Seppo Pulkkinen

Ridge Curve Approach to Extraction of Curvilinear Structures from Noisy Data

Turku Centre *for* Computer Science

TUCS Technical Report
No 1082, August 2013

Ridge Curve Approach to Extraction of Curvilinear Structures from Noisy Data

Seppo Pulkkinen
University of Turku, Department of Mathematics and Statistics
FI-20014 Turku, Finland
seppo.pulkkinen@utu.fi

TUCS Technical Report
No 1082, August 2013

Abstract

Extraction of curvilinear structures from noisy data is an essential task in many application fields such as data analysis, pattern recognition and machine vision. The approach considered in this paper assumes a random process in which the samples are obtained from a generative model. The model specifies a set of generating functions describing curvilinear structures as well as sampling noise and background clutter. It is shown that ridge curves of the marginal density induced by the model can be used to estimate the generating functions. Given a Gaussian kernel density estimate for the marginal density, ridge curves of the density estimate are parametrized as the solution to a differential equation. Finally, a predictor-corrector algorithm for tracing the ridge curve set of such a density estimate is developed. Efficiency and robustness of the algorithm are demonstrated by numerical experiments on synthetic datasets as well as observational datasets from seismology and cosmology.

Keywords: principal curve; filament; generative model; ridge curve; density estimation; predictor-corrector method

TUCS Laboratory
Turku Optimization Group (TOpGroup)

1 Introduction

Detection and extraction of curvilinear structures from noisy data is an essential task in many practical applications: extraction of blood vessels that form filament- and tree-like structures is an important task in medical imaging (see e.g. [6], [27] and [36]); in cosmological data, stars and galaxies form filament-like patterns (see e.g. [41] and [49]), and in astronomy, detection of solar flares involves finding filaments from solar images (see e.g. [47] and [53]). Identification of curvilinear structures from noisy data with background clutter is a typical task in remote sensing (see e.g. [5]) and seismology (see e.g. [17] and [50]). Real-time applications, where fitting curves into noisy data is an important task, include freeway traffic modeling (see e.g. [13] and [25]) and process monitoring (see e.g. [20]).

One of the most well-known approaches to extract curvilinear structures from noisy data is to use the so-called *principal curves*. This approach dates back to Hastie [32] and Hastie and Stuetzle [33]. A principal curve is defined as a curve passing through the "middle" of the data in a certain sense. Further variations of the principal curve approach have been developed, for instance, by Kégl and Krzyzak [37, 38] and Tibshirani [51]. All of these approaches, however, make rather restrictive assumptions. For instance, they attempt to fit a single curve with no self-intersections, or as the method of [37], require complicated parameter adjustments when intersecting or multiple curves are sought from the data.

In order to overcome the limitations of the original principal curve definition, locally defined variants of a principal curve have been proposed (see e.g. [18], [19], [26], [28–30] and [43]). This paper extends an earlier paper by the author [45] refining the ideas presented in [43]. The key idea in these two papers is to estimate the probability density from given data and extract curvilinear structures from the data from *ridge curves* of the density estimate. Since the definition of a ridge is based only on local derivative information, this approach does not suffer from the limitations of the earlier approaches. For projection of a sample point onto a ridge, a subspace-constrained variant of the standard mean-shift method (see e.g. [14] and [15]) is proposed in [43]. An improved Newton-based method for this purpose is developed in [45]. Recently, some extensions of ridge-based methods have been made for the more difficult problem of parametrization of principal curves by iteratively tracing ridge curves of the density (see e.g. [6–8]).

In [45] the author proposes a *generative model* for describing a random process that generates a noisy point set containing curvilinear structures. In the model, the data points are assumed to be sampled from a set of *generating functions* with additive noise. In this paper the model is extended to include background clutter that is often present in practical applications. Furthermore, it is shown in [45] that ridge curves of the *marginal density* induced by the model can be used to estimate the underlying generating functions. Differently to the earlier local principal curve approaches, where no statistical assumptions are made about the data-generating process, the proposed model provides a more disciplined approach.

For a computational implementation of the ridge curve approach, we consider *nonparametric* estimation of the marginal density by using *Gaussian* kernels (see e.g. [46]). This approach allows to estimate the density directly from the samples with no prior knowledge on the data-generating process, which is often the case in real-world tasks. We also discuss how to automatically choose the kernel *bandwidth* since this step is crucial for the practical applicability of the method.

The main contribution of this paper is the development of a computationally efficient and robust algorithm for tracing ridge curves of a Gaussian kernel density estimate. Adapting the theory of *gradient extremals* from theoretical chemistry (see e.g. [35]), it is shown that a ridge curve can be parametrized by tracing a solution curve of a differential equation. A predictor-corrector algorithm is developed for this purpose. The algorithm first finds a set of modes (maxima) of the density, and starting from each mode iteratively traces the ridge curve passing through it. Since the choice of the mode-finding and corrector methods largely determines the performance of the algorithm, the trust region Newton method developed in [45] is utilized for these purposes. This choice is motivated by the results of [45] showing that the Newton-based method is not only more efficient than the mean-shift method and its subspace-constrained variant previously proposed in [6–8] and [43] but also converges to a ridge point or mode under mild assumptions.

The main difficulty in tracing ridge curves is that they can have a very complex structure. Differently to the earlier ridge-based principal curve methods of [6–8], where this issue was not considered in detail, a detailed treatment for detection of different types of singular points along a ridge curve is given. The analysis is based on the theory of ridge curves from digital image processing (see e.g. [24]). In addition, we discuss some strategies for choosing the starting points. These considerations arise when the input data has multiple, possibly intersecting curvilinear structures.

The remaining of this paper is organized as follows. In Section 2 we describe the generative model and discuss how to use the ridge curves to estimate the generating functions. Sections 3 and 4 are devoted to the development of the ridge tracing algorithm. In Section 5 we demonstrate the performance and reliability of the proposed algorithm on synthetically generated point sets as well as two observational datasets from seismology and cosmology. Finally, Section 6 summarizes this paper with concluding remarks.

2 Probabilistic Model and Density Estimation

In this section we recall the probabilistic model describing a noisy point set containing curvilinear structures mixed with background clutter. The model is essentially the one described in [45], and it is closely related to the one described in [28–30].

Given a point set sampled from the model, our aim is to estimate the curvilinear structures directly from the data with no prior assumptions on the model parameters. To this end, we consider the marginal density induced by the model. For estimation of the curvilinear structures from the marginal density, we define the concept of a ridge curve. Finally, for a computational implementation of this approach, we consider nonparametric estimation of the marginal density by using Gaussian kernels.

2.1 The Model

In the model, the sample points are assumed to belong to some compact domain $\Omega \subset \mathbb{R}^d$, and they fall into two distinct categories. A sample either belongs to some curvilinear structure, that we call a *filament*, or is background clutter. The type of a sample point is modeled by the random variable

$$T = \begin{cases} 1, & \text{if the sample belongs to a filament,} \\ 0, & \text{if the sample is background clutter} \end{cases}$$

having probabilities

$$P(T = 1) = \rho \quad \text{and} \quad P(T = 0) = 1 - \rho \tag{1}$$

with some $\rho \in]0, 1]$.

We define the random variable $\boldsymbol{X}$ to represent the sample points. When a sample drawn from $\boldsymbol{X}$ is background clutter (i.e. when $T = 0$), it is assumed to be uniformly distributed in the domain Ω. That is,

$$\boldsymbol{X} \mid (T = 0) \sim \mathcal{U}(\Omega). \tag{2}$$

On the other hand, when a sample drawn from $\boldsymbol{X}$ belongs to some filament (i.e. when $T = 1$), we assume that it is sampled in a random process from some generating function parametrizing the filament. The generating functions $\{\boldsymbol{f}_i\}_{i=1}^n : \mathcal{D}_i \to \mathbb{R}^d$, where n is the number of filaments, are defined as continuous mappings from some compact and connected domains $\mathcal{D}_i \subset \mathbb{R}$. Given the condition that $T = 1$, the outcome of the random variable $\boldsymbol{X}$ depends on three random variables: I, Θ and $\boldsymbol{\varepsilon}$. The random variable I with domain $\{1, 2, \ldots, n\}$ specifies which filament the sample belongs into, and the random variable Θ gives coordinate along the specified filament. In addition, we assume that the sample is obtained from the generating function with additive noise represented by the random variable $\boldsymbol{\varepsilon}$.

Furthermore, we assume that the random variables I and $\boldsymbol{\varepsilon}$ are distributed according to

$$P(I = i) = w_i \quad \text{and} \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_d(0, \sigma^2) \tag{3}$$

with $\boldsymbol{w} > \boldsymbol{0}$ such that $\sum_{i=1}^n w_i = 1$ and with $\mathcal{N}_d(0, \sigma^2)$ denoting a d-dimensional normal distribution with zero mean and variance σ^2. We also assume that given

$i \in \{1, 2, \ldots, n\}$, the conditional variable $\Theta \mid (I = i)$ follows some distribution defined in the domain $\mathcal{D}_i$.

The above assumptions yield the conditional random variable

$$\boldsymbol{X} \mid (T = 1, I = i, \Theta = \theta) = \boldsymbol{f}_i(\theta) + \boldsymbol{\varepsilon}$$

having the density

$$p_{\boldsymbol{X}}(\boldsymbol{x} \mid T = 1, I = i, \Theta = \theta) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{f}_i(\theta)\|^2}{2\sigma^2}\right). \quad (4)$$

The above model is in fact a generative model, since it specifies a random process for obtaining the samples from a set of generating functions. When the generating functions are known a priori, the reliability of a filament extraction algorithm can be evaluated by comparing the estimates to the known functions. An example point set sampled from the model is plotted in Figure 1.

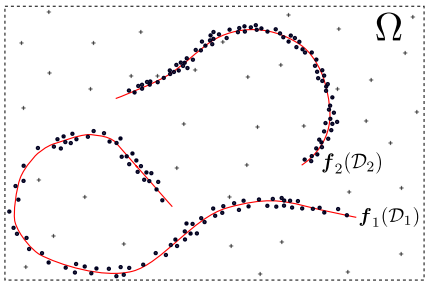


Figure 1: Filaments parametrized by two generating functions $\boldsymbol{f}_1 : \mathcal{D}_1 \to \mathbb{R}^2$ and $\boldsymbol{f}_2 : \mathcal{D}_2 \to \mathbb{R}^2$ with $\mathcal{D}_1 \subset \mathbb{R}$ and $\mathcal{D}_2 \subset \mathbb{R}$, noisy samples and background clutter.

Given a set of samples drawn from the above model, our aim is to estimate the functions $\boldsymbol{f}_i$ parametrizing the filaments. A typical situation arising in many real-world tasks is that there is not enough prior information to make parametric assumptions on the data-generating process. Therefore, in the following we consider *nonparametric* estimation of these functions directly from the samples represented by the *observed* random variable $\boldsymbol{X}$. To this end, we need to obtain a density for $\boldsymbol{X}$ that does not depend on the *latent* variables I, Θ and $\boldsymbol{\varepsilon}$.

By using the conditional random variables defined above, we can form the *joint density* and *marginalize* it to obtain a density that depends only on the random variable $\boldsymbol{X}$. Namely, by successively applying the relation between the joint and conditional densities we obtain

$$\begin{aligned} p_{\boldsymbol{X},T,I,\Theta}(\boldsymbol{x}, 1, i, \theta) &= p_{\boldsymbol{X}}(\boldsymbol{x} \mid T = 1, I = i, \Theta = \theta) p_{T,I,\Theta}(1, i, \theta) \\ &= p_{\boldsymbol{X}}(\boldsymbol{x} \mid T = 1, I = i, \Theta = \theta) p_{\Theta}(\theta \mid I = i) p_{T,I}(1, i) \\ &= p_{\boldsymbol{X}}(\boldsymbol{x} \mid T = 1, I = i, \Theta = \theta) p_{\Theta}(\theta \mid I = i) P(I = i) P(T = 1) \end{aligned}$$

and

$$p_{\boldsymbol{X},T,I,\Theta}(\boldsymbol{x}, 0, i, \theta) = p_{\boldsymbol{X}}(\boldsymbol{x} \mid T = 0) P(T = 0).$$

Then summing the joint density $p_{\boldsymbol{X},T,I,\Theta}(\boldsymbol{x}, t, i, \theta)$ over the domains of the discrete random variables T and I and integrating over the domain of the continuous variable Θ together with equations (1)-(4) yields the marginal density

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\rho}{(\sqrt{2\pi}\sigma)^d} \sum_{i=1}^{n} w_i \int_{\mathcal{D}_i} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{f}_i(\theta)\|^2}{2\sigma^2}\right) p_\Theta(\theta \mid I = i) d\theta + \frac{1-\rho}{V(\Omega)}, \quad (5)$$

where $V(\Omega)$ denotes the volume of the domain Ω.

The above density represents the observed density from a given set of samples from the model, and it depends only on $\boldsymbol{X}$. By marginalizing the joint density in this way we lose some information. As a result, ridge curves of the marginal density $p_{\boldsymbol{X}}$ give somewhat *biased* estimates of the generating functions $\boldsymbol{f}_i$. Nevertheless, as we shall see in the following, this approach allows a computationally tractable way of estimating the generating functions with a reasonably small bias.

2.2 Ridge Curves

Let us now define the concept of a ridge curve in order to estimate the generating functions $\boldsymbol{f}_i$ from the marginal density (5). A point on a ridge curve of a d-variate probability density is a (local) maximum on the cross-section of the density with respect to the hyperplane spanned by a subset of the Hessian eigenvectors. The eigenvectors in this subset correspond to the $d-1$ algebraically smallest eigenvalues of the Hessian matrix.

Definition 2.1. *A point $\boldsymbol{x} \in \mathbb{R}^d$ belongs to $\mathcal{R}_p$, the* set of ridge curves *of a twice differentiable probability density $p : \mathbb{R}^d \to \mathbb{R}$ if*

$$\nabla p(\boldsymbol{x})^T \boldsymbol{v}_i(\boldsymbol{x}) = 0, \quad \text{for all } 1 < i \leq d, \quad (6a)$$

$$\lambda_2(\boldsymbol{x}) < 0, \quad (6b)$$

$$\lambda_2(\boldsymbol{x}) < \lambda_1(\boldsymbol{x}), \quad (6c)$$

where $\lambda_1(\boldsymbol{x}) > \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$ and $\{\boldsymbol{v}_i(\boldsymbol{x})\}_{i=1}^d$ denote the eigenvalues and the corresponding eigenvectors of $\nabla^2 p(\boldsymbol{x})$, respectively.

A ridge curve is a connected set of ridge points lying on top of the density, as illustrated in Figure 2. As we shall see in Section 3, the ridge curve set $\mathcal{R}_p$ generally consists of set of multiple curves that are not connected to each other. From Definition 2.1 and Figure 2 we observe that a ridge curve passes through a set of modes (i.e. maxima) of the density. This property will be utilized in Sections 3 and 4, where an algorithm for obtaining the set $\mathcal{R}_p$ is developed. The algorithm starts tracing each ridge curve component from a mode, and each component curve of the set $\mathcal{R}_p$ is identified according to the modes it passes through.

The following result ensures that when we have a single generating function that parametrizes a line segment and no background clutter, the image of the generating function lies on the ridge curve of the marginal density (5). For the more general m-dimensional *ridge set* this result has been proven in [45].

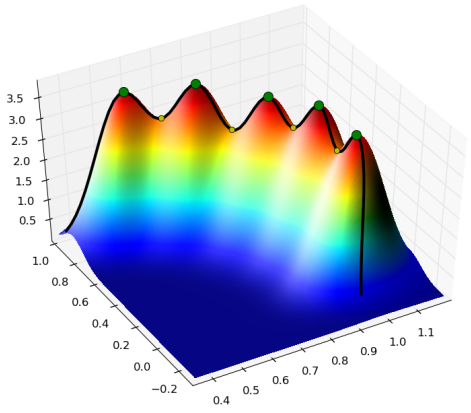


Figure 2: Ridge curve of a bivariate probability density.

Theorem 2.1. *Let $\mathcal{D} = [a, b]$ with some $a < b$ and let $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^d$ be defined as $\boldsymbol{f}(\theta) = \boldsymbol{x}_0 + \theta \boldsymbol{v}$ with some $\boldsymbol{x}_0 \in \mathbb{R}^d$ and $\boldsymbol{v} \in \mathbb{R}^d \setminus \{\boldsymbol{0}\}$. If p is defined by equation (5) with $n = 1$, $\rho = 1$, $\boldsymbol{f}_1 = \boldsymbol{f}$ and $\Theta \sim \mathcal{U}([a, b])$, then $\{\boldsymbol{f}(\theta) \mid \theta \in \mathcal{D}\} \subseteq \mathcal{R}_p$.*

The property that the ridge curves of the marginal density coincide with the generating functions does not generally hold when $n > 1$ or when one or more of the generating functions are nonlinear. Nevertheless, as we demonstrate by examples in Section 5, the ridge curves give accurate estimates of the generating functions when the data is sampled from the model with a sufficiently small amount of noise. To shed more light on this issue, in Appendix A we consider a special case where the model bias can be explicitly computed.

2.3 Kernel Density Estimation

For a computational implementation we consider estimation of the marginal density (5) from a given set of samples *nonparametrically* by using a *kernel density* estimate. The advantage of this approach is that it can be done directly from the data with no prior knowledge on the data-generating process. A widely used nonparametric estimation method is to use *Gaussian* kernels (see e.g. [46]). In such a density estimate, one Gaussian function is assigned for each sample point. The estimate requires choosing the matrix $\boldsymbol{H}$, for which a number of robust data-driven methods have been developed (see e.g. [11], [12], [22] and [23]).

Definition 2.2. *The* Gaussian kernel density estimate $\hat{p}$ *obtained by drawing a set of samples $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N \subset \mathbb{R}^d$ from some (unknown) probability density $p : \mathbb{R}^d \to \mathbb{R}$ is*

$$\hat{p}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{y}_i), \tag{7}$$

where the kernel $K_{\boldsymbol{H}} : \mathbb{R}^d \to]0, \infty[$ *is the* Gaussian *function*

$$K_{\boldsymbol{H}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{H}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{H}^{-1} \boldsymbol{x}\right) \tag{8}$$

with a symmetric and positive definite kernel bandwidth *matrix* $\boldsymbol{H}$.

The Gaussian kernel density estimate has a very favorable property of being a C^∞-function (i.e. infinitely many times continuously differentiable). As we shall see in the following section, the definition of a ridge curve of such a function is well-posed, and the set $\mathcal{R}_{\hat{p}}$ indeed defines a set of curves. With an appropriate choice of the bandwidth matrix $\boldsymbol{H}$, ridge curves of a Gaussian kernel density estimate give good estimates of the generating functions. To illustrate this fact, a Gaussian kernel density estimate obtained from a point set and the ridge curve of the density estimate are plotted in Figure 3.

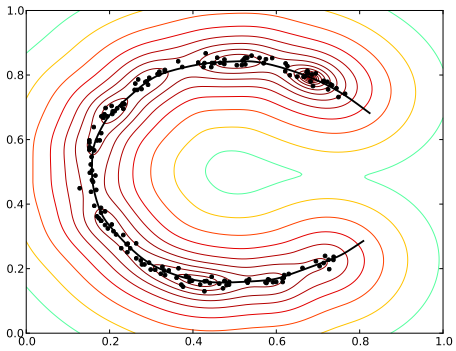


Figure 3: Contour lines and the ridge curve of a Gaussian kernel density estimate obtained from a point set generated from the model described in Subsection 2.1.

3 Theory of Ridge Curves

In this section we develop the necessary theory for extracting the ridge curve set $\mathcal{R}_{\hat{p}}$ of a given Gaussian kernel density estimate $\hat{p} : \mathbb{R}^d \to \mathbb{R}$. Based on the mathematical theory of ridge curves from digital image processing (see e.g. [24]), we first show that the definition of a ridge curve is well-posed for Gaussian kernel density estimates. Adapting the theory of gradient extremals from theoretical chemistry (see e.g. [35]), we then show that a ridge curve can be parametrized as the solution to a differential equation.

3.1 Existence of Ridge Curves

First of all, we need to ensure that Definition 2.1 is well-posed and that the set $\mathcal{R}_{\hat{p}}$ defines a set of curves. Furthermore, when this is the case and the curves have

endpoints, it is in our interest to analyze the behaviour of these curves at such points. This is essential in order to develop a ridge tracing algorithm that properly terminates when the followed ridge curve ends.

Motivated by applications in digital image processing, Damon [16] and Miller [40] give a rigorous analysis for ridge curves of C^∞-functions in a differential geometric framework. In [16] and [40], ridge curves are treated as a special case of the more general *critical curves*.

Definition 3.1. *Let $p \in C^\infty(\mathbb{R}^d, \mathbb{R})$ and let $\{\boldsymbol{v}_j\}_{j=1}^d : \mathbb{R}^d \to \mathbb{R}^d$ denote the eigenvectors of $\nabla^2 p$ corresponding to the eigenvalues $\lambda_1(\cdot) \geq \lambda_2(\cdot) \geq \cdots \geq \lambda_d(\cdot)$. The* set of critical curves *of p of index $i \in \{1, 2, \ldots, d\}$ is*

$$\mathcal{C}_p^i = \{\boldsymbol{x} \in \mathbb{R}^d \mid \nabla p(\boldsymbol{x})^T \boldsymbol{v}_j(\boldsymbol{x}) = 0 \text{ and } \lambda_j(\boldsymbol{x}) \neq \lambda_i(\boldsymbol{x}) \text{ for all } j \neq i\}.$$

Furthermore, the following definitions for different types of critical points are given in [16] and [40].

Definition 3.2. *Let $p \in C^\infty(\mathbb{R}^d, \mathbb{R})$ and let $\lambda_1(\cdot) \geq \lambda_2(\cdot) \geq \cdots \geq \lambda_d(\cdot)$ denote the eigenvalues of the Hessian $\nabla^2 p$. If $\boldsymbol{x} \in \mathcal{C}_p^i$ for some index i, then $\boldsymbol{x}$ is a*

(i) ridge point *of p if $i = 1$ and $\lambda_2(\boldsymbol{x}) < 0$.*

(ii) valley point *of p if $i = d$ and $\lambda_{d-1}(\boldsymbol{x}) > 0$.*

(iii) r-connector point *of p if $i = 1$ and $\lambda_2(\boldsymbol{x}) > 0$.*

(iv) v-connector point *of p if $i = d$ and $\lambda_{d-1}(\boldsymbol{x}) < 0$.*

(v) m-connector point *of p if $i > 1$ and $i < d$.*

It is not trivial to show when the set $\mathcal{R}_p$ defined by conditions (6a)-(6c) defines a set of curves. Fortunately, by generalizing the earlier results of [16] for bivariate functions, it has been shown in [40] that this property holds *generically* for C^∞-functions in higher dimensions as well. One of the main results of [40] is that the following properties hold generically in the sense that if some function $p \in C^\infty(\mathbb{R}^d, \mathbb{R})$ does not satisfy some property, then an arbitrarily small perturbation of p measured in the L_2-norm yields a function for which these properties are satisfied. Consequently, for any Gaussian kernel density estimate obtained from a point set generated from some random process, the following properties hold almost always. For a rigorous definition of genericity, we refer to [16] and [40].

Theorem 3.1. *For $p \in C^\infty(\mathbb{R}^d, \mathbb{R})$, the following properties are generically satisfied.*

(i) The set $\mathcal{C}_p = \bigcup_{i=1}^d \mathcal{C}_p^i$ consists of a discrete (i.e. finite or countably infinite) set of C^∞-curves. The curves in $\mathcal{R}_p$, which is a subset of $\mathcal{C}_p$, may have endpoints.

(ii) The curves in $\mathcal{R}_p$ do not intersect at any point and have no self-intersections.

(iii) *A connected component curve of $\mathcal{R}_p$ can have an endpoint $\boldsymbol{x}$ only when $\lambda_1(\boldsymbol{x}) = \lambda_2(\boldsymbol{x})$ or $\lambda_2(\boldsymbol{x}) = 0$.*

(iv) *When a ridge curve ends at a point $\boldsymbol{x}$ such that $\lambda_2(\boldsymbol{x}) = 0$, it is smoothly continued by an r-connector curve.*

(v) *When a ridge curve ends at a point $\boldsymbol{x}$ such that $\lambda_1(\boldsymbol{x}) = \lambda_2(\boldsymbol{x})$, it is smoothly continued by an m-connector curve.*

Remark 3.1. *When $d = 2$, the set $\mathcal{C}_p^1 \cup \mathcal{C}_p^2$ contains no m-connector points. Furthermore, property (v) in the above list then states that the ridge curve is continued by a v-connector curve.*

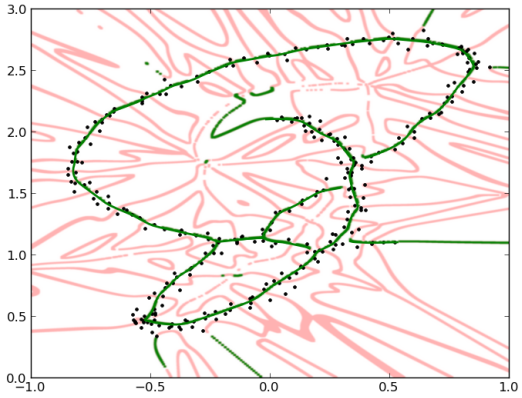


Figure 4: Critical curves (the set $\mathcal{C}_{\hat{p}}$) of a Gaussian kernel density estimate $\hat{p}$ obtained from a point set (light red). Critical curves belonging to $\mathcal{R}_{\hat{p}}$, the set of ridge curves, are plotted in dark green.

To illustrate the properties listed above, a point set and the set of critical and ridge curves of its Gaussian kernel density estimate $\hat{p}$ are plotted in Figure 4. The ridge curves in this case have endpoints and they are smoothly connected to critical curves. This figure also illustrates that in our application, only ridge curves are of interest since they give meaningful estimates for the generating functions of the data. Furthermore, it is apparent that the ridge tracing algorithm should be terminated when it enters a critical curve that is not in the set $\mathcal{R}_{\hat{p}}$.

3.2 Differential Equation Formulation

Given a function $p \in C^3(\mathbb{R}^d, \mathbb{R})$ with a nonempty ridge curve set $\mathcal{R}_p$ and a point $\boldsymbol{x}_0 \in \mathcal{R}_p$, we now derive the differential equation defining a ridge curve that passes through $\boldsymbol{x}_0$. Recalling Definition 2.1, the assumption that $\boldsymbol{x}_0 \in \mathcal{R}_p$ implies that

$$\nabla p(\boldsymbol{x}_0)^T \boldsymbol{v}_i(\boldsymbol{x}_0) = 0 \quad \text{for all } 1 < i \leq d, \tag{9}$$

where $\{\boldsymbol{v}_i(\boldsymbol{x}_0)\}_{i=1}^d$ denote the normalized eigenvectors of the Hessian $\nabla^2 p(\boldsymbol{x}_0)$ corresponding to the eigenvalues $\lambda_1(\boldsymbol{x}_0) > \lambda_2(\boldsymbol{x}_0) \geq \cdots \geq \lambda_d(\boldsymbol{x}_0)$. Since

$\nabla^2 p(\boldsymbol{x}_0)$ is symmetric, its eigenvectors are orthogonal, and consequently the gradient is parallel to the first eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_0)$. Thus, equation (9) implies that

$$\nabla^2 p(\boldsymbol{x}_0)\nabla p(\boldsymbol{x}_0) = \lambda_1(\boldsymbol{x}_0)\nabla p(\boldsymbol{x}_0), \tag{10}$$

which is equivalent to

$$\left[\nabla^2 p(\boldsymbol{x}_0) - \lambda_1(\boldsymbol{x}_0)\boldsymbol{I}\right] \nabla p(\boldsymbol{x}_0) = \boldsymbol{0}.$$

Again, by utilizing the property that the gradient $\nabla p(\boldsymbol{x}_0)$ is the first eigenvector of the Hessian $\nabla^2 p(\boldsymbol{x}_0)$ and normalizing the gradient, we obtain

$$\left[\nabla^2 p(\boldsymbol{x}_0) - \frac{\nabla p(\boldsymbol{x}_0)^T \nabla^2 p(\boldsymbol{x}_0) \nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|^2}\boldsymbol{I}\right] \frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} = \boldsymbol{0}.$$

By introducing the matrix

$$\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{I} - \frac{\nabla p(\boldsymbol{x}) \nabla p(\boldsymbol{x})^T}{\|\nabla p(\boldsymbol{x})\|^2} \tag{11}$$

projecting a given vector onto the subspace orthogonal to the gradient $\nabla p(\boldsymbol{x})$, this equation is equivalently written as

$$\boldsymbol{P}(\boldsymbol{x}_0)\nabla^2 p(\boldsymbol{x}_0)\frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} = \boldsymbol{0}. \tag{12}$$

Defining $\boldsymbol{x} : [0, \infty[\to \mathbb{R}^d$ as a curve passing through $\boldsymbol{x}_0$ and requiring that condition (12) holds along this curve yields the initial value problem

$$\frac{d}{d\theta}\left[\boldsymbol{P}(\boldsymbol{x}(\theta))\nabla^2 p(\boldsymbol{x}(\theta))\frac{\nabla p(\boldsymbol{x}(\theta))}{\|\nabla p(\boldsymbol{x}(\theta))\|}\right] = \boldsymbol{0}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{13}$$

Numerical solution of this initial value problem necessitates determining the tangent vector $\boldsymbol{x}'(\theta)$ for the solution curve. This can be done by utilizing the theory of *gradient extremal curves* developed in theoretical chemistry (see e.g. [9], [10], [34] and [35]). This is due to the fact that equation (10), and thus equation (13), are equivalent to the equations defining a gradient extremal curve passing through $\boldsymbol{x}_0$.[1] It has been shown, for instance, in [9] and [10] that calculating the derivative with respect to θ in equation (13) yields the equation

$$\boldsymbol{P}(\boldsymbol{x}(\theta))\boldsymbol{A}(\boldsymbol{x}(\theta))\boldsymbol{x}'(\theta) = \boldsymbol{0} \tag{14}$$

with

$$\boldsymbol{A}(\boldsymbol{x}) = \nabla^3 p(\boldsymbol{x})\nabla p(\boldsymbol{x}) + [\nabla^2 p(\boldsymbol{x})]^2 - \frac{\nabla p(\boldsymbol{x})^T \nabla^2 p(\boldsymbol{x}) \nabla p(\boldsymbol{x})}{\|\nabla p(\boldsymbol{x})\|^2}\nabla^2 p(\boldsymbol{x}). \tag{15}$$

[1] A gradient extremal point of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as a critical point of the gradient norm $\|\nabla f\|$ in a level set of f. A gradient extremal curve passes through such points.

Here we define the third derivative tensor as

$$[\nabla^3 p(\boldsymbol{x})]_{i,j,k} = \frac{\partial^3 p}{\partial x_i \partial x_j \partial x_k}(\boldsymbol{x})$$

and the tensor-vector product $\nabla^3 p(\boldsymbol{x})\nabla p(\boldsymbol{x})$ as

$$[\nabla^3 p(\boldsymbol{x})\nabla p(\boldsymbol{x})]_{i,k} = \sum_{j=1}^{d} [\nabla^3 p(\boldsymbol{x})]_{i,j,k} [\nabla p(\boldsymbol{x})]_j.$$

Whenever the matrix $\boldsymbol{P}(\boldsymbol{x}(\theta))\boldsymbol{A}(\boldsymbol{x}(\theta))$ has one-dimensional null space, the tangent vector $\boldsymbol{x}'(\theta)$ can be uniquely determined from equation (14) up to a scalar factor. When this is the case, the following result gives a formula for the tangent vector. In the following theorem and its proof given in Appendix B, we rephrase the result of [9] in our notation.

Theorem 3.2. *Let $p \in C^3(\mathbb{R}^d, \mathbb{R})$, $\boldsymbol{x}_0 \in \mathbb{R}^d$ and let*

$$\boldsymbol{P}(\boldsymbol{x}_0) = \boldsymbol{U}(\boldsymbol{x}_0)\boldsymbol{U}(\boldsymbol{x}_0)^T, \quad \text{where } \boldsymbol{U}(\boldsymbol{x}_0) \in \mathbb{R}^{d\times(d-1)} \tag{16}$$

be the eigendecomposition of the matrix $\boldsymbol{P}(\boldsymbol{x}_0)$ defined by equation (11). If $\nabla p(\boldsymbol{x}_0) \neq \boldsymbol{0}$ and the matrix $\boldsymbol{C}(\boldsymbol{x}_0) = \boldsymbol{U}(\boldsymbol{x}_0)^T \boldsymbol{A}(\boldsymbol{x}_0)\boldsymbol{U}(\boldsymbol{x}_0)$, where $\boldsymbol{A}(\cdot)$ is defined according to equation (15), is nonsingular, then the vector

$$\boldsymbol{u}^* = \frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} - \boldsymbol{U}(\boldsymbol{x}_0)\boldsymbol{C}(\boldsymbol{x}_0)^{-1}\boldsymbol{b}(\boldsymbol{x}_0) \tag{17}$$

with

$$\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})^T [\nabla^3 p(\boldsymbol{x})\nabla p(\boldsymbol{x})] \frac{\nabla p(\boldsymbol{x})}{\|\nabla p(\boldsymbol{x})\|} \tag{18}$$

and its scalar multiples are the only solutions to the equation

$$\boldsymbol{P}(\boldsymbol{x}_0)\boldsymbol{A}(\boldsymbol{x}_0)\boldsymbol{u} = \boldsymbol{0}. \tag{19}$$

The tangent vector given by equation (17) is not defined at a critical point (i.e. when $\nabla p(\boldsymbol{x}) = \boldsymbol{0}$). However, the following result covers this case. It shows that when an isolated critical point $\boldsymbol{x}_0$ of p (i.e. a critical point with a neighbourhood containing no other critical points of p) belonging to $\mathcal{R}_p$ is approached along a ridge curve, the tangent becomes parallel to the eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_0)$, and the limiting direction is well-defined. The proof of this result is given in Appendix B.

Theorem 3.3. *Let $p \in C^3(\mathbb{R}^d, \mathbb{R})$ and assume that there exists a continuous curve $\boldsymbol{x} : \mathcal{D} \to \mathbb{R}^d$ defined on some domain $\mathcal{D} \subset \mathbb{R}$ such that condition (6a) is satisfied for all $\boldsymbol{x}(\theta)$ with $\theta \in \mathcal{D}$. Further, assume that $\boldsymbol{x}(0) = \boldsymbol{x}_0$ for some isolated critical point $\boldsymbol{x}_0 \in \mathcal{R}_p$. If we define*

$$\boldsymbol{u}(\theta) = \frac{\nabla p(\boldsymbol{x}(\theta))}{\|\nabla p(\boldsymbol{x}(\theta))\|} - \boldsymbol{U}(\boldsymbol{x}(\theta))\boldsymbol{C}(\boldsymbol{x}(\theta))^{-1}\boldsymbol{b}(\boldsymbol{x}(\theta)), \tag{20}$$

where the matrix $\boldsymbol{U}(\cdot)$ *is defined according to* (16) *and the vector* $\boldsymbol{b}(\cdot)$ *is defined according to* (18)*, then*

$$\lim_{\theta \to 0} \left| \frac{\boldsymbol{u}(\theta)^T}{\|\boldsymbol{u}(\theta)\|} \boldsymbol{v}_1(\boldsymbol{x}_0) \right| = 1.$$

On the other hand, singularity of the matrix $\boldsymbol{C}(\cdot)$ may occur in two distinct ways, as shown in the following theorem. This result, whose proof is given in Appendix B, is a generalization from [9], where only the case when the matrix $\boldsymbol{C}(\cdot)$ has exactly one zero eigenvalue is considered.

Theorem 3.4. *Let* $p \in C^3(\mathbb{R}^d, \mathbb{R})$*,* $\boldsymbol{x}_0 \in \mathbb{R}^d$ *and let the matrices* $\boldsymbol{U}(\cdot)$*,* $\boldsymbol{A}(\cdot)$ *and* $\boldsymbol{C}(\cdot)$ *be defined as in Theorem 3.2 and assume that the matrix* $\boldsymbol{C}(\boldsymbol{x}_0)$ *is singular with eigenvalues* $\lambda_i = 0$ *for* $i \in I$*, where* $I \subset \{1, 2, \ldots, d-1\}$*. Let*

$$\boldsymbol{C}(\boldsymbol{x}_0) = \boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^T$$

with $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d-1}] \in \mathbb{R}^{(d-1)\times(d-1)}$ *and the diagonal matrix* $\boldsymbol{D} \in \mathbb{R}^{(d-1)\times(d-1)}$ *be the eigendecomposition of* $\boldsymbol{C}(\boldsymbol{x}_0)$ *and define the vector* $\boldsymbol{b}(\cdot)$ *according to equation* (18)*. If* $\boldsymbol{w}_i^T \boldsymbol{b}(\boldsymbol{x}_0) \neq 0$ *for some* $i \in I$*, then solutions to equation* (19) *with respect to* $\boldsymbol{u}$ *are of the form*

$$\boldsymbol{u}(\boldsymbol{\beta}) = \boldsymbol{U}(\boldsymbol{x}_0) \sum_{i \in I} \beta_i \boldsymbol{w}_i \tag{21}$$

with $\boldsymbol{\beta} \in \mathbb{R}^{|I|}$*. Otherwise, if* $\boldsymbol{w}_i^T \boldsymbol{b}(\boldsymbol{x}_0) = 0$ *for all* $i \in I$*, then solutions to equation* (19) *lie in the subspace spanned by the vector* $\boldsymbol{u}(\boldsymbol{\beta})$ *defined by equation* (21) *and the vector*

$$\tilde{\boldsymbol{u}} = \frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} - \boldsymbol{U}(\boldsymbol{x}_0) \sum_{\substack{i=1 \\ i \notin I}}^{d-1} \frac{\boldsymbol{w}_i^T \boldsymbol{b}(\boldsymbol{x}_0)}{d_{ii}} \boldsymbol{w}_i. \tag{22}$$

Theorem 3.4 gives rise to the following definition adapted from [9].

Definition 3.3. *Given the definitions of Theorem 3.4 and a function* $p \in C^3(\mathbb{R}^d, \mathbb{R})$ *and* $\boldsymbol{x}_0 \in \mathcal{R}_p$*, if the matrix* $\boldsymbol{C}(\boldsymbol{x}_0)$ *is singular and* $\boldsymbol{w}_i^T \boldsymbol{b} \neq 0$ *for some* $i \in I$*, then* $\boldsymbol{x}_0$ *is a* turning point *. Otherwise, if* $\boldsymbol{C}(\boldsymbol{x}_0)$ *is singular and* $\boldsymbol{w}_i^T \boldsymbol{b} = 0$ *for all* $i \in I$*, then* $\boldsymbol{x}_0$ *is a* bifurcation point*.*

Theorem 3.4 shows that the tangent vector for the solution curve of (13) cannot be usually uniquely determined when the matrix $\boldsymbol{C}(\boldsymbol{x}(\theta))$ becomes singular. Fortunately, this is not an issue in our application, since by Theorem 3.1 the ridge curve is smooth at such points for almost all Gaussian kernel density estimates.

It is important to note that according to Theorem 3.4 the ridge curve tangent becomes orthogonal to the gradient ∇p at at a turning point. Considering our application, Theorem 2.1 suggests that in this case the underlying generating function of the model deviates significantly from a linear function. Consequently, the

ridge curve is unlikely to give any meaningful estimate of any generating function, which gives a stopping criterion for tracing a ridge curve. On the other hand, bifurcation points seem to occur very rarely, and thus they are not an issue in practice.

4 Algorithm for Extracting the Ridge Curve Set

With the mathematical theory in place, we now develop an algorithm for obtaining the ridge curve set $\mathcal{R}_{\hat{p}}$ of a Gaussian kernel density estimate $\hat{p}$. Motivated by the fact that ridge curves of $\hat{p}$ pass through a set of its modes (cf. Definition 2.1), the algorithm first finds the modes (maxima) of $\hat{p}$. Then, by using these modes as starting points the algorithm constructs the set $\mathcal{R}_{\hat{p}}$ by tracing its component curves passing through these modes.

4.1 Definitions and Overview of the Algorithm

In practice, we are interested in ridge curves lying in areas of high probability density and consider two ridge curves separated when a low-density area lies between them. Low-density areas are of less interest because they are likely to represent background clutter or insignificant features in the data. Thus, we consider the set of ε-*separated* ridge curves

$$\mathcal{R}_{\hat{p},\varepsilon} = \mathcal{R}_{\hat{p}} \cap \{\boldsymbol{x} \in \mathbb{R}^d \mid \hat{p}(\boldsymbol{x}) > \varepsilon\}$$

that is a collection of ridge curve components separated by areas where the density $\hat{p}$ is smaller than some given threshold $\varepsilon > 0$. Given an $\varepsilon > 0$ and a mode $\boldsymbol{x}_0 \in \mathcal{R}_{\hat{p},\varepsilon}$, we then define the corresponding component of the ε-separated ridge curve set $\mathcal{R}_{\hat{p},\varepsilon}$ as the set

$$\begin{aligned}\mathcal{R}_{\hat{p},\varepsilon,\boldsymbol{x}_0} = \{\boldsymbol{y} \in \mathbb{R}^d \mid \exists \boldsymbol{x} \in C^\infty([0,1],\mathbb{R}^d) : \boldsymbol{x}(\theta) \in \mathcal{R}_{\hat{p}} \quad \forall \theta \in [0,1],\\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \text{ and } \boldsymbol{y} = \boldsymbol{x}(1)\} \cap \mathcal{R}_{\hat{p},\varepsilon}\end{aligned}$$

containing points on some ridge curve in the set $\mathcal{R}_{\hat{p},\varepsilon}$ passing through $\boldsymbol{x}_0$. This definition is justified by Theorem 3.1 which guarantees that generically each mode of $\hat{p}$ belonging to the set $\mathcal{R}_{\hat{p},\varepsilon}$ lies exactly on one smooth ridge curve.

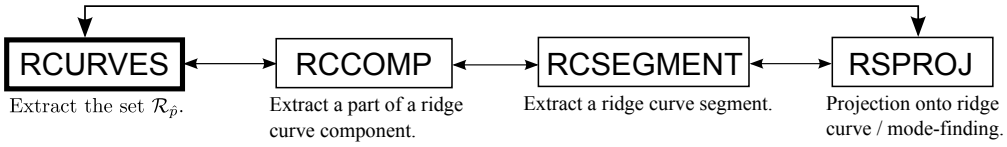


Figure 5: Components of the algorithm for extraction of the ridge curve set $\mathcal{R}_{\hat{p},\varepsilon}$.

The algorithmic framework for constructing the set $\mathcal{R}_{\hat{p},\varepsilon}$ consists of four algorithms, as shown in Figure 5. The main algorithm `RCURVES` first finds the modes

of $\hat{p}$ by starting a local maximization method from each sample point. Then, by using these modes as starting points, it constructs the set $\mathcal{R}_{\hat{p},\varepsilon}$ by successively applying the RCCOMP algorithm to extract its components. The RCCOMP algorithm, in turn, constructs a part of a ridge curve component originating from a given mode by successively invoking the RCSEGMENT algorithm. The RCSEGMENT algorithm traces ridge curve segment (i.e. a part of a ridge curve connecting at most two modes of the density) by using a predictor-corrector method adapted to the initial value problem (13). The RCURVES, RCCOMP and RCSEGMENT algorithms will be described in Subsections 4.2, 4.3 and 4.4, respectively.

The RSPROJ algorithm lies in the core of the algorithmic framework. Implementing the Newton-based method developed in [45], it projects a given point onto the m-dimensional *ridge set* of the density estimate $\hat{p}$. The ridge set of $\hat{p}$ is a generalization of its set of modes. Namely, ridge curves of $\hat{p}$ belong to its one-dimensional ridge set, and they pass through modes belonging to its zero-dimensional ridge set. Therefore the RSPROJ algorithm is used in the RCURVES algorithm for finding the modes of $\hat{p}$ and also in the RCSEGMENT algorithm as a corrector method that projects the predictor estimate back to the traced ridge curve.

For a numerical implementation of the algorithms, we consider a scaled version of the Gaussian kernel density estimate in order to avoid dependency on scaling of the data. By utilizing the Cholesky factorization $\boldsymbol{H} = \boldsymbol{L}\boldsymbol{L}^T$, the density estimate $\hat{p}$ defined by equations (7) and (8) can be written as

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N\sqrt{|\boldsymbol{H}|}} \sum_{i=1}^{N} K_{\boldsymbol{I}}(\boldsymbol{x} - \boldsymbol{L}^{-1}\boldsymbol{y}_i),$$

where $\boldsymbol{I}$ denotes the $d \times d$ identity matrix. The scaled density estimate $\tilde{p}$ is related to the original one via the identity $\tilde{p}(\boldsymbol{L}^{-1}\boldsymbol{x}) = \hat{p}(\boldsymbol{x})$.

In the following, we assume that the algorithms are supplied with a set of sample points $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N \subset \mathbb{R}^d$, a scaled Gaussian kernel density estimate $\tilde{p} : \mathbb{R}^d \to \mathbb{R}$ obtained from the point set and a low density threshold $\varepsilon > 0$.

4.2 The Main Algorithm (RCURVES)

The RCURVES algorithm (Algorithm 1) is the main algorithm for constructing the set $\mathcal{R}_{\tilde{p},\varepsilon}$. It produces a collection of discrete approximations of component curves of $\mathcal{R}_{\tilde{p},\varepsilon}$ that we shall denote as $\boldsymbol{X} \subset \mathcal{P}(\mathcal{R}_{\tilde{p},\varepsilon})$.

The first step of the algorithm is to find a set of modes $\boldsymbol{Z}^* \subset \mathcal{R}_{\tilde{p},\varepsilon}$ that are used as starting points for extracting the ridge curves. Starting from each sample point $\boldsymbol{y} \in \boldsymbol{Y}$, the RSPROJ algorithm is invoked to find a mode $\boldsymbol{y}^*$ (i.e. a point $\boldsymbol{y}^* \in \mathcal{R}_{\tilde{p},\varepsilon}$ such that $\lambda_1(\boldsymbol{y}^*) < 0$). The maximum trust region radius is chosen as $\Delta_{\max} = \frac{1}{2}$, and the stopping criterion threshold is chosen as $\varepsilon_{\text{corr}} = 10^{-6}$, where the meaning of these parameters is explained in [45]. The set $\boldsymbol{Z}^*$ is constructed so

Algorithm 1: RCURVES (extract ridge curve set).

input : point set $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N \subset \mathbb{R}^d$
Gaussian kernel density estimate $\tilde{p} : \mathbb{R}^d \to \mathbb{R}$
low probability density threshold $\varepsilon > 0$
parameters $\varepsilon_a, \varepsilon_c, \varepsilon_e$ and ε_r (see Algorithm 3)
output: collection of approximate ridge curves $\boldsymbol{X} \subset \mathcal{P}(\mathcal{R}_{\tilde{p},\varepsilon})$
$\boldsymbol{Z}^* \leftarrow \emptyset$
for $\boldsymbol{y} \in \boldsymbol{Y}$ **do**
  $\boldsymbol{y}^* \leftarrow \text{RSPROJ}(\tilde{p}, 0, \boldsymbol{y}, \frac{1}{2}, 10^{-6})$
  if $\|\boldsymbol{y}^* - \boldsymbol{z}\| > 10^{-5}$ for all $\boldsymbol{z} \in \boldsymbol{Z}^*$, $\boldsymbol{y}^* \in \mathcal{R}_{\tilde{p},\varepsilon}$ and $\lambda_1(\boldsymbol{y}^*) < 0$ **then**
    $\boldsymbol{Z}^* \leftarrow \boldsymbol{Z}^* \cup \{\boldsymbol{y}^*\}$
$\boldsymbol{X} \leftarrow \emptyset$
$\boldsymbol{M} \leftarrow \emptyset$
for $\boldsymbol{z}^* \in \boldsymbol{Z}^*$ **do**
  if $\|\boldsymbol{z}^* - \boldsymbol{x}\| > 10^{-5}$ for all $\boldsymbol{x} \in \boldsymbol{M}$ **then**
    $\boldsymbol{X}^+, \boldsymbol{M} \leftarrow \text{RCCOMP}(\tilde{p}, \boldsymbol{M}, \boldsymbol{z}^*, 1, \varepsilon, \varepsilon_a, \varepsilon_c, \varepsilon_e, \varepsilon_r)$
    if $\|\boldsymbol{x}^+_{|\boldsymbol{x}^+|-1} - \boldsymbol{z}^*\| > 10^{-5}$ **then**
      $\boldsymbol{X}^-, \boldsymbol{M} \leftarrow \text{RCCOMP}(\tilde{p}, \boldsymbol{M}, \boldsymbol{z}^*, -1, \varepsilon, \varepsilon_a, \varepsilon_c, \varepsilon_e, \varepsilon_r)$
    Construct the sequence $\tilde{\boldsymbol{X}}$ according to (23) and set $\boldsymbol{X} = \boldsymbol{X} \cup \tilde{\boldsymbol{X}}$.

that duplicate modes (within numerical precision) are not included. This is done in the RCURVES algorithm by adding $\boldsymbol{y}^*$ to the set of modes $\boldsymbol{Z}^*$ only when the condition

$$\|\boldsymbol{y}^* - \boldsymbol{z}\| > 10^{-5} \quad \text{for all } \boldsymbol{z} \in \boldsymbol{Z}^*$$

is satisfied and $\boldsymbol{y}^*$ is in the set $\mathcal{R}_{\tilde{p},\varepsilon}$ such that $\lambda_1(\boldsymbol{y}^*) < 0$.

Remark 4.1. *When the computational budget is limited and the aim is to find a single significant curve from the data, an alternative approach could be to use the method developed in [44]. Based on a homotopy continuation method, it finds a significant mode of a Gaussian kernel density estimate at a low computational cost. As demonstrated in [44], such a mode usually represents a region with a significant concentration of sample points that is clearly distinguishable from background clutter.*

From each starting point $\boldsymbol{z}^*$ in the set of modes $\boldsymbol{Z}^*$, RCURVES then calls the RCCOMP algorithm (Algorithm 2) twice to extract both parts of the ridge curve component $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{z}^*}$ originating from $\boldsymbol{z}^*$ into two opposite directions. Recalling Theorem 3.3, these directions are parallel to the eigenvector $\boldsymbol{v}_1(\boldsymbol{z}^*)$. Calling RCCOMP yields the sequences

$$\boldsymbol{X}^+ = (\boldsymbol{x}_0^+, \boldsymbol{x}_1^+, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{z}^*} \quad \text{and} \quad \boldsymbol{X}^- = (\boldsymbol{x}_0^-, \boldsymbol{x}_1^-, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{z}^*}$$

representing successive points along the ridge curve component.

Since the RCURVES algorithm extracts ridge curve components by using multiple starting points, it is essential to identify components that have already been extracted. Recalling that a ridge curve passes through a set of modes, a computationally convenient way to do this is to test whether the mode $\boldsymbol{z}^*$ chosen as the current starting point belongs to some already extracted component. Therefore, the algorithm maintains the set $\boldsymbol{M} \subset \mathbb{R}^d$ containing the modes visited during the executions of RCCOMP.

Before starting extraction of a ridge curve component corresponding to the sequence $\boldsymbol{X}^+$ from the starting point $\boldsymbol{z}^*$, the RCURVES algorithm tests whether $\boldsymbol{z}^*$ already belongs to the set of visited modes $\boldsymbol{M}$ (within numerical precision). The algorithm does this by testing the condition

$$\|\boldsymbol{z}^* - \boldsymbol{x}\| > 10^{-5} \quad \text{for all } \boldsymbol{x} \in \boldsymbol{M}$$

and skips ridge curve extraction from $\boldsymbol{z}^*$ when this condition is not satisfied.

Before starting the extraction of the second ridge curve component corresponding to the sequence $\boldsymbol{X}^-$, the RCURVES algorithm again tests whether the mode $\boldsymbol{z}^*$ has been visited. Namely, this can occur during the first call of RCCOMP if the ridge curve component forms a closed loop (that is when $\boldsymbol{x}^+_{|\boldsymbol{x}^+|-1} = \boldsymbol{z}^*$). Taking into account the limited numerical precision, the algorithm tests this by the condition

$$\|\boldsymbol{x}^+_{|\boldsymbol{X}^+|-1} - \boldsymbol{z}^*\| > 10^{-5}$$

and skips extraction of the second part of the ridge curve component when this condition is not satisfied.

In order to obtain a consistent ordering of points along the current ridge curve component, the sequences $\boldsymbol{X}^+$ and $\boldsymbol{X}^-$ representing the two parts of the component curve are at the end of the second loop in the RCURVES algorithm collected in the sequence $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \dots)$ defined as

$$\tilde{\boldsymbol{x}}_i = \begin{cases} \boldsymbol{x}^-_{|\boldsymbol{X}^-|-i-1}, & i = 0, 1, \dots, |\boldsymbol{X}^-| - 1, \\ \boldsymbol{x}^+_{i-|\boldsymbol{X}^-|}, & i = |\boldsymbol{X}^-|, |\boldsymbol{X}^-| + 1, \dots, |\boldsymbol{X}^-| + |\boldsymbol{X}^+| - 1. \end{cases} \tag{23}$$

Finally, the sequence $\tilde{\boldsymbol{X}}$ representing a discrete approximation of the ridge curve component $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{z}^*}$ is added to the set $\boldsymbol{X}$.

4.3 Extraction of a Ridge Curve Component (RCCOMP)

Given a mode $\boldsymbol{x}_0^* \in \mathcal{R}_{\tilde{p},\varepsilon}$ and a sign parameter $s^* \in \{-1, 1\}$, the RCCOMP algorithm (Algorithm 2) traces a part of a component curve of the set $\mathcal{R}_{\tilde{p},\varepsilon}$ passing through $\boldsymbol{x}_0^*$ (i.e. a subset of $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0^*}$). That is, starting from $\boldsymbol{x}_0^*$, the algorithm generates a sequence $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0^*}$ along the component curve $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0^*}$. The parameter s^* specifies whether tracing of the ridge curve is initiated along

the positive or negative tangent direction, which by Theorem 3.3 is parallel to the eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_0^*)$.

The RCCOMP algorithm constructs the ridge curve component by successively extracting and connecting its segments (i.e. parts of the curve connecting at most two modes). At the beginning of each iteration of RCCOMP, a segment starting from the current mode $\boldsymbol{x}^*$ along the direction specified by the sign parameter s^* is extracted by invoking the RCSEGMENT algorithm (Algorithm 3). This algorithm returns the sequence of points $\boldsymbol{X}^{**} = (\boldsymbol{x}_0^{**}, \boldsymbol{x}_1^{**}, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}^*}$, where $\boldsymbol{x}_0^{**} = \boldsymbol{x}^*$, along the segment, the endpoint of the segment $\boldsymbol{x}^{**} \in \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}^*}$, a scalar c indicating the type of the endpoint and the sign parameter s^{**} at the endpoint $\boldsymbol{x}^{**}$ (see Subsection 4.4). The sequence $\boldsymbol{X}^{**}$ is then appended to the sequence $\boldsymbol{X}$ representing the whole ridge curve component. In addition, the current mode $\boldsymbol{x}^*$ is marked as visited by adding it to the set of visited modes $\boldsymbol{M}$. When invoked from the RCURVES algorithm, RCCOMP takes the set $\boldsymbol{M}$ as input argument and upon termination returns the updated set $\boldsymbol{M}$ back to RCURVES.

Algorithm 2: RCCOMP (extract a part of a ridge curve component).

input : Gaussian kernel density estimate $\tilde{p} : \mathbb{R}^d \to \mathbb{R}$
visited modes $\boldsymbol{M} \subset \mathbb{R}^d$
a mode $\boldsymbol{x}_0 \in \mathcal{R}_{\tilde{p},\varepsilon}$ (i.e. a point $\boldsymbol{x}_0 \in \mathcal{R}_{\tilde{p},\varepsilon}$ such that $\lambda_1(\boldsymbol{x}_0) < 0$)
sign parameter $s^* \in \{-1, 1\}$
low probability density threshold $\varepsilon > 0$
parameters $\varepsilon_a, \varepsilon_c, \varepsilon_e$ and ε_r (see Algorithm 3)

output: subset of a ridge curve component $\boldsymbol{X} \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0^*}$
visited modes $\boldsymbol{M} \subset \mathbb{R}^d$

$\boldsymbol{X} \leftarrow \emptyset$
$\boldsymbol{x}^* \leftarrow \boldsymbol{x}_0^*$
while not terminated **do**
  $\boldsymbol{X}^{**}, \boldsymbol{x}^{**}, c, s^{**} \leftarrow$ RCSEGMENT$(\tilde{p}, \boldsymbol{x}^*, s^*, \varepsilon, \varepsilon_a, \varepsilon_c, \varepsilon_e, \varepsilon_r)$
  $\boldsymbol{X} \leftarrow (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_{|\boldsymbol{X}|-1}, \boldsymbol{x}_0^{**}, \boldsymbol{x}_1^{**}, \dots, \boldsymbol{x}_{|\boldsymbol{X}^{**}|-1}^{**})$
  $\boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{\boldsymbol{x}^*\}$
  if $c = 0$ **then**
    if $\|\boldsymbol{x}^{**} - \boldsymbol{x}\| > 10^{-5}$ for all $\boldsymbol{x} \in \boldsymbol{M}$ **then**
      $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{**}$; $s^* \leftarrow s^{**}$
    else Terminate.
  else Terminate.

When RCSEGMENT terminates at a point $\boldsymbol{x}^{**}$ that is a mode, it returns with $c = 0$. When $c = 0$, and the mode $\boldsymbol{x}^{**}$ has not been visited, that is, taking into account the limited numerical precision,

$$\|\boldsymbol{x}^{**} - \boldsymbol{x}\| > 10^{-5} \quad \text{for all } \boldsymbol{x} \in \boldsymbol{M}, \tag{24}$$

RCCOMP initiates tracing of a new ridge curve segment. To this end, the current mode $\boldsymbol{x}^*$ is set to $\boldsymbol{x}^{**}$. In order to ensure progress along the ridge curve, extraction of the next segment from $\boldsymbol{x}^{**}$ is started along the direction s^{**} returned upon termination of RCSEGMENT by setting the current direction s^* to s^{**}.

On the other hand, when $c = 0$ and the endpoint $\boldsymbol{x}^{**}$ is a mode that has already been visited (i.e. when condition (24) is not satisfied), the RCCOMP algorithm terminates. This situation occurs when the ridge curve component forms a closed loop. Also, when RCSEGMENT terminates by some other stopping criterion than crossing a mode (e.g. when it leaves a ridge curve or the density $\tilde{p}$ becomes smaller than ε), it returns with $c = 1$ and RCCOMP terminates.

Remark 4.2. *In principle, it would be sufficient to test that the mode $\boldsymbol{x}^{**}$ does not coincide with the starting point $\boldsymbol{x}_0^*$ since a ridge curve generally cannot intersect any other ridge curve and can only intersect itself when forming a closed loop (cf. Theorem 3.1). However, the more restrictive criterion (24) prevents extracting the same component curves of $\mathcal{R}_{\tilde{p},\varepsilon}$ multiple times when RCCOMP jumps from a component curve to another. This may happen when the step size τ_k is large and two curves are close to each other (cf. Figure 4).*

4.4 Tracing a Ridge Curve Segment (RCSEGMENT)

Finally, we describe the RCSEGMENT algorithm for tracing a ridge curve segment originating from a mode $\boldsymbol{x}_0 \in \mathcal{R}_{\tilde{p},\varepsilon}$. A ridge curve segment ends when it crosses another mode or leaves the set $\mathcal{R}_{\tilde{p},\varepsilon}$. The latter case occurs when the conditions defining a ridge curve become violated or when the ridge curve enters a region of low probability density. For tracing such a segment, RCSEGMENT implements a predictor-corrector method that traces a solution curve of the differential equation (13) satisfying the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ until either of these termination conditions is met.

4.4.1 Predictor-Corrector Algorithm

The algorithm generates a sequence of points $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon}$ along the ridge curve segment. At each iteration the algorithm takes a *predictor* step

$$\tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k + \tau_k s_k \boldsymbol{u}_k$$

along the normalized solution curve tangent $\boldsymbol{u}_k$ or its approximation with some step size $\tau_k > 0$ and sign parameter $s_k \in \{-1, 1\}$ in order to proceed along the solution curve.

The tangent vector $\boldsymbol{u}_k$ is chosen according to the rule

$$\boldsymbol{u}_k = \begin{cases} \boldsymbol{v}_1(\boldsymbol{x}_k), & \text{if } k = 0, \\ \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, & \text{otherwise}, \end{cases} \tag{25}$$

where

$$\boldsymbol{w}_k = \frac{\nabla\tilde{p}(\boldsymbol{x}_k)}{\|\nabla\tilde{p}(\boldsymbol{x}_k)\|} - \boldsymbol{U}(\boldsymbol{x}_k)\boldsymbol{C}(\boldsymbol{x}_k)^{-1}\boldsymbol{b}(\boldsymbol{x}_k)$$

with $\boldsymbol{b}(\cdot)$ defined according to (18). The latter choice in (25) is given by equation (17), and by the first choice, the algorithm avoids unnecessary computation of the third derivatives $\nabla^3\tilde{p}(\boldsymbol{x}_k)$ for the first iteration. Namely, by Theorem 3.3 the ridge curve tangent near a mode is approximately parallel to the eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_k)$. For the predictor estimate $\tilde{\boldsymbol{x}}_k$, the algorithm then tests the stopping criteria (27) given in Subsection 4.4.2. These criteria test whether the iteration has entered into a region where either of conditions (6a)-(6c) become violated.

The purpose of the sign parameter s_k is to ensure that the iteration proceeds forward along the solution curve. This is necessary since the orientation of the tangent vector $\boldsymbol{u}_k$ is not uniquely determined in either of the two cases in equation (25). At the first iteration $k = 0$, the sign is chosen as the user-supplied parameter $s_0 \in \{-1, 1\}$. For the subsequent iterations $k = 1, 2, \ldots$, s_k is chosen so that

$$s_k = \begin{cases} 1, & \text{if } s_{k-1}\boldsymbol{u}_{k-1}^T\boldsymbol{u}_k > 0, \\ -1, & \text{otherwise.} \end{cases} \tag{26}$$

After the predictor step, a *corrector* step is applied to project the predictor estimate back to the ridge curve. For this purpose, the algorithm uses the `RSPROJ` algorithm with ridge dimension $m = 1$. At each iteration of the predictor-corrector algorithm, the maximum trust region radius for `RSPROJ` is chosen as $\Delta_{\max} = \frac{\tau_k}{4}$.

4.4.2 Step Size Adaptation and Stopping Criteria

The ridge tracing algorithm uses an adaptive strategy for adjusting the predictor step size τ_k. Initially, τ_0 is set to $\frac{1}{10}$. For $k > 0$, after the predictor step the algorithm tests the conditions

$$\frac{|\nabla\tilde{p}(\tilde{\boldsymbol{x}}_k)^T\boldsymbol{v}_1(\tilde{\boldsymbol{x}}_k)|}{\|\nabla\tilde{p}(\tilde{\boldsymbol{x}}_k)\|} > 1 - \varepsilon_r \quad \text{and} \quad \lambda_2(\tilde{\boldsymbol{x}}_k) < 0 \tag{27}$$

with some small $\varepsilon_r \in]0, 1[$, where the first condition correponds to (6a), and the second condition corresponds to (6b). If either one of conditions (27) is not satisfied, the algorithm sets τ_k to $\frac{1}{2}\tau_k$ and updates the predictor estimate $\tilde{\boldsymbol{x}}_k$ accordingly. This is repeated as long as either of conditions (27) is not satisfied or $\tau_k < 10^{-6}$. The latter case indicates that the current iterate is near an endpoint of the ridge curve, and the algorithm terminates. On the other hand, when conditions (27) are both satisfied for the first predictor estimate $\tilde{\boldsymbol{x}}_k$, then the step size can be safely increased, and for the next iteration the algorithm chooses $\tau_{k+1} = 1.1\tau_k$.

In addition to the predictor conditions (27), the ridge tracing algorithm uses the stopping criteria

$$\tilde{p}(\boldsymbol{x}_k) < \varepsilon, \quad \frac{\lambda_1(\boldsymbol{x}_k)}{\lambda_2(\boldsymbol{x}_k)} > 1 - \varepsilon_e \quad \text{and} \quad \boldsymbol{u}_k^T\boldsymbol{v}_1(\boldsymbol{x}_k) < 1 - \varepsilon_a,$$

Algorithm 3: RCSEGMENT (extract ridge curve segment).

input : Gaussian kernel density estimate $\tilde{p} : \mathbb{R}^d \to \mathbb{R}$
a mode $\boldsymbol{x}_0 \in \mathcal{R}_{\tilde{p},\varepsilon}$ (i.e. a point $\boldsymbol{x}_0 \in \mathcal{R}_{\tilde{p},\varepsilon}$ such that $\lambda_1(\boldsymbol{x}_0) < 0$)
initial sign parameter $s_0 \in \{-1, 1\}$
low probability density threshold $\varepsilon > 0$
threshold parameters $\varepsilon_a, \varepsilon_c, \varepsilon_e, \varepsilon_r \in]0, 1[$

output: points $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots) \subset \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0}$ on a ridge curve segment
stopping criterion type $c \in \{0, 1\}$
(0=mode, 1=low density or iteration has left a ridge curve)
Returned when terminated at a mode:
the mode $\boldsymbol{x}^* \in \mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0}$
the current sign parameter $s^* \in \{-1, 1\}$

$\boldsymbol{X} \leftarrow (\boldsymbol{x}_0)$
$\boldsymbol{u}_0 \leftarrow \boldsymbol{v}_1(\boldsymbol{x}_0)$
$\tau_0 \leftarrow \frac{1}{10}$
for $k = 0, 1, \dots$ **do**
  if $\tilde{p}(\boldsymbol{x}_k) < \varepsilon$ or $\frac{\lambda_1(\boldsymbol{x}_k)}{\lambda_2(\boldsymbol{x}_k)} > 1 - \varepsilon_e$ **then** Terminate with $c = 1$.
  if $k > 0$ **then**
    if $\|\nabla\tilde{p}(\boldsymbol{x}_k)\| < 10^{-5}$ **then**
      $\boldsymbol{u}_k \leftarrow \boldsymbol{v}_1(\boldsymbol{x}_k)$
    else
      $\boldsymbol{u}_k \leftarrow \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}$, where $\boldsymbol{w}_k$ is defined by (25)
      if $\boldsymbol{u}_k^T \boldsymbol{v}_1(\boldsymbol{x}_k) < 1 - \varepsilon_a$ **then** Terminate with $c = 1$.
    if $s_{k-1}\boldsymbol{u}_{k-1}^T\boldsymbol{u}_k > 0$ **then** $s_k \leftarrow 1$ **else** $s_k \leftarrow -1$
    if conditions (28) are satisfied **then**
      $\boldsymbol{x}^* \leftarrow \text{RSPROJ}(\tilde{p}, 0, (\boldsymbol{x}_k + \boldsymbol{x}_{k-1})/2, \tau_k/2, 10^{-6})$
      $\boldsymbol{X} \leftarrow (\boldsymbol{x}_0, \dots, \boldsymbol{x}_{k-1}, \boldsymbol{x}^*)$
      if $s_{k-1}\boldsymbol{u}_{k-1}^T\boldsymbol{v}_1(\boldsymbol{x}^*) > 0$ **then** $s^* \leftarrow 1$ **else** $s^* \leftarrow -1$
      Terminate with $c = 0$.
  $\tilde{\boldsymbol{x}}_k \leftarrow \boldsymbol{x}_k + \tau_k s_k \boldsymbol{u}_k$
  $j \leftarrow 0$
  while $\frac{|\nabla\tilde{p}(\tilde{\boldsymbol{x}}_k)^T\boldsymbol{v}_1(\tilde{\boldsymbol{x}}_k)|}{\|\nabla\tilde{p}(\tilde{\boldsymbol{x}}_k)\|} \le 1 - \varepsilon_r$ or $\lambda_2(\tilde{\boldsymbol{x}}_k) \ge 0$ **do**
    if $\tau_k > 10^{-6}$ **then**
      $\tau_k \leftarrow \frac{1}{2}\tau_k$
      $\tilde{\boldsymbol{x}}_k \leftarrow \boldsymbol{x}_k + \tau_k s_k \boldsymbol{u}_k$
    else Terminate with $c = 1$.
    $j \leftarrow j + 1$
  if $j = 0$ **then** $\tau_{k+1} \leftarrow 1.1\tau_k$ **else** $\tau_{k+1} \leftarrow \tau_k$
  $\boldsymbol{x}_{k+1} \leftarrow \text{RSPROJ}(\tilde{p}, 1, \tilde{x}_k, \tau_k/4, 10^{-6})$.
  $\boldsymbol{X} \leftarrow (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_k, \boldsymbol{x}_{k+1})$

to detect if the iteration is leaving the set $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0}$. The first stopping criterion, where $\varepsilon > 0$, follows from the definition of the set $\mathcal{R}_{\tilde{p},\varepsilon,\boldsymbol{x}_0}$. The second stopping criterion with some $\varepsilon_e \in]0,1[$ tests whether the first and second eigenvalue of the Hessian become identical. When this is the case, the algorithm is near an endpoint of the ridge curve (cf. condition (6c) and Theorem 3.1). The third stopping criterion, where $\varepsilon_a \in]0,1[$, measures the cosine of the angle between the current ridge curve tangent $\boldsymbol{u}_k$ and the eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_k)$. When this measure is below the given threshold, the ridge curve tangent deviates significantly from the eigenvector $\boldsymbol{v}_1(\boldsymbol{x}_k)$, which indicates that the iteration is approaching a turning point. As pointed out in Subsection 3.2, the estimate of the underlying generating function given by the ridge curve cannot be considered reliable in this case.

The last stopping criteria test whether the iteration has crossed a mode of the density estimate. This can be detected by testing if the gradient changes direction along the ridge curve. Namely, before crossing a mode, the curve tangent is approximately parallel to the gradient and after crossing the mode approximately parallel to the negative gradient (cf. Theorem 3.3 and equation (10)). For $k > 0$, this yields the criteria

$$s_{k-1}\frac{\nabla\tilde{p}(\boldsymbol{x}_{k-1})^T\boldsymbol{u}_{k-1}}{\|\nabla\tilde{p}(\boldsymbol{x}_{k-1})\|} > 1-\varepsilon_c \quad \text{and} \quad s_k\frac{\nabla\tilde{p}(\boldsymbol{x}_k)^T\boldsymbol{u}_k}{\|\nabla\tilde{p}(\boldsymbol{x}_k)\|} < -(1-\varepsilon_c) \tag{28}$$

with some small $\varepsilon_c \in]0,1[$. When these criteria are met, the algorithm terminates and returns the mode $\boldsymbol{x}^*$ found by `RSPROJ` started from the midpoint of the current iterate $\boldsymbol{x}_k$ and the previous iterate $\boldsymbol{x}_{k-1}$. In analogy with equation (26), the algorithm also determines the sign parameter s^* at the mode $\boldsymbol{x}^*$ by comparing the directions of the previous tangent vector $s_{k-1}\boldsymbol{u}_{k-1}$ and the Hessian eigenvector $\boldsymbol{v}_1(\boldsymbol{x}^*)$ corresponding to the greatest Hessian eigenvalue at $\boldsymbol{x}^*$.

5 Numerical Tests

This section is devoted to demonstrating the applicability of the `RCURVES` algorithm (Algorithm 1) for extraction of curvilinear structures from noisy data. Illustrative examples on a representative selection of synthetic as well as two observational datasets from seismology and cosmology will be given. Some numerical results will also be provided to assess the computational performance of the algorithm.

5.1 Datasets

The `RCURVES` algorithm was run on kernel density estimates obtained from ten synthetic datasets and two observational datasets from seismology and cosmology. The synthetic datasets were generated from the model described in Section 2 with $\rho = 1$ (i.e. with no background clutter). The two- and three-dimensional test

datasets contain curvilinear structures having various shapes as well as intersections and closed loops. Most of the synthetic datasets are adapted from Kégl [3], and they were also used in [45]. The sample sizes N and the noise standard deviations σ for the synthetic datasets are listed in Table 1.

Dataset	N	σ	Dataset	N	σ
Arcs	2000	0.02	Circle	800	0.075
DistortedHalfCircle	800	0.02	DistortedSShape	800	0.015
HalfCircle	800	0.05	Jakob[1]	300	-
Ladder	3000	0.004	Spiral	1400	0.035
Spiral3d	1200	0.02	Zigzag	800	0.015

[1] The Jakob dataset does not have a known generating function or noise distribution.

Table 1: Sample sizes N and noise standard deviations σ used for generating the synthetic datasets.

5.1.1 The New Madrid Earthquake Dataset

Earthquake epicenters are typically clustered around seismic *faults* with a small number of "randomly" occurring earthquakes that can be considered as background clutter. Due to this fact, identification of faults from earthquake catalogs is a potential application for the proposed method (see e.g. [50] for an earlier approach to this problem). To illustrate this, a seismological dataset was obtained from the Center for Earthquake Research and Information (CERI) [2]. The dataset covers the New Madrid seismic region extending from Illinois to Arkansas. It contains the locations of observed earthquakes in this region from 1974 to 2013 with magnitude one and above, consisting of 6157 samples.

5.1.2 The Shapley Galaxy Dataset

In cosmology, galaxies typically form clusters and filamentary structures, and thus identification of such structures from galactic surveys is an important task. One of the most well-known example of this in our nearby universe is the *Shapley Supercluster* containing a rich variety of different galactic formations [21]. To illustrate the applicability of the proposed method to cosmological data, a dataset for the Shapley supercluster was obtained from the Center of Astrostatistics of Pennsylvania State University (CASt) [1]. The dataset consists of the angular sky coordinates and recession velocities of 4215 galaxies in the supercluster. As a preprocessing step, the original data was transformed into three-dimensional cartesian coordinates by utilizing the fact that recession velocities of galaxies are proportional to their radial distances [21].

5.2 Test Setup and Algorithm Parameters

All test runs were carried out on a machine with a 3.0GHz Core 2 Duo processor and 6Gb system memory running a 64-bit Linux operating system. The `RCURVES` algorithm and its subalgorithms were implemented in Fortran 95. Both cores of the test system were utilized for evaluating the kernel density estimate (7) because this operation can be trivially parallelized. In order to improve performance and numerical stability of the algorithm, the objective function was chosen as the logarithm of the density estimate (see [45] for additional justification of this choice). The algorithm was run with the experimentally chosen parameters

$$\varepsilon = -1, \quad \varepsilon_{\text{corr}} = 10^{-6}, \quad \varepsilon_a = 0.3, \quad \varepsilon_c = 0.25, \quad \varepsilon_e = 10^{-4} \quad \text{and} \quad \varepsilon_r = 0.01.$$

For each dataset, the marginal density (5) was estimated nonparametrically by using Gaussian kernels. The kernel bandwidth matrices $\boldsymbol{H}$ were computed by using the `Hpi` function implemented in the `ks` package [22] for the `R` software [4]. This function implements a multivariate generalization of the well-known univariate plug-in bandwidth selector by Wand and Jones [52]. The pilot bandwidth was chosen as the unconstrained bandwidth by Chacón and Duong [11], and the initial bandwidth was chosen as the normal scale bandwidth by Chacón et al. [12]. Since the `ks` package is capable of estimating derivatives of a density, the bandwidth matrix $\boldsymbol{H}$ was estimated for the first derivatives of the density p rather than the density itself. This choice is justified by the fact that modes and ridge curves are defined in terms of derivatives of the density (cf. Definition 2.1).

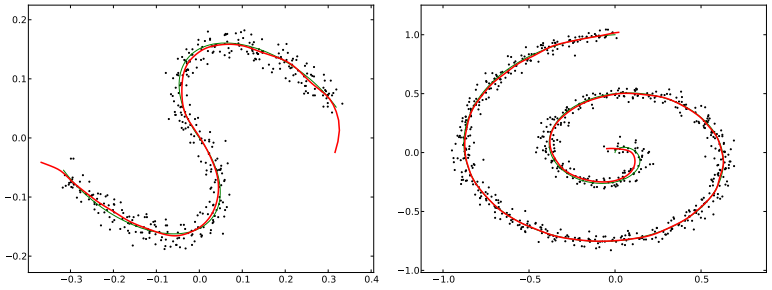
The numerical tests were carried out by applying the `RCURVES` algorithm in the scaled coordinate system as described in Subsection 4.1. This was done by applying the algorithm to the density estimate $\tilde{p}$ corresponding to the scaled samples $\tilde{\boldsymbol{y}}_i = \boldsymbol{L}^{-1}\boldsymbol{y}_i$, where the matrix $\boldsymbol{L}$ was obtained from the Cholesky factorization $\boldsymbol{H} = \boldsymbol{L}\boldsymbol{L}^T$. For each ridge curve point $\tilde{\boldsymbol{x}}$ in the scaled coordinate system, the corresponding point $\boldsymbol{x}$ in the original coordinate system was then obtained by applying the inverse transformation $\boldsymbol{x} = \boldsymbol{L}\tilde{\boldsymbol{x}}$.

5.3 Illustrative Examples

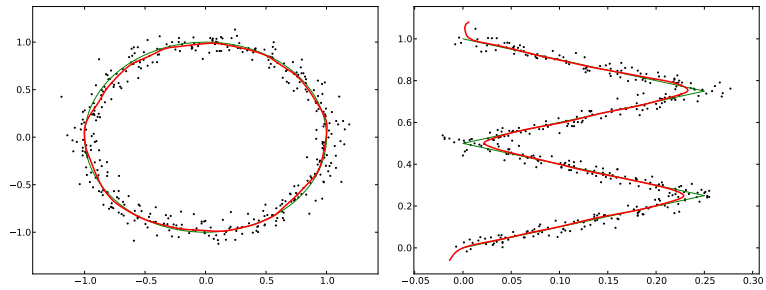
Some of the synthetic datasets having curvilinear structures with various shapes, closed loops and intersections are shown in Figures 6–8 (for the remaining datasets, see [45]). For all datasets, the ridge curves extracted by the `RCURVES` algorithm from kernel density estimates with bandwidths obtained from the `Hpi` bandwidth chooser seemed to generally give very good estimates of the underlying generating functions.

The deviations between the kernel density ridge curves and the generating functions shown in Figures 6–7 seem to be consistent with the error estimate given in Appendix A. The deviation is indeed proportional to the ratio between the noise variance and the curvature radius of the generating function. Furthermore, this

deviation is towards the center of curvature. The deviation is generally small, but as seen from Figures 6b and 6d, it can grow large when the generating function has sharp turns.



(a) DistortedSShape (b) Spiral



(c) Circle (d) Zigzag

Figure 6: Kernel density ridge curves (red) and generating functions (green) of the two-dimensional datasets having a single generating function.

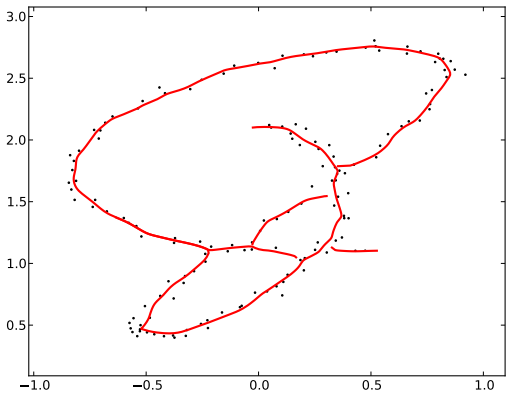


Figure 7: Kernel density ridge curves of the Jakob dataset.

The Jakob dataset plotted in Figure 7 is a good example showing that the `RCURVES` algorithm handles intersections properly and terminates ridge curve tracing when the followed ridge curve ends (cf. Figure 4). The only exception is

the point located near $(-0.25, 1.2)$, where the termination point of the lower ridge curve is very close to the upper one. However, it was verified that the algorithm terminated tracing of the lower ridge curve at the mode located near an endpoint of the lower ridge curve (cf. Remark 4.2).

However, when the generating functions have intersections, their complete parametrization cannot be recovered by the algorithm. This is a fundamental limitation of the ridge-curve based approach. Namely, by Theorem 3.1 we have the generic property that two connected ridge curve components cannot intersect each other. As a result, there can be exactly one connected ridge curve component passing through an intersection point, and the other components passing through such point are split into two parts.

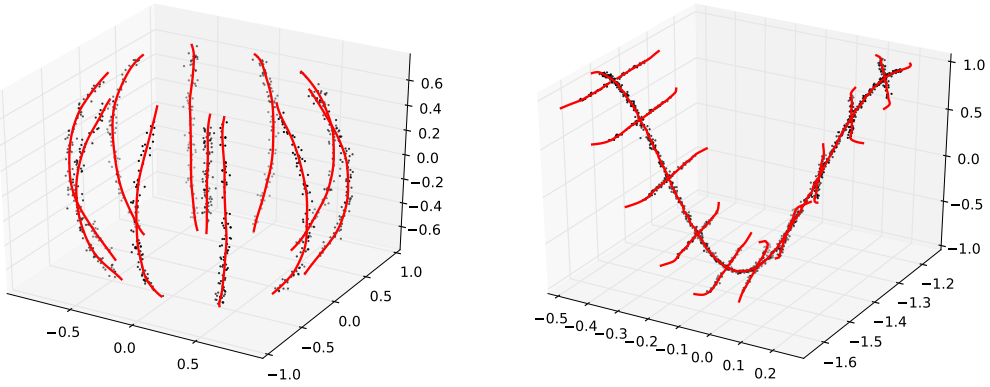
Another potential, but not serious shortcoming of the ridge curve-based approach is that ridge curves have a rather poor ability to extrapolate the estimates of the generating functions beyond the data. This can be seen from the arbitrary shape of the extracted curves beyond the endpoints of the generating functions especially in Figures 6d and 8c. Here the locality of the ridge curve definition, that gives the advantage of having an additional degree of freedom compared to most earlier principal curve approaches, seems to be a disadvantage. To the knowledge of the author, there does not seem to be a straightforward way to overcome this inherent limitation.

Finally we present the results for the earthquake and galaxy datasets. The New Madrid dataset and the extracted faults obtained by the `RCURVES` algorithm are plotted in Figure 9. The algorithm does an excellent job here, revealing all the visually distinguishable structures that could be interpreted as faults. Moreover, the result is not affected by the background clutter present in the data. This is due to the local nature of Gaussian kernels and the fact that samples with no significant concentration are automatically rejected by filtering out ridge curves lying on low-density areas.

The Shapley dataset and the filamentary structures extracted by the `RCURVES` algorithm in the transformed three-dimensional coordinates are plotted in Figure 10 for three different recession velocity ranges. In these examples, clustering of the points around filaments is not as obvious as in the previous examples, and there is some room for interpretation. Nevertheless, the algorithm seems to do a fairly good job in extracting the highly nonlinear shapes of the most prominent galaxy clusters. Finally, a subset of the Shapley dataset in the original sky coordinates is plotted in Figure 11. We can again observe that the algorithm identifies all the visually distinguishable galaxy concentrations despite the large amount of background clutter (i.e. galaxies not belonging to any cluster or filament).

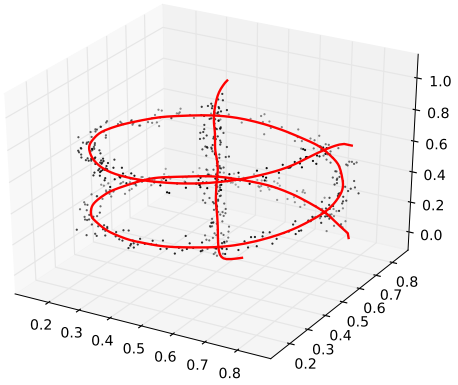
5.4 Performance Evaluation

To assess its computational performance, the `RCURVES` algorithm was compared to a variant where the Newton-based mode-finding and corrector methods were



(a) Arcs

(b) Ladder



(c) Spiral3d

Figure 8: The three-dimensional datasets and their kernel density ridge curves.

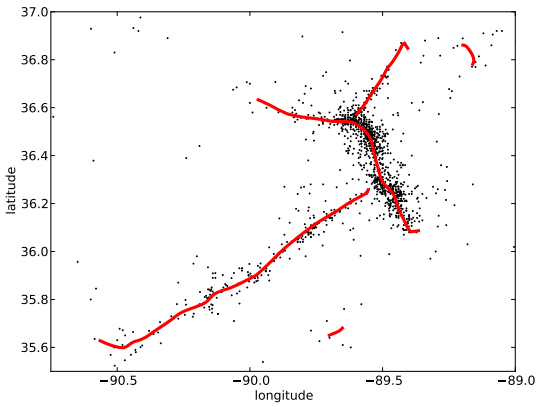
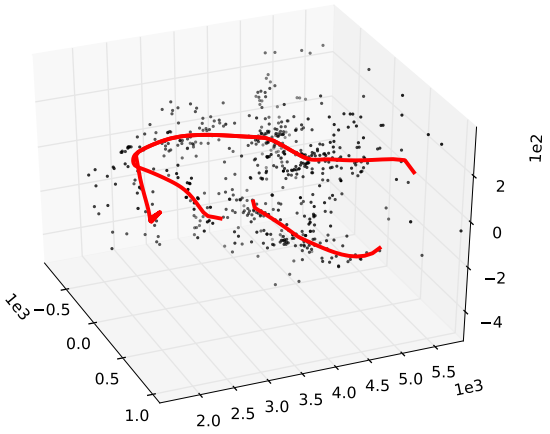
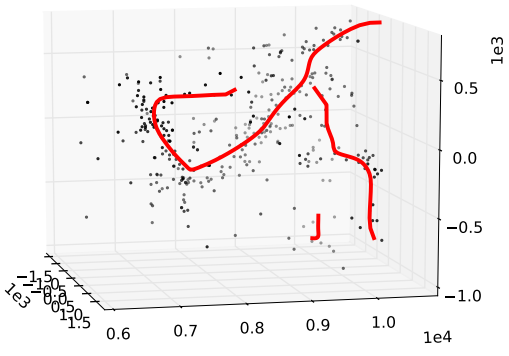


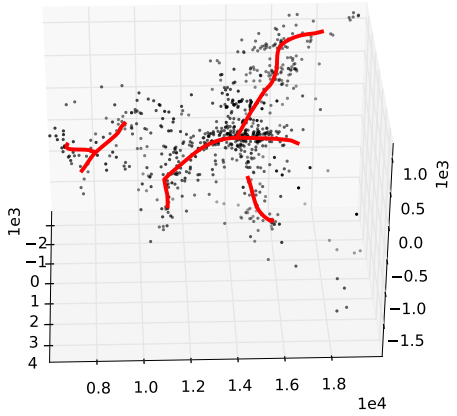
Figure 9: Faults extracted from the New Madrid dataset.



(a) velocity range $1500 - 6000$ km/s



(b) velocity range $6000 - 10500$ km/s



(c) velocity range $6000 - 20000$ km/s

Figure 10: Filaments extracted from the Shapley dataset in transformed coordinates.

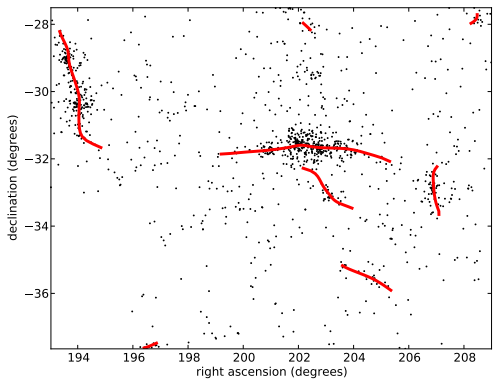


Figure 11: Filaments extracted from the Shapley dataset in sky coordinates, velocity range $9000 - 18000$ km/s.

replaced with the mean-shift method [14, 15] and its subspace-constrained variant [43], respectively. Apart from its more sophisticated stopping criteria, step size adaptation and the use of third derivatives of the objective function, this variant of the RCURVES algorithm is comparable to the earlier ridge-based principal curve methods proposed in [6–8]. The comparison with the mean-shift -based variant is also motivated by the fact that the mean-shift method is the most widely used approach to finding modes of Gaussian kernel density estimates, and more recently, finding ridges of such functions as in [43].

In the following, we refer to the Newton-based algorithm as RCURVES-NEWTON and the mean-shift variant as RCURVES-MS. The number of objective function evaluations, the more expensive third derivative evaluations and the used wall clock times (in seconds) for running these algorithms on kernel density estimates obtained from the synthetic datasets are listed in Table 2. Here one function evaluation means a combined evaluation of the function value, gradient and Hessian of the kernel density estimate. For the RCURVES-MS algorithm, computation of the mean-shift step is also combined with the function evaluation.

	RCURVES-MS			RCURVES-NEWTON		
Dataset	$\#f$	$\#\nabla^3$	time	$\#f$	$\#\nabla^3$	time
Arcs	208 929	618	12.469	18 272	618	2.251
Circle	107 059	263	2.388	9 212	263	0.300
DistortedHalfCircle	164 633	151	3.626	9 636	151	0.303
DistortedSShape	120 768	213	2.684	8 461	213	0.275
HalfCircle	175 874	179	3.888	8 999	179	0.301
Jakob	30 679	677	0.385	9 640	635	0.201
Ladder	328 057	1 999	29.596	34 218	2 013	6.924
Spiral	282 647	491	10.643	15 868	485	0.870
Spiral3d	231 018	534	9.653	11 878	534	1.070
Zigzag	108 312	216	2.773	9 118	208	0.302

Table 2: Function evaluations, third derivative evaluations and wall clock times used by the RCURVES-MS and RCURVES-NEWTON algorithms for kernel density estimates obtained from the synthetic test datasets.

It is evident from these results that the Newton-based RCURVES-NEWTON algorithm has superior performance. This suggests that the proposed algorithm is also superior to the mean-shift -based algorithms developed in [6–8] that are comparable to the RCURVES-MS algorithm. A detailed inspection of the computation times revealed that the rather dramatic performance difference is mostly explained by the slow convergence of the mean-shift method during the mode-finding step. This is due to the fact that the mean-shift method tends to have very slow convergence when applied to finding modes of a density estimate having highly elongated peaks (see e.g. [45]). As illustrated in Figure 3, this is typically the case in applications considered in this paper.

Another important observation from the above results is that the number of function evaluations strongly correlates with the used computation time. This indicates that the objective function evaluation dominates the total computational cost. It is indeed expensive because evaluation of a Gaussian kernel density estimate and its i-th derivative take $\mathcal{O}(Nd)$ and $\mathcal{O}(Nd^i)$ operations, respectively. Especially for the expensive mode-finding step whose objective function evaluations require $\mathcal{O}(N^2d^2)$ operations, the computational cost could be reduced to $\mathcal{O}(N)$ by using the *fast Gauss transform* [31] or related methods when the data dimension is low (say $d \leq 4$). This optimization is not implemented in the preliminary version of the `RCURVES` algorithm used in these tests, and it is left as future work.

6 Conclusions and Discussion

Extraction of curvilinear structures from noisy data is an essential task in many application areas such as data analysis, pattern recognition and machine vision. This paper contributes to the field by refining and extending the earlier approaches of [6–8] and [43] based on locally defined principal curves. In these papers, ridge curves of the probability density estimated from the data are used to estimate principal curves passing through the data. Building on this idea, this paper gives a more detailed treatment to the underlying data-generating process and presents several algorithmic enhancements to the earlier methods.

A probabilistic model describing a point set containing curvilinear structures mixed with background clutter was considered in this paper. In the model, such structures are concentrated around smooth generating functions with normally distributed noise, and the background clutter is assumed to be uniformly distributed. It was shown by examples that when the data is generated from this model, ridge curves of the marginal density induced by the model give good estimates of the underlying generating functions. The main observation is that the model bias is proportional to the ratio between the noise variance and curvature radii of the generating curves. In order to make the approach feasible for a computational implementation, estimation of the marginal density by using Gaussian kernels was considered.

The main contribution of this paper is the development of a robust and efficient method for tracing the ridge curve set of a Gaussian kernel density estimate. A ridge curve of such a density estimate was formulated as the solution to a differential equation. For tracing the solution curve set of this differential equation, an efficient and robust predictor-corrector algorithm was developed. The algorithm utilizes an efficient and provably convergent trust region Newton method developed in [45]. As a ridge curve is a generalization of a mode (maximum) of a density estimate, the Newton-based method is conveniently used both during the initial mode-finding step and as a corrector. Being rigorously based on the

mathematical theory of ridge curves, the algorithm also handles reliably different kinds of endpoints and singularities along ridge curves. Such points typically occur when the data contains multiple curvilinear structures.

As the proposed method is based on nonparametric density estimation, it is applicable to a wide range of real-world tasks where no prior information on the data-generating process is available. This was demonstrated by applying the method to observational datasets from seismology and cosmology. Coupled with a robust method for choosing the kernel density bandwidth, the proposed method provides a complete framework for extraction of curvilinear structures from noisy data.

Furthermore, due to major performance improvements over to the earlier ridge-based principal curve methods, the proposed method is likely to be computationally feasible for real-time applications such as process monitoring, traffic modeling and machine vision. A more extensive evaluation would, however, be needed to completely determine the practical applicability of the method and its competitiveness with other approaches based on different definitions of a principal curve.

Finally, it is worth noting that ridge- and valley- following methods have been proposed for global optimization (see e.g. [39]). A related approach to avoid getting trapped into local minima has been proposed for some optimization problems appearing in machine vision in [48]. Since the predictor-corrector method developed in this paper is applicable to extraction of ridges from any function satisfying the conditions of Theorem 3.1 (not necessarily a C^∞-function), it could be used in the aforementioned applications as well.

Acknowledgements. The author was financially supported by the TUCS Graduate Programme. He would also like to thank Prof. Marko Mäkelä and Doc. Napsu Karmitsa for their valuable comments.

References

[1] CASt: Shapley galaxy dataset. http://astrostatistics.psu.edu/datasets/Shapley_galaxy.html. visited on 28/7/2013.

[2] New Madrid Earthquake Catalog. http://www.ceri.memphis.edu/seismic/catalogs/cat_nm.html. visited on 28/7/2013.

[3] Principal Curves. http://www.iro.umontreal.ca/~kegl/research/pcurves. visited on 23/5/2013.

[4] The R Project for Statistical Computing. http://www.r-project.org. visited on 23/5/2013.

[5] 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.

[6] E. Baş. *Extracting structural information on manifolds from high dimensional data and connectivity analysis of curvilinear structures in 3D biomedical images*. PhD thesis, Northeastern University, Boston, MA, 2011.

[7] E. Baş and D. Erdogmus. Principal curves as skeletons of tubular objects. *Neuroinformatics*, 9(2-3):181–191, 2011.

[8] E. Baş, D. Erdogmus, R. W. 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.

[9] J. M. Bofill, W. Quapp, and M. Caballero. The variational structure of gradient extremals. *Journal of Chemical Theory and Computation*, 8(3):927–935, 2012.

[10] K. Bondensgård and F. Jensen. Gradient extremal bifurcation and turning points: An application to the H2CO potential energy surface. *Journal of Chemical Physics*, 104(20):8025–8031, 1996.

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

[12] J. E. Chacón, T. Duong, and M. P. Wand. Asymptotics for general multivariate kernel density derivative estimators. *Statistica Sinica*, 21:807–840, 2011.

[13] 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), 2004.

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

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

[16] J. Damon. Generic structure of two-dimensional images under Gaussian blurring. *SIAM Journal on Applied Mathematics*, 59(1):97–138, 1998.

[17] A. Dasgupta and A. E. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302, 1998.

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

[19] P. Delicado and M. Huerta. Principal curves of oriented points: theoretical and computational improvements. *Computational Statistics*, 18(2):293–315, 2003.

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

[21] M. J. Drinkwater, Q. A. Parker, D. Proust, E. Slezak, and H. Quintana. The large scale distribution of galaxies in the Shapley supercluster. *Publications of the Astronomical Society of Australia*, 21(1):89–96, 2004.

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

[23] T. Duong and M. L. Hazelton. Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32(3):485–506, 2005.

[24] D. Eberly. *Ridges in Image and Data Analysis*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.

[25] J. Einbeck and J. Dwyer. Using principal curves to analyse traffic patterns on freeways. *Transportmetrica*, 7(3):229–246, 2011.

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

[27] N. Flasque, M. Desvignes, J.-M. Constans, and M. Revenu. Acquisition, segmentation and tracking of the cerebral vascular tree on 3D magnetic resonance angiography images. *Medical Image Analysis*, 5(3):173–183, 2001.

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

[29] C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. The geometry of nonparametric filament estimation. *Journal of the American Statistical Association*, 107(498):788–799, 2012.

[30] C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Nonparametric ridge estimation, 2012. arXiv:1212.5156.

[31] L. Greengard and J. Strain. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

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

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

[34] M. Hirsch and W. Quapp. The reaction pathway of a potential energy surface as curve with induced tangent. *Chemical Physics Letters*, 395(1-3):150–156, 2004.

[35] D. K. Hoffman, R. S. Nord, and K. Ruedenberg. Gradient extremals. *Theoretica Chimica Acta*, 69(4):265–279, 1986.

[36] A. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, 2000.

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

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

[39] A. Lucia, P. A. DiMaggio, and P. Depa. A geometric terrain methodology for global optimization. *Journal of Global Optimization*, 29(3):297–314, 2004.

[40] J. Miller. *Relative Critical Sets in R^n and Applications to Image Analysis*. PhD thesis, University of North Carolina, 1998.

[41] D. Novikov, S. Colombi, and O. Doré. Skeleton as a probe of the cosmic web: the two-dimensional case. *Monthly Notices of the Royal Astronomical Society*, 366(4):1201–1216, 2006.

[42] J. M. Ortega. *Numerical Analysis: A Second Course*. SIAM, Philadelphia, 1990.

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

[44] S. Pulkkinen, M. M. Mäkelä, and N. Karmitsa. A continuation approach to mode-finding of multivariate Gaussian mixtures and kernel density estimates. *Journal of Global Optimization*, 56(2):459–487, 2013.

[45] S. Pulkkinen, M. M. Mäkelä, and N. Karmitsa. A generative model and a generalized trust region Newton method for noise reduction. *Computational Optimization and Applications*, 2013. to appear, doi: 10.1007/s10589-013-9581-4.

[46] D. W. Scott. *Multivariate Density Estimation: Theory Practice and Visualization*. John Wiley and Sons, New York, 1992.

[47] F. Y. Shih and A. J. Kowalski. Automatic extraction of filaments in hα solar images. *Solar Physics*, 218(1-2):99–122, 2003.

[48] C. Sminchisescu and B. Triggs. Building roadmaps of minima and transitions in visual models. *International Journal of Computer Vision*, 61(1):81–101, 2005.

[49] T. Sousbie, C. Pichon, S. Colombi, D. Novikov, and D. Pogosyan. The 3D skeleton: tracing the filamentary structure of the universe. *Monthly Notices of the Royal Astronomical Society*, 383(4):1655–1670, 2008.

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

[51] R. Tibshirani. Principal curves revisited. *Statistics and Computation*, 2(4):183–190, 1992.

[52] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, London, 1995.

[53] V. Zharkova, S. Ipson, A. Benkhalil, and S. Zharkov. Feature recognition in solar images. *Artificial Intelligence Review*, 23(3):209–266, 2005.

A Model Bias: a Simple Example

To investigate the bias of the model developed in Section 2, in this appendix we consider a simple example with a generating curve on a plane. In this example, a single generating function $\boldsymbol{f} : [0, 2\pi] \to \mathbb{R}^2$ parametrizes the unit circle

$$\boldsymbol{f}(\theta) = (\cos\theta, \sin\theta).$$

With the assumptions of Section 2, assuming that no background clutter is present and that the random variable Θ is uniformly distributed in the interval $[0, 2\pi]$, we obtain the marginal density

$$p(\boldsymbol{x}) = \frac{1}{4\pi^2\sigma^2} \int_0^{2\pi} G_\sigma(\boldsymbol{x}, \theta) d\theta, \tag{29}$$

where we have introduced the short-hand notation

$$G_\sigma(\boldsymbol{x}; \theta) = \exp\left(-\frac{(x_1 - \cos\theta)^2 + (x_2 - \sin\theta)^2}{2\sigma^2}\right).$$

Due to symmetry, it suffices to consider the model bias at a point lying on the x-axis. That is, we let

$$\boldsymbol{x} = (x_1, x_2), \quad 0 < x_1 < 1 \quad \text{and} \quad x_2 = 0.$$

The first component of the gradient $\nabla p(\boldsymbol{x})$ is given by

$$\frac{\partial p}{\partial x_1}(\boldsymbol{x}) = -\frac{1}{4\pi^2\sigma^4} \int_0^{2\pi} (x_1 - \cos\theta) G_\sigma(\boldsymbol{x}, \theta) d\theta. \tag{30}$$

The second component of the gradient is zero since $x_2 = 0$ and due to the antisymmetry of the sine function we have

$$\begin{aligned} \frac{\partial p}{\partial x_2}(\boldsymbol{x}) &= -\frac{1}{4\pi^2\sigma^4} \int_0^{2\pi} (x_2 - \sin\theta) G_\sigma(\boldsymbol{x}, \theta) d\theta \\ &= \frac{1}{4\pi^2\sigma^4} \left[\int_0^{\pi} \sin\theta \cdot G_\sigma(\boldsymbol{x}, \theta) d\theta - \int_0^{\pi} \sin\theta \cdot G_\sigma(\boldsymbol{x}, \theta) d\theta \right] = 0. \end{aligned} \tag{31}$$

A straightforward calculation shows that the first diagonal component of the Hessian $\nabla^2 p(\boldsymbol{x})$ is given by

$$[\nabla^2 p(\boldsymbol{x})]_{1,1} = \frac{1}{4\pi^2\sigma^4} \int_0^{2\pi} \left[\frac{(x_1 - \cos\theta)^2}{\sigma^2} - 1 \right] G_\sigma(\boldsymbol{x}, \theta) d\theta \tag{32}$$

and the other components satisfy

$$[\nabla^2 p(\boldsymbol{x})]_{1,2} = [\nabla^2 p(\boldsymbol{x})]_{2,1} = [\nabla^2 p(\boldsymbol{x})]_{2,2} = 0. \tag{33}$$

It can be shown that the Hessian element $[\nabla^2 p(x_1, 0)]_{1,1}$ has exactly one root in the interval $[0, 1]$ and that $[\nabla^2 p(x_1, 0)]_{1,1} < 0$ when $\sigma \in]0, \frac{\sqrt{2}}{2}[$ and $x_1 \in [x_1^*, 1]$, where x_1^* denotes such root (we omit the proof). In view of equations (32) and (33), this implies that the normalized eigenvectors of the Hessian $\nabla^2 p(\boldsymbol{x})$ are

$$\boldsymbol{v}_1(\boldsymbol{x}) = (0, 1) \quad \text{and} \quad \boldsymbol{v}_2(\boldsymbol{x}) = (1, 0)$$

for all $\boldsymbol{x} \in \mathbb{R}^2$ such that $x_1 \in]x_1^*, 1]$ and $x_2 = 0$. By equations (32) and (33), the first eigenvector $\boldsymbol{v}_1(\boldsymbol{x})$ corresponds to the eigenvalue $\lambda_1(\boldsymbol{x}) = 0$ and the second eigenvector $\boldsymbol{v}_2(\boldsymbol{x})$ corresponds to the eigenvalue $\lambda_2(\boldsymbol{x}) < 0$ when $x_1 \in]x_1^*, 1]$.

Combining the above observations with equations (30) and (31) and conditions (6a)–(6c), we make the following observation for any ridge point of p on the positive x-axis. Namely, the x-coordinate of such point is a root of the function $[\nabla p(\boldsymbol{x})]_1$ with $\boldsymbol{x} = (x_1, 0)$ such that $x_1 \in]x_1^*, 1]$. Finding a ridge point of p then requires finding the root x_1^*, for which we cannot obtain a closed-form expression. However, by applying numerical integration and root-finding, the root x_1^* for a given value of σ can be approximately computed.

The dependence of the model bias relative to the noise standard deviation σ on σ is plotted in Figures 12 and 13. The results show that the ridge curve gives quite an accurate estimate of the actual generating function. For instance, in the interval $[0, 0.35]$, the distance between the ridge point and the generating curve grows linearly, and with $\sigma = 0.35$ it is only 0.2σ. As seen from Figure 13, such a value of σ corresponds to a rather large amount of noise.

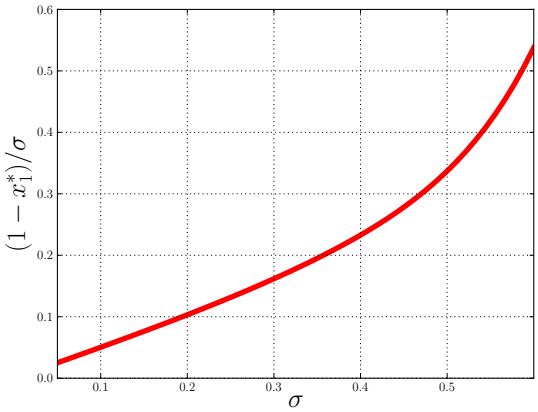


Figure 12: Distance between the generating curve $\boldsymbol{f}(\theta)$ and the x-coordinate x_1^* of a ridge point of the marginal density (29) relative to noise standard deviation σ in the interval $[0, 0.6]$ as a function of σ.

The conclusion is that the bias is generally small and proportional to the ratio between the noise deviation σ and the curvature radius of the generating function. Furthermore, the bias occurs towards the curvature center. As observed in Section 5, this property seems to apply to more complex generating functions and also when the marginal density is estimated by using Gaussian kernels.

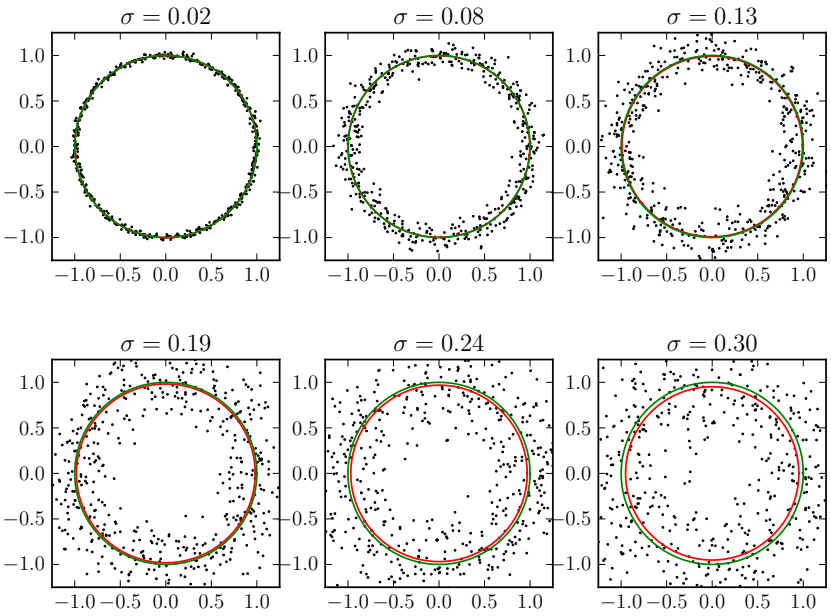


Figure 13: Circular data distributions with different values of σ, generating functions (green) and ridge curves of the marginal density (red).

B Proofs of Technical Results

Proof of Theorem 3.2. We seek for a solution of the form

$$\boldsymbol{u}(\alpha, \boldsymbol{v}) = \alpha \frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} + \boldsymbol{U}(\boldsymbol{x}_0)\boldsymbol{v}, \quad \alpha \in \mathbb{R}, \quad \boldsymbol{v} \in \mathbb{R}^{d-1} \tag{34}$$

to equation (19). First, we note the identity

$$\boldsymbol{A}(\boldsymbol{x}_0)\nabla p(\boldsymbol{x}_0) = [\nabla^3 p(\boldsymbol{x}_0)\nabla p(\boldsymbol{x}_0)]\nabla p(\boldsymbol{x}_0) \tag{35}$$

following from equations (10) and (15). With this identity, substituting equations (16) and (34) into (19) and premultiplying the resulting equation by $\boldsymbol{U}(\boldsymbol{x}_0)^T$ shows that a vector $\boldsymbol{u}(\alpha, \boldsymbol{v})$ of the form (34) is a solution to equation (19) if and only if

$$-\alpha \boldsymbol{U}(\boldsymbol{x}_0)^T [\nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x}_0)] \frac{\nabla p(\boldsymbol{x}_0)}{\|\nabla p(\boldsymbol{x}_0)\|} = \boldsymbol{U}(\boldsymbol{x}_0)^T \boldsymbol{A}(\boldsymbol{x}_0)\boldsymbol{U}(\boldsymbol{x}_0)\boldsymbol{v}. \tag{36}$$

Since the matrix $\boldsymbol{C}(\boldsymbol{x}_0) = \boldsymbol{U}(\boldsymbol{x}_0)^T \boldsymbol{A}(\boldsymbol{x}_0)\boldsymbol{U}(\boldsymbol{x}_0)$ was assumed to be nonsingular, the vector $\boldsymbol{v}$ can be solved from equation (36) for any $\alpha \in \mathbb{R}$. Substituting such a vector, that we denote as $\boldsymbol{v}^*(\alpha)$, into equation (34) and choosing $\alpha = 1$ yields the vector $\boldsymbol{u}^*$ defined by equation (17) as a solution to equation (19). Furthermore, all solutions to (19) are of the form $\boldsymbol{u}(\alpha, \boldsymbol{v}^*(\alpha)) = \alpha \boldsymbol{u}^*$. Namely, the vectors $\boldsymbol{u}(\alpha, \boldsymbol{v})$ span $\mathbb{R}^d$ by the definition of the matrix $\boldsymbol{U}(\boldsymbol{x}_0)$ (see equation (16)) and the vector $\boldsymbol{v}^*(\alpha)$ yielding a solution to (36) is uniquely determined for any scalar α. □

Proof of Theorem 3.4. As in the proof of Theorem 3.2, we seek for a solution of the form (34) to equation (19) and utilize equation (36) for this purpose. With the definitions of Theorem 3.4, equation (36) can be equivalently written as

$$-\alpha \boldsymbol{b}(\boldsymbol{x}_0) = \boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^T\boldsymbol{v},$$

and premultiplying this equation by $\boldsymbol{W}^T$ yields

$$-\alpha \boldsymbol{W}^T\boldsymbol{b}(\boldsymbol{x}_0) = \boldsymbol{D}\boldsymbol{W}^T\boldsymbol{v}. \tag{37}$$

The assumption that the eigenvalues λ_i of the matrix $\boldsymbol{C}(\boldsymbol{x}_0)$ satisfy the condition $\lambda_i = 0$ for $i \in I$ implies that $d_{ii} = 0$ for $i \in I$. Consequently, when $\boldsymbol{w}_i^T\boldsymbol{b}(\boldsymbol{x}_0) \neq 0$ for some $i \in I$, equation (37) has a solution with respect to $\boldsymbol{v}$ only when $\alpha = 0$. Furthermore, we note that in this case all solutions to equation (37) are of the form $\boldsymbol{v}(\boldsymbol{\beta}) = \sum_{i\in I}\beta_i\boldsymbol{w}_i$ with $\boldsymbol{\beta} \in \mathbb{R}^{|I|}$. Then substituting such a vector $\boldsymbol{v}(\boldsymbol{\beta})$ into equation (34) and letting $\alpha = 0$ gives the vector $\boldsymbol{u}(\boldsymbol{\beta}) = \boldsymbol{U}(\boldsymbol{x}_0)\sum_{i\in I}\beta_i\boldsymbol{w}_i$ as a solution to (19).

On the other hand, when $\boldsymbol{w}_i^T\boldsymbol{b} = 0$ for all $i \in I$, equation (37) has two solution spaces. Since $d_{ii} = 0$ for all $i \in I$, the first one corresponds to the choice $\alpha = 0$, in which case the possible solution vectors $\boldsymbol{v}$ are of the form $\boldsymbol{v}(\boldsymbol{\beta}) = \sum_{i\in I}\beta_i\boldsymbol{w}_i$ with $\boldsymbol{\beta} \in \mathbb{R}^{|I|}$. Substituting such α and $\boldsymbol{v}(\boldsymbol{\beta})$ into equation (34) yields the vector $\boldsymbol{u}(\boldsymbol{\beta})$ (or any its scalar multiple) defined by equation (21) as a solution to (19).

The second solution space of (37) corresponding to the choice $\alpha \neq 0$ is spanned by the vector

$$\boldsymbol{v}(\alpha) = -\alpha\sum_{\substack{i=1\\ i\notin I}}^{d-1}\frac{\boldsymbol{w}_i^T\boldsymbol{b}(\boldsymbol{x}_0)}{d_{ii}}\boldsymbol{w}_i.$$

This can be seen by substituting the vector

$$\boldsymbol{v}(\alpha) = -\alpha\sum_{\substack{i=1\\ i\notin I}}^{d-1}\beta_i\boldsymbol{w}_i$$

into (37) and solving for the coefficients β_i. Substituting the vector $\boldsymbol{v}(\alpha)$ into (34) then shows that any scalar multiple of the vector $\tilde{\boldsymbol{u}}$ defined by equation (22) is a solution to (19). Clearly, this is also the case for any linear combination of the vectors defined by equations (21) and (22). □

Finally we prove Theorem 3.3. For this we need two auxiliary lemmata. The first one follows from the continuity of eigenvalues of a matrix with respect to its elements (see e.g. [42], Theorem 3.1.2) and the continuity of the Hessian $\nabla^2 p$ when p is a C^2-function.

Lemma B.1. *If $p \in C^2(\mathbb{R}^d, \mathbb{R})$, then there exist continuous functions $\{\lambda_i\}_{i=1}^d : \mathbb{R}^d \to \mathbb{R}$ representing the eigenvalues of the Hessian $\nabla^2 p$.*

Lemma B.2. *Let $p \in C^3(\mathbb{R}^d, \mathbb{R})$ and let $\boldsymbol{x}_0 \in \mathcal{R}_p$ be an isolated critical point of p. If we define*

$$\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{I} - \frac{\nabla p(\boldsymbol{x})\nabla p(\boldsymbol{x})^T}{\|\nabla p(\boldsymbol{x})\|^2} = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})^T,$$

then there exists a neighbourhood $\mathcal{N}$ of $\boldsymbol{x}_0$ such that the matrix $\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})^T \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{U}(\boldsymbol{x})$ is nonsingular for all $\boldsymbol{x} \in (\mathcal{R}_p \cap \mathcal{N}) \setminus \{\boldsymbol{x}_0\}$. Furthermore, for all $\boldsymbol{x} \in (\mathcal{R}_p \cap \mathcal{N}) \setminus \{\boldsymbol{x}_0\}$ the matrix $\boldsymbol{C}(\boldsymbol{x})$ can be written as

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})^T [\nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x})] \boldsymbol{U}(\boldsymbol{x}) + \sum_{i=2}^{d} [\lambda_i(\boldsymbol{x})^2 - \lambda_1(\boldsymbol{x}) \lambda_i(\boldsymbol{x})] \boldsymbol{v}_i(\boldsymbol{x}) \boldsymbol{v}_i(\boldsymbol{x})^T. \quad (38)$$

Proof. As a symmetric matrix, the Hessian $\nabla^2 p(\boldsymbol{x})$ admits the eigendecomposition $\nabla^2 p(\boldsymbol{x}) = \boldsymbol{V}(\boldsymbol{x}) \boldsymbol{\Lambda}(\boldsymbol{x}) \boldsymbol{V}(\boldsymbol{x})^T$, where

$$\begin{aligned} \boldsymbol{V}(\boldsymbol{x}) &= [\boldsymbol{v}_1(\boldsymbol{x}), \boldsymbol{v}_2(\boldsymbol{x}), \ldots, \boldsymbol{v}_d(\boldsymbol{x})] \in \mathbb{R}^{d \times d}, \\ \boldsymbol{\Lambda}(\boldsymbol{x}) &= \operatorname{diag}[\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \ldots, \lambda_d(\boldsymbol{x})] \in \mathbb{R}^{d \times d} \end{aligned}$$

with normalized eigenvectors $\{\boldsymbol{v}_i(\boldsymbol{x})\}_{i=1}^d$ corresponding to the eigenvalues $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$ of $\nabla^2 p(\boldsymbol{x})$. By using this decomposition and equations (10) and (15) we obtain that

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{x}) &= \nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x}) + [\nabla^2 p(\boldsymbol{x})]^2 - \frac{\nabla p(\boldsymbol{x})^T \nabla^2 p(\boldsymbol{x}) \nabla p(\boldsymbol{x})}{\|\nabla p(\boldsymbol{x})\|^2} \nabla^2 p(\boldsymbol{x}) \\ &= \nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x}) + \boldsymbol{V}(\boldsymbol{x}) \boldsymbol{\Lambda}(\boldsymbol{x})^2 \boldsymbol{V}(\boldsymbol{x})^T - \lambda_1(\boldsymbol{x}) \boldsymbol{V}(\boldsymbol{x}) \boldsymbol{\Lambda}(\boldsymbol{x}) \boldsymbol{V}(\boldsymbol{x})^T \\ &= \nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x}) + \boldsymbol{V}(\boldsymbol{x}) [\boldsymbol{\Lambda}(\boldsymbol{x})^2 - \lambda_1(\boldsymbol{x}) \boldsymbol{\Lambda}(\boldsymbol{x})] \boldsymbol{V}(\boldsymbol{x})^T \\ &= \nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x}) + \sum_{i=1}^{d} [\lambda_i(\boldsymbol{x})^2 - \lambda_1(\boldsymbol{x}) \lambda_i(\boldsymbol{x})] \boldsymbol{v}_i(\boldsymbol{x}) \boldsymbol{v}_i(\boldsymbol{x})^T \end{aligned} \quad (39)$$

for all $\boldsymbol{x} \in \mathcal{R}_p$ such that $\nabla p(\boldsymbol{x}) \neq \boldsymbol{0}$.

By condition (6a), the gradient $\nabla p(\boldsymbol{x})$ is orthogonal to the eigenvectors $\{\boldsymbol{v}_i(\boldsymbol{x})\}_{i=2}^d$ of $\nabla^2 p(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{R}_p$. We can also observe that these vectors span the $d-1$-dimensional eigenspace of the matrix $\boldsymbol{P}(\boldsymbol{x})$ corresponding to the eigenvalue one. Thus, we may assume without loss of generality for $\boldsymbol{x} \in \mathcal{R}_p$ that $\boldsymbol{u}_i(\boldsymbol{x}) = \boldsymbol{v}_{i+1}(\boldsymbol{x})$ for $i = 1, 2, \ldots, d-1$, where $\boldsymbol{u}_i(\boldsymbol{x})$ denotes the i-th column of the orthogonal matrix $\boldsymbol{U}(\boldsymbol{x})$. With this assumption and equation (39) we then have

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})^T \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{U}(\boldsymbol{x}) =& \boldsymbol{U}(\boldsymbol{x})^T [\nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x})] \boldsymbol{U}(\boldsymbol{x}) \\ &+ \sum_{i=2}^{d} [\lambda_i(\boldsymbol{x})^2 - \lambda_1(\boldsymbol{x}) \lambda_i(\boldsymbol{x})] \boldsymbol{v}_i(\boldsymbol{x}) \boldsymbol{v}_i(\boldsymbol{x})^T \end{aligned} \quad (40)$$

for all $\boldsymbol{x} \in \mathcal{R}_p$ such that $\nabla p(\boldsymbol{x}) \neq \boldsymbol{0}$.

By the assumption that $\boldsymbol{x}_0 \in \mathcal{R}_p$, we have $\lambda_1(\boldsymbol{x}_0) > \lambda_2(\boldsymbol{x}_0)$ and $\lambda_2(\boldsymbol{x}_0) < 0$. Thus, $\lambda_i(\boldsymbol{x}_0)^2 - \lambda_1(\boldsymbol{x}_0) \lambda_i(\boldsymbol{x}_0) \neq 0$ for all $i = 2, 3, \ldots, n$. In addition, it was assumed that $\boldsymbol{x}_0$ is an isolated critical point (i.e. $\nabla p(\boldsymbol{x}_0) = \boldsymbol{0}$ and we may choose the neighbourhood $\mathcal{N}$ so that $\nabla p(\boldsymbol{x}) \neq \boldsymbol{0}$ for all $\boldsymbol{x} \in \mathcal{N}$) and $p \in C^3(\mathbb{R}^d, \mathbb{R})$. This implies the limit

$$\lim_{\boldsymbol{x} \to \boldsymbol{x}_0} \boldsymbol{U}(\boldsymbol{x})^T [\nabla^3 p(\boldsymbol{x}) \nabla p(\boldsymbol{x})] \boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{0},$$

and the convergence is uniform in any compact neighbourhood $\mathcal{N}$ of $\boldsymbol{x}_0$. From the above observations and Lemma B.1 we conclude that the matrix $\boldsymbol{C}(\boldsymbol{x})$ is nonsingular in $(\mathcal{R}_p \cap \mathcal{N}) \setminus \{\boldsymbol{x}_0\}$ for any sufficiently small neighbourhood $\mathcal{N}$ of $\boldsymbol{x}_0$. □

Proof of Theorem 3.3. Since $p \in C^3(\mathbb{R}^d, \mathbb{R})$ and $\boldsymbol{x}_0 \in \mathcal{R}_p$ is an isolated critical point of p, the assumptions of Lemma B.2 are satisfied. Thus, we can write the matrix $\boldsymbol{C}(\cdot)$ according to equation (38) in some neighbourhood $(\mathcal{R}_p \cap \mathcal{N}) \setminus \{\boldsymbol{x}_0\}$ and it is nonsingular in this neighbourhood. Consequently, because

$$\lim_{\theta \to 0} \boldsymbol{U}(\boldsymbol{x}(\theta))^T [\nabla^3 p(\boldsymbol{x}(\theta)) \nabla p(\boldsymbol{x}(\theta))] \boldsymbol{U}(\boldsymbol{x}(\theta)) = \boldsymbol{0}$$

by the assumptions that $p \in C^3(\mathbb{R}^d, \mathbb{R})$ and $\nabla p(\boldsymbol{x}_0) = \boldsymbol{0}$ and the fact that the columns of the matrix $\boldsymbol{U}(\cdot)$ are orthonormal, the limit

$$\lim_{\theta \to 0} \boldsymbol{C}(\boldsymbol{x}(\theta))^{-1} = \sum_{i=2}^{d} [\lambda_i(\boldsymbol{x}_0)^2 - \lambda_1(\boldsymbol{x}_0)\lambda_i(\boldsymbol{x}_0)]^{-1} \boldsymbol{v}_i(\boldsymbol{x}_0) \boldsymbol{v}_i(\boldsymbol{x}_0)^T \tag{41}$$

exists by Lemma B.2.

Since condition (6a) was assumed to be satisfied for all $\boldsymbol{x}(\theta)$ with $\theta \in \mathcal{D}$, there exists an interval $\mathcal{I} \subseteq \mathcal{D}$ such that

$$\boldsymbol{v}_1(\boldsymbol{x}(\theta)) = \pm \frac{\nabla p(\boldsymbol{x}(\theta))}{\|\nabla p(\boldsymbol{x}(\theta))\|} \quad \text{for all } \theta \in \mathcal{I} \setminus \{0\}. \tag{42}$$

This holds because $\boldsymbol{x}_0$ was assumed to be an isolated critical point, and by Lemma B.1 and the assumption that $\boldsymbol{x}_0 \in \mathcal{R}_p$ we have $\lambda_1(\cdot) > \lambda_2(\cdot)$ in some neighbourhood of $\boldsymbol{x}_0$. Consequently, the eigenvector $\boldsymbol{v}_1(\cdot)$ is uniquely determined in such a neighbourhood. Also, without loss of generality we may assume that the sign in equation (42) positive.

On the other hand, we have

$$\lim_{\theta \to 0} \|\boldsymbol{b}(\boldsymbol{x}(\theta))\| = \lim_{\theta \to 0} \left\| \boldsymbol{U}(\boldsymbol{x}(\theta))^T [\nabla^3 p(\boldsymbol{x}(\theta)) \nabla p(\boldsymbol{x}(\theta))] \boldsymbol{v}_1(\boldsymbol{x}(\theta)) \right\| = 0 \tag{43}$$

by the assumptions that $p \in C^3(\mathbb{R}^d, \mathbb{R})$ and $\nabla p(\boldsymbol{x}_0) = \boldsymbol{0}$ and by equation (42).

By the limits (41) and (43), the second term on the right-hand side of equation (20) converges to zero. Consequently, by using equation (42) we obtain that

$$\lim_{\theta \to 0} \|\boldsymbol{u}(\theta)\| = \lim_{\theta \to 0} \left\| \boldsymbol{v}_1(\boldsymbol{x}(\theta)) - \boldsymbol{U}(\boldsymbol{x}(\theta)) \boldsymbol{C}(\boldsymbol{x}(\theta))^{-1} \boldsymbol{b}(\boldsymbol{x}(\theta)) \right\| = 1.$$

By similar arguments we obtain that

$$\lim_{\theta \to 0} \left| \frac{\boldsymbol{u}(\theta)^T}{\|\boldsymbol{u}(\theta)\|} \boldsymbol{v}_1(\boldsymbol{x}_0) \right| = \lim_{\theta \to 0} \left| \frac{\boldsymbol{v}_1(\boldsymbol{x}(\theta))^T}{\|\boldsymbol{u}(\theta)\|} \boldsymbol{v}_1(\boldsymbol{x}_0) \right| = 1,$$

which concludes the proof.

□

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 A, 20520 TURKU, Finland | www.tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics

Turku School of Economics
- Institute of Information Systems Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

ISBN 978-952-12-2905-3
ISSN 1239-1891