angles since not conformal in general), and is consistent under composition of transformations. These properties are useful when transforming coordinates in manifold unwrapping and dimensionality reduction, implying transitive inverse consistent mappings.

The Euler-Lagrange equation for solving the large deformation diffeomorphic mapping is studied in [63], [167], and [110] for variational formulation of image matching. This setting parametrizes the transformation by means of velocity vectors $\mathbf{v}$ tangent to each displacement vector $\mathbf{u}$. In our version of this model, input training data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and corresponding transformed coordinates $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$, both coordinates in $\mathbb{R}^d$, are connected via the diffeomorphic change of coordinate $\phi : \mathbb{R}^d \to \mathbb{R}^d$. ϕ is the solution of the ordinary differential equation (ODE)

$$\frac{\mathrm{d}\phi(\mathbf{x}, t)}{\mathrm{d}t} = \mathbf{v}(\phi(\mathbf{x}, t), t), \tag{A.1}$$

where $t \in [0, 1]$ is the curve parametrization for the differential equation trajectory, and the initial point $\phi(\mathbf{x}, 0) = \mathbf{x}$ corresponds to the identity transform. Here, $\phi(\mathbf{x}, t)$ is the Lagrangian trajectory defined as the position at *time* t, which was at $\mathbf{x}$ at time 0 [50]. The final transformation (solution of the differential equation) $\phi(\cdot, 1)$ is therefore controlled through the velocity field $\mathbf{v}(\cdot, t)$, and is given as:

$$\phi(\mathbf{x}, 1) = \mathbf{x} + \int_0^1 \mathbf{v}(\phi(\mathbf{x}, \tau), \tau)\, \mathrm{d}\tau. \tag{A.2}$$

Since such $\phi(\mathbf{x}, t)$ is not unique, the optimal diffeomorphic match is constructed by minimizing deformation energy $\|L\mathbf{v}\|^2$, where L is a linear differential operator on the velocity field. This is analogous to thin-plate splines [23]. In addition, we want to minimize the distance between the curvilinear coordinates $\mathbf{c}$ and the endpoints of the transformation $\phi(\mathbf{x}, 1)$, re-

sulting in the following optimization problem for the velocity field of (A.2):

$$\hat{\mathbf{v}}(\mathbf{x},t) = \underset{\mathbf{v}(\mathbf{x},t)}{\arg\min} \int_{\mathbb{R}^d\times[0,1]} \|L\mathbf{v}(\mathbf{x},t)\|^2 \, \mathrm{d}\mathbf{x}\, \mathrm{d}t + \sum_{i=1}^{n} [\mathbf{c}_i - \phi(\mathbf{x}_i,1)]^T [\mathbf{c}_i - \phi(\mathbf{x}_i,1)]. \tag{A.3}$$

This optimization problem poses two problems: the infinite dimensional parameter space of the velocity field and the continuous nature of the integral. The first issue is elegantly alleviated by noticing that the minimizer of (A.3) must take the following form [110] (for reasons similar to kernel regression emerging from the representation theorem for RKHS):

$$\hat{\mathbf{v}}(\mathbf{x},t) = \sum_{i=1}^{n} k(\phi(\mathbf{x}_i,t),\mathbf{x}) \sum_{j=1}^{n} (\mathbf{K}^{-1}(t))_{ij} \dot{\hat{\phi}}(\mathbf{x}_j,t), \tag{A.4}$$

where $\mathbf{K}_{ij}(t) = k(\phi(\mathbf{x}_i,t),\phi(\mathbf{x}_j,t))$, where k is the Green's function for L^TL; that is, $L^TLk(\mathbf{x}-\mathbf{y}) = \delta(\mathbf{x}-\mathbf{y})$.

With this observation, the following equivalent optimization problem is obtained [110]:

$$\begin{aligned} \dot{\hat{\mathbf{\Phi}}}(t) = \underset{\mathbf{\Phi(t)}}{\arg\min} \quad & \int_0^1 \dot{\mathbf{\Phi}}(t)\mathbf{K}^{-1}(t)\dot{\mathbf{\Phi}}(t)dt + \sum_{i=1}^{n} [\mathbf{c}_i - \phi(\mathbf{x}_i,1)]^T [\mathbf{c}_i - \phi(\mathbf{x}_i,1)] \\ \text{subject to} \quad & \mathbf{\Phi}(0) = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T]^T \end{aligned} \tag{A.5}$$

This reformulation has reduced the problem from finding the velocity field on the entire space $\mathbb{R}^d$ to finding n trajectories.

Finally we can approximate these n velocity fields trajectories on $t \in [0,1]$ to be piece-wise constant on M sub-intervals (as in Euler/RK2 numerical integration), $\{t_m = 1/M\}_{m=0}^{M}$, resulting in:

$$\int_0^1 \dot{\mathbf{\Phi}}(t)\mathbf{K}^{-1}(t)\dot{\mathbf{\Phi}}(t)dt \approx M \sum_{m=0}^{M-1} \left(\mathbf{\Phi}(t_m+1) - \mathbf{\Phi}(t_m)\right)^T \mathbf{K}^{-1}(t_m) \left(\mathbf{\Phi}(t_m+1) - \mathbf{\Phi}(t_m)\right). \tag{A.6}$$

A.2 Projecting Out-of-Sample Test Data

We continue to use the landmark matching framework [110], which essentially results in a kernel regression type transformation for projecting points that are not in the original landmark/training set. The out-of-sample test data point projection rule consists of a weighted sum of estimated velocities at each training sample, multiplied with the inverse pairwise kernel matrix seen earlier during training. The estimated velocity at time $t \in [t_{i-1}, t_i]$ for a training point $\mathbf{x}_i$ is given as

$$\dot{\hat{\phi}}(\mathbf{x}_n, t) = \frac{\hat{\phi}(\mathbf{x}_n, t_i) - \hat{\phi}(\mathbf{x}_n, t_{i-1})}{(1/M)}, \tag{A.7}$$

and we get the velocity field for an out-of-sample data point $\tilde{\mathbf{x}}$ as

$$\hat{\mathbf{v}}(\tilde{\mathbf{x}}, t) = \sum_{i=1}^{n} k(\hat{\phi}(\mathbf{x}_n, t), \tilde{\mathbf{x}}) \sum_{j=1}^{n} (\mathbf{K}^{-1}(t))_{ij} \dot{\hat{\phi}}(\mathbf{x}_j, t). \tag{A.8}$$

With this we can estimate the final transformation for the out of sample point

$$\hat{\mathbf{c}}(\tilde{\mathbf{x}}) = \hat{\phi}(\tilde{\mathbf{x}}, 1) = \tilde{\mathbf{x}} + \int_0^1 \hat{\mathbf{v}}(\hat{\phi}(\tilde{\mathbf{x}}, \tau), \tau) \mathrm{d}\tau \approx \tilde{\mathbf{x}} + M \sum_{m=0}^{M-1} \hat{\mathbf{v}}(\hat{\phi}(\tilde{\mathbf{x}}, t_m), t_m), \tag{A.9}$$

using Euler/RK2 integration with the same step length as in training. The inverse mapping will start from some $\tilde{\mathbf{c}}$ and integrate *backwards* in time:

$$\hat{\mathbf{x}}(\tilde{\mathbf{c}}) = \hat{\phi}(\tilde{\mathbf{x}}, 0) = \tilde{\mathbf{c}} + \int_1^0 \hat{\mathbf{v}}(\hat{\phi}(\tilde{\mathbf{x}}, \tau), \tau) \mathrm{d}\tau \approx \tilde{\mathbf{c}} - M \sum_{m=1}^{M} \hat{\mathbf{v}}(\hat{\phi}(\tilde{\mathbf{x}}, t_m), t_m). \tag{A.10}$$

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] R. J. Adler, S. Krishnan, J. E. Taylor, and S. Weinberger. The reach of randomly embedded manifolds. *arXiv preprint arXiv:1503.01733*, 2015.

[3] S.-i. Amari and H. Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007.

[4] L. Anderson, É. Aubourg, S. Bailey, F. Beutler, V. Bhardwaj, M. Blanton, A. S. Bolton, J. Brinkmann, J. R. Brownstein, A. Burden, et al. The clustering of galaxies in the sdss-iii baryon oscillation spectroscopic survey: baryon acoustic oscillations in the data releases 10 and 11 galaxy samples. *Monthly Notices of the Royal Astronomical Society*, 441(1):24–62, 2014.

[5] E. Arias-Castro, D. Mason, and B. Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 2015.

[6] J. Ashburner et al. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

[7] E. Ataer-Cansizoglu and D. Erdogmus. A mode-based clustering algorithm without mode seeking. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1925–1928. IEEE, 2012.

[8] J. Back, G. J. Barker, S. B. Boyd, J. Einbeck, M. Haigh, B. Morgan, B. Oakley, Y. Ramachers, and D. Roythorne. Implementation of a local

principal curves algorithm for neutrino interaction reconstruction in a liquid argon volume. *The European Physical Journal C*, 74(3):1–15, 2014.

[9] S. Balakrishnan, S. Narayanan, A. Rinaldo, A. Singh, and L. Wasserman. Cluster trees on manifolds. In *Advances in Neural Information Processing Systems*, pages 2679–2687, 2013.

[10] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

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

[12] M. Barile and E. W. Weisstein. T2-space. Visited on 25/02/15.

[13] E. Bas, D. Erdogmus, R. Draft, and J. W. Lichtman. Local tracing of curvilinear structures in volumetric color images: Application to the brainbow analysis. *Journal of Visual Communication and Image Representation*, 23(8):1260–1271, 2012.

[14] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.

[15] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[16] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[17] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[18] R. E. Bellman and S. E. Dreyfus. *Applied dynamic programming*. Princeton university press, 2015.

[19] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[20] M. Bernstein, V. De Silva, J. C. Langford, and J. B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Technical report, Department of Psychology, Stanford University, 2000.

[21] A. S. Bijral, N. Ratliff, and N. Srebro. Semi-supervised learning with density based distances. *arXiv preprint arXiv:1202.3702*, 2012.

[22] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[23] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):567–585, 1989.

[24] Z. I. Botev, J. F. Grotowski, D. P. Kroese, et al. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[25] J. Böttger, A. Schäfer, G. Lohmann, A. Villringer, and D. S. Margulies. Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain. *IEEE transactions on visualization and computer graphics*, 20(3):471–480, 2014.

[26] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[27] A. Brun, C.-F. Westin, M. Herberthson, and H. Knutsson. Fast manifold learning based on riemannian normal coordinates. In *Image Analysis*, pages 920–929. Springer, 2005.

[28] C. J. Burges. Dimension reduction: A guided tour. *Machine Learning*, 2(4):275–365, 2009.

[29] C. J. Burges. Geometric methods for feature extraction and dimensional reduction-a guided tour. In *Data mining and knowledge discovery handbook*, pages 53–82. Springer, 2009.

[30] A. C. Cameron, J. B. Gelbach, and D. L. Miller. Robust inference with multiway clustering. *Journal of Business & Economic Statistics*, 2012.

[31] V. Camion and L. Younes. Geodesic interpolating splines. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 513–527. Springer, 2001.

[32] P. Campadelli, E. Casiraghi, C. Ceruti, and A. Rozza. Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015, 2015.

[33] F. P. Carli, L. Ning, and T. T. Georgiou. Convex clustering via optimal mass transport. *arXiv preprint arXiv:1307.5459*, 2013.

[34] J. E. Chacón. Clusters and water flows: a novel approach to modal clustering through morse theory. *arXiv preprint arXiv:1212.1384*, 2012.

[35] J. E. Chacón and T. Duong. Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *Test*, 19(2):375–398, 2010.

[36] J. E. Chacón, T. Duong, et al. Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting. *Electronic Journal of Statistics*, 7:499–532, 2013.

[37] J. E. Chacón, T. Duong, and M. Wand. Asymptotics for general multivariate kernel density derivative estimators. *Statistica Sinica*, pages 807–840, 2011.

[38] K.-Y. Chang and J. Ghosh. A unified model for probabilistic principal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):22–41, 2001.

[39] K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

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

[41] Y.-C. Chen, C. R. Genovese, and L. Wasserman. Asymptotic theory for density ridges. *arXiv preprint arXiv:1406.5663*, 2014.

[42] Y.-C. Chen, C. R. Genovese, and L. Wasserman. Generalized mode and ridge estimation. *arXiv preprint arXiv:1406.1803*, 2014.

[43] Y.-C. Chen, C. R. Genovese, L. Wasserman, et al. Asymptotic theory for density ridges. *The Annals of Statistics*, 43(5):1896–1928, 2015.

[44] Y.-C. Chen, S. Ho, P. E. Freeman, C. R. Genovese, and L. Wasserman. Cosmic web reconstruction through density ridges: method and algorithm. *Monthly Notices of the Royal Astronomical Society*, 454(1):1140–1156, 2015.

[45] Y.-C. Chen, S. Ho, A. Tenneti, R. Mandelbaum, R. Croft, T. DiMatteo, P. E. Freeman, C. R. Genovese, and L. Wasserman. Investigating galaxy-filament alignments in hydrodynamic simulations using density ridges. *Monthly Notices of the Royal Astronomical Society*, 454(3):3341–3350, 2015.

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

[47] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.

[48] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. 3d brain mapping using a deformable neuroanatomy. *Physics in medicine and biology*, 39(3):609, 1994.

[49] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on*, 5(10):1435–1447, 1996.

[50] G. E. Christensen, P. Yin, M. W. Vannier, K. Chao, J. Dempsey, and J. F. Williamson. Large-deformation image registration using fluid

landmarks. In *Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium*, pages 269–273. IEEE, 2000.

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

[52] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[53] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[54] A. Cuevas, M. Febrero, and R. Fraiman. Cluster analysis: a further approach based on density estimation. *Computational Statistics & Data Analysis*, 36(4):441–459, 2001.

[55] P. Delicado. Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, 77(1):84–116, 2001.

[56] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on neural networks*, 8(1):148–154, 1997.

[57] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[58] M. P. Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-hall Englewood Cliffs, 1976.

[59] P. Dollár, V. Rabaud, and S. Belongie. Non-isometric manifold learning: Analysis and an algorithm. In *Proceedings of the 24th international conference on Machine learning*, pages 241–248. ACM, 2007.

[60] D. Dong and T. J. McAvoy. Nonlinear principal component analysis—based on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1):65–78, 1996.

[61] R. P. Duin, A. L. Fred, M. Loog, and E. Pekalska. Mode seeking clustering by knn and mean shift evaluated. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 51–59. Springer, 2012.

[62] T. Duong et al. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21(7):1–16, 2007.

[63] P. Dupuis, U. Grenander, and M. I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of applied mathematics*, 56(3):587, 1998.

[64] R. C. Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[65] J. Einbeck, G. Tutz, and L. Evers. Local principal curves. *Statistics and Computing*, 15(4):301–313, 2005.

[66] K. Elakkia and P. Narendran. Survey of medical image segmentation using removal of gaussian noise in medical image. *International Journal of Engineering Science*, 7593, 2016.

[67] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[68] D. Erdogmus and U. Ozertem. Nonlinear coordinate unfolding via principal curve projections with application to nonlinear bss. In *International Conference on Neural Information Processing*, pages 488–497. Springer, 2007.

[69] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[70] A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3032–3042, 2015.

[71] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008.

[72] P. T. Fletcher, C. Lu, and S. Joshi. Statistics of shape via principal geodesic analysis on lie groups. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–95. IEEE, 2003.

[73] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

[74] K. Florek, J. Łukaszewicz, J. Perkal, H. Steinhaus, and S. Zubrzycki. Sur la liaison et la division des points d'un ensemble fini. In *Colloquium Mathematicae*, volume 2, pages 282–285. Institute of Mathematics Polish Academy of Sciences, 1951.

[75] A. L. N. Fred. Finding consistent clusters in data partitions. In *In Proc. 3d Int. Workshop on Multiple Classifier*, pages 309–318. Springer, 2001.

[76] A. L. N. Fred and A. K. Jain. Evidence accumulation clustering based on the k-means algorithm. In *Structural, Syntactic, and Statistical Pattern Recognition, LNCS 2396:442–451*, pages 442–451. Springer-Verlag, 2002.

[77] A. L. N. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):835–850, 2005.

[78] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[79] O. Freifeld, S. Hauberg, and M. J. Black. Model transport: Towards scalable transfer learning on manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1378–1385. IEEE, 2014.

[80] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[81] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[82] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theor.*, 21(1):32–40, Sept. 2006.

[83] R. Fukunaga. Statistical pattern recognition. 1990.

[84] C. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Finding singular features. *arXiv preprint arXiv:1606.00265*, 2016.

[85] C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Nonparametric ridge estimation. *The Annals of Statistics*, 42(4):1511–1545, Aug. 2014.

[86] C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, L. Wasserman, et al. On the path density of a gradient field. *The Annals of Statistics*, 37(6A):3236–3271, 2009.

[87] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 456–463. IEEE, 2003.

[88] S. Gerber and R. Whitaker. Regularization-free principal curve estimation. *The Journal of Machine Learning Research*, 14(1):1285–1302, 2013.

[89] Y. A. Ghassabeh, T. Linder, and G. Takahara. On the convergence and applications of mean shift type algorithms. In *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–5. IEEE, 2012.

[90] Y. A. Ghassabeh, T. Linder, and G. Takahara. On some convergence properties of the subspace constrained mean shift. *Pattern Recognition*, 46(11):3140–3147, 2013.

[91] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[92] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[93] F. Godtliebsen, J. S. Marron, and P. Chaudhuri. Statistical significance of features in digital images. *Image and Vision Computing*, 22(13):1093–1104, 2004.

[94] K. Gopalakrishnan, H. Balakrishnan, and R. Jordan. Clusters and communities in air traffic delay networks. In *2016 American Control Conference (ACC)*, pages 3782–3788. IEEE, 2016.

[95] A. Gorban and A. Zinovyev. Elastic principal graphs and manifolds and their practical applications. *Computing*, 75(4):359–379, 2005.

[96] A. N. Gorban, B. Kégl, D. C. Wunsch, A. Y. Zinovyev, et al. *Principal manifolds for data visualization and dimension reduction*, volume 58. Springer, 2008.

[97] J. C. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.

[98] I. Griva, S. G. Nash, and A. Sofer. *Linear and nonlinear optimization*. Siam, 2009.

[99] I. Guyon, U. Von Luxburg, and R. C. Williamson. Clustering: Science or art. In *NIPS 2009 workshop on clustering theory*, pages 1–11, 2009.

[100] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[101] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[102] S. Hauberg. Principal curves on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[103] S. Hauberg, A. Feragen, R. Enficiaud, and M. Black. Scalable robust principal component analysis using grassmann averages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016.

[104] S. Hauberg, O. Freifeld, and M. J. Black. A geometric take on metric learning. In *Advances in Neural Information Processing Systems*, pages 2024–2032, 2012.

[105] N.-B. Heidenreich, A. Schindler, and S. Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA Advances in Statistical Analysis*, 97(4):403–433, 2013.

[106] D. J. Henderson and C. F. Parmeter. Canonical higher-order kernels for density derivative estimation. *Statistics & Probability Letters*, 82(7):1383–1387, 2012.

[107] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[108] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[109] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[110] S. C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *Image Processing, IEEE Transactions on*, 9(8):1357–1370, 2000.

[111] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

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

[113] S. Kisilevich, F. Mansmann, and D. Keim. P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st international conference and exhibition on computing for geospatial research & application*, page 38. ACM, 2010.

[114] T. Kohonen and P. Somervuo. Self-organizing maps of symbol strings. *Neurocomputing*, 21(1):19–30, 1998.

[115] E. Kopylova, J. A. Navas-Molina, C. Mercier, Z. Z. Xu, F. Mahé, Y. He, H.-W. Zhou, T. Rognes, J. G. Caporaso, and R. Knight. Open-source sequence clustering methods improve the state of the art. *mSystems*, 1(1):e00003–15, 2016.

[116] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

[117] J.-Y. Kwok and I.-H. Tsang. The pre-image problem in kernel methods. *IEEE transactions on neural networks*, 15(6):1517–1525, 2004.

[118] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

[119] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[120] J. M. Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer, 1997.

[121] J. Li, S. Ray, and B. G. Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(Aug):1687–1723, 2007.

[122] Q. Li and J. S. Racine. *Nonparametric econometrics: theory and practice*. Princeton University Press, 2007.

[123] T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 577–582. IEEE, 2007.

[124] T. Lin and H. Zha. Riemannian manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):796–809, 2008.

[125] G. Menardi. A review on modal clustering. *International Statistical Review*, 2015.

[126] C. Meyer, S. Race, and K. Valakuzhy. Determining the number of clusters via iterative consensus clustering. 2013.

[127] Z. Miao, B. Wang, W. Shi, and H. Wu. A method for accurate road centerline extraction from a classified image. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4762–4771, 2014.

[128] A. Mills, T. Shardlow, and S. Marsland. Computing the geodesic interpolating spline. In *International Workshop on Biomedical Image Registration*, pages 169–177. Springer, 2006.

[129] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1-2):91–118, 2003.

[130] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52(1-2):91–118, 2003.

[131] J. N. Myhre and R. Jenssen. Mixture weight influence on kernel entropy component analysis and semi-supervised learning using the lasso. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.

[132] J. N. Myhre, K. Ø. Mikalsen, S. Løkse, and R. Jenssen. Consensus clustering using knn mode seeking. In *Image Analysis*, pages 175–186. Springer, 2015.

[133] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[134] J. Ning, L. Zhang, D. Zhang, and C. Wu. Robust mean-shift tracking with corrected background-weighted histogram. *IET Computer Vision*, 6(1):62–69, 2012.

[135] U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12(4):1249–1286, 2011.

[136] X. Pan, D. Papailiopoulos, S. Oymak, B. Recht, K. Ramchandran, and M. I. Jordan. Parallel correlation clustering on big graphs. In *Advances in Neural Information Processing Systems*, pages 82–90, 2015.

[137] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

[138] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[139] N. Pitelis, C. Russell, and L. Agapito. Learning a manifold as an atlas. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1642–1649. IEEE, 2013.

[140] N. Pitelis, C. Russell, and L. Agapito. Semi-supervised learning using an unsupervised atlas. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 565–580. Springer, 2014.

[141] J. C. Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[142] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[143] C. E. Rasmussen. *Gaussian processes for machine learning*. Citeseer, 2006.

[144] S. Ray and B. G. Lindsay. The topography of multivariate normal mixtures. *Annals of Statistics*, pages 2042–2065, 2005.

[145] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[146] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[147] L. Rossi, A. Torsello, and E. R. Hancock. Unfolding kernel embeddings of graphs: enhancing class separation through manifold learning. *Pattern Recognition*, 48(11):3357–3370, 2015.

[148] W. Rossmann. Lectures on differential geometry, 2004.

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

[150] H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 19–34. Springer, 2014.

[151] H. Sasaki, Y.-K. Noh, and M. Sugiyama. Direct density-derivative estimation and its application in kl-divergence approximation. In *AISTATS*, 2015.

[152] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[153] H. K. Seifoddini. Single linkage versus average linkage clustering in machine cells formation applications. *Computers & Industrial Engineering*, 16(3):419–426, 1989.

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

[155] T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics*, pages 3960–3984, 2009.

[156] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, C. M., and R. Moore. Real-Time Human Pose Recognition in Parts from Single Depth Images. *Communications of the ACM*, 56(1):116–124, 2013.

[157] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[158] R. S. Singh. Applications of estimators of a density and its derivatives to certain statistical problems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 357–363, 1977.

[159] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth. Medical image segmentation on gpus–a comprehensive review. *Medical image analysis*, 20(1):1–18, 2015.

[160] T. Sousbie. The persistent cosmic web and its filamentary structure–i. theory and implementation. *Monthly Notices of the Royal Astronomical Society*, 414(1):350–383, 2011.

[161] A. Strehl and J. Ghosh. Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[162] W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 2012.

[163] K. Sun, E. Bruno, and S. Marchand-Maillet. Stochastic unfolding. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.

[164] P. K. Tadavani, B. Alipanahi, and A. Ghodsi. Manifold unfolding by isometric patch alignment with an application in protein structure determination. *Perspectives on Big Data Analysis: Methodologies and Applications*, 622:177, 2014.

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

[166] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 4th edition, 2009.

[167] A. Trouvé. Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision*, 28(3):213–221, 1998.

[168] L. W. Tu. *An introduction to manifolds*, volume 200. Springer, 2008.

[169] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[170] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[171] N. Vassilas, T. Tsenoglou, and D. Ghazanfarpour. Mean shift-based preprocessing methodology for improved 3d buildings reconstruction. *World Academy of Science, Engineering and Technology, International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 9(5):615–620, 2015.

[172] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Computer vision–ECCV 2008*, pages 705–718. Springer, 2008.

[173] S. Vega-Pons and J. Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[174] V. V. Vikjord and R. Jenssen. Information theoretic clustering using a k-nearest neighbors approach. *Pattern Recognition*, 47(9):3070–3081, 2014.

[175] T. Vojir, J. Noskova, and J. Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49:250–258, 2014.

[176] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[177] M. P. Wand and M. C. Jones. *Kernel smoothing*. Crc Press, 1994.

[178] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[179] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, volume 6, pages 1683–1686, 2006.

[180] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 106. ACM, 2004.

[181] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 2012.

[182] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[183] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. *arXiv preprint arXiv:1511.06335*, 2015.

[184] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907–916. ACM, 2009.

[185] J. Yi, T. Yang, R. Jin, A. K. Jain, and M. Mahdavi. Robust ensemble clustering by matrix completion. In *2012 IEEE 12th International Conference on Data Mining*, pages 1176–1181. IEEE, 2012.

[186] S. You, E. Bas, D. Erdogmus, and J. Kalpathy-Cramer. Principal curved based retinal vessel segmentation towards diagnosis of retinal diseases. In *Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on*, pages 331–337. IEEE, 2011.

[187] P. Zadeh, R. Hosseini, and S. Sra. Geometric mean metric learning. In *Proceedings of the 33th international conference on Machine learning*, 2016.

[188] M. Zayed and J. Einbeck. Constructing economic summary indexes via principal curves. In *Proceedings of the 19th International Conference on Computational Statistics, COMPSTAT*, pages 1709–1716, 2010.

[189] H. Zhang and R. H. Zamar. A natural framework for sparse hierarchical clustering. *arXiv preprint arXiv:1409.0745*, 2014.

[190] Z. Zhang, J. Wang, and H. Zha. Adaptive manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):253–265, 2012.

[191] Z.-y. Zhang and H.-y. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University (English Edition)*, 8(4):406–424, 2004.