Seppo Pulkkinen | Marko M. Mäkelä | Napsu Karmitsa

A Generalized Trust Region Newton Method Applied to Noise Reduction

Turku Centre *for* Computer Science

TUCS Technical Report
No 1061, October 2012

A Generalized Trust Region Newton Method Applied to Noise Reduction

Seppo Pulkkinen
University of Turku, Department of Mathematics and Statistics
FI-20014 Turku, Finland
seppo.pulkkinen@utu.fi
Marko M. Mäkelä
University of Turku, Department of Mathematics and Statistics
FI-20014 Turku, Finland
makela@utu.fi
Napsu Karmitsa
University of Turku, Department of Mathematics and Statistics
FI-20014 Turku, Finland
napsu@karmitsa.fi

TUCS Technical Report
No 1061, October 2012

Abstract

In practical applications related to, for instance, machine learning, data mining and pattern recognition, one is commonly dealing with noisy data lying near some low-dimensional manifold. A well-established tool for extracting the intrinsically low-dimensional structure from such data is principal component analysis (PCA). Due to the inherent limitations of this linear method, its extensions to extraction of nonlinear structures have attracted increasing research interest in recent years. Assuming a generative model for noisy data, we develop a probabilistic approach for separating the data-generating nonlinear functions from noise. We demonstrate that ridges of the marginal density induced by the model are viable estimators for the generating functions. For projecting a given point onto a ridge of its estimated marginal density, we develop a generalized trust region Newton method and prove its convergence to a ridge point. Accuracy of the model and computational efficiency of the projection method are assessed via numerical experiments where we utilize Gaussian kernels for nonparametric estimation of the underlying densities of the test datasets.

Keywords: principal manifold; noise reduction; generative model; ridge; density estimation; trust region; Newton method

TUCS Laboratory
Turku Optimization Group (TOpGroup)

1 Introduction

Machine learning, data mining and pattern recognition are typical tasks, where one is commonly dealing with noisy data lying near some low-dimensional manifold. Extraction of the intrinsically low-dimensional structure from such data is an essential task in many applications. The usual tool for this purpose is *principal component analysis* (PCA) that can be used to project a point set onto its *principal plane* [17]. Since this linear method is inadequate for complex nonlinear data, several attempts have been made to extend it to nonlinear *principal manifolds*. In particular, the so called *principal curves* (i.e. one-dimensional principal manifolds) have been applied to a variety of real-world applications. Examples include skeletonization of optical characters [19], detection of fault lines from seismological data [31], extraction of blood vessels from medical images [2], freeway traffic monitoring [8] and sensor fault detection [14]. Another related application is extraction of filaments from cosmological data [11]. Extraction of higher-dimensional principal manifolds has been applied, for instance, to visual recognition [21], astrophysical data analysis and low-dimensional visualization of microarray data in bioinformatics [12] and process monitoring [37].

Since the pioneering work of Hastie [15] and Hastie and Stuetzle [16], principal curve and manifold extraction has sparked a considerable amount of research interest (see e.g. [20] and [33]). Recently, Ozertem and Erdogmus [25] and Baş and Erdogmus [2–4] have proposed a novel definition that naturally extends the definition of a principal curve to higher-dimensional principal manifolds and addresses several limitations of the earlier definitions. That is, while the earlier approaches either attempt to fit a single globally defined principal curve or manifold to the point set under restrictive assumptions or require complicated parameter adjustments, the approach of [2–4] and [25] is based on locally defined principal manifolds and curves under rather nonrestrictive assumptions. The principal manifold of a point set is defined heuristically via the *critical set* that is a generalization of the set of modes (maxima) of the underlying probability density of the point set. As a modification of the standard *mean-shift* method (see e.g. [9] and [10]), a *subspace-constrained* mean-shift method for projecting a point set onto the critical set of its underlying density is proposed. For estimating the density, the authors suggest using either a Gaussian kernel density estimate or a Gaussian mixture model.

Extending the work in [2–4] and [25], we make a twofold contribution to extraction of principal manifolds from noisy point sets. Differently to the previous studies, we give a precise definition for the underlying density of a point set as the *marginal density* induced by a *generative model*. This model explicitly defines a set of smooth *generating functions* and a noise distribution. Assuming that the points are sampled from this model, we show by examples that when the amount of noise is sufficiently small, the *ridge set* of the marginal density is a viable estimator for the underlying generating functions. In order to make this approach

feasible for a practical implementation, we consider *nonparametric* estimation of the marginal density by Gaussian kernels.

When a point set is sampled from our model, we show that reconstruction of noise-free samples can be done by projecting the sample points onto the ridge set of their estimated marginal density. To this end, we propose a novel generalization of the *trust region* Newton method by Moré and Sorensen [22] and prove its convergence to a ridge point from a given starting point. Since a ridge point is a generalization of a mode, the proposed method is applicable to finding modes of a density as a special case. We demonstrate by numerical experiments that the rapidly converging Newton-based method gives a significant improvement in computational efficiency when compared to the mean-shift method and its subspace-constrained variant that typically exhibit only linear convergence rates [6]. This improvement is particularly relevant for real-time applications where performance is imperative. Another advantage is that whereas the mean-shift -based methods are only applicable to Gaussian mixtures and kernel density estimates, our method is applicable to general twice continuously differentiable densities.

The remainder of this paper is organized as follows. In Section 2, we develop the generative model for a noisy point set and the model for extraction of its generating functions. The trust region Newton method for projecting a point onto the ridge set of its underlying density is described in Section 3. Numerical test results and illustrations are given in Section 4. Finally, Section 5 concludes this paper and points out potential directions of future research.

2 Generative Model and Ridge Estimation

In this section, we develop the probabilistic framework for extraction of underlying low-dimensional structure from a noisy point set. First, in Subsection 2.1 we define the generative model that specifies a set of generating functions and a noise distribution for a point set having such structure. Second, assuming that the point set is sampled from this model, we consider ridges of the marginal density induced by the model as estimators to the generating functions in Subsection 2.2. Third, to make estimation of the generating functions via ridges of the marginal density amenable for a practical implementation, we consider estimation of the density by Gaussian kernels in Subsection 2.3.

2.1 Generative Model

In what follows, we describe the model for a noisy point set with a low-dimensional structure. The model specifies of a set of generating functions and a noise distribution. The generating functions, that we shall denote as $\{\boldsymbol{f}_i\}_{i=1}^n : \mathcal{D}_i \to \mathbb{R}^d$ with some compact and connected domains $\mathcal{D}_i \subset \mathbb{R}^m$, are differentiable func-

tions that parametrize a set of m-dimensional *coordinate patches* embedded in the d-dimensional space $\mathbb{R}^d$. The points sampled from the model are considered as instances of a random variable $\boldsymbol{X}$ whose outcome depends on the random variables I, $\boldsymbol{\Theta}$ and an independent random variable $\boldsymbol{\varepsilon}$ according to

$$(\boldsymbol{X} \mid I = i, \boldsymbol{\Theta} = \boldsymbol{\theta}) = \boldsymbol{f}_i(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}, \quad i = 1, 2, \ldots, n. \tag{1}$$

In other words, with a random choice of generating function index $I = i \in \{1, 2, \ldots, n\}$ and randomly chosen coordinates $\boldsymbol{\Theta} = \boldsymbol{\theta} \in \mathcal{D}_i$, a sample is generated from $\boldsymbol{f}_i$ with additive noise represented by the random variable $\boldsymbol{\varepsilon}$.

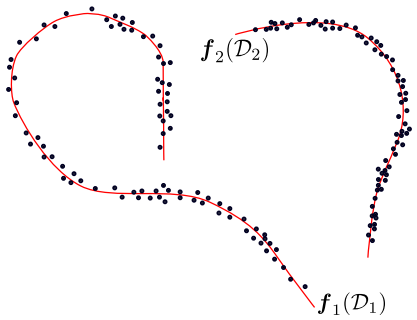


Figure 1: Coordinate patches parametrized by generating functions $\boldsymbol{f}_1 : \mathcal{D}_1 \to \mathbb{R}^2$ and $\boldsymbol{f}_2 : \mathcal{D}_2 \to \mathbb{R}^2$ with $\mathcal{D}_1 \subset \mathbb{R}$ and $\mathcal{D}_2 \subset \mathbb{R}$ (red curves) and noisy samples (black points).

For the sake of simplicity, we assume that for each sample the generating function $\boldsymbol{f}_i$ is chosen among the n generating functions with equal probability. Given $I = i$ with some $i \in \{1, 2, \ldots, n\}$, the random variable $\boldsymbol{\Theta}$ is uniformly distributed in the domain $\mathcal{D}_i$, and the noise represented by the random variable $\boldsymbol{\varepsilon}$ is normally distributed with standard deviation σ.

Assumption 2.1. *The random variables I, $\boldsymbol{\Theta}$ and $\boldsymbol{\varepsilon}$ are distributed according to*

$$p_I(I = i) = \frac{1}{n}, \quad (\boldsymbol{\Theta} \mid I = i) \sim \mathcal{U}(\mathcal{D}_i) \quad i = 1, 2 \ldots, n \quad \textit{and } \boldsymbol{\varepsilon} \sim \mathcal{N}_d(0, \sigma).$$

In order to make the generating functions well-defined, we assume that their Jacobian matrices are of full rank and that the coordinate system of each generating function has unit scaling.

Assumption 2.2. *The Jacobian $\boldsymbol{J}_{\boldsymbol{f}_i}$ of $\boldsymbol{f}_i$ is of full rank and has unit scaling for all $i = 1, 2 \ldots, n$ and $\boldsymbol{x} \in \mathcal{D}_i$. That is,* $\det(\boldsymbol{J}_{\boldsymbol{f}_i}(\boldsymbol{x})^T \boldsymbol{J}_{\boldsymbol{f}_i}(\boldsymbol{x})) = 1$.

Assuming that we only have a set of samples obtained from the model without any a priori information on the unobserved (latent) variables I and $\boldsymbol{\Theta}$ of the data-generating process, we need to obtain a density that only depends on the observed variable $\boldsymbol{X}$. To this end, we note that given $I = i$ with some $i \in \{1, 2, \ldots, n\}$

and $\boldsymbol{\Theta} = \boldsymbol{\theta}$ with $\boldsymbol{\theta} \in \mathcal{D}_i$, by equation (1) and Assumption 2.1 the conditional probability density of the random variable $\boldsymbol{X}$ is given by

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

By repeatedly using the definition of conditional probability density, we observe that the joint density $p_{\boldsymbol{X},I,\boldsymbol{\Theta}}$ satisfies the relation

$$\begin{aligned} p_{\boldsymbol{X},I,\boldsymbol{\Theta}}(\boldsymbol{x}, i, \boldsymbol{\theta}) &= p_{\boldsymbol{X}}(\boldsymbol{x} \mid I = i, \boldsymbol{\Theta} = \boldsymbol{\theta}) p_{I,\boldsymbol{\Theta}}(i, \boldsymbol{\theta}) \\ &= p_{\boldsymbol{X}}(\boldsymbol{x} \mid I = i, \boldsymbol{\Theta} = \boldsymbol{\theta}) p_{\boldsymbol{\Theta}}(\boldsymbol{\theta} \mid I = i) p_I(i), \end{aligned} \tag{3}$$

where by Assumption 2.1

$$p_{\boldsymbol{\Theta}}(\boldsymbol{\theta} \mid I = i) = \frac{1}{V(\mathcal{D}_i)}, \quad p_I(i) = \frac{1}{n}, \quad i = 1, 2 \ldots, n \tag{4}$$

and $V(\mathcal{D}_i)$ denotes the volume of the domain $\mathcal{D}_i$.

In order to get rid of the undesired variables, we can now *marginalize* the joint density (3) by integrating it with respect to $\boldsymbol{\theta}_i$ over the domain $\mathcal{D}_i$ and summing over the domain of I that is $\{1, 2, \ldots, n\}$. By equations (2)–(4), we obtain the expression

$$\begin{aligned} p_{\boldsymbol{X}}(\boldsymbol{x}) &= \sum_{i=1}^{n} \int_{\mathcal{D}_i} p_{\boldsymbol{X},I,\boldsymbol{\Theta}}(\boldsymbol{x}, i, \boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \sum_{i=1}^{n} \int_{\mathcal{D}_i} p_{\boldsymbol{X}}(\boldsymbol{x} \mid I = i, \boldsymbol{\Theta} = \boldsymbol{\theta}) p_{\boldsymbol{\Theta}}(\boldsymbol{\theta} \mid I = i) p_I(i) d\boldsymbol{\theta} \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^d n} \sum_{i=1}^{n} \frac{1}{V(\mathcal{D}_i)} \int_{\mathcal{D}_i} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{f}_i(\boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} \end{aligned} \tag{5}$$

for the marginal density of the observed variable $\boldsymbol{X}$. This density represents the observed density of the points sampled from the model.

2.2 Ridge Set of the Marginal Density as an Estimator

As an estimator of the noise-free generating functions from the marginal density (5), we now define the ridge set. Our definition is a specialization of the more general definition of a critical set given in [25]. An m-dimensional ridge set of a d-variate probability density is a set characterized by two properties. First, the gradient of the density at a ridge point is orthogonal to at least $d - m$ eigenvectors corresponding to the $d - m$ smallest eigenvalues of the Hessian matrix. Second, a ridge point is a (local) maximum of a cross-section of the density with respect to a hyperplane spanned by these eigenvectors. In order to make this definition well-posed, we require that the m greatest Hessian eigenvalues at a ridge point are distinct from each other and the remaining ones.

Definition 2.1. *A point $\boldsymbol{x} \in \mathbb{R}^d$ belongs to the m-dimensional* ridge set R_p^m*, where $0 \leq m < d$, of a twice differentiable probability density $p : \mathbb{R}^d \to \mathbb{R}$ if*

$$\nabla p(\boldsymbol{x})^T \boldsymbol{v}_i(\boldsymbol{x}) = 0, \quad \text{for all } i > m, \tag{6a}$$

$$\lambda_{m+1}(\boldsymbol{x}) < 0, \tag{6b}$$

$$\lambda_1(\boldsymbol{x}) > \lambda_2(\boldsymbol{x}) > \cdots > \lambda_{m+1}(\boldsymbol{x}), \quad \text{if } m > 0, \tag{6c}$$

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

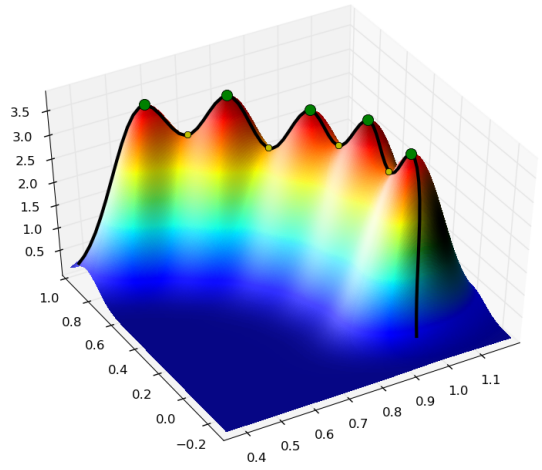


Figure 2: Ridge curve (i.e. a one-dimensional ridge set) of a probability density.

From the above definition we immediately observe that the set R_p^0 consists of the modes (maxima) of a probability density p, and hence the ridge set is a generalization of a set of modes. In addition, the above definition implies the following inclusion property which states that lower-dimensional ridge sets are contained within higher dimensional ones.

Proposition 2.1. *If $p : \mathbb{R}^d \to \mathbb{R}$ is a twice differentiable probability density, then $R_p^m \subseteq R_p^{m+1}$ for all $m = 0, 1, \ldots, d-1$.*

It is often more convenient to operate on the logarithm of a probability density than the density itself. If a probability density is bounded away from zero, then a straighforward calculation shows that the ridge sets of the density and its logarithm coincide. For the more general critical set, this has been proven in [25].

Proposition 2.2. *If $p : \mathbb{R}^d \to \mathbb{R}$ is a probability density with $p(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$, then $R_p^m = R_{\log p}^m$.*

In the special case where $n = 1$ and the generating function $\boldsymbol{f}_1$ parametrizes a line segment or a hyperrectangular region, the following result guarantees that the image of $\boldsymbol{f}_1$ lies in the ridge set of the marginal density (5). The proof of this result is given in Appendix A.

Theorem 2.1. *Let* $m > 0$, $\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^d \mid a_i \leq x_i \leq b_i, i = 1, 2, \ldots, m\}$ *with* $a_i < b_i$, $i = 1, 2, \ldots, m$ *and let* $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^d$ *be defined as*

$$\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{x}_0 + \sum_{i=1}^{m} \theta_i \boldsymbol{v}_i \tag{7}$$

with some $\boldsymbol{x}_0 \in \mathbb{R}^d$ *and mutually orthogonal vectors* $\{\boldsymbol{v}_i\}_{i=1}^m \subset \mathbb{R}^d \setminus \{\boldsymbol{0}\}$. *If* p *is defined by equation* (5) *with* $n = 1$ *and* $\boldsymbol{f}_1 = \boldsymbol{f}$, *then* $\{\boldsymbol{f}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \mathcal{D}\} \subseteq R_p^m$.

However, when $n > 1$ or when $n = 1$ and the generating function $\boldsymbol{f}_1$ has nonzero curvature (i.e. its second derivatives are not identically zero), the ridge set of the marginal density generally does not coincide with the images of the generating functions but entails some *model bias*. Estimating the bias for arbitrary generating functions is a difficult if not an impossible task due to the large degree of freedom in the choice of different generating functions. However, in Section 4 we demonstrate via numerical experiments that the model bias is expected to be small when the data points are sampled from the model of Subsection 2.1.

2.3 Estimation of the Marginal Density

In practice, we cannot use the marginal density (5) directly since it would require a priori knowledge on the generating functions and the noise distribution of the model. Nevertheless, given a sufficiently large number of samples from the marginal density, we can attempt to estimate it from the samples. A particularly well-suited tool for this purpose is *nonparametric* estimation which has the advantage of not requiring any assumptions on the functional form of the density.

One of the most widely used approaches for nonparametric density estimation is to use *Gaussian kernel density estimates* [29, 35]. In this density model, one Gaussian function is assigned for each sample point. This model only requires choosing the bandwidth parameter h, for which robust data-driven methods have been described in the literature (see e.g. [13], [18], [26] and [30]). Being a linear combination of Gaussian functions, the Gaussian kernel density estimate is a C^∞-function (i.e. infinitely many times differentiable). By virtue of this property, the ridge set of such a density is well-defined. For the reasons to be discussed in Section 3, we shall consider the logarithm of the density estimate that is also a C^∞-function.

Definition 2.2. *The* Gaussian log-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*

$$\hat{p}(\boldsymbol{x}) = \log\left(\frac{1}{N}\sum_{i=1}^{N} K_h(\|\boldsymbol{x} - \boldsymbol{y}_i\|)\right),$$

where the kernel $K_h : [0, \infty[\rightarrow]0, \infty[$ *is the* Gaussian *function*

$$K_h(r) = \frac{1}{(2\pi)^{\frac{d}{2}} h^d} \exp\left(-\frac{r^2}{2h^2}\right) \tag{8}$$

with kernel bandwidth $h > 0$.

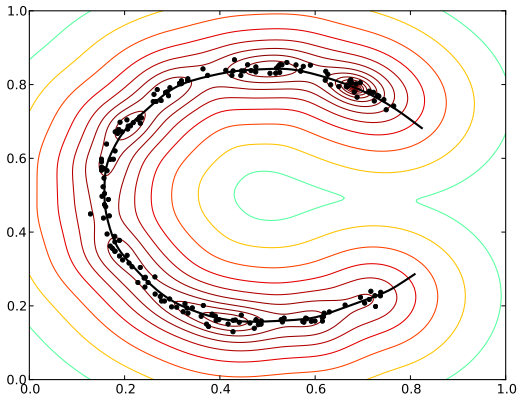


Figure 3: Contour lines and the ridge curve of a log-kernel density estimate obtained from a point set generated from the model of Subsection 2.1.

Figure 3 illustrates a point set generated from the model described in Subsection 2.1. In this example, the ridge curve (i.e. a one-dimensional ridge set) of a Gaussian log-kernel density estimate obtained from the point set with appropriately chosen bandwidth parameter h gives a good approximation of the underlying generating curve. The accuracy of the estimate improves as N, the number of samples is increased. For an elaborate analysis on this aspect, we refer to Chacón et al. [7] who have shown that under rather generic assumptions, a kernel density estimate and its derivatives converge to the estimated density and its derivatives, respectively, when N approaches infinity. The kernel K_h and the marginal density $p_{\boldsymbol{X}}$ of our model can be trivially shown to satisfy the assumptions stated in [7]. Being a C^∞-function, these results also hold for the log-kernel density $\hat{p}$.

3 Projection onto Ridge Set

Rather than estimating the generating functions $\boldsymbol{f}_i$ itself, which necessitates obtaining a parametrization of the ridge set, for the remainder of this paper we consider a simpler problem of projecting the sample points $\boldsymbol{Y}$ onto the ridge set of the log-kernel density $\hat{p}$. By such a projection, we obtain noise-free estimates of the sample points as if they were sampled from the generating functions $\boldsymbol{f}_i$. Once the sample points have been projected, parametrizations of the generating functions can be obtained, for instance, by constructing a neighbour graph. Such approaches

based on the assumption that the points lie directly on a low-dimensional manifold have been described, for instance, in [4], [5], [27], [28], [32] and [36]. Thus, the projection can be utilized as a preprocessing step for those methods.

Assuming that we have a log-kernel density estimate $\hat{p} : \mathbb{R}^d \to \mathbb{R}$ estimating the logarithm of the marginal density of a point set sampled from the model of Subsection 2.1, we now develop a method for projecting a given sample point onto an m-dimensional ridge set of $\hat{p}$. That is, in Subsections 3.1 and 3.2 we develop a trust region Newton method for approximate projection and prove its convergence to a ridge point in Subsection 3.3. The method is a generalization of the standard trust region Newton method (see e.g. [23]), since when $m = 0$, it reduces to the standard method for finding maxima.

3.1 Trust Region Newton Method

Recalling Definition 2.1, a point lying on an m-dimensional ridge set of a probability density is its local maximum in the hyperplane spanned by the Hessian eigenvectors corresponding to the $d - m$ smallest eigenvalues. This property suggests the idea of projecting a given point $\boldsymbol{x}_0$ onto the ridge set by seeking for a maximum of the density in the coordinate system induced by these eigenvectors. Based on this intuitive idea illustrated in Figure 4, we now describe a Newton-type method that maximizes the objective function $\hat{p}$ by successively maximizing its quadratic approximation in the subspace that is a local linearization of this nonlinear coordinate system. In order to ensure convergence, the method incorporates a trust region approach.

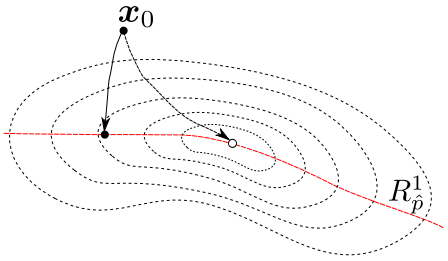


Figure 4: A given point $\boldsymbol{x}_0$ and its projection (black circle) onto the ridge curve $R^1_{\hat{p}}$ of a log-kernel density estimate $\hat{p}$ by subspace-constrained maximization of $\hat{p}$. By unconstrained maximization we obtain the nearest mode (white circle) that is in the zero-dimensional ridge set $R^0_{\hat{p}}$.

A reasonable requirement for the coordinate system described above is that when we have one linear generating function, the coordinate axes become straight lines. The following theorem that follows from the proof of Theorem 2.1 in Appendix A guarantees this property. With Proposition 2.2, this motivates our choice for the log-density estimate that estimates the logarithm of the marginal density rather than the density itself.

Theorem 3.1. *If $p : \mathbb{R}^d \to \mathbb{R}$ is defined as in Theorem 2.1, then for all $\boldsymbol{x} \in \mathbb{R}^d$ the eigenvectors of the Hessian $\nabla^2 \log p(\boldsymbol{x})$ corresponding to the $d - m$ smallest eigenvalues are orthogonal to the subspace spanned by the vectors $\{\boldsymbol{v}_i\}_{i=1}^m$.*

As in the standard trust region Newton method (see e.g. [23]), the objective function is near the current iterate $\boldsymbol{x}_k$ approximated by a quadratic model, which in our case takes the form

$$Q_k(\boldsymbol{s}) = \hat{p}(\boldsymbol{x}_k) + \nabla \hat{p}(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}^T \nabla^2 \hat{p}(\boldsymbol{x}_k) \boldsymbol{s}. \tag{9}$$

The iteration formula is $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$, where the step $\boldsymbol{s}_k$ maximizes Q_k and yields a sufficient increase of the objective function $\hat{p}$.

The main difference to the standard trust region method is that the maximization of the quadratic model is constrained to the subspace spanned by the Hessian eigenvectors $\{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_{i=m+1}^d$ corresponding to the $d - m$ smallest eigenvalues at $\boldsymbol{x}_k$. With this modification, a candidate for the step $\boldsymbol{s}_k$ is obtained as a solution to the subproblem

$$\max_{\boldsymbol{s}} Q_k(\boldsymbol{s}) \text{ s.t. } \begin{cases} \|\boldsymbol{s}\| \leq \Delta_k, \\ \boldsymbol{s} \in \text{span}(\boldsymbol{v}_{m+1}(\boldsymbol{x}_k), \boldsymbol{v}_{m+2}(\boldsymbol{x}_k), \ldots, \boldsymbol{v}_d(\boldsymbol{x}_k)), \end{cases} \tag{10}$$

where $\Delta_k \in]0, \Delta_{\max}]$ denotes the current trust region radius with some user-specified upper bound $\Delta_{\max}$. The initial trust region radius Δ_0 is chosen as $\Delta_0 = \frac{1}{4}\Delta_{\max}$. The parameter $\Delta_{\max}$ plays an important role in controlling the accuracy of the projection. By decreasing $\Delta_{\max}$, a more accurate projection along the nonlinear coordinate system induced by the subset of Hessian eigenvectors can be obtained when desired.

After the solution to the trust region subproblem (10) is obtained, the ratio

$$\rho_k = \frac{\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) - \hat{p}(\boldsymbol{x}_k)}{Q_k(\boldsymbol{s}_k) - Q_k(\boldsymbol{0})} \tag{11}$$

between the actual change in the objective function value and the change predicted by the quadratic model is computed. Based on this ratio, the trust region radius Δ_k is adjusted according to the rules [23]

$$\Delta_{k+1} = \begin{cases} \frac{1}{2}\Delta_k, & \text{if } \rho_k < \frac{1}{4}, \\ \min\{2\Delta_k, \Delta_{\max}\}, & \text{if } \|\boldsymbol{s}_k\| = \Delta_k \text{ and } \rho_k > \frac{3}{4}, \\ \Delta_k, & \text{otherwise.} \end{cases} \tag{12}$$

These rules ensure that the trust region radius Δ_k remains below the upper bound $\Delta_{\max}$, the iterates remain in a range where the quadratic model gives satisfactory approximations and the radius does not get too small. Finally, following [23], if $\rho_k > \frac{1}{10}$, we choose $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ as the next iterate. Otherwise, the current iterate remains unchanged (i.e. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$) and the above steps are carried out with the decreased trust region radius Δ_{k+1}.

The GTRN (generalized trust region Newton) algorithm with the stopping criterion described below is listed as Algorithm 1. The algorithm projects a given point onto the m-dimensional ridge set of a Gaussian log-kernel density estimate. For solving the trust region subproblem (10), the algorithm invokes the TRSREG algorithm that will be described in Subsection 3.2.

Algorithm 1: GTRN (generalized trust region Newton).

input : Gaussian log-kernel density estimate $\hat{p} : \mathbb{R}^d \to \mathbb{R}$
starting point $\boldsymbol{x}_0 \in \mathbb{R}^d$
ridge set dimension $0 \leq m < d$
maximum trust region radius $\Delta_{\max} > 0$
stopping criterion threshold $\varepsilon_{pr} > 0$
output: ridge point $\boldsymbol{x}^* \in R^m_{\hat{p}}$.
$\Delta_0 \leftarrow \frac{1}{4}\Delta_{\max}$
for $k = 0, 1, \ldots$ **do**
  Evaluate $\hat{p}(\boldsymbol{x}_k)$, $\nabla\hat{p}(\boldsymbol{x}_k)$, $\nabla_{pr}\hat{p}(\boldsymbol{x}_k)$ and $\nabla^2\hat{p}(\boldsymbol{x}_k)$.
  Compute eigenvalues $\lambda_1(\boldsymbol{x}_k) \geq \lambda_2(\boldsymbol{x}_k) \geq \cdots \geq \lambda_d(\boldsymbol{x}_k)$ and normalized eigenvectors $\{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_{i=1}^d$ of $\nabla^2\hat{p}(\boldsymbol{x}_k)$.
  if $\|\nabla_{pr}\hat{p}(\boldsymbol{x}_k)\| < \varepsilon_{pr}$ **then** terminate with $\boldsymbol{x}^* = \boldsymbol{x}_k$.
  $\boldsymbol{s}_k \leftarrow \text{TRSREG}(\nabla\hat{p}(\boldsymbol{x}_k), \{\lambda_i(\boldsymbol{x}_k)\}_{i=m+1}^d, \{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_{i=m+1}^d, \Delta_k)$
  $\rho_k \leftarrow \dfrac{\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) - \hat{p}(\boldsymbol{x}_k)}{Q_k(\boldsymbol{s}_k) - Q_k(\boldsymbol{0})}$
  if $\rho_k < \frac{1}{4}$ **then**
    $\Delta_{k+1} \leftarrow \frac{1}{2}\Delta_k$
  else if $\rho_k > \frac{3}{4}$ and $\|\boldsymbol{s}_k\| = \Delta_k$ **then**
    $\Delta_{k+1} \leftarrow \min\{2\Delta_k, \Delta_{\max}\}$
  if $\rho_k > \frac{1}{10}$ **then**
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$
  else
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$
$\boldsymbol{x}^* \leftarrow \boldsymbol{x}_{k+1}$

The iteration terminates when

$$\|\nabla_{pr}\hat{p}(\boldsymbol{x}_k)\| < \varepsilon_{pr}, \tag{13}$$

where $\varepsilon_{pr} > 0$ is some small threshold value and

$$\nabla_{pr}\hat{p}(\boldsymbol{x}_k) = \nabla\hat{p}(\boldsymbol{x}_k) - \sum_{i=1}^{m} \boldsymbol{v}_i^T(\boldsymbol{x})\nabla\hat{p}(\boldsymbol{x})\boldsymbol{v}_i(\boldsymbol{x})$$

is the projection of the gradient onto the subspace spanned by the normalized Hessian eigenvectors corresponding to the $d - m$ smallest eigenvalues. When

this criterion is satisfied, condition (6a) approximately holds. This criterion is in practice sufficient to test convergence to a ridge point. Namely, we will show in Subsection 3.3 that when the above iteration with the algorithm of Subsection 3.2 for solving problem (10) converges to a point $\boldsymbol{x}^*$ satisfying conditions (6a) and (6c) and the $d-m$ smallest eigenvalues $\{\lambda_i(\boldsymbol{x}^*)\}_{i=m+1}^d$ of the Hessian $\nabla^2\hat{p}(\boldsymbol{x}^*)$ are nonzero, the point $\boldsymbol{x}^*$ also satisfies condition (6b), and thus $\boldsymbol{x}^*$ is in the ridge set $R_{\hat{p}}^m$.

3.2 Solution to the Trust Region Subproblem

In this subsection we describe a generalization of the Moré and Sorensen algorithm [22] for solving the trust region subproblem (10). For notational convenience, we drop the subscripts k and consider the equivalent problem

$$\begin{aligned} \max_{\boldsymbol{s}} \quad & \boldsymbol{g}^T\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{s}^T\boldsymbol{H}\boldsymbol{s} \\ \text{s.t.} \quad & \|\boldsymbol{s}\| \leq \Delta \text{ and } \boldsymbol{s} \in \text{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \ldots, \boldsymbol{v}_d), \end{aligned} \tag{14}$$

where $\Delta > 0$, $0 \leq m < d$, $\boldsymbol{g} \in \mathbb{R}^d$ and $\{\boldsymbol{v}_i\}_{i=m+1}^d \subset \mathbb{R}^d$ denote the normalized eigenvectors of a matrix $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ corresponding to the $d-m$ smallest eigenvalues $\lambda_{m+1} \geq \lambda_{m+2} \geq \cdots \geq \lambda_d$.

We will utilize the following lemma for the formulation of the algorithm. This result is a generalization of Lemmata 2.1 and 2.3 given in [22], and it follows from the KKT conditions of problem (14). Its proof is given in Appendix A.

Lemma 3.1. *Let $\boldsymbol{g} \in \mathbb{R}^d$ and let $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ be a symmetric matrix and define*

$$Q(\boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{H}\boldsymbol{s}.$$

Let $0 \leq m < d$, $\Delta > 0$ and let $\lambda_{m+1} \geq \lambda_{m+2} \geq \cdots \geq \lambda_d$ and $\{\boldsymbol{v}_i\}_{i=m+1}^d$ denote the $d-m$ smallest eigenvalues and the corresponding normalized eigenvectors of $\boldsymbol{H}$, respectively. A vector $\boldsymbol{s}^* \in \mathbb{R}^d$ is a solution to the problem

$$\max_{\boldsymbol{s}} Q(\boldsymbol{s}) \quad \text{s.t. } \|\boldsymbol{s}\| \leq \Delta \text{ and } \boldsymbol{s} \in \text{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2} \ldots, \boldsymbol{v}_d)$$

if $\boldsymbol{s}^*$ is feasible and the conditions

$$\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa\boldsymbol{I})\boldsymbol{V}^T\boldsymbol{s}^* = -\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{g}, \tag{15}$$

$$\kappa(\Delta - \|\boldsymbol{s}^*\|) = 0, \tag{16}$$

$$\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa\boldsymbol{I})\boldsymbol{V}^T \quad \textit{is negative semidefinite} \tag{17}$$

hold for some $\kappa \geq 0$, where

$$\begin{aligned} \boldsymbol{V} &= [\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \ldots, \boldsymbol{v}_d] \in \mathbb{R}^{d\times(d-m)}, \\ \boldsymbol{\Lambda} &= \text{diag}[\lambda_{m+1}, \lambda_{m+2}, \ldots, \lambda_d] \in \mathbb{R}^{(d-m)\times(d-m)} \end{aligned}$$

and $\boldsymbol{I}$ is the $(d-m)\times(d-m)$ identity matrix.

Remark 3.1. *If* $\boldsymbol{s}^* \in \text{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \ldots, \boldsymbol{v}_d)$ *and* $\lambda_{m+1} - \kappa < 0$*, then condition* (15) *is equivalent to the condition*

$$\boldsymbol{s}^* = -\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{g}.$$

As suggested by Lemma 3.1 and Remark 3.1, we seek for a solution of the form

$$\boldsymbol{s}(\kappa) = -\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{g} = -\sum_{i=m+1}^{d} \frac{\boldsymbol{g}^T\boldsymbol{v}_i}{\lambda_i - \kappa}\boldsymbol{v}_i \tag{18}$$

to problem (14) with some $\kappa \geq 0$. First, by Remark 3.1 we note that if $\lambda_{m+1} < 0$, then condition (15) is satisfied by choosing $\boldsymbol{s}^* = \boldsymbol{s}(\kappa)$ for any $\kappa \geq 0$ and condition (17) is satisfied with $\kappa = 0$. With $\kappa = 0$, the step $\boldsymbol{s}^* = \boldsymbol{s}(\kappa)$ also satisfies condition (16). Thus, if $\lambda_{m+1} < 0$ and the step $\boldsymbol{s}(0)$ is in the feasible range, that is $\|\boldsymbol{s}(0)\| \leq \Delta$, then $\boldsymbol{s}^* = \boldsymbol{s}(0)$ is a solution to problem (14).

Otherwise, if either $\lambda_{m+1} \geq 0$ or $\|\boldsymbol{s}(0)\| > \Delta$, condition (17) is satisfied for any $\kappa \geq \max\{\lambda_{m+1}, 0\}$. From equation (18) we note that

$$\|\boldsymbol{s}(\kappa)\| = \|\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{g}\| = \left[\sum_{i=m+1}^{d} \frac{(\boldsymbol{g}^T\boldsymbol{v}_i)^2}{(\lambda_i - \kappa)^2}\right]^{\frac{1}{2}}, \tag{19}$$

and thus if $\boldsymbol{g}^T\boldsymbol{v}_{m+1} \neq 0$, then

$$\lim_{\kappa \to \lambda_{m+1}} \|\boldsymbol{s}(\kappa)\| = \infty \quad \text{and} \quad \lim_{\kappa \to \infty} \|\boldsymbol{s}(\kappa)\| = 0 \tag{20}$$

and $\|\boldsymbol{s}(\kappa)\|$ is a continuous nonincreasing function of κ for all $\kappa > \lambda_{m+1}$. Consequently, the equation $\|\boldsymbol{s}(\kappa)\| = \Delta$ has a solution in the interval $]\kappa_{\min}, \infty[$, where $\kappa_{\min} = \max\{\lambda_{m+1}, 0\}$. Clearly, the step $\boldsymbol{s}^* = \boldsymbol{s}(\kappa^*)$, where κ^* is a solution to this equation, satisfies condition (16). Furthermore, we have $\lambda_{m+1} - \kappa^* < 0$, and thus condition (17) is satisfied. Finally, since by Remark 3.1 and equation (18) condition (15) holds, $\boldsymbol{s}^* = \boldsymbol{s}(\kappa^*)$ is a solution to problem (14).

Remark 3.2. *If* $\boldsymbol{g}^T\boldsymbol{v}_{m+1} = 0$*, then the limiting value* (20) *for* $\kappa \to \lambda_{m+1}$ *does not necessarily hold, and consequently the equation* $\|\boldsymbol{s}(\kappa)\| = \Delta$ *may not have a solution. We omit the analysis of this special case since it very rarely occurs in practice and refer to [22] and [23].*

Numerical difficulties may arise in the solution of the equation $\|\boldsymbol{s}(\kappa)\| = \Delta$ when κ is close to λ_{m+1}. As suggested in [22], the solution becomes numerically more tractable if we consider the root-finding problem

$$\frac{1}{\Delta} - \frac{1}{\|\boldsymbol{s}(\kappa)\|} = 0 \tag{21}$$

instead.

As pointed out in [22], and also verified by our numerical experiments, a properly safeguarded Newton method is an efficient and reliable method for solving equation (21) in the interval $]\kappa_{\min}, \infty[$, where $\kappa_{\min} = \max\{\lambda_{m+1}, 0\}$. For the Newton iteration, the objective function and its derivative are

$$\phi(\kappa) = \frac{1}{\Delta} - \frac{1}{\|\boldsymbol{s}(\kappa)\|}, \quad \phi'(\kappa) = \left[\sum_{i=m+1}^{d} \frac{(\boldsymbol{g}^T\boldsymbol{v}_i)^2}{(\lambda_i - \kappa)^3}\right] \times \left[\sum_{i=m+1}^{d} \frac{(\boldsymbol{g}^T\boldsymbol{v}_i)^2}{(\lambda_i - \kappa)^2}\right]^{-\frac{3}{2}}$$

and the iteration formula is

$$\kappa^{(k+1)} = \kappa^{(k)} - \frac{\phi(\kappa^{(k)})}{\phi'(\kappa^{(k)})}.$$

Based on the above discussion and the safeguarding and stopping criteria given below, the `TRSREG` algorithm for solving problem (14) is listed as Algorithm 2.

Algorithm 2: TRSREG (trust region subproblem).

input : vector $\boldsymbol{g} \in \mathbb{R}^d$
eigenvalues $\lambda_{m+1} \geq \lambda_{m+2} \geq \cdots \geq \lambda_d$ of $\boldsymbol{H} \in \mathbb{R}^{d\times d}$
normalized eigenvectors $\{\boldsymbol{v}_i\}_{i=m+1}^{d} \subset \mathbb{R}^d$ of $\boldsymbol{H} \in \mathbb{R}^{d\times d}$
upper bound $\Delta > 0$ for $\|\boldsymbol{s}^*\|$
output: vector $\boldsymbol{s}^* \in \mathbb{R}^d$ that is a solution to (14)
if $\lambda_{m+1} < 0$ and $\|\boldsymbol{s}(0)\| \leq \Delta$ **then**
  Return with $\boldsymbol{s}^* = \boldsymbol{s}(0)$.
else
  $\kappa_{\min} \leftarrow \max\{\lambda_{m+1}, 0\}$
  $\kappa_{\max} \leftarrow \infty$
  $\kappa^{(0)} \leftarrow \max\{1.01\lambda_{m+1}, 10^{-10}\}$
  for $k = 0, 1, \ldots$ **do**
    $\kappa^{(k+1)} \leftarrow \kappa^{(k)} - \frac{\phi(\kappa^{(k)})}{\phi'(\kappa^{(k)})}$
    $\kappa^{(k+1)} \leftarrow \min\{\max\{\kappa^{(k+1)}, \kappa_{\min}\}, \kappa_{\max}\}$
    if $\phi(\kappa^{(k+1)}) < 0$ **then**
      $\kappa_{\max} \leftarrow \kappa^{(k+1)}$
    else
      $\kappa_{\min} \leftarrow \kappa^{(k+1)}$
    if $|\Delta - \|\boldsymbol{s}(\kappa^{(k+1)})\|| < 10^{-5}\Delta$ **then** terminate with $\boldsymbol{s}^* = \boldsymbol{s}(\kappa^{(k+1)})$.
  Return with $\boldsymbol{s}^* = \boldsymbol{s}(\kappa^{(k+1)})$.

Following the approach of [22], for safeguarding the Newton iteration, we enforce the bounds $\kappa_{\min}$ and $\kappa_{\max}$ such that $\kappa_{\min} \leq \kappa^{(k)} \leq \kappa_{\max}$ for all $k = 1, 2 \ldots$. The initial values are chosen as

$$\kappa_{\min} = \max\{\lambda_{m+1}, 0\}, \quad \kappa^{(0)} = \max\{1.01\lambda_{m+1}, 10^{-10}\} \quad \text{and} \quad \kappa_{\max} = \infty,$$

where the constants for $\kappa^{(0)}$ are chosen experimentally based on our numerical experiments. For each iteration step, we use the safeguarding and updating rules

1. $\kappa^{(k+1)} = \min\{\max\{\kappa^{(k+1)}, \kappa_{\min}\}, \kappa_{\max}\}$
2. If $\phi(\kappa^{(k+1)}) < 0$, then set $\kappa_{\max} = \kappa^{(k+1)}$. Otherwise, set $\kappa_{\min} = \kappa^{(k+1)}$.

As the stopping criterion for the Newton iteration we use the condition

$$|\Delta - \|\boldsymbol{s}(\kappa^{(k+1)})\|| < 10^{-5}\Delta,$$

where the constant 10^{-5} is chosen experimentally.

3.3 Convergence Analysis

We now prove convergence of a sequence generated by Algorithm 1 to a ridge point of a Gaussian log-kernel density estimate. This is done by adapting the convergence results of the classical trust region Newton method of [22] to a local minimum of a twice continuously differentiable function. The property that superlevel sets of our objective function are compact plays a key role in our analysis.

Lemma 3.2. *If $\hat{p} : \mathbb{R}^d \to \mathbb{R}$ is a Gaussian log-kernel density estimate defined according to Definition 2.2, then its* superlevel *set*

$$\mathcal{L}_c = \{\boldsymbol{x} \in \mathbb{R}^d \mid \hat{p}(\boldsymbol{x}) \geq c\}$$

is compact for all $c \in \mathbb{R}$.

Proof. Let $c \in \mathbb{R}$ such that $\mathcal{L}_c$ is nonempty, and let $\boldsymbol{z} \in \mathcal{L}_c$. First, we note the inequality

$$\|\boldsymbol{x} - \boldsymbol{z}\| - \|\boldsymbol{x} - \boldsymbol{y}\| \leq \|\boldsymbol{x} - \boldsymbol{y}\| + \|\boldsymbol{y} - \boldsymbol{z}\| - \|\boldsymbol{x} - \boldsymbol{y}\| = \|\boldsymbol{y} - \boldsymbol{z}\| \leq R$$

for all $R > 0$, $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{y} \in B(\boldsymbol{z}; R)$, where $B(\boldsymbol{z}; R)$ denotes a closed ball centered at $\boldsymbol{z}$. This inequality implies that

$$\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq (\|\boldsymbol{x} - \boldsymbol{z}\| - R)^2$$

for all $R > 0$, $\boldsymbol{x} \in \mathbb{R}^d \setminus B(\boldsymbol{z}; R)$ and $\boldsymbol{y} \in B(\boldsymbol{z}; R)$. Consequently, we have

$$\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2h^2}\right) \leq \exp\left(-\frac{(\|\boldsymbol{x} - \boldsymbol{z}\| - R)^2}{2h^2}\right) \tag{22}$$

for all $R > 0$, $\boldsymbol{x} \in \mathbb{R}^d \setminus B(\boldsymbol{z}; R)$, $\boldsymbol{y} \in B(\boldsymbol{z}; R)$ and $h > 0$. Let us now define the function

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}_i\|^2}{2h^2}\right)$$

with some $h > 0$ and $\{\boldsymbol{y}_i\}_{i=1}^N \subset \mathbb{R}^d$. Since $\lim_{r\to\infty} \exp\left(-\frac{r^2}{2h^2}\right) = 0$, from equation (22) we obtain that for all $R > 0$ such that $\boldsymbol{y}_i \in B(\boldsymbol{z}; R)$ for all $i = 1, 2, \ldots, N$, there exists $R' \geq R$ such that

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}_i\|^2}{2h^2}\right) \leq \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{(\|\boldsymbol{x}-\boldsymbol{z}\|-R)^2}{2h^2}\right) < c$$

for all $\boldsymbol{x} \in \mathbb{R}^d \setminus B(\boldsymbol{z}; R')$. Since $\log r \leq r$ for all $r > 0$ and $\exp r > 0$ for all $r \in \mathbb{R}$, we have $\hat{p}(\boldsymbol{x}) \leq \tilde{p}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^d$ if $\hat{p} : \mathbb{R}^d \to \mathbb{R}$ is a Gaussian log-kernel density estimate defined by the samples $\{\boldsymbol{y}_i\}_{i=1}^N$ and kernel bandwidth h. This implies that $\mathcal{L}_c \subseteq B(\boldsymbol{z}; R')$ for any such R'. Consequently, the set $\mathcal{L}_c$ is also bounded as a subset of the bounded set $B(\boldsymbol{z}; R')$. Finally, compactness of the set $\mathcal{L}_c$ follows from the continuity of $\hat{p}$. Namely, under $\hat{p}$, the inverse image of the closed interval $[c, \infty[$, which is the set $\mathcal{L}_c$, is closed. □

Let $\{\boldsymbol{x}_k\}$ denote a sequence generated by Algorithm 1. To facilitate the proofs of the convergence results, we introduce the basis spanned by normalized eigenvectors $\{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_{i=m+1}^d$ corresponding to the $d-m$ smallest eigenvalues $\lambda_{m+1}(\boldsymbol{x}_k) \geq \lambda_{m+2}(\boldsymbol{x}_k) \geq \cdots \geq \lambda_d(\boldsymbol{x}_k)$ of the Hessian $\nabla^2\hat{p}(\boldsymbol{x}_k)$ and the basis matrix

$$\boldsymbol{V}_k = [\boldsymbol{v}_{m+1}(\boldsymbol{x}_k), \boldsymbol{v}_{m+2}(\boldsymbol{x}_k), \ldots, \boldsymbol{v}_d(\boldsymbol{x}_k)] \in \mathbb{R}^{d\times(d-m)}.$$

In this basis we define the quadratic model

$$\tilde{Q}_k(\tilde{\boldsymbol{s}}) = \tilde{\nabla}\hat{p}(\boldsymbol{x}_k)^T\tilde{\boldsymbol{s}} + \frac{1}{2}\tilde{\boldsymbol{s}}^T\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)\tilde{\boldsymbol{s}} \tag{23}$$

with

$$\tilde{\nabla}\hat{p}(\boldsymbol{x}_k) = \boldsymbol{V}_k^T\nabla\hat{p}(\boldsymbol{x}_k), \quad \tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) = \boldsymbol{V}_k^T\nabla^2\hat{p}(\boldsymbol{x}_k)\boldsymbol{V}_k$$

and $\tilde{\boldsymbol{s}}_k = \boldsymbol{V}_k^T\boldsymbol{s}_k$. For the convergence proofs, we will utilize the following lemma whose proof is given in Appendix A.

Lemma 3.3. *At each iteration of Algorithm 1, there exists a constant $\kappa_k \geq 0$ such that the conditions*

$$[\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) - \kappa_k\boldsymbol{I}]\tilde{\boldsymbol{s}}_k = -\tilde{\nabla}\hat{p}(\boldsymbol{x}_k), \tag{24}$$

$$\kappa_k(\Delta_k - \|\tilde{\boldsymbol{s}}_k\|) = 0, \tag{25}$$

$$\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) - \kappa_k\boldsymbol{I} \quad \textit{is negative semidefinite} \tag{26}$$

are satisfied. At each iteration, the conditions

$$\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) + \hat{p}(\boldsymbol{x}_k) = Q_k(\boldsymbol{s}_k), \tag{27}$$

$$\|\tilde{\boldsymbol{s}}_k\| = \|\boldsymbol{s}_k\|, \tag{28}$$

$$\boldsymbol{s}_k = \boldsymbol{V}_k\tilde{\boldsymbol{s}}_k \tag{29}$$

also hold.

By condition (26), we can form the Cholesky factorization

$$\tilde{\nabla}^2 \hat{p}(\boldsymbol{x}_k) - \kappa_k \boldsymbol{I} = -\boldsymbol{R}_k^T \boldsymbol{R}_k, \tag{30}$$

for some upper triangular matrix $\boldsymbol{R}_k \in \mathbb{R}^{(d-m)\times(d-m)}$. Substituting equations (24) and (30) into equation (23) and then substituting equations (25) and (30) into the resulting expression yields

$$\begin{aligned}\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) &= -\tilde{\boldsymbol{s}}_k^T[\tilde{\nabla}^2 \hat{p}(\boldsymbol{x}_k) - \kappa_k \boldsymbol{I}]\tilde{\boldsymbol{s}}_k + \frac{1}{2}\tilde{\boldsymbol{s}}_k^T(-\boldsymbol{R}_k^T \boldsymbol{R}_k + \kappa_k \boldsymbol{I})\tilde{\boldsymbol{s}}_k \\ &= \frac{1}{2}(\|\boldsymbol{R}_k \tilde{\boldsymbol{s}}_k\|^2 + \kappa_k \Delta_k^2). \end{aligned} \tag{31}$$

Thus, since Algorithm 1 imposes the condition $\rho_k > \frac{1}{10}$ for accepting a new iterate $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ for all $k \geq 0$ and $Q_k(\boldsymbol{0}) = \hat{p}(\boldsymbol{x}_k)$, by equations (11), (27) and (31) we have the bound

$$\begin{aligned}\hat{p}(\boldsymbol{x}_{k+1}) - \hat{p}(\boldsymbol{x}_k) &> \frac{1}{10}[Q_k(\boldsymbol{s}_k) - Q_k(\boldsymbol{0})] \\ &= \frac{1}{10}\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) = \frac{1}{20}(\|\boldsymbol{R}_k \tilde{\boldsymbol{s}}_k\|^2 + \kappa_k \Delta_k^2) \end{aligned} \tag{32}$$

for all k such that $\rho_k > \frac{1}{10}$. Having established the above properties, we are now ready to carry out the proofs of our convergence results.

Lemma 3.4. *Assume that $\{\boldsymbol{x}_k\}$ is a convergent subsequence of some sequence generated by Algorithm 1 such that $\rho_k > \frac{1}{10}$ for all k. Then the corresponding sequence of parameters $\{\kappa_k\}$ has a subsequence that converges to zero.*

Proof. Let $\{\boldsymbol{x}_k\}$ be a convergent subsequence of some sequence generated by Algorithm 1 such that $\rho_k > \frac{1}{10}$ for all k. Let us assume by contradiction that the corresponding sequence $\{\kappa_k\}$ is bounded away from zero. That is, there exists $\varepsilon_1 > 0$ and an index k_0 such that $\kappa_k \geq \varepsilon_1 > 0$ for all $k \geq k_0$. Then equation (31) and the condition that $\|\boldsymbol{s}_k\| \leq \Delta_k$ with equation (28) imply that

$$\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) = \frac{1}{2}(\|\boldsymbol{R}_k \tilde{\boldsymbol{s}}_k\|^2 + \kappa_k \Delta_k^2) \geq \frac{1}{2}\kappa_k \Delta_k^2 \geq \frac{1}{2}\varepsilon_1 \|\tilde{\boldsymbol{s}}_k\|^2 \tag{33}$$

for all $k \geq k_0$. On the other hand, by Taylor's theorem and equation (9) we obtain that for all k there exists $\xi_k \in [0, 1]$ such that

$$\begin{aligned}\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) &= \hat{p}(\boldsymbol{x}_k) + \nabla \hat{p}(\boldsymbol{x}_k)^T \boldsymbol{s}_k + \frac{1}{2}\boldsymbol{s}_k^T \nabla^2 \hat{p}(\boldsymbol{x}_k + \xi_k \boldsymbol{s}_k)\boldsymbol{s}_k \\ &= Q_k(\boldsymbol{s}_k) + \frac{1}{2}\boldsymbol{s}_k^T[\nabla^2 \hat{p}(\boldsymbol{x}_k + \xi_k \boldsymbol{s}_k) - \nabla^2 \hat{p}(\boldsymbol{x}_k)]\boldsymbol{s}_k. \end{aligned}$$

This implies that for all k, we have

$$|\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) - Q_k(\boldsymbol{s}_k)| \leq \frac{1}{2}\|\boldsymbol{s}_k\|^2 \max_{0\leq\xi\leq 1} \|\nabla^2 \hat{p}(\boldsymbol{x}_k + \xi \boldsymbol{s}_k) - \nabla^2 \hat{p}(\boldsymbol{x}_k)\|. \tag{34}$$

Since $Q_k(\boldsymbol{s}_k) = \tilde{Q}_k(\tilde{\boldsymbol{s}}_k) + \hat{p}(\boldsymbol{x}_k)$ by equation (27) and $Q_k(\mathbf{0}) = \hat{p}(\boldsymbol{x}_k)$, equations (11) and (27) and inequalities (33) and (34) yield

$$|\rho_k - 1| = \frac{|\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) - \hat{p}(\boldsymbol{x}_k) - [Q_k(\boldsymbol{s}_k) - Q_k(\mathbf{0})]|}{Q_k(\boldsymbol{s}_k) - Q_k(\mathbf{0})} = \frac{|\hat{p}(\boldsymbol{x}_k + \boldsymbol{s}_k) - Q_k(\boldsymbol{s}_k)|}{\tilde{Q}_k(\tilde{\boldsymbol{s}}_k)}$$
$$\leq \frac{1}{\varepsilon_1} \max_{0 \leq \xi \leq 1} \|\nabla^2 \hat{p}(\boldsymbol{x}_k + \xi \boldsymbol{s}_k) - \nabla^2 \hat{p}(\boldsymbol{x}_k)\| \tag{35}$$

for all $k \geq k_0$.

By the continuity of $\hat{p}$ and the assumption that the sequence $\{\boldsymbol{x}_k\}$ converges, we have $\lim_{k\to\infty}[\hat{p}(\boldsymbol{x}_{k+1}) - \hat{p}(\boldsymbol{x}_k)] = 0$. Hence, by the assumption that κ_k is bounded away from zero, inequality (32) implies that the sequence $\{\Delta_k\}$ converges to zero. Consequently, the sequence $\{\|\boldsymbol{s}_k\|\}$ also converges to zero since the steps $\boldsymbol{s}_k$ satisfy the condition $\|\boldsymbol{s}_k\| \leq \Delta_k$ for all k.

On the other hand, if we let $r > 0$, the Hessian $\nabla^2 \hat{p}$ is uniformly continuous on the set

$$S_r = \{\boldsymbol{x} \in \mathbb{R}^d \mid \min_{\boldsymbol{y} \in \mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}} \|\boldsymbol{x} - \boldsymbol{y}\| \leq r\}$$

as a continuous function in $\mathbb{R}^d$. This is due to the fact that the set S_r is compact by compactness of the set $\mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}$ due to Lemma 3.2. In addition, since Algorithm 1 generates a sequence of nondecreasing function values $\hat{p}(\boldsymbol{x}_k)$, the iterates $\boldsymbol{x}_k$ lie in the set $\mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}$. Thus, since the sequence $\{\|\boldsymbol{s}_k\|\}$ converges to zero and we have $\|\boldsymbol{x}_k + \xi \boldsymbol{s}_k - \boldsymbol{x}_k\| = \|\xi \boldsymbol{s}_k\| \leq \|\boldsymbol{s}_k\|$ for all $\xi \in [0, 1]$, there exists an index k_1 and $r > 0$ such that $\boldsymbol{x}_k + \xi \boldsymbol{s}_k \in S_r$ for all $k \geq k_1$ and $\xi \in [0, 1]$. By the uniform continuity of $\nabla^2 \hat{p}$ in S_r, for all $\varepsilon_2 > 0$ there then exists an index k_2 such that

$$\frac{1}{\varepsilon_1} \max_{0 \leq \xi \leq 1} \|\nabla^2 \hat{p}(\boldsymbol{x}_k + \xi \boldsymbol{s}_k) - \nabla^2 \hat{p}(\boldsymbol{x}_k)\| < \varepsilon_2 \tag{36}$$

for all $k \geq \max\{k_1, k_2\}$. Then by inequalities (35) and (36), we have $\rho_k \geq \frac{1}{4}$ for all $k \geq \max\{k_0, k_1, k_2\}$. Consequently, the updating rules (12) yield that $\{\Delta_k\}$ is bounded away from zero, which leads to a contradiction. □

Lemma 3.5. *Assume that $\{\boldsymbol{x}_k\}$ is a subsequence of some sequence generated by Algorithm 1 with $0 \leq m < d$ such that $\{\boldsymbol{x}_k\}$ converges to some point $\boldsymbol{x}^*$. Also assume that $\rho_k > \frac{1}{10}$ for all k and*

(i) the corresponding sequence $\{\kappa_k\}$ converges to zero,

(ii) the sequence $\{\lambda_{m+1}(\boldsymbol{x}_k)\}$ converges to $\lambda_{m+1}(\boldsymbol{x}^*)$,

(iii) the sequences $\{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_k$ converge to $\boldsymbol{v}_i(\boldsymbol{x}^*)$ for all $i = 1, 2, \ldots, m$,

where $\{\boldsymbol{v}_i(\boldsymbol{x})\}_{i=1}^d$ denote the normalized eigenvectors of $\nabla^2 \hat{p}(\boldsymbol{x})$ corresponding to the eigenvalues $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$. Then $\boldsymbol{x}^* \in \tilde{R}_{\hat{p}}^m$, where

$$\tilde{R}_{\hat{p}}^m = \{\boldsymbol{x} \in \mathbb{R}^d \mid \nabla \hat{p}(\boldsymbol{x})^T \boldsymbol{v}_i(\boldsymbol{x}) = 0 \text{ for all } i > m \text{ and } \lambda_{m+1}(\boldsymbol{x}) \leq 0\}.$$

Proof. Let $0 \leq m < d$ and let $\{\boldsymbol{x}_k\}$ be a convergent subsequence of some sequence generated by Algorithm 1 with a limit point $\boldsymbol{x}^*$ such that $\rho_k > \frac{1}{10}$ for all k and assumptions (i)–(iii) are satisfied. Equations (24) and (30) imply that

$$\begin{aligned}\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\|^2 = \|\boldsymbol{R}_k^T\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2 &\leq \|\boldsymbol{R}_k\|^2\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2 \\ &= \|\boldsymbol{R}_k^T\boldsymbol{R}_k\|\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2 \\ &= \|\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) - \kappa_k\boldsymbol{I}\|\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2 \\ &\leq [\|\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)\| + \kappa_k]\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2\end{aligned}$$

for all k. Since Algorithm 1 generates a sequence of nondecreasing objective function values $\hat{p}(\boldsymbol{x}_k)$ and the superlevel set $\mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}$ is compact by Lemma 3.2, the sequence $\{\|\nabla^2\hat{p}(\boldsymbol{x}_k)\|\}$ is bounded from above due to the continuity of $\nabla^2\hat{p}$. Consequently, the sequence $\{\|\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)\|\}$ is also bounded. That is, there exists a constant $M > 0$ such that $\|\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)\| \leq M$ for all k. Thus, from the above inequality we obtain

$$\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2 \geq \frac{\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\|^2}{\|\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)\| + \kappa_k} \geq \frac{\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\|^2}{M + \kappa_k} \tag{37}$$

for all k. By inequality (32), the sequence $\{\|\boldsymbol{R}_k\tilde{\boldsymbol{s}}_k\|^2\}$ converges to zero. This implies that the sequence $\{\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\|\}$ converges to zero by the above inequality since the sequence $\{\kappa_k\}$ converges to zero by assumption (i).

In order to show that $\nabla\hat{p}(\boldsymbol{x}^*)^T\boldsymbol{v}_i(\boldsymbol{x}^*) = 0$ for all $i > m$, we note that by the property that $\text{span}(\boldsymbol{v}_1(\boldsymbol{x}_k), \boldsymbol{v}_2(\boldsymbol{x}_k), \ldots, \boldsymbol{v}_d(\boldsymbol{x}_k)) = \mathbb{R}^d$ and by assumption (iii),

$$\begin{aligned}\lim_{k\to\infty}\nabla\hat{p}(\boldsymbol{x}_k) &= \lim_{k\to\infty}\sum_{i=1}^{d}\boldsymbol{v}_i(\boldsymbol{x}_k)\boldsymbol{v}_i^T(\boldsymbol{x}_k)\nabla\hat{p}(\boldsymbol{x}_k) \\ &= \sum_{i=1}^{m}\boldsymbol{v}_i(\boldsymbol{x}^*)\boldsymbol{v}_i^T(\boldsymbol{x}^*)\nabla\hat{p}(\boldsymbol{x}^*) + \lim_{k\to\infty}\sum_{i=m+1}^{d}\boldsymbol{v}_i(\boldsymbol{x}_k)\boldsymbol{v}_i^T(\boldsymbol{x}_k)\nabla\hat{p}(\boldsymbol{x}_k).\end{aligned}$$

The sequence $\{\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\|\}$ converges to zero by inequality (37). Thus, by the definitions of the matrices $\boldsymbol{V}_k$ and the gradient $\tilde{\nabla}\hat{p}$, for the second term of the above equation we obtain

$$\begin{aligned}\lim_{k\to\infty}\|\sum_{i=m+1}^{d}\boldsymbol{v}_i(\boldsymbol{x}_k)\boldsymbol{v}_i^T(\boldsymbol{x}_k)\nabla\hat{p}(\boldsymbol{x}_k)\| &= \lim_{k\to\infty}\|\boldsymbol{V}_k\boldsymbol{V}_k^T\nabla\hat{p}(\boldsymbol{x}_k)\| \\ &= \lim_{k\to\infty}\|\boldsymbol{V}_k\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\| \\ &\leq \lim_{k\to\infty}\|\boldsymbol{V}_k\|\|\tilde{\nabla}\hat{p}(\boldsymbol{x}_k)\| = 0,\end{aligned}$$

which implies that $\nabla\hat{p}(\boldsymbol{x}^*) \in \text{span}(\boldsymbol{v}_1(\boldsymbol{x}^*), \boldsymbol{v}_2(\boldsymbol{x}^*), \ldots, \boldsymbol{v}_m(\boldsymbol{x}^*))$. Consequently, by the orthogonality of the eigenvectors, $\boldsymbol{v}_i(\boldsymbol{x}^*)$ we have $\nabla\hat{p}(\boldsymbol{x}^*)^T\boldsymbol{v}_i(\boldsymbol{x}^*) = 0$ for all $i > m$.

Finally, the condition $\lambda_{m+1}(\boldsymbol{x}^*) \leq 0$ follows from the fact that condition (17) of Lemma 3.1 is satisfied for all k by the construction of Algorithms 1 and 2. That is, the matrix $\boldsymbol{V}_k(\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I})\boldsymbol{V}_k^T$, where

$$\boldsymbol{\Lambda}_k = \operatorname{diag}[\lambda_{m+1}(\boldsymbol{x}_k), \lambda_{m+2}(\boldsymbol{x}_k), \ldots, \lambda_d(\boldsymbol{x}_k)] \in \mathbb{R}^{(d-m)\times(d-m)},$$

is negative semidefinite for all k. Thus, $\lambda_{m+1}(\boldsymbol{x}_k) - \kappa_k \leq 0$ for all k. Recalling that the sequence $\{\kappa_k\}$ converges to zero by assumption (i), by assumption (ii) we have

$$\lim_{k\to\infty} [\lambda_{m+1}(\boldsymbol{x}_k) - \kappa_k] = \lambda_{m+1}(\boldsymbol{x}^*) \leq 0,$$

which concludes the proof. □

With Lemmata 3.2–3.5, we can now provide conditions that a sequence generated by Algorithm 1 has a subsequence converging to a ridge point of a Gaussian log-kernel density estimate $\hat{p}$. This is done by using the following result about continuity of eigenvalues of the Hessian $\nabla^2\hat{p}$ (see Appendix B).

Theorem 3.2. *If $\hat{p} : \mathbb{R}^d \to \mathbb{R}$ is a Gaussian log-kernel density estimate, then there exist continuous functions $\{\lambda_i\}_{i=1}^d : \mathbb{R}^d \to \mathbb{R}$ representing the eigenvalues of $\nabla^2\hat{p}$ such that $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^d$.*

Our main convergence result guarantees existence of a convergent subsequence of any sequence generated by Algorithm 1. Such a subsequence converges to a point $\boldsymbol{x}^* \in \tilde{R}_{\hat{p}}^m$.

Theorem 3.3. *If $\{\boldsymbol{x}_k\}$ is a sequence generated by Algorithm 1 with $0 \leq m < d$ from a given starting point $\boldsymbol{x}_0 \in \mathbb{R}^d$, it has a subsequence that converges to a point $\boldsymbol{x}^* \in \tilde{R}_{\hat{p}}^m$.*

Proof. Let $\{\boldsymbol{x}_k\}$ be a sequence generated by Algorithm 1 from a given starting point $\boldsymbol{x}_0$. By construction of Algorithm 1, the sequence $\{\hat{p}(\boldsymbol{x}_k)\}$ is nondecreasing, and thus the iterates $\boldsymbol{x}_k$ lie on the superlevel set $\mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}$. Since the set $\mathcal{L}_{\hat{p}(\boldsymbol{x}_0)}$ is compact by Lemma 3.2, the sequence $\{\boldsymbol{x}_k\}$ has a convergent subsequence. Moreover, the updating rules (12) and the fact that $\rho_k > \frac{1}{10}$ when $\|\boldsymbol{s}_k\|$ is sufficiently small yield that any convergent subsequence of $\{\boldsymbol{x}_k\}$ has a subsequence satisfying the condition $\rho_k > \frac{1}{10}$ for all k. Thus, for notational simplicity we can assume that the sequence $\{\boldsymbol{x}_k\}$ itself is a convergent sequence satisfying this condition.

By Lemma 3.4 the sequence $\{\boldsymbol{x}_k\}$ has a subsequence $\{\boldsymbol{x}_{k_j}\}_j$ whose corresponding sequence of parameters $\{\kappa_{k_j}\}_j$ converges to zero. Hence the sequence $\{\kappa_{k_j}\}$ satisfies assumption (i) of Lemma 3.5. The property that the sequence $\{\lambda_{m+1}(\boldsymbol{x}_{k_j})\}_j$ satisfies assumption (ii) of Lemma 3.5 follows from Theorem 3.2. In addition, by compactness of the unit sphere and the property that the vectors $\{\boldsymbol{v}_i(\boldsymbol{x}_{k_j})\}_{i=1}^m$ lie on the unit sphere, for all $i = 1, 2, \ldots, m$ the sequences $\{\boldsymbol{v}_i(\boldsymbol{x}_{k_j})\}_j$ have convergent subsequences satisfying assumption (iii) of Lemma 3.5. The claim then follows from Lemma 3.5 applied to an appropriately chosen subsequence of $\{\boldsymbol{x}_{k_j}\}$ and from the definition of the set $\tilde{R}_{\hat{p}}^m$. □

If we make the following additional assumptions, then Theorem 3.3 guarantees that the limit point $\boldsymbol{x}^*$ of a convergent subsequence of some sequence $\{\boldsymbol{x}_k\}$ generated by Algorithm 1 is not only in the set $\tilde{R}^m_{\hat{p}}$ but also in the ridge set $R^m_{\hat{p}}$.

Assumption 3.1. *There exists a neighbourhood U of $\boldsymbol{x}^*$ such that $\boldsymbol{x}_0 \in U$ and condition (6c), where $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$ denote the eigenvalues of $\nabla^2\hat{p}(\boldsymbol{x})$, holds for all $\boldsymbol{x} \in U$.*

Assumption 3.2. *The iterates $\{\boldsymbol{x}_k\}$ generated by Algorithm 1 lie in the neighbourhood U for all $k = 0, 1, \ldots$.*

Assumption 3.3. *The limit point $\boldsymbol{x}^*$ satisfies the condition $\lambda_{m+1}(\boldsymbol{x}^*) < 0$.*

Corollary 3.1. *If $\{\boldsymbol{x}_k\}$ is a convergent subsequence of some sequence generated by Algorithm 1 with $0 \leq m < d$ from a given starting point $\boldsymbol{x}_0 \in \mathbb{R}^d$ and Assumptions 3.1–3.3 are satisfied, then the limit point $\boldsymbol{x}^*$ of $\{\boldsymbol{x}_k\}$ is in $R^m_{\hat{p}}$.*

By Definition 2.1 and continuity of the eigenvalues of $\nabla^2\hat{p}$ due to Theorem 3.2, the first assumption is satisfied if the starting point $\boldsymbol{x}_0$ is sufficiently close to the limiting point $\boldsymbol{x}^*$. In this case, the second assumption is satisfied as well. According to our numerical experiments with Gaussian kernel density estimates, the third assumption is satisfied in all but some pathological cases.

Assumption 3.1 also guarantees continuity of the eigenvectors of the Hessian $\nabla^2\hat{p}$ corresponding to the m greatest eigenvalues in the neighbourhood U. This is stated in the following theorem (see Appendix B).

Theorem 3.4. *If $m > 0$, then for any open neighbourhood U of $\boldsymbol{x}_0$ satisfying Assumption 3.1, there exists a set of continuous eigenvectors $\{\boldsymbol{v}_i\}_{i=1}^m : U \to \mathbb{R}^d$ of $\nabla^2\hat{p}$ corresponding to the eigenvalues $\{\lambda_i\}_{i=1}^m$ defined as in Theorem 3.2.*

Consequently, by the orthogonality of the eigenvectors of $\nabla^2\hat{p}$, the basis induced by the eigenvectors corresponding to the remaining $d - m$ smallest eigenvalues is continuous in U. Together with Assumption 3.2, this guarantees that we obtain a projection with a meaningful interpretation since the projection is done in the coordinate system induced by these eigenvectors (c.f. Figure 4).

4 Numerical Results

In this section we demonstrate the feasibility of our approach via numerical tests on synthetically generated datasets. First, we show in Subsection 4.2 that when a point set is generated according to the model of Subsection 2.1, ridge set projections of the points give good approximations of projections onto the underlying generating functions. Second, in Subsection 4.3 we demonstrate that when applied to ridge set projection, the generalized trust region Newton (abbreviated as `GTRN`) algorithm of Section 3 has a significant performance advantage over the

subspace-constrained mean-shift (abbreviated as SCMS) algorithm of [25]. The performance advantage is particularly large when the GTRN algorithm is compared to the standard mean-shift algorithm for finding modes (i.e. projecting onto a zero-dimensional ridge set).

4.1 Test Setup, Datasets and Density Estimates

We used eight different synthetically generated datasets in our tests. According to the model of Subsection 2.1, we used the known generating functions of the datasets and sampled uniformly distributed points from these functions with normally distributed random noise. For each dataset, we then picked an appropriate bandwidth parameter h for the density estimate by hand. The sample sizes N, noise standard deviations σ and kernel density bandwidths h for each dataset are listed in Table 1. The Spiral3d dataset was specifically generated for this paper, the Helix dataset is adapted from [25] and the other datasets are those generated by Kégl for testing his principal curve algorithm [1]. Kernel density ridge set projections obtained with the GTRN algorithm for these datasets are shown in Figures 5–6 in Appendix C.

Dataset	N	σ	h	Dataset	N	σ	h
Circle	800	0.075	0.09	Helix	4000	0.02	0.09
DistortedSShape	800	0.015	0.025	HalfCircle	800	0.05	0.08
DistortedHalfCircle	800	0.02	0.03	Zigzag	800	0.015	0.025
Spiral	1400	0.035	0.07	Spiral3d	1200	0.02	0.05

Table 1: Sample sizes N, noise standard deviations σ and kernel density bandwidths h used in the numerical tests.

In our numerical tests, we used FORTRAN 95-based implementations of the GTRN and SCMS algorithms, of which the latter is based on the pseudocode given in [25]. The algorithms were run on one core of a 3.0 GHz Core 2 Duo processor running a 64-bit Linux operating system. For plotting Figures 5–6 and carrying out the tests of Subsections 4.2 and 4.3, the GTRN algorithm was run with the experimentally chosen parameters

$$\varepsilon_{pr} = 10^{-6}, \quad k_{\max} = 200 \quad \text{and} \quad \Delta_{\max} = 3h, \tag{38}$$

where $k_{\max}$ is the maximum number of iterations allowed. For the SCMS algorithm the relevant parameters are the stopping criterion parameters ε_{pr} and $k_{\max}$. For both algorithms, we used the projected gradient stopping criterion (13).

4.2 Model and Estimation Error

We recall from Sections 2 and 3 that the projection method operates on the kernel density estimate of the marginal density whose ridge set gives an approximation of

the underlying generating function. Due to multiple steps involved in this process, it is therefore essential to distinguish between different sources of error. First, estimation of the underlying function by the ridge set of the marginal density introduces model bias. Second, additional errors are introduced by the kernel density estimation particularly when the number of sample points is not sufficiently large. Third, the trust region Newton method produces only an approximate projection by using successive linearizations of the nonlinear coordinate system. However, as we will show by the following experiments, the combined error from all of these steps is nevertheless reasonably small with appropriate test setup.

In order to gain insight into the model and density estimation errors, for each two-dimensional dataset of Table 1, we projected each point onto the ridge set of the kernel density obtained from the dataset. Each of these datasets has a single univariate generating function satisfying Assumption 2.2. For each projected point denoted by $\{\boldsymbol{x}_i^*\}_{i=1}^N$, we computed its distance to the generating curve. This distance is given by the expression

$$d_i = \min_{\boldsymbol{\theta}\in\mathcal{D}} \|\boldsymbol{x}_i^* - \boldsymbol{f}(\boldsymbol{\theta})\|,$$

where $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^2$ with $\mathcal{D} \subset \mathbb{R}$ denotes the generating function. Based on this distance, we define the average and maximum error

$$\bar{E} = \frac{1}{N\sigma}\sum_{i=1}^{N} d_i \quad \text{and} \quad E_{\max} = \max_{i=1,2,\ldots,N} \frac{d_i}{\sigma}, \tag{39}$$

respectively, relative to the noise standard deviation σ.

In order to assess the model error, by using the known generating curves $\boldsymbol{f}$ and noise deviations σ, we projected the sample points directly onto ridge sets of the marginal densities defined by equation (5). We approximated the generating curves with line segments connecting 1000 evenly distributed points sampled from the curve. Since the marginal density and its gradient and Hessian have closed form expressions along the approximating line segments[1], the ridge set of the density obtained by integrating over the line segments gives an accurate approximation of the ridge set of the true marginal density. The error measures $\bar{E}$ and $E_{\max}$ for projections onto ridge sets of the kernel density estimates and directly onto the ridge sets of the marginal densities are listed in Table 2.

Consistent with Figures 5 and 6, these results indicate that even when the marginal density is approximated by Gaussian kernels, the errors are within reasonable limits. However, in certain special cases larger errors can be observed. Figure 6d shows that the deviation between the projected points and the generating curve is quite large near the "corners" where the generating function has sharp turns, which also contributes to the error measures in the last row of Table 2. As

[1]We omit a detailed derivation of these expressions. They can be calculated by adapting the proof of Theorem 2.1.

seen from Figure 6b, the error grows large near the center of the spiral where its curvature radius is smaller, which contributes to the overall errors as well.

	Kernel density estimate		Marginal density	
	$\bar{E}$	$E_{\max}$	$\bar{E}$	$E_{\max}$
Circle	0.180	0.573	0.038	0.056
DistortedHalfCircle	0.149	1.467	0.043	1.197
DistortedSShape	0.192	1.518	0.051	1.367
HalfCircle	0.094	1.676	0.036	1.665
Spiral	0.158	1.929	0.039	1.106
Zigzag	0.213	3.065	0.042	2.310

Table 2: Average error $\bar{E}$ and maximum error $E_{\max}$ of projections onto ridge sets of kernel density estimates and marginal densities, respectively.

Excluding the error due to density estimation, the right columns of Table 2 represent the model bias, which is small compared to the estimation error. Again, the error depends on the amount of curvature in the generating function. In particular, the maximum error $E_{\max}$ is large with the Zigzag dataset having sharp turns. These observations seem to reflect the general behaviour of the underlying model. That is, the model bias is proportional to the ratio between the noise deviation σ and the curvature radius of the generating curve. Moreover, as seen from Figure 6, the bias is towards the center of curvature. We believe that such behaviour is also expected with datasets having higher-dimensional and multiple generating functions.

4.3 Performance Comparison with the Mean-Shift Algorithm

As the mean-shift method is the de facto standard method for finding modes and more recently, finding ridges of Gaussian mixtures and kernel density estimates, we compare the performance of our method with this method. For zero-dimensional ridge sets (i.e. modes) we shall present a performance comparison of our method with the standard mean-shift algorithm of [9] and [10] and for one- and higher-dimensional ridge sets we shall present a comparison with the `SCMS` algorithm.

In our setting, the number of function evaluations is the most important factor contributing to the performance of both algorithms. This is due to the fact that d, the dimension of the point sets is small compared to N, the number of data points. Since evaluation of each of the N components of a kernel density estimate involves distance computation and evaluation of a Gaussian function, its evaluation is computationally expensive. On the other hand, since the data dimension is small in our tests, the cost of linear algebra operations compared to the evaluation of the density estimate is negligible. Our test setup reflects potential applications

since particularly in many machine vision applications the dimension is relatively small (two or three) and the number of data points can be very large.

For each dataset, we projected each point onto the ridge set of the density estimate and counted the number of used function evaluations. By a function evaluation we mean a combined evaluation of the density function value, its gradient and Hessian. Combining these evaluations into a single loop is possible due to the special structure of Algorithm 1 and the Gaussian kernel density estimate. For the SCMS algorithm, we combined the mean-shift step with the loop that evaluates the gradient and Hessian. In order to give a picture of the overall performance of both algorithms, we also measured total CPU times from ten repeated test runs. The function evaluation counts and the CPU times (in seconds) are listed in Table 3. Evidently, rapid convergence of the Newton-based GTRN algorithm accounts for these results, where it outperforms the SCMS algorithm by a wide margin. Moreover, the measured CPU times strongly correlate with the used function evaluations, which shows that evaluation of the objective function is indeed the computationally most expensive operation in both algorithms.

	SCMS		GTRN	
	num. eval.	CPU time	num. eval.	CPU time
Circle	13 965	7.85	3 783	2.13
DistortedHalfCircle	12 161	6.81	3 684	2.08
DistortedSShape	10 561	5.93	3 384	1.91
HalfCircle	9 878	5.54	3 341	1.88
Helix	24 520	91.94	14 463	54.24
Spiral	15 984	15.19	5 701	5.44
Spiral3d	12 070	14.37	5 253	6.26
Zigzag	12 214	7.12	3 796	2.26

Table 3: Function evaluations and CPU times used by the SCMS and GTRN algorithms for ridge set projection.

Since the GTRN algorithm encompasses mode-finding (i.e. projection onto zero-dimensional ridge set) as a special case, we compared its mode-finding performance on the test datasets with the standard mean-shift algorithm (i.e. the SCMS algorithm with $m = 0$). The results of the performance comparison are listed in Table 4. Interestingly, for all datasets the mean-shift algorithm uses extremely high function evaluation counts and consequently long CPU times. A closer inspection revealed that this is due to very slow convergence of the mean-shift algorithm. As pointed out in [6], its convergence rate depends on the scaling of the Hessian eigenvalues and is particularly slow when the greatest eigenvalue of the Hessian of the density is near zero. This is exactly the case in our tests where the probability density has a clearly distinguishable ridge structure (c.f. Figure 3). Thus, our results suggest that for Gaussian kernel density estimates,

the performance of the GTRN algorithm is superior to mean-shift algorithm in mode-finding particularly when the probability density has a curved structure with elongated peaks. Though we did not numerically verify it, most likely this is also the case for nonzero-dimensional ridge set projection when the random noise has anisotropic covariance structure. That is, when the random variable $\boldsymbol{\varepsilon}$ in the model of Subsection 2.1 has a general covariance matrix instead of our choice $\sigma^2\boldsymbol{I}$.

	Mean-shift		GTRN	
	num. eval.	CPU time	num. eval.	CPU time
Circle	85 496	35.51	4 541	2.54
DistortedHalfCircle	137 335	57.05	4 826	2.68
DistortedSShape	114 078	47.42	4 770	2.67
HalfCircle	119 016	49.46	4 699	2.62
Spiral	220 654	159.11	8 170	7.72
Spiral3D	163 564	107.56	7 363	8.71
Zigzag	111 563	47.07	4 772	2.73

Table 4: Function evaluations and CPU times used by the mean-shift and GTRN algorithms for mode-finding.

5 Conclusions and Discussion

Nonlinear principal manifold extraction from noisy point sets is an essential problem in many practical applications related to, for instance, data mining, machine learning and pattern recognition. Adapting the earlier approaches of [2–4] and [25], we defined the principal manifold of a point set via the ridge set of its underlying probability density. This paper extends the ideas and methods presented in those papers in two ways.

First, differently to the earlier papers where the assumptions on a point set having a low-dimensional structure were not discussed in detail, we explicitly formulated a generative model for such a point set. The model describes a data-generating process where the points are sampled from a set of generating functions with normally distributed additive noise. In our approach, the ridge set of the marginal density induced by the model is interpreted as an estimator for the underlying generating functions. As we showed by examples, the model provides tools for assessing the accuracy of this estimator. Furthermore, the error can be expected to be reasonably small when the points are sampled from the model with a sufficiently small amount of noise. The model is also useful for practical applications since its assumptions are not too restrictive. To further demonstrate its potential for practical applications, we showed that the marginal density can be estimated nonparametrically by Gaussian kernels without any a priori information on the generating functions or parameters of the data-generating model.

Second, we developed a novel trust region Newton method for projecting a given point onto a ridge set of the kernel density estimate of its underlying density. Since modes (maxima) of a probability density are in its zero-dimensional ridge set, the projection method encompasses mode-finding as a special case. We established convergence of the method to a ridge point from a given starting point. Due to the rapid convergence rate inherited from the standard trust region Newton method, we observed in our numerical experiments that the proposed method has a significant performance advantage over the mean-shift method and its subspace-constrained variant introduced in [25]. The performance improvement is particularly relevant for real-time applications appearing, for instance, in machine vision, robotics and traffic and process monitoring where performance is imperative. Another advantage is that whereas the mean-shift -based methods are only applicable to Gaussian mixtures and kernel density estimates, the proposed method is applicable to general twice continuously differentiable densities.

Although the proposed method only produces a projected set of points and not a coordinate system, it can nevertheless be useful in conjunction with other dimensionality reduction methods. Namely, many of the existing approaches such as Isomap [32], locally linear embedding (LLE) [27,28], Laplacian eigenmaps [5] and maximum variance unfolding (MVU) [36] that build a coordinate system by constructing a neighbourhood graph of the points are sensitive to noise [34]. Thus, the projection obtained by the proposed method could be utilized as a preprocessing step for those methods in order to reduce the undesired noise. However, one shortcoming of the method is the assumption that the intrinsic dimensionality of the data is known a priori. A worthy topic of future research would be development of a method for automatically determining the intrinsic dimensionality of a given point set.

Acknowledgements. The first author was financially supported by the TUCS Graduate Programme. We would also like to thank Dr. Balázs Kégl for providing his datasets for our numerical tests.

References

[1] Principal curves. http://www.iro.umontreal.ca/~kegl/research/pcurves. visited on 12/10/2012.

[2] 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, 2011.

[3] E. Baş and D. Erdogmus. Connectivity of projected high dimensional data charts on one-dimensional curves. *Signal Processing*, 91(10):2404–2409, 2011.

[4] E. Baş and D. Erdogmus. Sampling on locally defined principal manifolds. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2276–2279, 2011.

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

[6] M. Á. Carreira-Perpiñán. Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):767–776, 2007.

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

[8] 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.

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

[10] 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.

[11] 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.

[12] A. N. Gorban, B. Kégl, D. C. Wunsch, and A. Y. Zinovyev. *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin, Heidelberg, 2008.

[13] P. Hall, S. J. Sheather, M. C. Jones, and J. S. Marron. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika*, 78(2):263–269, 1991.

[14] M.-F. Harkat, S. Djelel, N. Doghmane, and M. Benouaret. Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis. *International Journal of Automation and Computing*, 4:149–155, 2007.

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

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

[17] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, Berlin, 1986.

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

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

[20] 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.

[21] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, 2002.

[22] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[23] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

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

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

[26] B. U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990.

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

[28] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[29] D. W. Scott. *Multivatiate Density Estimation: Theory Practice and Visualization*. John Wiley and Sons, 1992.

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

[31] 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.

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

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

[34] F. S. Tsai. Comparative study of dimensionality reduction techniques for data visualization. *Journal of Artificial Intelligence*, 3(3):119–134, 2010.

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

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

[37] D. J. H. Wilson, G. W. Irwin, and G. Lightbody. RBF principal manifolds for process monitoring. *IEEE Transactions on Neural Networks*, 10(6):1424–1434, 1999.

A Proofs of Theorem 2.1 and Lemmata 3.1 and 3.3

Theorem 2.1 Let $m > 0$, $\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^d \mid a_i \leq x_i \leq b_i, i = 1, 2, \ldots, m\}$ with $a_i < b_i$, $i = 1, 2, \ldots, m$ and let $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^d$ be defined as

$$\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{x}_0 + \sum_{i=1}^{m} \theta_i \boldsymbol{v}_i$$

with some $\boldsymbol{x}_0 \in \mathbb{R}^d$ and mutually orthogonal vectors $\{\boldsymbol{v}_i\}_{i=1}^m \subset \mathbb{R}^d \setminus \{\boldsymbol{0}\}$. If p is defined by equation (5) with $n = 1$ and $\boldsymbol{f}_1 = \boldsymbol{f}$, then $\{\boldsymbol{f}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \mathcal{D}\} \subseteq R_p^m$.

Proof. Without loss of generality we can assume that $\boldsymbol{x}_0 = \boldsymbol{0}$ and

$$\boldsymbol{f}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \theta_i \boldsymbol{e}_i, \tag{40}$$

where $\boldsymbol{e}_i \in \mathbb{R}^d$ denotes a unit vector along the i-th coordinate axis. With some noise standard deviation $\sigma > 0$, the gradient and Hessian of the marginal density (5) induced by the model of Subsection 2.1 with this function are

$$\nabla p(\boldsymbol{x}) = -\frac{C_\sigma}{V(\mathcal{D})} \int_{\mathcal{D}} \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta}) \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} \tag{41}$$

and

$$\begin{aligned}\nabla^2 p(\boldsymbol{x}) =& \frac{C_\sigma}{V(\mathcal{D})} \int_{\mathcal{D}} \left[\frac{\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta}) \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})}{\sigma^2} - \boldsymbol{I}\right] \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta}\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} \\ =& \frac{C_\sigma}{V(\mathcal{D})\sigma^2} \int_{\mathcal{D}} \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta}) \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})^T \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} - \frac{p(\boldsymbol{x})}{\sigma^2} \boldsymbol{I},\end{aligned} \tag{42}$$

where

$$\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{x} - \sum_{i=1}^{m} \theta_i \boldsymbol{e}_i \quad \text{and} \quad C_\sigma = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^{d+2}},$$

respectively, and $\boldsymbol{I}$ denotes the $d \times d$ identity matrix.

By Proposition 2.2, it suffices to consider the logarithm of the density p. For the Hessian of the logarithm of p, we obtain the expression

$$\nabla^2 \log p(\boldsymbol{x}) = \frac{\nabla^2 p(\boldsymbol{x})}{p(\boldsymbol{x})} - \frac{\nabla p(\boldsymbol{x}) \nabla p(\boldsymbol{x})^T}{p(\boldsymbol{x})^2}. \tag{43}$$

Since $\sum_{j=1}^m \theta_j \boldsymbol{e}_j^T \boldsymbol{e}_i = 0$ for all $i = m+1, m+2, \ldots, d$, from equations (41) and (42) we obtain

$$\begin{aligned}\nabla p(\boldsymbol{x})^T \boldsymbol{e}_i &= -\frac{C_\sigma}{V(\mathcal{D})} \int_{\mathcal{D}} \left(\boldsymbol{x} - \sum_{j=1}^{m} \theta_j \boldsymbol{e}_j\right)^T \boldsymbol{e}_i \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} \\ &= -\boldsymbol{x}^T \boldsymbol{e}_i \frac{C_\sigma}{V(\mathcal{D})} \int_{\mathcal{D}} \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} = -\frac{x_i p(\boldsymbol{x})}{\sigma^2}\end{aligned} \tag{44}$$

and

$$\begin{aligned}\nabla^2 p(\boldsymbol{x})\boldsymbol{e}_i = \frac{C_\sigma}{V(\mathcal{D})\sigma^2}\int\limits_{\mathcal{D}} \left(\boldsymbol{x} - \sum_{j=1}^m \theta_j \boldsymbol{e}_j\right)\left(\boldsymbol{x} - \sum_{j=1}^m \theta_j \boldsymbol{e}_j\right)^T \boldsymbol{e}_i \cdot \\ \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} - \frac{p(\boldsymbol{x})}{\sigma^2}\boldsymbol{e}_i \\ = \boldsymbol{x}^T \boldsymbol{e}_i \frac{C_\sigma}{V(\mathcal{D})\sigma^2}\int\limits_{\mathcal{D}} \boldsymbol{r}(\boldsymbol{x};\boldsymbol{\sigma}) \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} - \frac{p(\boldsymbol{x})}{\sigma^2}\boldsymbol{e}_i \\ = -\frac{x_i}{\sigma^2}\nabla p(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{\sigma^2}\boldsymbol{e}_i\end{aligned}$$

for all $i = m+1, m+2, \ldots, d$. Substituting these expressions into equation (43) yields

$$\nabla^2 \log p(\boldsymbol{x})\boldsymbol{e}_i = -\frac{\boldsymbol{e}_i}{\sigma^2} \tag{45}$$

for all $i = m+1, m+2, \ldots, d$, which shows that the vectors $\boldsymbol{e}_{m+1}, \boldsymbol{e}_{m+2}, \ldots, \boldsymbol{e}_d$ span the $d-m$-dimensional eigenspace of the matrix $\nabla^2 \log p(\boldsymbol{x})$ corresponding to the eigenvalue $-\frac{1}{\sigma^2}$ for all $\boldsymbol{x} \in \mathbb{R}^d$. Conseqently, condition (6b) for $\log p$ holds for all $\boldsymbol{x} \in \mathbb{R}^d$. Furthermore, by equation (44) and the fact that $p(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$, condition (6a) for $\log p$ holds if and only if $\boldsymbol{x} \in \operatorname{span}(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m)$.

On the other hand, from equations (41)–(43) we obtain

$$\begin{aligned}\boldsymbol{e}_i^T \nabla^2 \log p(\boldsymbol{x})\boldsymbol{e}_i &= \frac{\boldsymbol{e}_i^T \nabla^2 p(\boldsymbol{x})\boldsymbol{e}_i}{p(\boldsymbol{x})} - \frac{\boldsymbol{e}_i^T \nabla p(\boldsymbol{x}) \nabla p(\boldsymbol{x})^T \boldsymbol{e}_i}{p(\boldsymbol{x})^2} \\ &= -\frac{1}{\sigma^2} + F_{\sigma,i}(\boldsymbol{x})\end{aligned}$$

for all $i = 1, 2, \ldots, m$, where

$$\begin{aligned}F_{\sigma,i}(\boldsymbol{x}) = &\frac{C_\sigma}{p(\boldsymbol{x})V(\mathcal{D})\sigma^2}\int\limits_{\mathcal{D}} (x_i - \theta_i)^2 \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta} - \\ &\frac{C_\sigma}{p(\boldsymbol{x})^2 V(\mathcal{D})}\left[\int\limits_{\mathcal{D}} (x_i - \theta_i) \exp\left(-\frac{\|\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\theta})\|^2}{2\sigma^2}\right) d\boldsymbol{\theta}\right]^2.\end{aligned}$$

Since it can be shown that $F_{\sigma,i}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and $i = 1, 2, \ldots, m$, we observe that

$$\boldsymbol{e}_i^T \nabla^2 \log p(\boldsymbol{x})\boldsymbol{e}_i = -\frac{1}{\sigma^2} + F_{\sigma,i}(\boldsymbol{x}) > -\frac{1}{\sigma^2} = \boldsymbol{e}_j^T \nabla^2 \log p(\boldsymbol{x})\boldsymbol{e}_j$$

for all $i = 1, 2, \ldots, m$ and $j = m+1, m+2, \ldots, d$. This shows that the eigenvalues of $\nabla^2 \log p$ corresponding to the eigenvectors in the subspace spanned by the vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m$ are strictly greater than the remaining ones for all $\boldsymbol{x} \in \mathbb{R}^d$. Thus, the eigenvalues of $\nabla^2 \log p$ satisfy condition (6c) for all $\boldsymbol{x} \in \mathbb{R}^d$. Since we showed above that for $\log p$, condition (6b) holds for all $\boldsymbol{x} \in \mathbb{R}^d$ and condition (6a) holds if and only if $\boldsymbol{x} \in \operatorname{span}(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m)$, this implies that $R^m_{\log p} = \operatorname{span}(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m)$. By equation (40) and Proposition 2.2, we then obtain that $\{f(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \mathcal{D}\} \subseteq R^m_{\log p} = R^m_p$. □

For the proofs of Lemmata 3.1 and 3.3, we need the following lemma.

Lemma A.1. *Let $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ be a symmetric matrix and define*

$$\boldsymbol{V} = [\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \dots, \boldsymbol{v}_d] \in \mathbb{R}^{d\times(d-m)}$$

and

$$\boldsymbol{\Lambda} = \operatorname{diag}[\lambda_{m+1}, \lambda_{m+2}, \dots, \lambda_d] \in \mathbb{R}^{(d-m)\times(d-m)},$$

where $\{\boldsymbol{v}_i\}_{i=m+1}^d$ denote the orthonormal eigenvectors of $\boldsymbol{H}$ corresponding to its eigenvalues $\lambda_{m+1} \geq \lambda_{m+2} \geq \cdots \geq \lambda_d$. If we define $\tilde{\boldsymbol{H}} = \boldsymbol{V}^T\boldsymbol{H}\boldsymbol{V}$, then $\tilde{\boldsymbol{H}} = \boldsymbol{\Lambda}$. Furthermore, if we let $\boldsymbol{g} \in \mathbb{R}^d$ and $\boldsymbol{s} \in \operatorname{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \dots, \boldsymbol{v}_d)$ and define $\tilde{\boldsymbol{s}} = \boldsymbol{V}^T\boldsymbol{s}$ and

$$Q(\boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{H}\boldsymbol{s} \quad \text{and} \quad \tilde{Q}(\tilde{\boldsymbol{s}}) = \tilde{\boldsymbol{g}}^T\tilde{\boldsymbol{s}} + \frac{1}{2}\tilde{\boldsymbol{s}}^T\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}}$$

with $\tilde{\boldsymbol{g}} = \boldsymbol{V}^T\boldsymbol{g}$, then we have $Q(\boldsymbol{s}) = \tilde{Q}(\tilde{\boldsymbol{s}})$, $\|\boldsymbol{s}\| = \|\tilde{\boldsymbol{s}}\|$ and $\boldsymbol{s} = \boldsymbol{V}\tilde{\boldsymbol{s}}$.

Proof. In order to show that $\tilde{\boldsymbol{H}} = \boldsymbol{\Lambda}$, we define the matrices

$$\boldsymbol{W} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_d] \in \mathbb{R}^{d\times d} \quad \text{and} \quad \boldsymbol{U} = \operatorname{diag}[\lambda_1, \lambda_2, \dots, \lambda_d] \in \mathbb{R}^{d\times d},$$

where $\{\boldsymbol{v}_i\}_{i=1}^d$ denote the eigenvectors of $\boldsymbol{H}$ corresponding to its eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. By the definition $\tilde{\boldsymbol{H}} = \boldsymbol{V}^T\boldsymbol{H}\boldsymbol{V}$ and the eigendecomposition $\boldsymbol{H} = \boldsymbol{W}\boldsymbol{U}\boldsymbol{W}^T$, we have

$$\tilde{\boldsymbol{H}} = \boldsymbol{V}^T\boldsymbol{H}\boldsymbol{V} = \boldsymbol{V}^T\boldsymbol{W}\boldsymbol{U}\boldsymbol{W}^T\boldsymbol{V} = \boldsymbol{V}^T\left[\sum_{i=1}^d \lambda_i\boldsymbol{v}_i\boldsymbol{v}_i^T\right]\boldsymbol{V} = \sum_{i=m+1}^d \boldsymbol{e}_{i-m}\boldsymbol{e}_{i-m}^T\lambda_i = \boldsymbol{\Lambda},$$

where $\boldsymbol{e}_i$ denotes a $d-m$-dimensional unit vector along the i-th coordinate axis.

Since the vectors $\{\boldsymbol{v}_i\}_{i=m+1}^d$ span an orthonormal basis, we have $\boldsymbol{s} = \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{s}$ for all $\boldsymbol{s} \in \operatorname{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \dots, \boldsymbol{v}_d)$. Thus, by the definition $\tilde{\boldsymbol{s}} = \boldsymbol{V}^T\boldsymbol{s}$, for any such $\boldsymbol{s}$ we obtain

$$Q(\boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{H}\boldsymbol{s} = \boldsymbol{g}^T\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{H}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{s} = \tilde{\boldsymbol{g}}^T\tilde{\boldsymbol{s}} + \frac{1}{2}\tilde{\boldsymbol{s}}^T\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} = \tilde{Q}(\tilde{\boldsymbol{s}})$$

and also $\boldsymbol{s} = \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{s} = \boldsymbol{V}\tilde{\boldsymbol{s}}$.

Finally, for $\boldsymbol{s} \in \operatorname{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \dots, \boldsymbol{v}_d)$ and $\tilde{\boldsymbol{s}} = \boldsymbol{V}^T\boldsymbol{s}$, by expressing $\boldsymbol{s}$ as a linear combination of the orthonormal basis vectors $\{\boldsymbol{v}_i\}_{i=1}^d$ that span $\mathbb{R}^d$, we obtain

$$\|\boldsymbol{s}\|^2 = \|\sum_{i=1}^d \boldsymbol{v}_i^T\boldsymbol{s}\boldsymbol{v}_i\|^2 = \|\sum_{i=m+1}^d \boldsymbol{v}_i^T\boldsymbol{s}\boldsymbol{v}_i\|^2 = \sum_{i=m+1}^d (\boldsymbol{v}_i^T\boldsymbol{s})^2 = \|\boldsymbol{V}^T\boldsymbol{s}\|^2 = \|\tilde{\boldsymbol{s}}\|^2.$$

□

Lemma 3.1 Let $\boldsymbol{g} \in \mathbb{R}^d$ and let $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ be a symmetric matrix and define

$$Q(\boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{H}\boldsymbol{s}.$$

Let $0 \leq m < d$, $\Delta > 0$ and let $\lambda_{m+1} \geq \lambda_{m+2} \geq \cdots \geq \lambda_d$ and $\{\boldsymbol{v}_i\}_{i=m+1}^d$ denote the $d - m$ smallest eigenvalues and the corresponding normalized eigenvectors of $\boldsymbol{H}$, respectively. A vector $\boldsymbol{s}^* \in \mathbb{R}^d$ is a solution to the problem

$$\max_{\boldsymbol{s}} Q(\boldsymbol{s}) \quad \text{s.t. } \|\boldsymbol{s}\| \leq \Delta \text{ and } \boldsymbol{s} \in \text{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2} \ldots, \boldsymbol{v}_d) \tag{46}$$

if $\boldsymbol{s}^*$ is feasible and the conditions

$$\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})\boldsymbol{V}^T\boldsymbol{s}^* = -\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{g}, \tag{47}$$

$$\kappa(\Delta - \|\boldsymbol{s}^*\|) = 0, \tag{48}$$

$$\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})\boldsymbol{V}^T \quad \text{is negative semidefinite} \tag{49}$$

hold for some $\kappa \geq 0$, where

$$\boldsymbol{V} = [\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \ldots, \boldsymbol{v}_d] \in \mathbb{R}^{d \times (d-m)},$$
$$\boldsymbol{\Lambda} = \text{diag}[\lambda_{m+1}, \lambda_{m+2}, \ldots, \lambda_d] \in \mathbb{R}^{(d-m)\times(d-m)}$$

and $\boldsymbol{I}$ is the $(d - m) \times (d - m)$ identity matrix.

Proof. Define $\tilde{\boldsymbol{g}} = \boldsymbol{V}^T\boldsymbol{g}$ and $\tilde{\boldsymbol{H}} = \boldsymbol{V}^T\boldsymbol{H}\boldsymbol{V}$. By Theorem 4.1 of [23], a vector $\tilde{\boldsymbol{s}}^* \in \mathbb{R}^{d-m}$ is a solution to the problem

$$\max_{\tilde{\boldsymbol{s}}} \tilde{Q}(\tilde{\boldsymbol{s}}) \quad \text{s.t. } \|\tilde{\boldsymbol{s}}\| \leq \Delta, \tag{50}$$

where

$$\tilde{Q}(\tilde{\boldsymbol{s}}) = \tilde{\boldsymbol{g}}^T\tilde{\boldsymbol{s}} + \frac{1}{2}\tilde{\boldsymbol{s}}^T\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}}$$

if

$$(\tilde{\boldsymbol{H}} - \kappa \boldsymbol{I})\tilde{\boldsymbol{s}}^* = -\tilde{\boldsymbol{g}}, \tag{51}$$

$$\kappa(\Delta - \|\tilde{\boldsymbol{s}}^*\|) = 0, \tag{52}$$

$$\tilde{\boldsymbol{H}} - \kappa \boldsymbol{I} \quad \text{is negative semidefinite} \tag{53}$$

for some $\kappa \geq 0$.

Assume then that $\boldsymbol{s}^* \in \mathbb{R}^d$ is in the feasible set of problem (46), that is, $\|\boldsymbol{s}^*\| \leq \Delta$ and $\boldsymbol{s}^* \in \text{span}(\boldsymbol{v}_{m+1}, \boldsymbol{v}_{m+2}, \ldots, \boldsymbol{v}_d)$ and that conditions (47)–(49) are satisfied with some $\kappa \geq 0$. Let $\tilde{\boldsymbol{s}}^* = \boldsymbol{V}^T\boldsymbol{s}$. By Lemma A.1 we have $\tilde{\boldsymbol{H}} = \boldsymbol{\Lambda}$. Thus, by premultiplying condition (47) by $\boldsymbol{V}^T$ we obtain $(\tilde{\boldsymbol{H}} - \kappa \boldsymbol{I})\boldsymbol{V}^T\boldsymbol{s}^* = -\boldsymbol{V}^T\boldsymbol{g}$. By the definitions of $\tilde{\boldsymbol{s}}^*$ and $\tilde{\boldsymbol{g}}$, this is equivalent to condition (51). In addition, since by Lemma A.1 we have $\|\boldsymbol{s}^*\| = \|\tilde{\boldsymbol{s}}^*\|$, condition (48) implies condition (52) and the condition that $\|\tilde{\boldsymbol{s}}\| \leq \Delta$. Finally, since the matrices $\boldsymbol{V}(\boldsymbol{\Lambda} - \kappa \boldsymbol{I})\boldsymbol{V}^T$ and $\boldsymbol{\Lambda} - \kappa \boldsymbol{I}$ have the same nonzero eigenvalues and $\boldsymbol{\Lambda} = \tilde{\boldsymbol{H}}$ by Lemma A.1, condition (49) implies condition (53). Hence, $\tilde{\boldsymbol{s}}^*$ is a solution to problem (50).

By Lemma A.1 we have $Q(\boldsymbol{s}) = \tilde{Q}(\tilde{\boldsymbol{s}})$. Assume then that there exists $\boldsymbol{u}^* \in \mathbb{R}^d$ that is in the feasible set of problem (46) such that $Q(\boldsymbol{u}^*) > Q(\boldsymbol{s}^*)$. By Lemma A.1, this implies that $Q(\boldsymbol{u}^*) = \tilde{Q}(\tilde{\boldsymbol{u}}^*) > \tilde{Q}(\tilde{\boldsymbol{s}}^*) = Q(\boldsymbol{s}^*)$, where $\tilde{\boldsymbol{u}}^* = \boldsymbol{V}^T\boldsymbol{u}^*$. This leads to a contradiction with the above fact that $\tilde{\boldsymbol{s}}^*$ is a solution to problem (50), and thus $\boldsymbol{s}^*$ is a solution to problem (46). □

Lemma 3.3 At each iteration of Algorithm 1, there exists a constant $\kappa_k \geq 0$ such that the conditions

$$[\tilde{\nabla}^2 \hat{p}(\boldsymbol{x}_k) - \kappa_k \boldsymbol{I}]\tilde{\boldsymbol{s}}_k = -\tilde{\nabla}\hat{p}(\boldsymbol{x}_k), \tag{54}$$

$$\kappa_k(\Delta_k - \|\tilde{\boldsymbol{s}}_k\|) = 0, \tag{55}$$

$$\tilde{\nabla}^2 \hat{p}(\boldsymbol{x}_k) - \kappa_k \boldsymbol{I} \quad \text{is negative semidefinite} \tag{56}$$

are satisfied. At each iteration, the conditions

$$\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) + \hat{p}(\boldsymbol{x}_k) = Q_k(\boldsymbol{s}_k), \tag{57}$$

$$\|\tilde{\boldsymbol{s}}_k\| = \|\boldsymbol{s}_k\|, \tag{58}$$

$$\boldsymbol{s}_k = \boldsymbol{V}_k \tilde{\boldsymbol{s}}_k \tag{59}$$

also hold.

Proof. The steps $\boldsymbol{s}_k$ and parameters κ_k at each iteration of Algorithm 1 are chosen as $\boldsymbol{s}_k = \boldsymbol{s}^*$ and $\kappa_k = \kappa^*$ such that conditions (15)–(17) are satisfied. The vector $\boldsymbol{g}$ and matrix $\boldsymbol{H}$ of problem (14) correspond to the gradient and Hessian of problem (9), and hence the matrices $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ of conditions (15)–(17) correspond to the matrices

$$\boldsymbol{V}_k = [\boldsymbol{v}_{m+1}(\boldsymbol{x}_k), \boldsymbol{v}_{m+2}(\boldsymbol{x}_k), \ldots, \boldsymbol{v}_d(\boldsymbol{x}_k)] \in \mathbb{R}^{d\times(d-m)}$$

and

$$\boldsymbol{\Lambda}_k = \operatorname{diag}[\lambda_{m+1}(\boldsymbol{x}_k), \lambda_{m+2}(\boldsymbol{x}_k), \ldots, \lambda_d(\boldsymbol{x}_k)] \in \mathbb{R}^{(d-m)\times(d-m)},$$

respectively, where $\{\boldsymbol{v}_i(\boldsymbol{x}_k)\}_{i=m+1}^d$ denote the eigenvectors of $\nabla^2\hat{p}(\boldsymbol{x}_k)$ corresponding to the eigenvalues $\lambda_{m+1}(\boldsymbol{x}_k) \geq \lambda_{m+2}(\boldsymbol{x}_k) \geq \cdots \geq \lambda_d(\boldsymbol{x}_k)$.

Thus, by condition (15) for all $k \geq 0$ we have

$$\boldsymbol{V}_k[\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I}]\boldsymbol{V}_k^T \boldsymbol{s}_k = -\boldsymbol{V}_k \boldsymbol{V}_k^T \nabla\hat{p}(\boldsymbol{x}_k). \tag{60}$$

Premultiplying equation (60) by $\boldsymbol{V}_k^T$ and using the definitions $\tilde{\boldsymbol{s}}_k = \boldsymbol{V}_k^T \boldsymbol{s}_k$ and $\tilde{\nabla}\hat{p}(\boldsymbol{x}_k) = \boldsymbol{V}_k^T \nabla\hat{p}(\boldsymbol{x}_k)$ yields

$$(\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I})\tilde{\boldsymbol{s}}_k = -\tilde{\nabla}\hat{p}(\boldsymbol{x}_k).$$

By the definition $\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) = \boldsymbol{V}_k^T \nabla^2\hat{p}(\boldsymbol{x}_k)\boldsymbol{V}_k$ we can apply Lemma A.1 from which we obtain that $\boldsymbol{\Lambda}_k = \tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)$. Thus, condition (54) holds for all k.

By condition (17), for all k we have that $\boldsymbol{V}_k(\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I})\boldsymbol{V}_k^T$ is negative semidefinite. Since this matrix and the matrix $\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I}$ have the same nonzero eigenvalues, by the definition of $\tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k)$ and Lemma A.1 this is equivalent to the condition that $\boldsymbol{\Lambda}_k - \kappa_k \boldsymbol{I} = \tilde{\nabla}^2\hat{p}(\boldsymbol{x}_k) - \kappa_k \boldsymbol{I}$ is negative semidefinite, which shows that condition (56) holds for all k.

Since $\boldsymbol{s}_k \in \operatorname{span}(\boldsymbol{v}_{m+1}(\boldsymbol{x}_k), \boldsymbol{v}_{m+2}(\boldsymbol{x}_k), \ldots, \boldsymbol{v}_d(\boldsymbol{x}_k))$ for all k, by Lemma A.1 we have $\tilde{Q}_k(\tilde{\boldsymbol{s}}_k) + \hat{p}(\boldsymbol{x}_k) = Q_k(\boldsymbol{s}_k)$, where $\tilde{Q}_k(\tilde{\boldsymbol{s}}_k)$ and $Q_k(\boldsymbol{s}_k)$ are defined according to equations (23) and (9), respectively. Thus, condition (57) holds for all k. By Lemma A.1, conditions (58) and (59) also hold for all k. Finally, by condition (16), we have $\kappa_k(\Delta_k - \|\boldsymbol{s}_k\|) = 0$ for all k. By condition (58), this implies that condition (55) holds. □

B Continuity of Eigenvalues and Eigenvectors

In this appendix we recall the classical results about continuity of eigenvalues and eigenvectors of a matrix. Theorems 3.2 and 3.4 follow from these results.

Theorem B.1 ([24], Theorem 3.1.2)**.** *The roots of the eigenvalue equation*

$$\det(\boldsymbol{M} - \lambda \boldsymbol{I}) = 0 \tag{61}$$

with respect to λ are continuous functions of the elements of the matrix $\boldsymbol{M} \in \mathbb{C}^{d\times d}$. That is, there exist continuous functions $\{\lambda_i\}_{i=1}^d : \mathbb{C}^{d\times d} \to \mathbb{C}$ satisfying equation (61) for all $\boldsymbol{M} \in \mathbb{C}^{d\times d}$.

It follows directly from Theorem B.1 that the eigenvalues of the Hessian of the C^∞-function $\hat{p}(\boldsymbol{x})$ are continuous with respect to $\boldsymbol{x}$ due to continuity of the Hessian. This is due to the fact that the functions $\{\lambda_i \circ \nabla^2 \hat{p}\}_{i=1}^d$, where the functions $\{\lambda_i\}_{i=1}^d$ are defined as in Theorem B.1, are continuous. Consequently, the eigenvalues sorted in descending order are also continuous.

Theorem 3.2 If $\hat{p} : \mathbb{R}^d \to \mathbb{R}$ is a Gaussian log-kernel density estimate, then there exist continuous functions $\{\lambda_i\}_{i=1}^d : \mathbb{R}^d \to \mathbb{R}$ representing the eigenvalues of $\nabla^2 \hat{p}$ such that $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x}) \geq \cdots \geq \lambda_d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

It is shown in [24] that if a matrix has a simple eigenvalue (i.e. an eigenvalue λ that is a simple root of equation (61)), then the eigenvector corresponding to this eigenvalue is locally continuous with respect to the elements of the matrix.

Theorem B.2 ([24], Theorem 3.1.3)**.** *Assume that $\lambda_0 \in \mathbb{C}$ is a simple eigenvalue of a matrix $\boldsymbol{M}_0 \in \mathbb{C}^{d\times d}$, and $\boldsymbol{v}_0 \neq \boldsymbol{0}$ is the corresponding eigenvector. Then there exist functions $\lambda : \mathbb{C}^{d\times d} \to \mathbb{C}$ and $\boldsymbol{v} : U \to \mathbb{C}^d$ defined in some neighbourhood U of $\boldsymbol{M}_0$ such that*

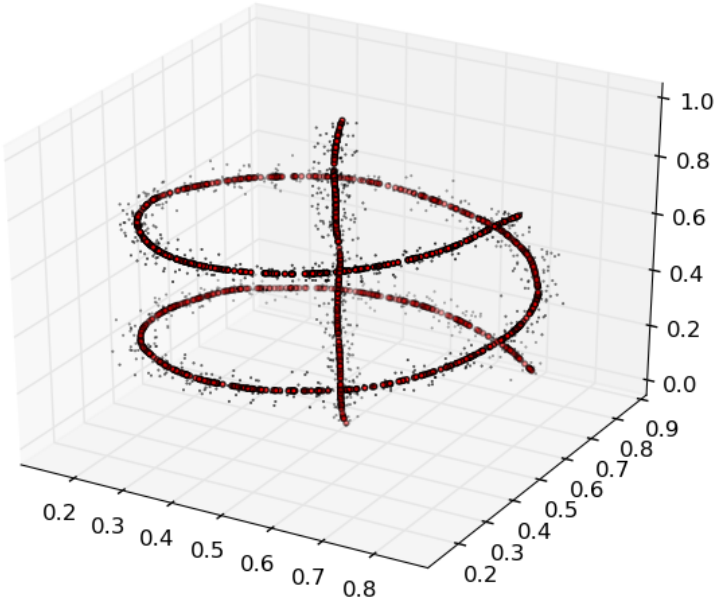
$$\boldsymbol{M}\boldsymbol{v}(\boldsymbol{M}) = \lambda(\boldsymbol{M})\boldsymbol{v}(\boldsymbol{M})$$

for all $\boldsymbol{M} \in U$ and λ and $\boldsymbol{v}$ are continuous at $\boldsymbol{M}_0$.

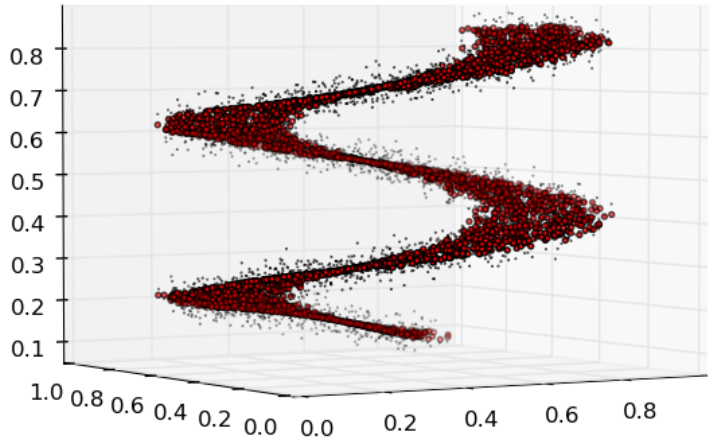
In analogy with Theorem 3.2, continuity of eigenvectors with respect to matrix elements implies continuity of eigenvectors of the Hessian $\nabla^2 \hat{p}$ with respect to $\boldsymbol{x}$. Under Assumption 3.1, continuity of the eigenvectors of $\nabla^2 \hat{p}$ corresponding to the m greatest eigenvalues in the neighbourhood U of Assumption 3.1 is thus guaranteed.

Theorem 3.4 If $m > 0$, then for any open neighbourhood U of $\boldsymbol{x}_0$ satisfying Assumption 3.1, there exists a set of continuous eigenvectors $\{\boldsymbol{v}_i\}_{i=1}^m : U \to \mathbb{R}^d$ of $\nabla^2 \hat{p}$ corresponding to the eigenvalues $\{\lambda_i\}_{i=1}^m$ defined as in Theorem 3.2.

C Projections of the Test Datasets

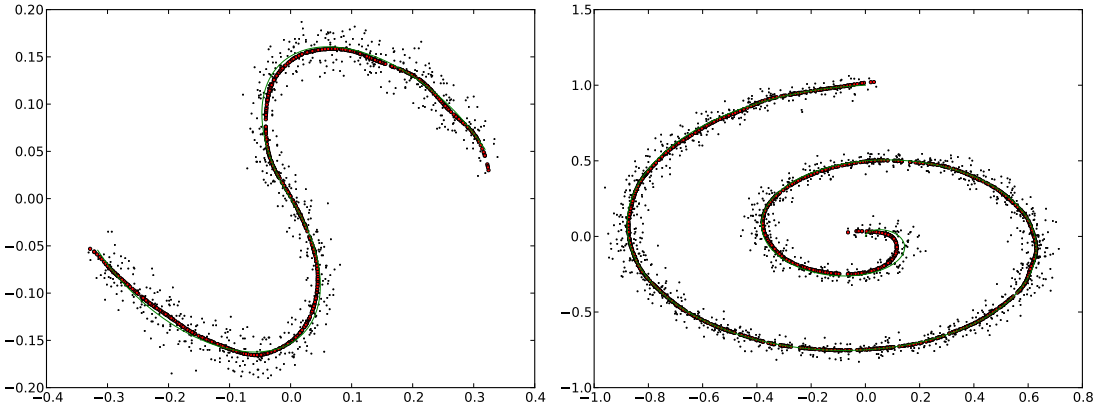


(a) Spiral3d

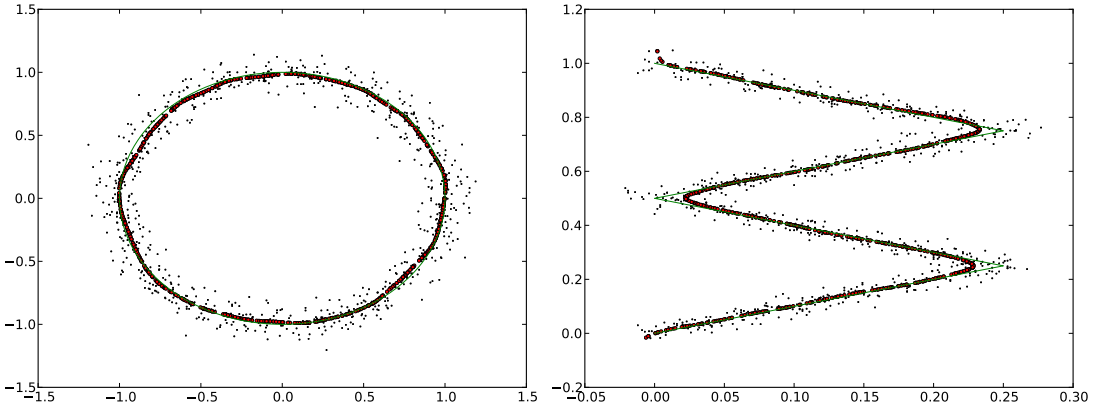


(b) Helix

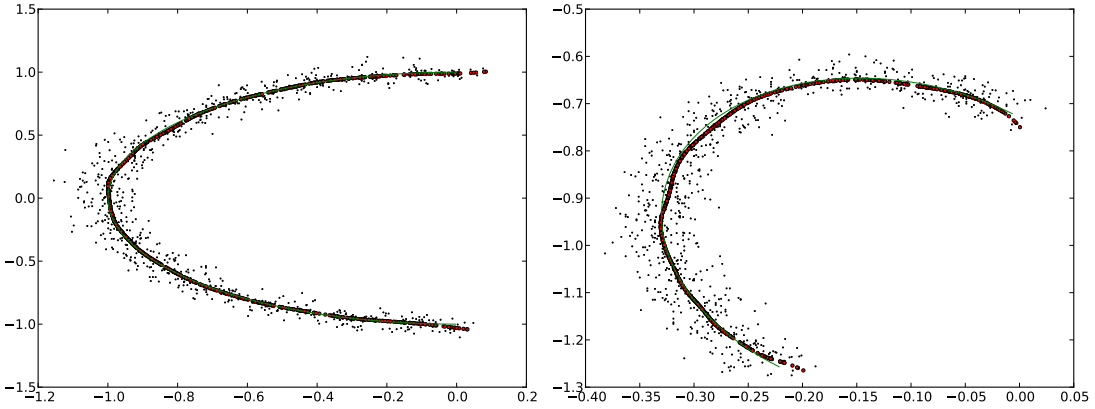
Figure 5: Kernel density ridge projections of the Spiral3d and Helix datasets.



(a) DistortedSShape (b) Spiral



(c) Circle (d) Zigzag



(e) HalfCircle (f) DistortedHalfCircle

Figure 6: Generating curves and kernel density ridge projections of two-dimensional datasets listed in Table 1.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

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

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978-952-12-2804-9
ISSN 1239-1891