Anomaly detection with kernel density estimation on manifolds

Fan Cheng
Monash University
Email: Fan.Cheng@monash.edu

Anastasios Panagiotelis
University of Sydney
Email: Anastasios.Panagiotelis@sydney.edu.au

Rob J Hyndman
Monash University
Email: Rob.Hyndman@monash.edu

14 June 2022

Anomaly detection with kernel density estimation on manifolds

Abstract

Manifold learning can be used to obtain a low-dimensional representation of the underlying manifold given the high-dimensional data. However, kernel density estimates of the low-dimensional embedding with a fixed bandwidth fail to account for the way manifold learning algorithms distort the geometry of the underlying Riemannian manifold. We propose a novel kernel density estimator for any manifold learning embedding by introducing the estimated Riemannian metric of the manifold as the variable bandwidth matrix for each point. The geometric information of the manifold guarantees a more accurate density estimation of the true manifold, which subsequently could be used for anomaly detection. To compare our proposed estimator with a fixed-bandwidth kernel density estimator, we run two simulations with 2-D metadata mapped into a 3-D swiss roll or twin peaks shape and a 5-D semi-hypersphere mapped in a 100-D space, and demonstrate that the proposed estimator could improve the density estimates given a good manifold learning embedding and has higher rank correlations between the true and estimated manifold density. A shiny app in R is also developed for various simulation scenarios. The proposed method is applied to density estimation in statistical manifolds of electricity usage with the Irish smart meter data. This demonstrates our estimator's capability to fix the distortion of the manifold geometry and to be further used for anomaly detection in high-dimensional data.

Keywords: manifold learning, variable bandwidth, Riemannian metric, highest density region, Gaussian kernels

1 Introduction

Anomaly detection has been an important and diverse area where anomalies or outliers are detected in a given data. It often involves a data analysis process to uncover the unusual patterns and has been widely applied to machine learning (Omar, Ngadi & Jebur 2013), network intrusions identification (Ahmed, Naser Mahmood & Hu 2016; Bhuyan, Bhattacharyya & Kalita 2013), medical imaging (Fernando et al. 2022), fraudulent attacks (Ahmed, Mahmood & Islam 2016), cyber-security (Ten, Hong & Liu 2011), energy consumption (Cheng, Hyndman & Panagiotelis 2021), the last of which explored detecting households with anomalous electricity usage distributions instead of raw data. In the case of non-Euclidean sample space, the observations lie on a Riemannian manifold embedded in a very high-dimensional ambient space, which makes it computationally expensive or impossible to detect anomalies. To address this problem, we propose a kernel density estimator of the low-dimensional statistical manifold embedding and find outliers as the data with the lowest density estimates.

Kernel density estimation [KDE; Parzen (1962); Chen (2017)] is one of the most popular methods to calculate the probability density function from a sample dataset. KDE is flexible to learn the shape of the underlying density from the data controlled by the bandwidth and the selection of bandwidth is crucial in KDE (Jones 1990; Terrell & Scott 1992). There is extensive research on bandwidth selection, two main categories of which are cross-validation (Jones & R. F. Kappenman 1992; Sain, Baggerly & Scott 1994) and plug-in methods (Wand, Jones, et al. 1994; Duong & Hazelton 2003). For multivariate data, variable kernel density estimator [VKDE; Jones (1990); Terrell & Scott (1992)] has been studied with an adaptive bandwidth matrix to control the amount of smoothing on the location of the estimated point [balloon estimator; Terrell & Scott (1992)] or the data point [sample smoothing estimator; Terrell & Scott (1992)]. However, these density estimators are based on random samples in the Euclidean space.

For samples points lying on a manifold with the differentiable structure called the Riemannian manifold, Pelletier (2005) proposed a kernel density estimator based on the kernel weight depending on the distance between the estimated points and the sample data points. The idea of the estimator is to use a strictly positive function of the geodesic distance on the manifold and then normalize it with the volume density function of the Riemannian manifold for curvature. (Henry & Rodriguez 2009a) In many application scenarios, we tend to find that the sample points on a manifold are embedded in a much higher-dimensional space, and the kernel density estimator from Pelletier (2005) is not directly applicable. This is when we introduce manifold learning to reduce the input data dimension. For these high-dimensional data set, various manifold learning algorithms including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP (see details of these algorithms in Cheng, Hyndman &

Panagiotelis (2021)), are applied to get a low-dimensional embedding. We propose a kernel density estimator to be applied to the low-dimensional embedding and define outliers as those with the lowest densities.

In manifold learning, the underlying idea is that the data lies on a low-dimensional smooth manifold that is embedded in a high-dimensional space. One of the fundamental objectives of manifold learning is to explore the geometry of the dataset, including the distances between points and volumes of regions of data. These intrinsic geometric attributes of the data, such as distances, angles, and areas, however, can be distorted in the low-dimensional embedding, leading to failure to recover the geometry of the manifold (Goldberg et al. 2008). To tackle this problem and measure the distortion incurred in manifold learning, Perrault-Joncas & Meila (2013) propose the Learn metric algorithm to augment any existing embedding output with geometric information in the Riemannian metric of the manifold itself. By applying this algorithm, the outputs of different manifold learning methods can be unified and compared under the same framework, which would highly benefit in improving the effectiveness of the embedding.

The Riemannian metric using the method of Perrault-Joncas & Meila (2013) gives some idea of the distortion of an embedding. Mapping the points through a non-linear function "stretches" some regions of space and "shrinks" others. The Riemannian gives us an idea of the direction and angle of this stretching, which is informative for learning the manifold. In variable kernel density estimate, the bandwidth matrix H is also defined to control the amount of smoothing for each data point. Therefore, we propose to the use Riemannian metric as the bandwidth matrix in, we could further get the kernel density estimation of the manifold M. This kernel density estimate can then be used to produce the highest density region plots(Hyndman 1996) for outlier visualization.

The rest of the paper is organized as follows. In Section 2, we present the proposed algorithm to detect anomalies based on variable kernel density estimates of manifold embeddings. In this section, we provide justification for the use of the Riemannian metric as the bandwidth of variable kernel density estimation, including the comparison with fixed bandwidth. Section 4 is composed of two simulations with the proposed algorithm; the first deals with 2-dimensional meta data mapped into a 3-D swiss roll or twin peaks data and the second with a 5-D semi-hypersphere mapped in a 100-D space. Section 5 contains the application to visualize and identify anomalies in the Irish smart meter dataset. Conclusions and discussions are presented in Section 6. Readers interested in the notions of Riemannian geometry mentioned in this paper could use the Appendix in Appendix A as a reference.

2 Kernel density estimation on manifolds

In this section, we introduce the proposed method to detect anomalies based on the kernel density estimates of manifold learning embeddings where the Riemannian matrix is used as the pointwise variable bandwidth to measure the direction and angle of the distortion of the low-dimensional embeddings. Perrault-Joncas & Meila (2013) gives us an idea of how to measure the direction and angle of the distortion using the Riemannian metric which is a positive definite square matrix for each data point. To learn the distribution of the low-dimensional embedding, we use the kernel density estimation with the bandwidth matrix being the Riemannian metric. The outliers could then be defined as the points with the lowest density estimates. The proposed schematic is shown in Figure 1. The highlighted two steps in Figure 1 are the main contributions of this main chapter, replacing the bandwidth matrix $\boldsymbol{H}_i$ with the Riemannian metric g_i for each point in variable kernel density estimate, and computing the highest density region plots based on the density estimates, $\hat{f}(\boldsymbol{x})$, for anomaly detection.

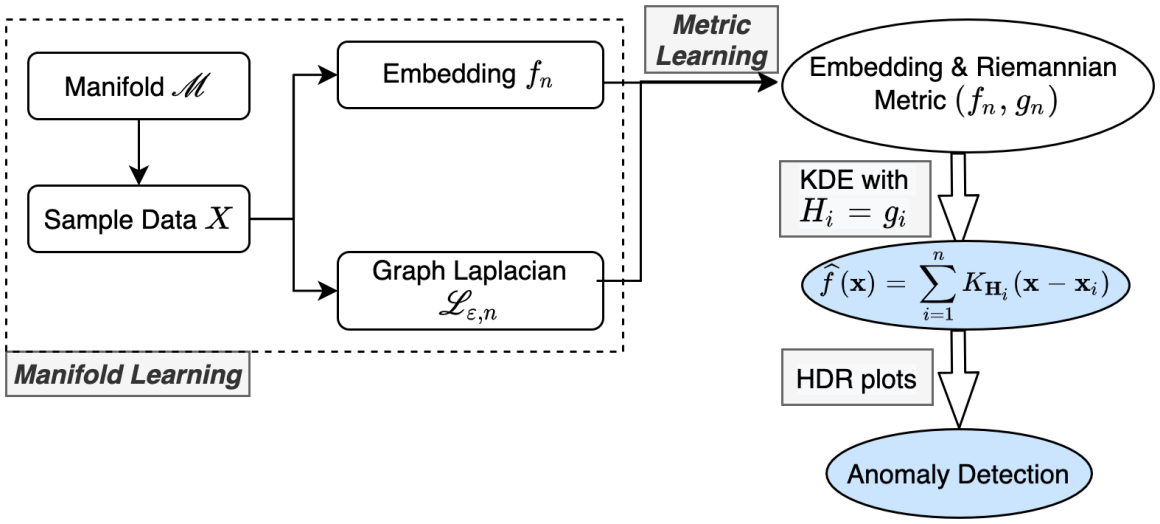

Figure 1: *The proposed schematic for variable kernel density estimation with recovered geometry.*

To start with, we introduce the notations used in this manuscript. Then we introduce the multivariate kernel density estimation method with variable bandwidth matrix and the metric learning algorithm to derive the pointwise Riemannian metric. Readers familiar with these topics could skip the corresponding subsections. Finally, we propose our novel algorithm to estimate densities and detect anomalies for high-dimensional data set.

2.1 Multivariate kernel density estimation

Multivariate kernel density estimation has gained lots of attention in exploratory data analysis. It is a non-parametric technique to estimate the density of the data based on weighted kernels centered at the data. Scott (1992) and Scott (2015) covered a wide range of techniques for estimating the kernel

densities of multivariate data. In general, given a multivariate random sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N \in \mathcal{R}^d$ drawn from a density f, the kernel density estimate of f at point $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)' \in \mathcal{R}^d$ is given by

$$\hat{f}(\boldsymbol{x}; \boldsymbol{H}) = \frac{1}{N} \sum_{i=1}^{N} K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{X}_i), \tag{1}$$

where $\boldsymbol{H}$ is called the bandwidth matrix and it is a $d \times d$ symmetric positive definite matrix, $K(\boldsymbol{x})$ is the kernel function, and $K_{\boldsymbol{H}}(\boldsymbol{x}) = |\boldsymbol{H}|^{-1/2} K(\boldsymbol{H}^{-1/2}\boldsymbol{x})$. When using the standard gaussian kernel, we have $K(\boldsymbol{x}) = (2\pi)^{-d/2} \exp(-\frac{1}{2}\boldsymbol{x}'\boldsymbol{x})$ and hence

$$K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{X}_i) = (2\pi)^{-d/2} |\boldsymbol{H}|^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{X}_i)' \boldsymbol{H}^{-1} (\boldsymbol{x} - \boldsymbol{X}_i)\right],$$

with $|\boldsymbol{H}|$ being the determinant of matrix $\boldsymbol{H}$. Intuitively, KDE smooths out each data point $\boldsymbol{X}_i$ into a smooth bump with the shape determined by the kernel function. Then an average over all data points is taken as the estimated density $\hat{f}$. KDE is a local smooth estimator because the nearby points contributes more to the density value compared to points that are far away from the estimated data point $\boldsymbol{x}$.

The choice of the the bandwidth matrix $\boldsymbol{H}$ is crucial in the accuracy of $\hat{f}$ because it controls the smoothing across the sample data points. For a given kernel function $K(\boldsymbol{x})$, the shape of the local smoothing at each point is determined. Then the performance of the density estimate $\hat{f}$ depends on the choice of $\boldsymbol{H}$. When $\boldsymbol{H}$ is too small, the density estimates will apprear to be wiggly. On the contrary, when $\boldsymbol{H}$ is too large, the density estimates might smooth out some local feathers of f. So the accuracy of $\hat{f}$ depends strongly on the bandwidth matrix. Many bandwidth selection methods has been proposed in the literature, including the rule-of-thumb, cross-validation and plug-in methods (See Heidenreich, Schindler & Sperlich 2013; Scott 2015, for details). For univariate kernel density estimation, the bandwidth selection problem has been thoroughly investigated (See Jones, Marron & Sheather 1992; Cao, Cuevas & González Manteiga 1994; Jones, Marron & Sheather 1996; Wand & Jones 1994, for reviews). The generalization to multivariate case could mostly be found in Duong & Hazelton (2003), Duong (2004), and Chacón & Duong (2010). In this paper, we consider the multivariate kernel density estimation.

Note that the bandwidth matrix in (1) is a global smoothing parameter for all data points. However, when the local data structure is not universal for all sample data, an adaptive bandwidth matrix that is varying rather than fixed at each data point is needed. This is ususally referred to as variable kernel density estimation [VKDE; Breiman, Meisel & Purcell (1977); Jones (1990); Terrell & Scott (1992)], where the bandwidth is varied depending on either the location of the sample points or that of the estimated points (Section 6.6 of Scott 2015). In this paper, the densities are estimated at

the sample points themselves, so we only need to consider the case where the bandwidth changes for each sample point $\boldsymbol{X}_i$ and will refer to this as the variable/adaptive kernel density estimation unless otherwise stated.

In multivariate VKDE, the estimator in Equation (2) is similar to that in (1) except that the bandwidth is a variable matrix $\boldsymbol{H}_i$ for each $\boldsymbol{X}_i$,

$$\hat{f}(\boldsymbol{x};\boldsymbol{H}_i) = \frac{1}{N}\sum_{i=1}^{N} K_{\boldsymbol{H}_i}(\boldsymbol{x}-\boldsymbol{X}_i), \tag{2}$$

Compared to a global fixed $\boldsymbol{H}$, a variable bandwidth matrix is able to account for the pointwise adaptive local structure at each point and generate an estimate more applicable to the case where the distortion at each point is different at each point in the manifold learning embedding.

For computational purposes, Duong (2007) implements the kernel density estimates for data from 1 to 6 dimensions in the R package *ks*. However, for variable kernel density estimation where the bandwidth is no longer a constant matrix, the *ks* implementation is limited to 2-dimension and the bandwidth can either be a diagonal matrix or a vector with the diagonal elements, which limits the full flexibility of the kernel function. More importantly, the estimator in Equation (2) is only applicable when the sample space is Euclidean. In the cases of a Riemannian manifold, the underlying density is supported on a single manifold. The fixed bandwidth matrix assumes that the smoothing is the same across the entire manifold and there is no local distortion at each sample point. Therefore, there is a need for an adaptive density estimator for Riemannian manifolds as described in Section 2.2.

2.2 Kernel density estimator on Riemannian manifolds

Consider a compact Riemannian manifold (M, g) of dimension d without boundary and a probability distribution with density f on the manifold. Assume (M, d_g) is a complete metric space, where d_g is the Riemannian distance induced by the Riemannian, g, and the strictly positive injectivity radius (Chavel 2006) of the manifold, $inj_g(M)$. Denote $\mathbf{X}_i, i = 1, 2, \ldots, N$, where $\mathbf{X}_i \in \mathcal{R}^d$, as i.i.d. random objects on M with density f. For each point $p \in M$, Pelletier (2005) proposed a kernel density estimator of f to be

$$\hat{f}(p) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{h^d \theta_{\mathbf{X}_i}(p)} K\left(\frac{d_g(p, \mathbf{X}_i}{h}\right), \tag{3}$$

where h is the global bandwidth satisfying $h \leq h_0$ for fixed h_0 such that $0 \leq h_0 \leq inj_g(M)$, $K : \mathcal{R}_+ \to \mathcal{R}$ is a non-negative map, and $\theta_q(p)$ is the volume density function on the manifold. Pelletier (2005) mention that the volume density function $\theta_q(p)$ defined for p in a neighborhood

of q on the manifold, with the geodesic normal coordinates at q, is equal to the determinant of the Riemannian metric g at the logarithm map at q, $\exp_q^{-1}(p)$. The expression in (3) is also proved to be consistent with the kernel density estimators in the Euclidean case as

$$\hat{f}(p) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^d}K(\frac{\|p-\mathbf{X}_i\|}{h}),$$

and converges at the same rate as the Euclidean kernel density estimator (Henry & Rodriguez 2009a).

2.2.1 Volume density function

However, in application, we usually found the observations lie on a d-dimensional manifold embedded in a s-dimensional space, where $d \ll s$. Then manifold learning can be applies to find the d-dimensional representation of the manifold M. Therefore, based on the estimator in (3), we propose a kernel density estimator of f with the manifold embedding $y_i, i = 1, 2, \ldots, N$ to be

$$\hat{f}(p) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i^d|g(\exp_p^{-1}(y_i))|}K(\frac{d_g(p,y_i)}{h_i}), \tag{4}$$

where $h_i, i = 1, 2, \ldots, N$ is the pointwise variable bandwidth matrix and h_i is positive definite with dimension $d \times d$.

An important goal of manifold learning is to recover the local or global features of the data. In order to recover the geometry of the manifold, Perrault-Joncas & Meila (2013) propose a method to augment any existing reasonable embedding and allow for the computation of geometric values to be calculated with an estimation of the Riemannian metric g. This could also be used to fix the geometric distortion in the embedding. It is worth noticing that the Riemannian metric from Perrault-Joncas & Meila (2013) is a $d \times d$ positive definite matrix for each data point. Therefore, we propose to take the geometric distortion into consideration when estimating the kernel densities of the embedding with Equation (4). In this way, we present a consistent kernel density estimator with pointwise variable bandwidth as the Riemannian metric.

2.2.2 Riemannian metric as variable bandwidth

The Riemannian metric g is a symmetric and positive definite tensor field which defines an inner product $<,>_g$ on the tangent space T_pM for every point $p \in M$. If the inner product of the tangent space is known for a given geometry, the Riemannian metric is a good measure to recover the geometry of the manifold. The Metric Learning algorithm (Perrault-Joncas & Meila 2013) then augment the embedded manifold with the Riemannian metric and produce a Riemannian manifold (M, g).

To recover the original geometry of the manifold, we need to know what the inner product corresponds to in the embedding. The inner product between two vectors $u, v \in T_pM$, $< u, v >_g = g_{ij}u^i v^j$[1], can be used to define some geometric quantities, such as the vector norm $\|u\| = \sqrt{< u, v >_g}$ and the angle between two vectors $\cos\theta = \frac{<u,v>_g}{\|u\|\|v\|}$ in the tangent space. Therefore, for each point $p \in M$ in any coordinate system, the Riemannian metric g is a $d \times d$ symmetric positive definite matrix, where d is the dimension of the manifold.

The line element and volume element of the full manifold or a subset of the manifold can also be computed from g. The arc length of a curve $c \in M$ is defined as

$$l(c) = \int_a^b \sqrt{g_{ij}\frac{dx^i}{dt}\frac{dx^j}{dt}}dt,$$

where $(x^1, \ldots, x^d)$ are the coordinates of chart (U, x) and $c(t)$ is a function mapping $[a, b]$ to M. While the volume of $V \subset M$ is computed by

$$Vol(V) = \int_V \sqrt{\|g\|}dx^1 \ldots dx^d.$$

Both the concepts of distance and volume are relevant to kernel density estimation.

Perrault-Joncas & Meila (2013) propose the Learn metric algorithm which mainly involves four steps: weighted neighborhood graph construction; geometric graph Laplacian calculation; manifold learning; and Riemannian metric calculation. We restate these four steps in Algorithm 1. By applying an existing manifold learning algorithm to the data $X \in \mathbb{R}^r$ with n observations, a low-dimensional embedding $f_n \in \mathbb{R}^d$ can be computed. Most manifold learning methods involve the construction of the nearest neighbor graph based on which the Laplace-Beltrami operator Δ_M is built. The Laplacian is quite useful because it can be coordinate-free while containing all the important geometry. Perrault-Joncas & Meila (2013) have stated one way to compute the approximated Δ_M with a discrete consistent estimator, the geometric graph Laplacian $\mathcal{L}_{\varepsilon,n}$ (Zhou & Belkin 2011), where ε is the radius parameter for the nearest neighbor graph. The graph Laplacian together with the embedding can be used in the Metric Learning algorithm to achieve the augmented embedding with the Riemannian metric (f_n, g_n).

As pointed out by Perrault-Joncas & Meila (2013), if the embedding dimension s is larger than the manifold intrinsic dimension d, the rank of the embedding metric $h_n(p)$ is d; otherwise, the Riemannian metric g_n will be returned. This algorithm is also implemented in a Python library *megaman* (McQueen et al. 2016). It is designed to apply the manifold learning methods to large-scale data sets, as well as compute the Riemannian metric of the manifold.

[1]Here the Einstein notation is used where superscripts denote summation over i and j

Algorithm 1: Learn metric algorithm in Perrault-Joncas & Meila 2013

Input : high-dimensional data $x_i \in \mathcal{R}^s$ for all $i = 1, \ldots, N$
Output : low-dimensional data $y_i \in \mathcal{R}^d$ and its Riemannian metric h_i for all $i = 1, \ldots, N$
parameter : embedding dimension d, bandwidth parameter $\sqrt{\varepsilon}$, manifold learning algorithm
optimization parameter: manifold learning parameters

1: Construct a weighted neighborhood graph $\mathcal{G}_{w,\varepsilon}$ with weight matrix W where $w_{i,j} = \exp(-\frac{1}{\varepsilon}\|x_i - x_j\|^2)$ for data points $x_i, x_j \in \mathbb{R}^s$;
2: Calculate the $N \times N$ geometric graph Laplacian $\widetilde{\mathcal{L}}_{\varepsilon,N}$ by

$$\widetilde{\mathcal{L}}_{\varepsilon,N} = 1/(c\varepsilon)(\widetilde{D}^{-1}\widetilde{W} - I_N),$$

where $\widetilde{D} = diag\widetilde{W}\mathbf{1}$, $\widetilde{W} = D^{-1}WD^{-1}$, and $D = diagW\mathbf{1}$;
3: Embed each data point $x_i \in \mathcal{R}^s$ to embedding coordinates $\boldsymbol{y} = (\boldsymbol{y}^1, \ldots, \boldsymbol{y}^d)'$ by any existing manifold learning algorithm;
4: Obtain the matrix $\tilde{\boldsymbol{h}}$ of all data point by applying the graph Laplacian $\widetilde{\mathcal{L}}_{\sqrt{\varepsilon},N}$ to the embedding coordinates matrix $\boldsymbol{y}$ with each element vector in $\tilde{\boldsymbol{h}}$ being

$$\tilde{\boldsymbol{h}}^{ij} = \frac{1}{2}\left[\tilde{\mathcal{L}}_{\varepsilon,N}\left(y^i \cdot \boldsymbol{y}^j\right) - \boldsymbol{y}_i \cdot \left(\tilde{\mathcal{L}}_{\varepsilon,n}\boldsymbol{y}^j\right) - \boldsymbol{y}^j \cdot \left(\tilde{\mathcal{L}}_{\varepsilon,n}\boldsymbol{y}^i\right)\right],$$

where $i, j = 1, \ldots, d$ and the $\cdot$ calculation is the elementwise product between two vectors;
5: Calculate the Riemannian metric $\boldsymbol{h}$ as the rank d pseudo inverse of $\tilde{\boldsymbol{h}}$ with

$$\boldsymbol{h} = Udiag1/(\Lambda[1:d])U',$$

where $[U, \Lambda]$ is the eigendecomposition of matrix $\tilde{\boldsymbol{h}}(x)$, and U is the matrix of column eigenvectors ordered by the eigenvalues Λ in descending order.

3 Outlier detection on manifold learning embedding

Now we present our proposed algorithm for anomaly detection based on variable kernel density estimates in 2. There are mainly four steps involved in the algorithm. The first step is to apply the Learn metric algorithm described in 1 to the input high-dimensional data $x_i, i = 1, \ldots, N$ where $x_i \in \mathcal{R}$ to get the low-dimensional embedding y_i, where $y_i \in \mathcal{R}$ and the Riemannian metric $\boldsymbol{H}_i$ with dimension $d \times d$ for each observation. Two parameters $\sqrt{\varepsilon} = 0.4$ and $c = 0.25$ are given for Gaussian kernels. Then we could use the pointwise Riemannian metric to calculate the kernel density estimate with our proposed estimator in Equation (4). The top outliers of size *n_outliers* are obtained by ordering the embedding points y_i according to their density estimates $f_N(\boldsymbol{y}_i)$.

If the embedding dimension $d = 2$, the HDR plots (Hyndman 1996) could be used to identify the relative location of outliers in the embedding and find which observations lie in the highest density region of specified coverage, eg. 1%, 50%, 99%, >99%.

Algorithm 2: Variable kernel density estimates with Riemannian metric

Input : high-dimensional data x_i for all $i = 1, \ldots, N$
Output : outliers embedding coordinates $y_1, \ldots, y_{n\_outliers}$ with their estimated densities $f_1, \ldots, f_{n\_outliers}$
parameter: number of outliers $n\_outliers$, embedding dimension d

1: For all $i = 1, \ldots, N$, compute the d-dimensional embeddings y_i with any exsiting manifold leanring algorithms and the corresponding Riemannian metric $\boldsymbol{g}_i$ using the Learn metric algorithm with inputs d and $\sqrt{\varepsilon} = 0.4$ and $c = 0.25$ for heat kernels;
2: Set the variable bandwidth for each observation as $\boldsymbol{h}_i = \boldsymbol{g}_i$;
3: Compute the kernel density estimates $f_N(\boldsymbol{y}_i)$ for all $i = 1, \ldots, N$ using Equation refequ:denestimator;
4: Reorder the embedding coordinates $\boldsymbol{y}$ according to the density estimates $f_N(y)$ and subset the top $n\_outliers$ as the outliers.

4 Simulations

In this section, we examine two scenarios for both low and high dimensions to test our proposed algorithm. For visualization purposes, Section 4.1 presents a 2-D meta data example. We first simulate the data of size $N = 2,000$ from a mixture of four Gaussian kernels with the same covariance but different means, each consisting of 500 points. Different mapping functions are then applied to the 2-D meta data to be mapped in a 3-D feature space, which gives the higher-dimensional input for different manifold learning algorithms, including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP. The embedded dimension is set as $d = 2$, the same as the meta data dimension. This enables us to compare the manifold learning embedding with the true meta data. We could now apply Algorithm 2 to get the density estimates of all data points and further detect anomalies. As a high-dimensional example, the second simulation in Section 4.2 is based on a 5-D meta data of size $N = 2,000$ embedded in a 100-D space and the corresponding embedding dimension is $d = 5$.

4.1 3-D mapping from a 2-D Gaussian Mixture Model

We first generate a 2-dimensional data of size $N = 2000$ from a Gaussian mixture model with four components with different means $\boldsymbol{\mu_1} = (0.25, 0.25)', \boldsymbol{\mu_2} = (0.25, 0.75)', \boldsymbol{\mu_3} = (0.75, 0.25)', \boldsymbol{\mu_4} = (0.75, 0.75)'$ and the same variance-covariance matrix $\boldsymbol{\Sigma}_i = diag(0.02, 0.02), i = 1, 2, 3, 4$. The mixture proportions are equally set as $\pi_i = 0.25, i = 1, 2, 3, 4$. Then the mixture Gaussian mixture density function is a weighted linear combination of the four component Gaussian densities as

$$P(\boldsymbol{X} = \boldsymbol{x}) = \sum_{i=1}^{4} \pi_i \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{-1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i})\}. \tag{5}$$

Figure 2 shows the 2-dimensional meta data and the colors indicate the true density of all data points calculated from (5), with brighter colors showing higher densities and darker colors showing lower densities. We then define outliers as points with lowest densities shown in black and typical points with highest densities shown in yellow. Based on the true density plot, the outliers are scattered in the middle and the outer area of the whole structure, while typical points are near the means of four kernels. These are *true outliers* to be compared with outliers from the kernel density estimates.

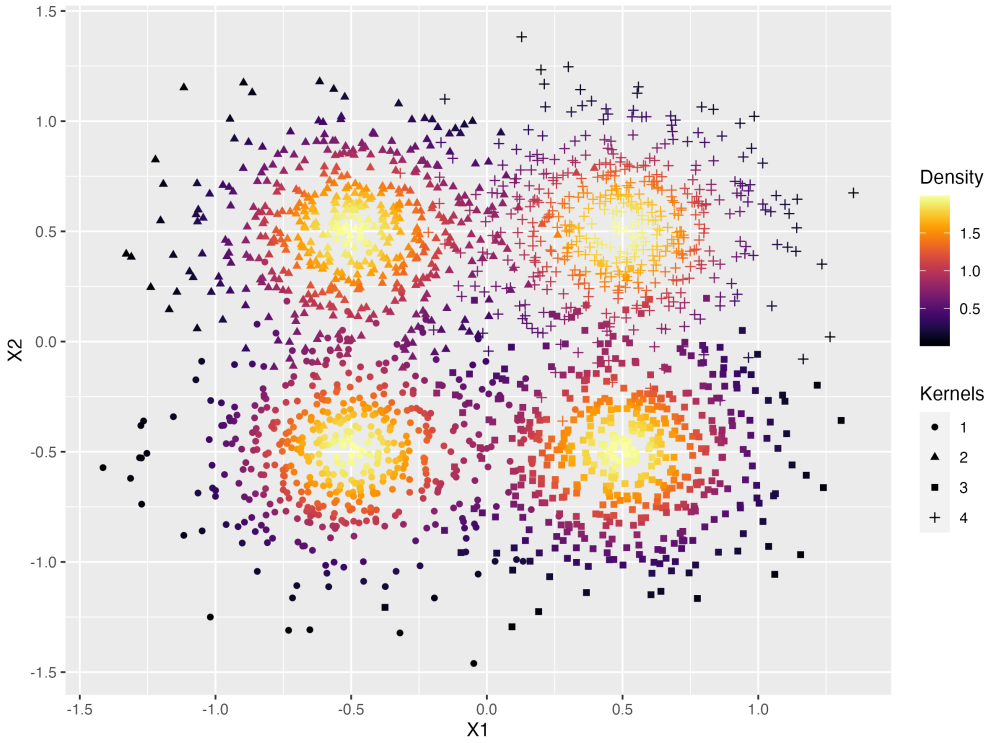

Figure 2: *True density of the Gaussian mixture model of four kenels with means* $(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)$ *and the same variance-covariance matrix* $diag(0.02, 0.02)$*. The colors indicate the density of the data and lower density points in darker colors are scattered both in the outer and center areas. The shapes indicate the four kernels.*

4.1.1 Swiss roll mapping

Given the 2-D meta data, multiple mapping functions could be applied to embed the data in a 3-D space. One of the most famous examples in manifold learning is the swiss roll data, with the mapping function in (6). The two-dimensional meta data $(\boldsymbol{X}_1, \boldsymbol{X}_2)'$ is transformed into the three-dimensional data $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})'$, shown in the left plot of Figure 3. The four colors in the mappings represent the four Gaussian kernels used to generate the meta data $(\boldsymbol{X}_1, \boldsymbol{X}_2)'$.

$$\begin{cases} X = X_1 \cos X_1, \\ Y = X_2, \\ Z = X_1 \sin X_1. \end{cases} \tag{6}$$

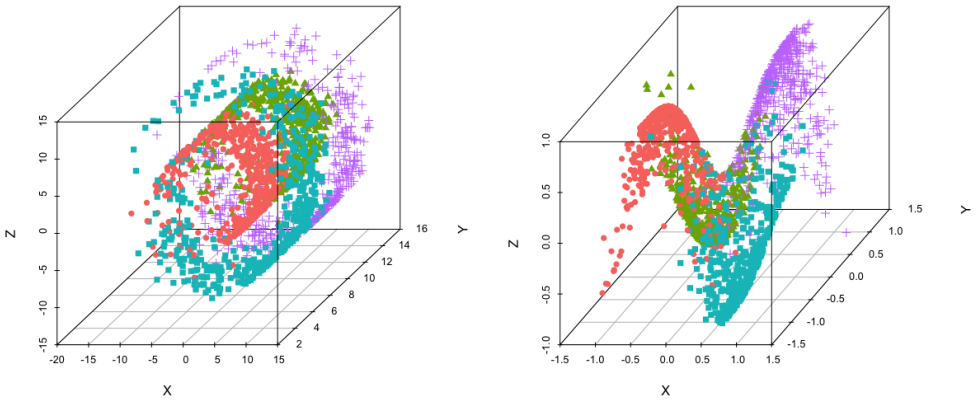

Figure 3: *3-D Mappings of the meta data with colors and shapes indicating the four kernels. Left: swiss roll mapping. Right: twin peak mapping.*

Now we are able to apply different manifold learning algorithms to $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})'$ and reduce the dimension back to $d = 2$, and further estimate the density of the 2-D embedding. According to the density estimates, we could rank the data points and then identify which observations lie in the highest density region of specified coverage, eg. 1%, 5%, 50%, 99%, >99%. For each of the five manifold learning methods, namely ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, Figure 4 presents the 2-D embedding plot in the same row, with the colors indicating the densities levels, the left column for true densities from the Gaussian mixture model, the middle column for highest density region plots with densities from our proposed variable KDE method, and the right for similar HDR plots with densities from KDE with fixed bandwidth. The top twenty outliers with the lowest densities are highlighted in black with point indexes in blue. From Figure 2 and the data generating process, we know that there are four highest density regions. However, in all manifold learning embeddings colored with true densities (left column in Figure 4), except for LLE, the number of highest density regions are not the same as the meta data. When comparing the number of HDRs for variable and fixed bandwidth (middle and right column in Figure 4), our proposed method with variable bandwidth outperforms fixed bandwidth for ISOMAP, LLE, and Laplacian Eigenmaps (top three rows in Figure 4). In terms of the top 20 outliers found rowwise, variable bandwidth could find most outliers lying on the left area of the embedding in ISOMAP and UMAP, and both methods in LLE embedding could find the outliers in the outer area, but for the other methods, both variable and fixed bandwidth are not detecting true outliers accurately. For t-SNE and UMAP embedding, the embedding structure is highly distorted and the points are clustered together in a discontinuous way, which is also shown in the clustered outliers.

To further compare the accuracy of the estimated densities for all data points, we calculate the correlation between the rank of true densities and the estimated densities and present in Table 1. It

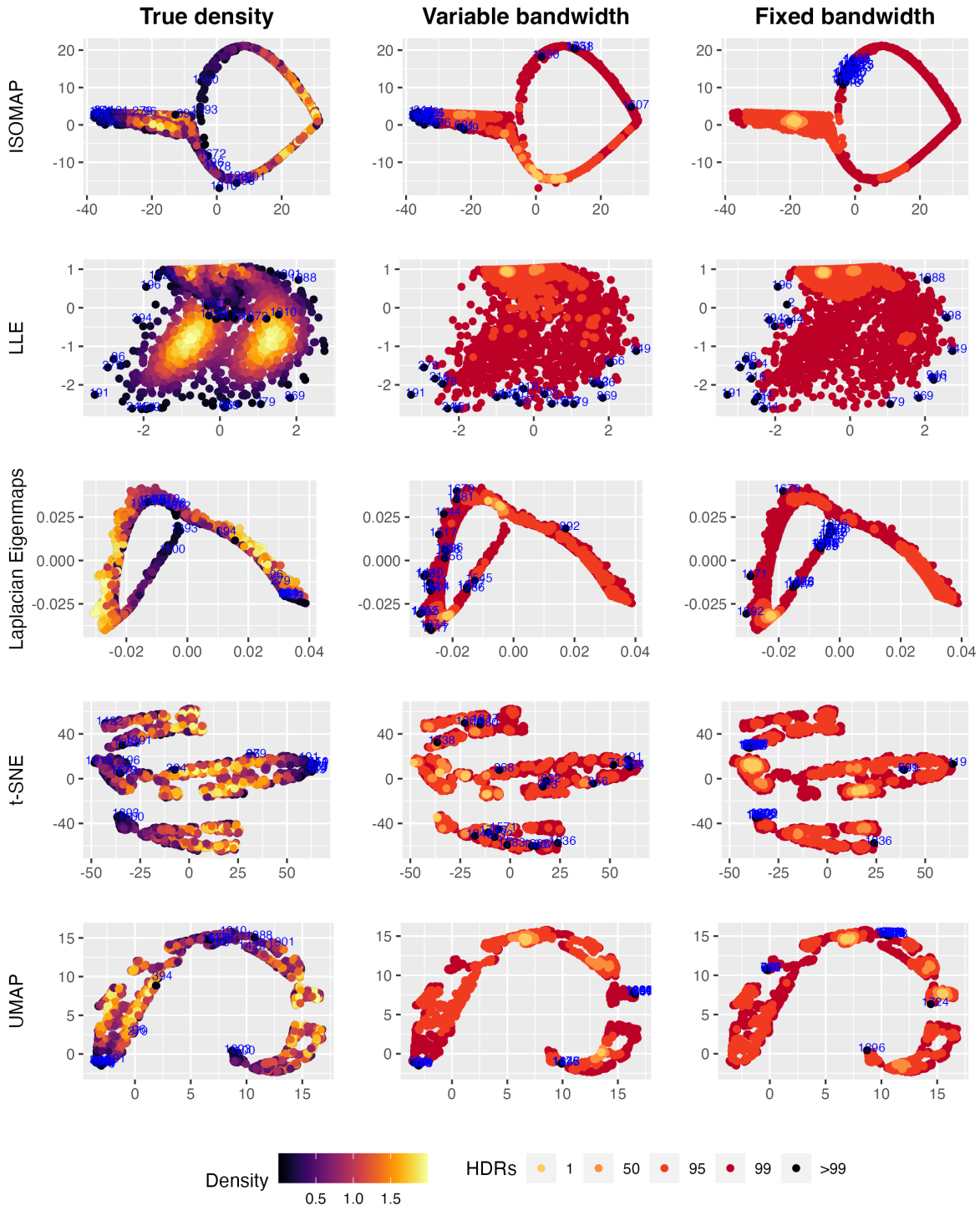

Figure 4: *Highest density region plots of five manifold learning embeddings of the swiss roll data. Colors are indicating densities from left: true densities from the Gaussian mixture model; middle: KDE with Riemannian matrix as variable bandwidth; and right: KDE with fixed bandwidth. Variable KDE performs better in finding kernel structures with ISOMAP, LLE, and Laplacian Eigenmaps, and in locating outliers with ISOMAP and LLE. The t-SNE and UMAP embeddings are highly distorted and the outliers found are clustered.*

Table 1: *Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the swiss roll data. Variable bandwidth KDE outperfoms for LLE and UMAP, and LLE gives the highest rank correlation.*

	ISOMAP	LLE	Laplacian.Eigenmaps	t.SNE	UMAP
Variable bandwidth	0.0696	**0.400**	-0.2357	0.023	**0.0138**
Fixed bandwidth	**0.2798**	0.351	**0.0141**	**0.367**	-0.0110

can be seen that the rank correlation of our proposed method with variable bandwidth is higher for LLE and UMAP, although the correlation for UMAP is very close to zero. The highest correlation comes from our method in LLE embedding, which is mainly due to it being closest to the rectangular structure of the meta data shown in Figure 2. For Laplacian Eigenmaps, our method has wrongly estimated the left area with lower densities even though their true densities are very high in yellow, leading to a negative correlation. The negative correlation would occur typically when the highest or lowest true density areas are not well estimated. As for the estimates in highly distorted embedding, including ISOMAP, t-SNE, and UMAP, the rank correlations are quite low. This shows that our proposed method could improve the kernel density estimate of manifold learning embedding by considering the distortion using the Riemannian metric. However, if the distortion is too severe, eg. ISOMAP, or when the embedding is discontinuous, eg. t-SNE and UMAP, the density estimates are not as reliable for outlier detection.

4.1.2 Twin peaks mapping

For comparison, we use the same 2-D meta data in Figure 2 with a different mapping function, twin peaks mapping in Equation (7), with the corresponding 3-D structure shown in the right plot of Figure 3.

$$\begin{cases} X = X_1, \\ Y = X_2, \\ Z = \sin(\pi X_1)\tanh(3X_2). \end{cases} \tag{7}$$

Similar to Figure 4, different manifold learning embeddings are obtained and used to detect outliers with true densities and two bandwidth selection methods shown in Figure 5. In general, the four highest density regions in yellow are identified in almost all manifold learning embeddings except for ISOMAP with fixed bandwidth and t-SNE. For ISOMAP, our proposed variable KDE, compared with the true density, gives the most accurate mixture kernel structure with the lowest estimated densities (darker colored points) in the outside and the center of the embedding, and the kernel means (yellow points) with highest densities are also clearly identified. In contrast, the fixed bandwidth KDE failed to identify the lowest density area in the center. Both variable and fixed bandwidth KDE are quite close with LLE and Laplacian Eigenmaps, but in Laplacian Eigenmaps

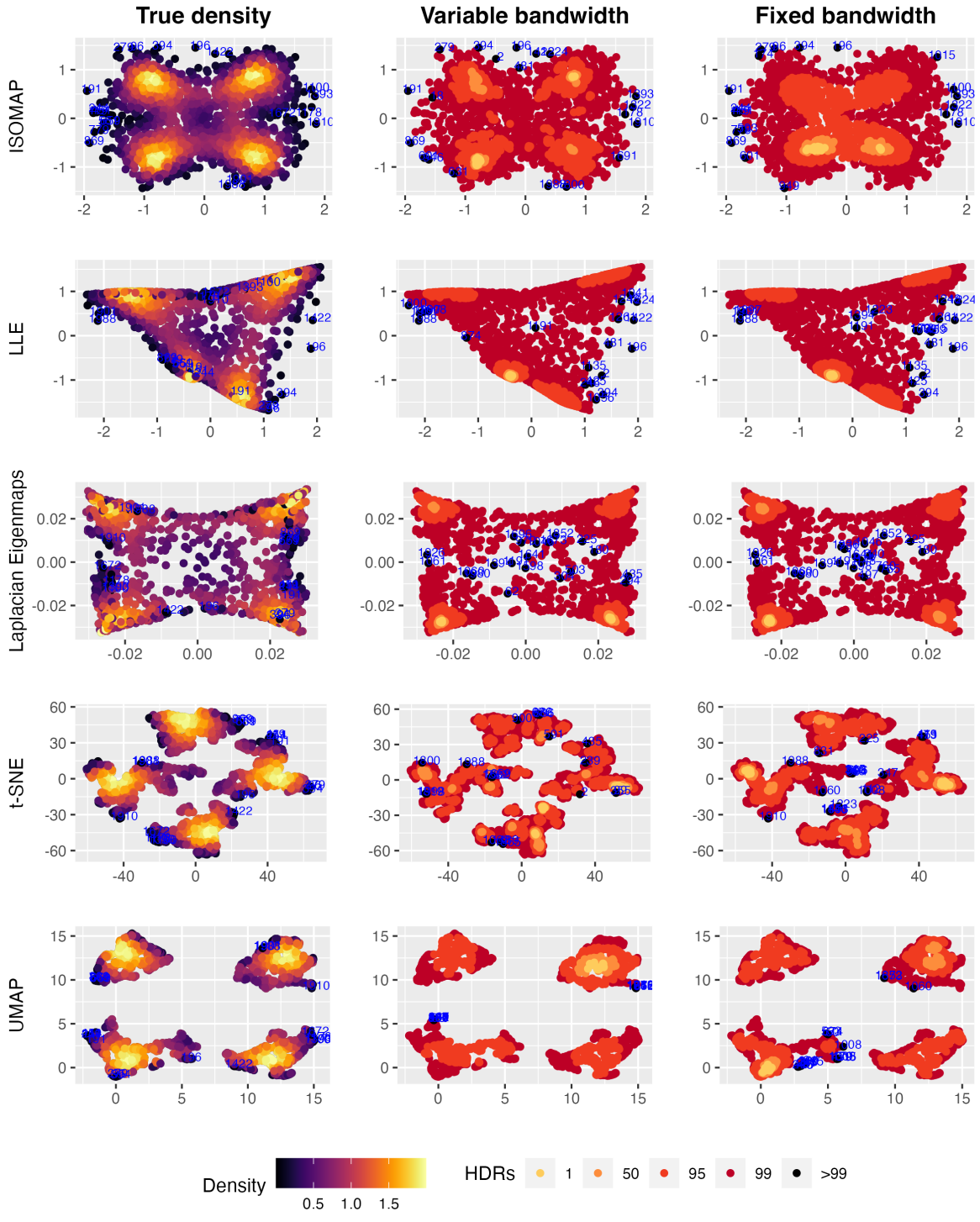

Figure 5: *Highest density region plots of four manifold learning embeddings of the twin peak data. Variable KDE preforms better in finding kernel structures with ISOMAP and LLE, and in locating outliers with t-SNE and UMAP.*

Table 2: *Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the twin peak data. Variable bandwidth KDE outperfoms for LLE and UMAP, and LLE gives the highest rank correlation.*

	ISOMAP	LLE	Laplacian.Eigenmaps	t.SNE	UMAP
Variable bandwidth	**0.899**	0.385	0.620	0.259	**0.659**
Fixed bandwidth	0.626	**0.399**	**0.622**	**0.663**	0.653

embedding, the top outliers are indexed in the middle instead of the true outer areas due to the large distortion in the middle. For t-SNE and UMAP, there are four clusters in the embedding and UMAP does a better job in finding the HDRs than t-SNE. Also due to the clusters in the embedding, the outliers found in UMAP are clustered.

We can gain further insight by comparing the correlation between ranks of true densities and estimated densities from variable and fixed bandwidth KDE by Table 2. Again the highest correlations appear from embedding with higher quality, including ISOMAP, Laplacian Eigenmaps, and UMAP. The rank correlations between variable and fixed bandwidth are equivalent to the third decimal place in Laplacian Eigemaps and UMAP. As for t-SNE, the four clusters in the embedding are less separated than in UMAP and our proposed method has misidentified the kernel cores, leading to a lower rank correlation in variable bandwidth. Since the embeddings from twin peak data generally capture the rectangular structure in the meta data than those from the swiss roll data, the rank correlations are much higher in Table 2 than in Table 1, with the lowest correlation being 0.259. This again suggests that the accuracy of outlier detection is highly related to the quality of manifold learning embedding.

In Figure 6, we plot the estimated density against the true density of the ISOMAP embedding for KDE with variable and fixed bandwidth, with colors and shapes showing the four kernels in the meta data. The linear positive relationship between the true densities and variable KDEs on the left handside is stronger than that of the fixed bandwidth KDEs. Combined with the top-right subplot in Figure 5, we could tell that most points are underestimated near the true kernel cores, which also suggests that the fixed bandwidth tries to smooth across all the data points and fails to fix the local distortions in the manifold learning process like the proposed pointwise variable bandwidth.

4.2 100-D mapping from a 5-D semi-hypersphere

As a high-dimensional experiment, we generate the meta data from a 5-dimensional semi-hypersphere, transform it into a 100-dimensional space, and then embed it in $d = 5$ with manifold learning. First, we simulate $N = 2,000$ points, $(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_4)'$, from a 4-dimensional Gaussian mixture model with two mixture components, $\mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0, 0, 0, 0)'$,

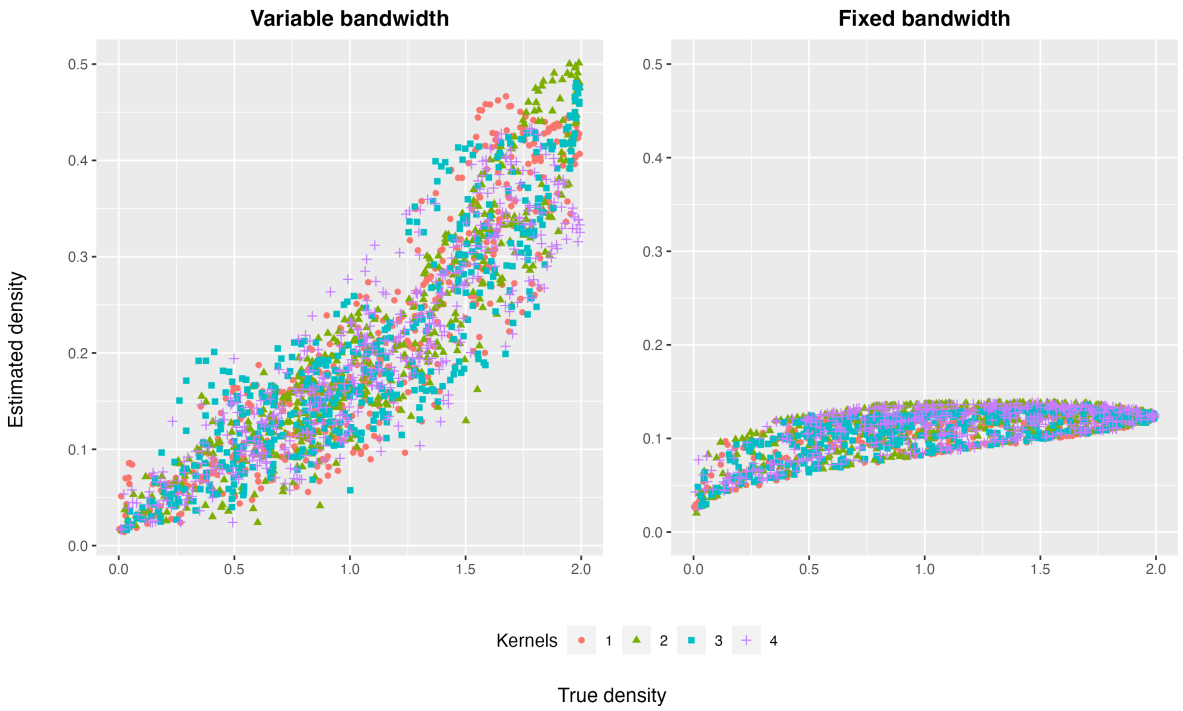

Figure 6: *Scatterplot of true density and estimated density of ISOMAP embedding for KDE with both variable and fixed bandwidth. The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Variable bandwidth shows a strong linear positive relationship.*

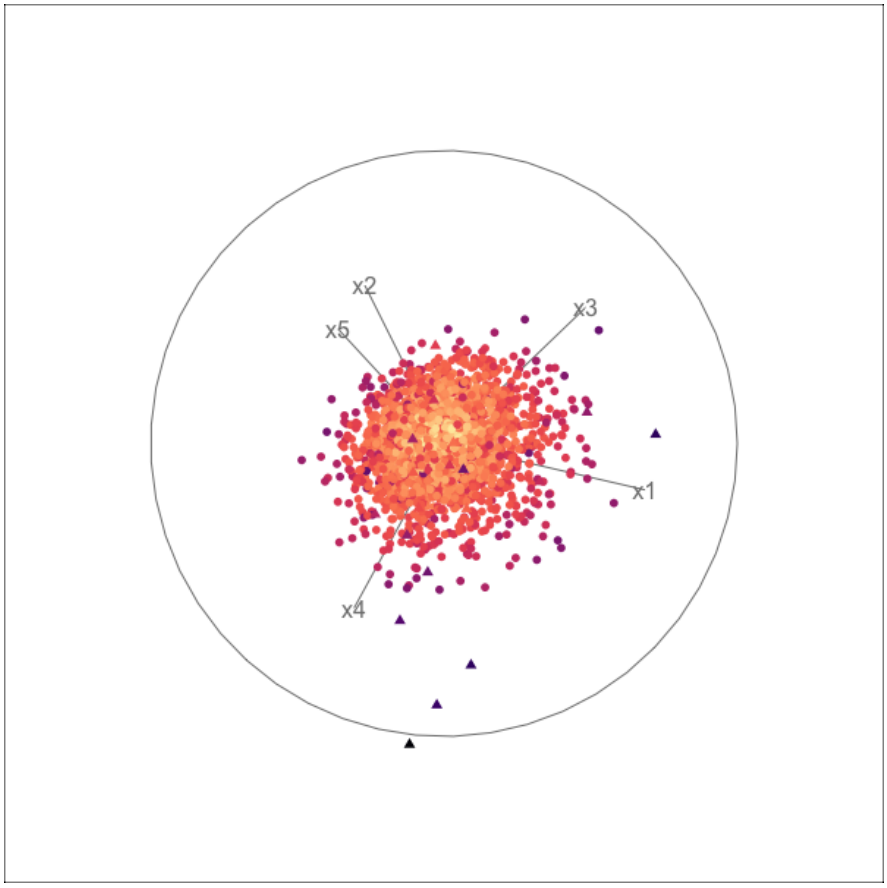

Figure 7: *Scatterplot display of the animation of a 5-D tour path with shapes indexing the Gaussian mixture component and the colors showing the distance to the kernel cores.*

$\mathbf{\Sigma}_1 = diag(1,1,1,1)$, and $\mathbf{\Sigma}_1 = diag(2,2,2,2)$. In order to manually add anomalies to be distant points from the means, the mixture proportions are set as $\pi_1 = 0.99$ and $\pi_2 = 0.01$. The fifth dimension is calculated to satisfy the five-dimensional semi-hypersphere surface equation $X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 = r^2$ where $X_5 > 0$ and r is set as 7. The Gaussian mixture densities could be calculated using Equation (5) as the true density of the 5-d meta data. Figure 7 shows a scatterplot display when animating a 5-D tour path with the R package *tourr* [REFERENCE]. The round and triangular point shapes index the two mixture components $\mathsf{N}(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1)$ and $\mathsf{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2)$, and the colors shows the distance between the simulated $4-D$ data point from Gaussian mixture model and the kernel means $(0,0,0,0)'$. The more distant from the point to the kernel cores, the lower the true densities, which shows in a darker color in Figure 7. It can be seen that the most distant points are in a triangular shape, meaning that they are simulated from $\mathsf{N}(\boldsymbol{\mu}_2, \mathbf{\Sigma}_2)$. The dark colors also indicate that they are the true outliers because of their low densities.

Then we initial the other 95 dimensions in the high-dimensional space as zero columns and further rotate the 100-dimensional data of size N(denote the transpose of the data matrix as $\boldsymbol{X}_0$ with dimension $100 \times N$) to get rid of the zeros so that it could be passed to the manifold learning algorithms. The rotation matrix is derived from the QR decomposition of a 100×100 matrix $\boldsymbol{A}$ with all components randomly generated from a uniform distribution $\mathcal{U}(0,1)$. For any real matrix $\boldsymbol{A}$ of dimension $p \times q$, the QR decomposition could decompose the matrix into the multiplication of two matrix $\boldsymbol{Q}$ and $\boldsymbol{R}$ so that $\boldsymbol{A} = \boldsymbol{QR}$, where the dimension of Q is a matrix with unit norm orthogonal vectors, $\boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}$, and $\boldsymbol{R}$ is an upper triangular matrix. Matrix $\boldsymbol{Q}$ satisfies $\boldsymbol{X}_0'\boldsymbol{X} = (\boldsymbol{QX}_0)'(\boldsymbol{QX}_0)$, meaning that the pairwise Euclidean distances between data points in $\boldsymbol{X}_0'$ is equivalent to that of $(\boldsymbol{QX}_0)'$. Therefore, we use matrix Q as the rotation matrix for where the rotated data matrix $\boldsymbol{X} = (\boldsymbol{QX}_0)'$ of dimension $N \times 100$ is now the input data for the manifold learning algorithms. Again, we set the embedding dimension to be equal to the meta data dimension $d = 5$.

In Figure 8, the estimated densities are compared with the true density on the x-axis for four manifold learning embeddings, ISOMAP, LLE, Laplacian Eigenmaps, and UMAP. Note that we exclude t-SNE algorithm in this section because it is designed mainly for low-dimensional visualization purposes, and it is only applicable to embedding dimensions within three. Similar to Figure 7, the point shapes show the two mixture component in the meta data, and the colors represent the distance to the kernel means, with distant outliers shown in darker colors. For well-estimated densities, the true outliers with low true densities will also have low estimated densities, which suggests that darker-colored points should appear in the bottom-left corner in Figure 8. This is true for both ISOMAP and LLE, partly true for Laplacian Eigenmaps, but not in UMAP where these outliers have relatively high density estimates. For variable bandwidth KDE, there is a strong positive linear relationship with the true densities for ISOMAP, LLE, and Laplacian Eigenmaps,

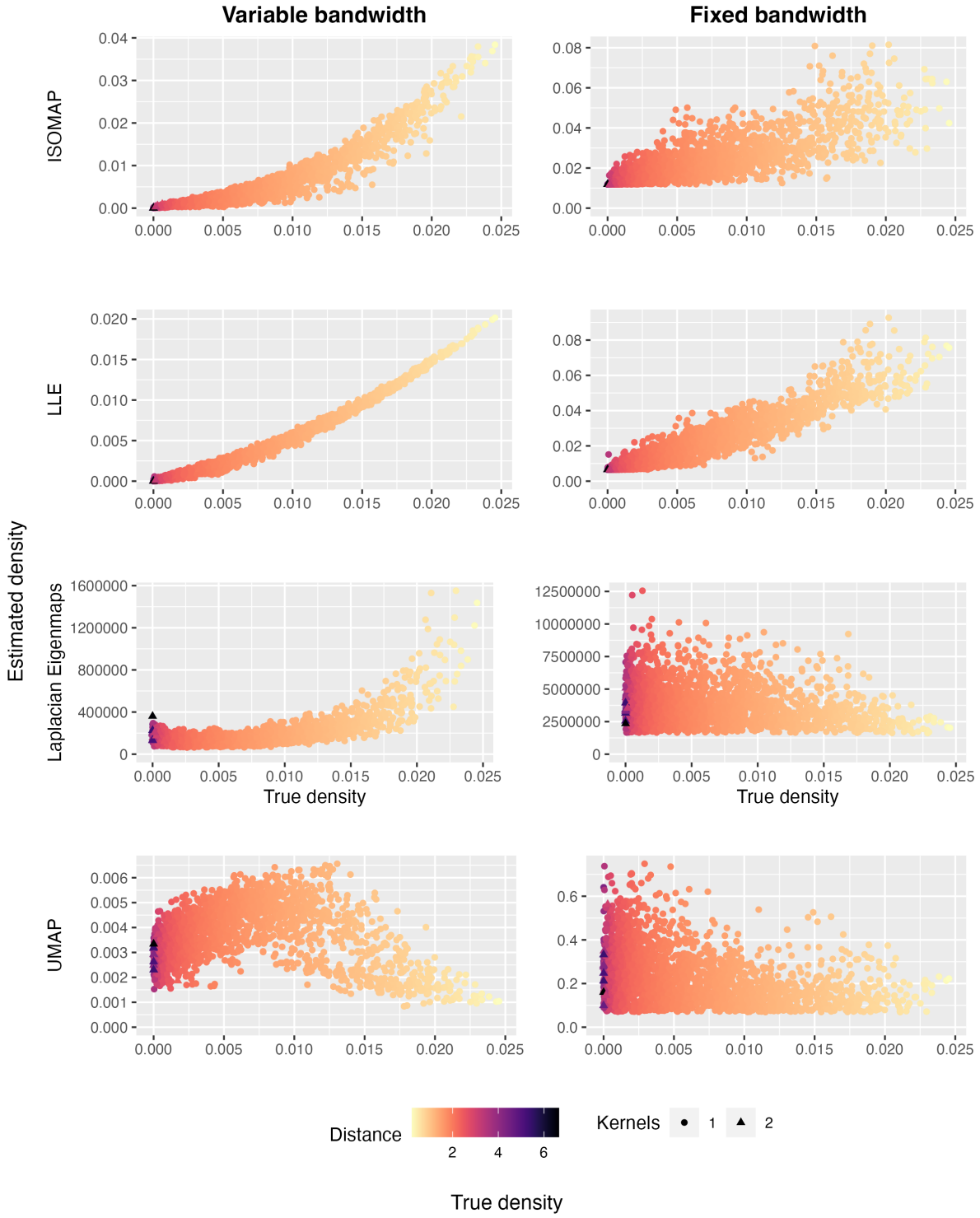

Figure 8: *Scatterplot of true density and estimated density of different embeddings for KDE with both variable and fixed bandwidth. The point shapes indicates the two Gaussian mixture components and the colors shows the distance to the kernel cores.*

Table 3: *Correlation between true density and estimated density for four manifold learning embeddings.*

	ISOMAP	LLE	Laplacian.Eigenmaps	UMAP
Variable bandwidth	**0.921**	**0.981**	**0.662**	**-0.130**
Fixed bandwidth	0.806	0.940	-0.181	-0.341

and the relationship is stronger than the fixed bandwidth. This suggests that our proposed KDE with variable bandwidth is more accurate than the fixed bandwidth in estimating the manifold learning embedding densities. In KDE with fixed bandwidth, the bandwidth is often too large to smooth across all data points, especially when there is severe distortion in the embedding data. By introducing the pointwise variable Riemannian metric in kernel density estimation, it is reasonable to believe that it could fix the distortion introduced by these three manifold learning algorithms.

5 Application

5.1 Irish smart meter dataset

In this application, we use the smart meter data from the *CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010* in Ireland (Commission for Energy Regulation (CER) 2012) between 14 July 2009 and 31 December 2010. The CER dataset[2] records the half-hourly electricity consumption of individual residential and commercial properties, not including energy for cooling or heating systems. We selected the 3,639 residential data with no missing values during the data collection period for a total of 535 days.

For the electricity consumption data of residential individuals, it would be worthwhile to explore the distribution of electricity demand rather than the raw consumption data to study the usage patterns of different households or different periods or the week (Hyndman, Liu & Pinson 2018). Cheng, Hyndman & Panagiotelis (2021) propose two non-Euclidean distance estimators to enable manifold learning algorithms in statistical manifolds with each observation as a distribution. Cheng, Hyndman & Panagiotelis (2021) use the same smart meter data for identifying outliers with kernel bandwidth estimation but fail to consider the distortion and information loss in the 2-dimensional embeddings given that the input data dimension is much higher. By introducing the Riemannian metric as the bandwidth matrix, we could take into account the distortion in the 2-D embedding and further improve the accuracy of the density estimation.

In this section, we first calculated the same empirical distributions of the 336 half-hourly periods of the week for each household, and apply the total variation distance estimator proposed in Cheng, Hyndman & Panagiotelis (2021) in the statistical manifold learning to get the 2-D embedding of

[2]accessed via the Irish Social Science Data Archive - www.ucd.ie/issda.

all households. Algorithm 2 is then used to obtain density estimates with the pointwise variable Riemannian metric as the bandwidth matrix and detect outliers. The data processing steps have been clearly stated in the application section of Cheng, Hyndman & Panagiotelis (2021) and they are skipped here. Unlike the simulations in Section 4, we know nothing about the true density of the electricity distributions for all periods of the week and all households, so it is impossible to compare the estimated densities with the true meta data density as in Figure 8. However, we could generate all the density estimates with the existing KDE method with fixed bandwidth, which is an optimal method for density estimation, and compare the densities from our proposed method with them.

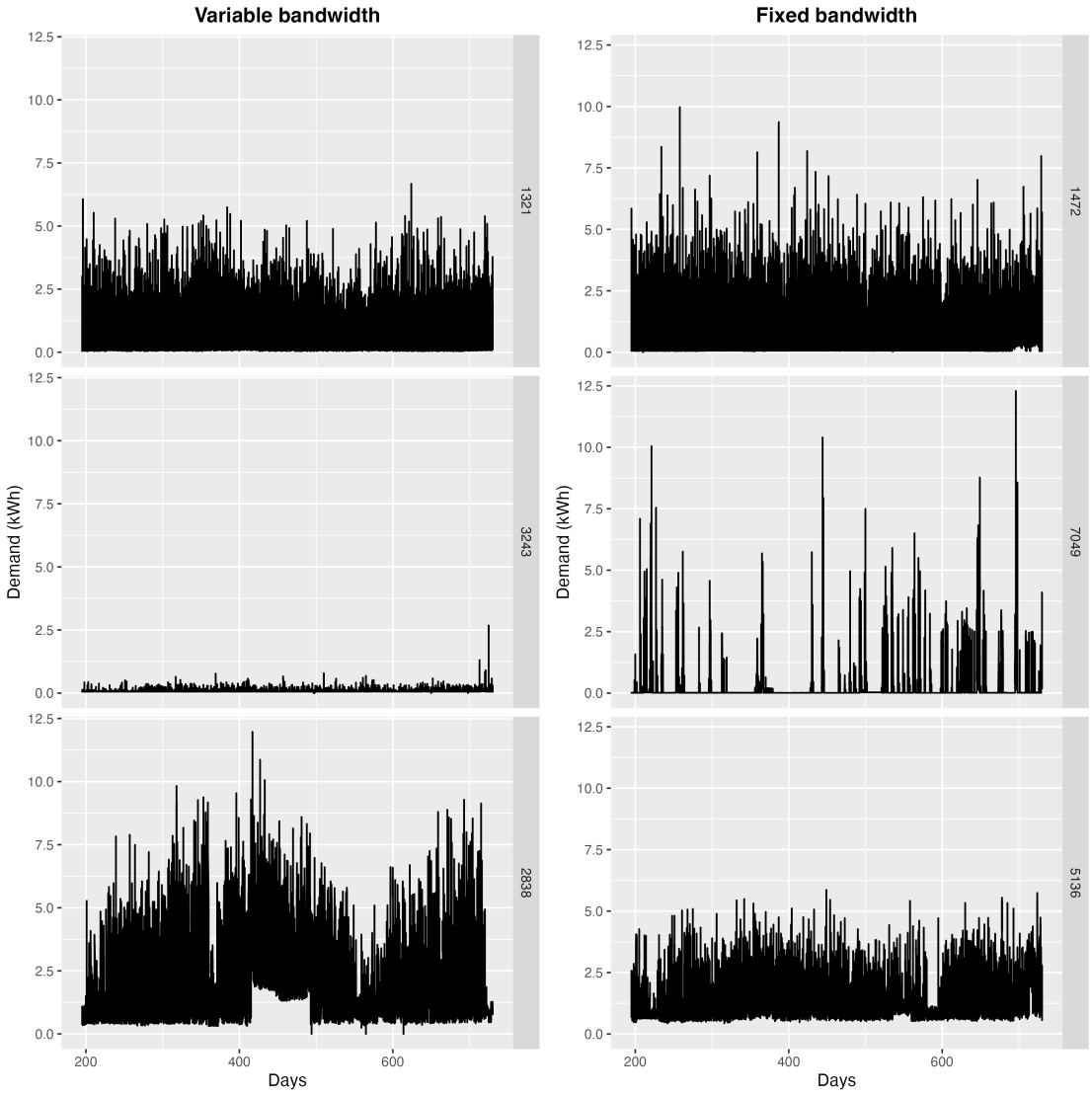

Figure 9: *Electricity usage plots of all 535 days for the most typical household and two anomalies in rows and two bandwidth selection methods in columns.*

Figure 9 shows the electricity usage data of three households for both density estimation methods respectively, with the top one being the most typical household with the highest density and the bottom two being the top two outliers with the lowest densities. The typical households in the top row are close except that there are a few spikes for the one with fixed bandwidth. As to the

anomalies, variable bandwidth tends to capture the unusual electricity demand volume when the usage is very low or high. It could also capture the unusual usage pattern when there are sudden spikes in ID 3243 or very high base electricity usage for the middle time periods in ID 2838. In contrast, fixed bandwidth KDE is more sensitive to spikes even when the spikes happen in a certain time window in 7049, or when the usage has an obvious time-of-week pattern with a few low electricity usage periods.

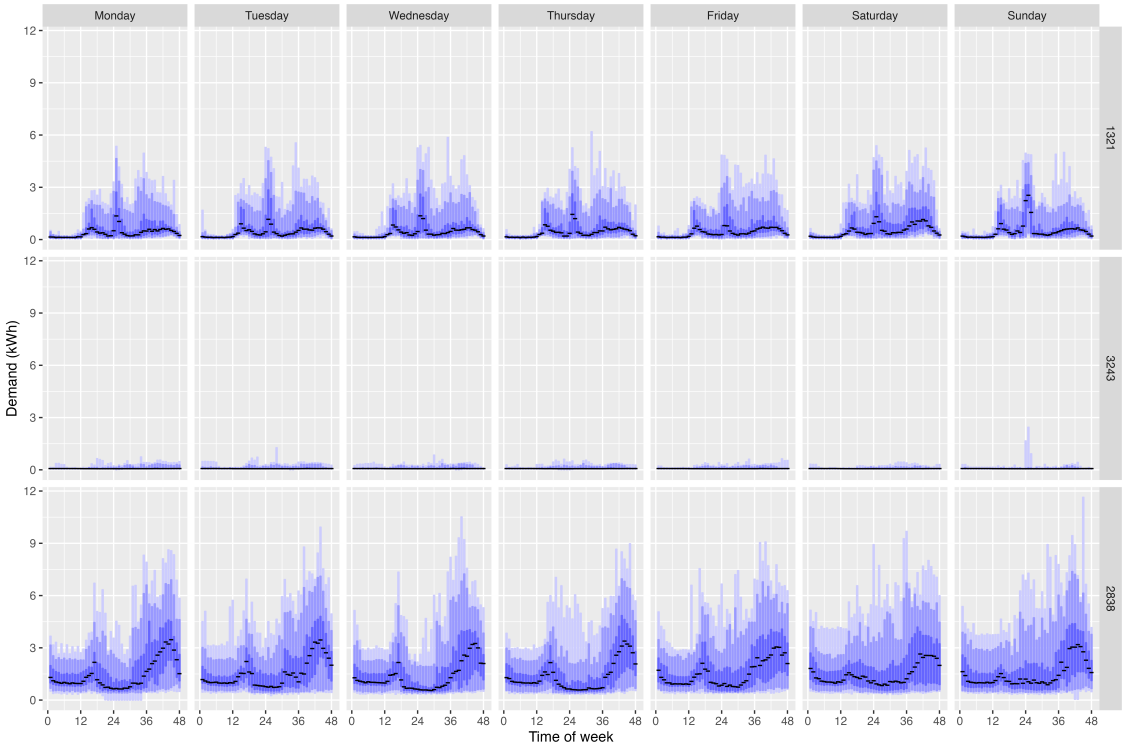

Figure 10: *Two smart-meter demand examples, ID 1003 and ID 1539, from the Irish smart meter data set.*

Further insights could be gained by comparing the quantile region plots of electricity demand against the time of the week for the same typical or anomalous households in Figure 10 and Figure 11. Again the distribution of both typical households in the top panel has shown a repeated period-of-the-week usage pattern, with higher usage during mealtime on all seven days of the week and slightly higher usage for weekends. However, this repeated pattern in a week window is clearly for the typical household ID 1321. As for the distributions for outliers, the middle row outliers from variable bandwidth have spikes only on Tuesday and Sunday noons, while the fixed bandwidth has an increasing electricity demand across the day of the week. The bottom row outliers both have a repeated time usage pattern, but the electricity usage amount is higher with the highest median above 3kWh. These findings show the difference in finding typical and anomalous households with different bandwidth selections.

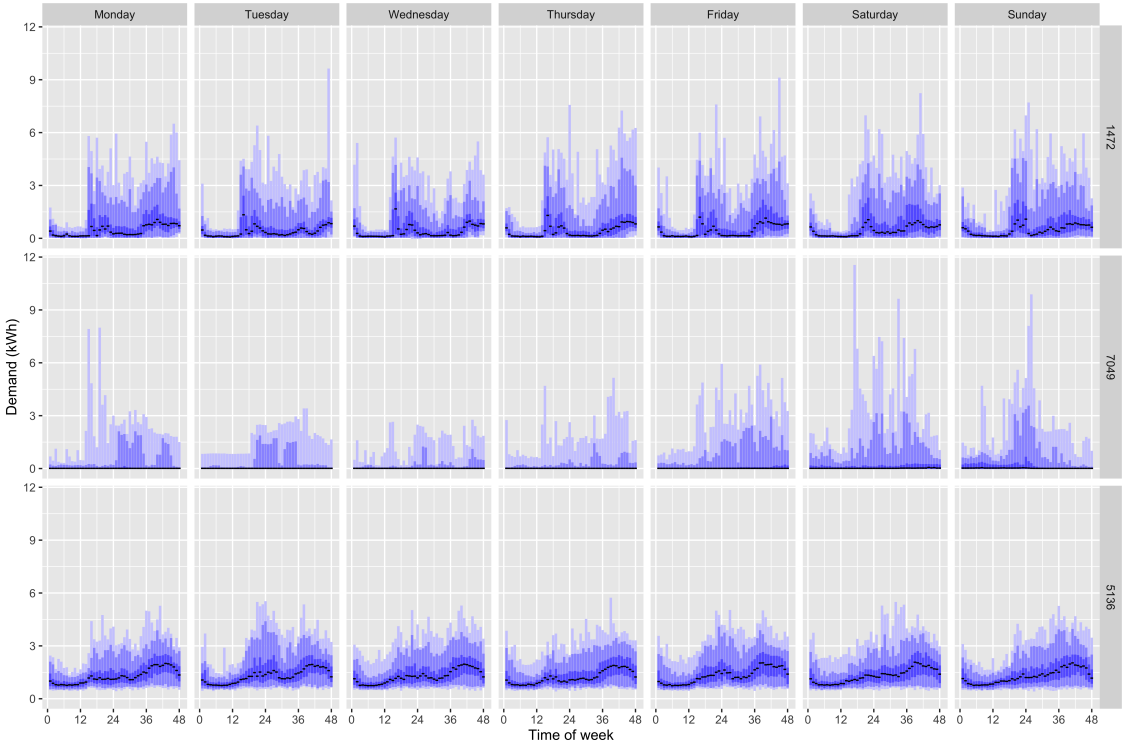

Figure 11: *Two smart-meter demand examples, ID 1003 and ID 1539, from the Irish smart meter data set.*

6 Conclusion

In this paper, we propose a new method to estimate the density of manifold learning embedding and further identify outliers based on the densities. The Riemannian metric measure the direction and angle of the distortion when mapping data points through a non-linear function in manifold learning algorithms. By introducing the Riemannian metric as the pointwise variable bandwidth matrix in kernel density estimation, the local distortion in the low-dimensional embedding could be used to estimate densities, leading to a more accurate description of the data distributions. We compare our proposed method with fixed bandwidth KDE by two simulation settings, 2-D meta data mapped as a 3-D swiss roll or twin peaks data and 5-D semi-hypersphere mapped in 100-D space, and show that variable bandwidth could improve the density estimation given a good manifold learning embedding.

As an empirical example, we explore the distributions of different households and time periods of the week in the Irish smart meter data. Five manifold learning algorithms, including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, are applied to get the 2-D embeddings, and KDE with both variable and fixed bandwidth are used to get the density estimates. We compare both density estimates by looking at the distributions of the most typical households with the highest densities and the most anomalous households with the lowest densities. Both methods could identify the typical households with certain usage patterns, while the outliers are anomalous in different ways.

There are several open questions to be explored. The first involves the selection of tuning parameters for the manifold algorithm so that a maximal level of embedding quality is achieved, where embedding quality is measured using one of the metrics discussed in the online supplementary material of Cheng, Hyndman & Panagiotelis (2021). The scaling of the Riemannian metric to get the closest range of the true densities is also worth exploring. The scale of distortion in each point in manifold learning could vary a lot. If we could smooth across all data points, eg. multiply the Riemannian matrix with the ratio between its determinant and the sum of all Riemannian matrix determinants, the global density estimates could be potentially smoothed. The density estimates on the edges of the whole data structure could be improved because most outer area points tend to be detected as outliers. The choice of manifold learning algorithms also has a large impact on the embedding accuracy, which we have seen will affect the density estimation and outlier detection. However, the outperformance of VKDE with Riemannian bandwidth than the fixed bandwidth has been shown in the higher dimensional simulation data and the electricity usage data, which are more related to real-life data sets.

Acknowledgment

This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH HPC Cluster.

A Appendix: Notions about Riemannian geometry

In this appendix, we present some notions about the Riemannian geometry used in this paper.

A.1 Differentiable manifolds

In topology, a *homeomorphism* is a bijective map between two topological spaces that is continuous in both directions. A *Hausdorff space* is a topological space where any two distinct points can be separated by disjoint neighborhoods. A d-dimensional (topological) *manifold* M is a connected Hausdorff space $(M, \mathcal{T}_M)$ where the neighborhood U for each point p is homeomorphic to an open subset V of the Euclidean space $\mathbb{R}^d$. Such a homeomorphism $\varphi : U \to V$ together with U gives a (coordinate) *chart*, denoted as (U, φ), with the corresponding local coordinates $(x_1(p), \ldots, x_d(p)) := \varphi(p)$. A collection of charts $\{U_\alpha, \varphi_\alpha\}$ ranging over the manifold M is called an *atlas*, denoted as $\mathcal{A}$. The manifold M is a *differentiable manifold* if there exists an atlas of M, $\{U_\alpha, \varphi_\alpha\}$, such that the *transition maps* between any two charts,

$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \to \varphi_\beta(U_\alpha \cap U_\beta),$$

are differentiable of class C^∞ (smooth).

Let φ be an injective map $E \to \varphi(E)$. Then φ is an embedding of E into M if and only if $\varphi : E \to \varphi(E)$ is a homeomorphism, and $\varphi(E)$ is called an embedded submanifold of M with the subspace topology.

A.2 Tangent vector and tangent space

The tangent vector at point p can be intuitively viewed as the velocity of a curve passing through the point p or as the directional derivatives at p. Here we define the tangent vector via the velocity of curves.

For any $p \in M$, let $\gamma_1 : (-\epsilon_1, \epsilon_1) \to M$ and $\gamma_2 : (-\epsilon_2, \epsilon_2) \to M$ be two smooth curve through p, i.e. $\gamma_1(0) = \gamma_2(0) = p$. γ_1 and γ_2 are *equivalent* if and only if there exists a chart, (U, φ), at p such that

$$(\varphi \circ \gamma_1)'(0) = (\varphi \circ \gamma_2)'(0).$$

A *tangent vector* to a manifold M at point p, v_p, is any equivalent class of the differentiable curves initialized at p. The set of all tangent vectors at p defines the *tangent space* of M at p, denoted as T_pM. The tangent space is a vector space of dimension d, equal to the dimension of M, and it does not depend on the chart φ locally at p. The collection of all tangent spaces defines the *tangent bundle* $TM = \cup_{p \in M} T_pM$. Tangent vectors can also be seen as the directional derivatives at p. For a given

coordinate chart $\varphi = (x_1, \ldots, x_d)$, the tangent vectors defining partial derivatives are denoted as $\frac{\partial}{\partial x_1}(p), \ldots, \frac{\partial}{\partial x_d}(p)$, which define a *basis* of the tangent space.

The tangent space T_pM also admits a dual space $T_p^\star M$ called the *cotangent space* with corresponding *cotangent vector* $z_p : T_p^\star M \to \mathbb{R}^d$, and basis denoted as $dx_1(p), \ldots, dx_d(p)$.

A.3 Riemannian metric and geodesic distance

A Riemannian metric g_p defined on the tangent space T_pM at each point p is a local inner product $T_pM \times T_pM \to \mathbb{R}$, where g_p is $d \times d$ symmetric positive definite and varies smoothly at p. Generally, we omit the subscript p and refer to g as the Riemannian metric. The inner product for two vectors $u, v \in T_pM$ is written as $< u, v >_g = g_{ij}u^i v^j$ using the Einstein summation convention where implicit summation over all indices, $\sum_{i,j}$, is assumed. A differentiable manifold M endowed with the Riemannian metric g on each tangent space T_pM is called a *Riemannian manifold* (M, g).

The Riemannian metric g can be used to define the norm of a vector u, $\|u\| = \sqrt{< u, v >_g}$ and the angle between two vectors u and v, $\cos\theta = \frac{<u,v>_g}{\|u\|\|v\|}$, which are the geometric quantities induced by g. g could also be used to define the line element $dl^2 = g_{ij}dx_i dx_j$ and the volume element $dV_g = \sqrt{\det(g)}dx_1 \ldots dx_d$, where $(x_1, \ldots, x_d)$ are the local coordinates of the chart (U, φ). For a curve $\gamma : I \to M$, the length of the curve is

$$l(\gamma) = \sqrt{\int_0^1 \|\gamma'(t)\|_g^2 dt} = \sqrt{\int_0^1 g_{ij}\frac{dx_i}{dt}\frac{dx_j}{dt}dt},$$

where $\gamma(I) \subset U$. The volume of $W \subset U$ is defined as

$$Vol(W) = \int_W \sqrt{\det(g)}dx_1 \ldots dx_d,$$

which is also called the *Riemannian measure* on M.

The *geodesics* of M are the smooth curves that locally joins the points along the shortest path on the manifold. A curve $\gamma : I \to M$ is a geodesic if for all indices i, j, k, the second-order ordinary differential equation is satisfied,

$$\frac{d^2x_i}{dt^2} + \Gamma_{jk}^i \frac{dx_j}{dt}\frac{dx_k}{dt} = 0,$$

where Γ_{jk}^i is the *Christoffel symbol* defined by

$$\Gamma_{jk}^i = \frac{1}{2}\sum_l g_{il}\left(\frac{\partial g_{il}}{\partial x_k} + \frac{\partial g_{kl}}{\partial x_j} - \frac{\partial g_{jk}}{\partial x_l}\right).$$

The geodesics have a constant speed with norm $\|\gamma'(t)\|$, and they are the local minimizers of the arc length functional $l : \gamma \to \sqrt{\int_0^1 \|\gamma'(t)\|_g^2 dt}$ when the curves are defined over the interval $[0, 1]$. The

geodesic distance d_g is the length of the shortest geodesic between two points on the manifold. For a point $p \in M$, when the geodesic distance starting at p is not minimized, we call such set of points the *cut locus* of p, and the *injectivity radius* at $p \in M$ is the distance to the cut locus. The injectivity radius of (M, g), $inj_g M$ is the infimum of the injectivity radii over all points on the manifold.

A.4 Pushforward and pullback metric

Pushforward and pullback are two notions corresponding to the notions of tangent and cotangent vectors. Let f be an embedding from the Riemannian manifold (M, g) to another smooth manifold E. The pushforward $h = \varphi * g$ of the Riemannian metric g along $\varphi \equiv f^{-1}$ is a linear map $f_* : TM \to TE$, which maps the tangent vectors u_p at point $p \in M$ to the tangent vectors $f_* u_p$ at the mapping point $f(p) \in E$, and satisfies that for $u, v \in T_f(p)E$,

$$< u, v >_{\varphi * g_p} = < df_p^{-1} u, df_p^{-1} v >_{g_p},$$

where df_p^{-1} is the Jacobian of f^{-1}. The tangent vectors $f_\star u_p$ are equivalent to the velocity vector of a curve $\gamma : I \to M$ passing through point p at time zero with a constant speed $\gamma^{-1}(0) = u_p$. Similarly, the pullback maps the cotangent vectors $z_{f(p)}$ at $f(p) \in E$ to cotangent vectors at $p \in M$ acting on tangent vectors $u \in T_p M$.

A.5 Exponential map and logarithmic map

Denote $B(p, r) \subset T_p M$ as an open ball centered at point p with raidus r. Then $B(0_p, r) = exp_p^{-1}(B(p, r))$ is an open neighborhood of 0_p in the tangent space at p, $T_p M$. Define exp_p as the *exponential map* at point p, where exp_p is a differentiable, bijective map of differentiable inverse (i.e. *diffeomorphism*) that maps a tangent vector $u \in B(0_p, r)$ to the endpoint of the geodesic $\gamma : I \to M$ satisfying $\gamma(0) = p, \gamma^{-1}(0) = u$, and $\gamma(1) = exp_p(u)$. The exponential map moves point p from p at speed u to an endpoint after covering the length of $\|u\|$ along the geodesic in one time unit. The inverse of the exponential map is called the *logarithm map*, $\log_p(q) := \exp_p^{-1}(q)$, which gives the vector from point p to q. Also define the *geodesic ball* centered at p of radius $r > 0$ as the image by the exponential map of $B(0_p, r) \subset T_p M$ with $r < inj_g M$. Then we could interpolate a geodesic γ between two points p and q with the exponential map and the logarithmic map, $\gamma(t) = \exp_p(t \log_p(q))$, and the geodesic distance is given by $d_g(p, q) = \| \log_p(q) \|_g$.

A.6 Volume density function

For any $u \in T_p M$, let $\gamma(u) : t \to \exp(\frac{tu}{\|u\|})$ and let $w_i, i = 1, \ldots, d$ be Jacobi fields along γ such that $w_i(0) = 0$, for all $i = 1, \ldots, d$, $Dw_1/dt(0) = u/\|u\|$ and $Dw_i/dt(0)$ forms an orthonomal basis of

the tangent space T_pM. Define $\theta_p(q) : T_pM \to \mathbb{R}$ the volume density function on M as

$$\theta_p(u) = \|u\|^{d-1} \det(w_1(\|u\|), \ldots, w_d(\|u\|)).$$

Rewrite $\theta_p(u) = \theta_p(\log_p(u))$. The exponential map $\exp_p : T_pM \to M$ induced by pullback of a volume form $\exp_p^* Vol$ on T_pM, and θ_p is its density with respect to the Lebesgue measure of the Euclidean structure on T_pM, which gives in normal coordinates

$$d \exp_p^* Vol(u) = \theta_p(u) du.$$

A.7 Example: S^1

Manifold: S^1 with center 0 and radius $r = 1$

Polar coordinates: (r, θ)

Uniformly distributed angle: $\theta \sim U(0, \pi/2)$ with $f(\theta) = 2/\pi$ for $\theta \in [0, \pi/2]$.

Cartesian coordinates: (x, y) where $x = \cos(\theta) \in [0, 1]$ and $y = \sin(\theta) \in [0, 1]$

Density of θ: $f(\theta) = 2/\pi$

Density of x: $f(x) = \frac{2}{\pi\sqrt{1-x^2}}$

The CDF of X is given by

$$P(X \leq x) = P(\cos(\theta) \leq x) = P(\theta \geq \arccos(x)) = \int_{\arccos(x)}^{\pi/2} f(\theta) d\theta = \frac{2}{\pi}(\frac{\pi}{2} - \arccos(x)) = 1 - \frac{2}{\pi}\arccos(x).$$

Then the density of X is the derivative of the CDF w.r.t. x

$$f(x) = -\frac{2}{\pi}(-\frac{1}{\sqrt{1-x^2}}) = \frac{2}{\pi\sqrt{1-x^2}}.$$

Given $(x, y) \in \mathbb{R}^2$ as sample points from S^1, we could estimate the density of each sample point $X_i \in S^1, i = 1, \ldots, N$ and compare it with its true density $f(x)$.

We use the density estimator in Pelletier (2005). For point $p \in S^1$, the density at p is estimated by

$$f_N(p) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r^d \theta_{X_i}(p)} K(\frac{d_g(p, X_i)}{r}),$$

where $d = 1$, h is the bandwidth, $\theta_{X_i}(p)$ is the volume density function at sample point X_i, and $d_g(p, X_i)$ is the geodesic distance between p and X_i along the manifold S^1.

Since the manifold is known (S^1), we know the geodesic distance is the arc length between two points on the manifold, which is given by $L(\gamma) = r(\beta - \alpha)$ with $r = 1$. Therefore, the geodesic distance is

$$d_g(p, X_i) = \arccos(p) - \arccos(X_i)$$

As stated in Pelletier (2005), in terms of *geodesic normal coordinates* at X_i, the volume density function $\theta_{X_i}(p)$ equals the determinant of the metric g expressed in these coordinates at the logarithm map $\exp_{X_i}^{-1}(p)$, i.e. $|g(\exp_p^{-1}(x_i))|$.

[When using normal coordinates(orthogonal basis) for tangent vector in $T_{X_i}S^1$, the exponential map has *geodesic normal coordinates*.]

The Riemannian metric g for S^1 is given by the inner products on the tangent spaces of S^1. For each $X_i \in M$,

$$g_{X_i} : T_{X_i}M \times T_{X_i}M \to \boldsymbol{R},$$

is smooth in a local neighborhood U of M.

Then the volume density function $\theta_{X_i}(p)$ is the inner product of the tangent vectors produced by the logarithm map from X_i to p, which is given by

$$\theta_{X_i}(p) = \sqrt{1 - X_i^2}.$$

Therefore, the estimator for $p \in S^1$ is given by

$$f_N(p) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{r\sqrt{1 - X_i^2}} K\left(\frac{\|\arccos(p) - \arccos(X_i))\|}{r}\right),$$

where the bandwidth r is selected by the plug-in bandwidth selector implemented by the R package *ks*.

A.7.1 How to derive volume density function $\theta_{X_i}(p)$

A.7.1.1 Example of S^2 in Henry & Rodriguez (2009b). Let S^2 be the two-dimensional sphere of radius 1 and $p \in S^2$. Let v and w be vectors such that $\|v\| = \|w\| = 1$ and $\{v, w, p\}$ is the orthonormal basis of $\mathbb{R}^3$. Consider the exponential chart (U, Ψ) induced by the parametrisation $\Psi^{-1} : B_\pi(0) \to S^2 \backslash \{-p\}$ given by

$$\Psi^{-1}(s, t) = \cos(r)p + \frac{\sin(r)}{r}(sv + tw),$$

if $(s,t) \neq (0,0)$ and $\Psi^{-1}(0,0) = p$ where $r = \sqrt{s^2+t^2}$. Note that $r = d_g(p,q)$ if $q = \exp_p(sv+tw)$. Then we take the partial derivatives of $\Psi^{-1}(s,t)$ with respect to the basis of the tangent space, s and t, and have that

$$A(s,t) = \frac{\partial}{\partial \Psi_1}\bigg|_{\Psi^{-1}(s,t)} = \frac{-\sin(r)s}{r}p + \frac{\cos(r)rs^2 + t^2\sin(r)}{r^3}v + \frac{(\cos(r)r - \sin(r))st}{r^3}w,$$

$$B(s,t) = \frac{\partial}{\partial \Psi_2}\bigg|_{\Psi^{-1}(s,t)} = \frac{-\sin(r)t}{r}p + \frac{(\cos(r)r - \sin(r))st}{r^3}v + \frac{\cos(r)rt^2 + \sin(r)s^2}{r^3}w.$$

The coefficients for v and w gives the local coordinates of two tangent vectors. Further, since the riemannian metric is defined as the inner product between two tangent vectors, each component of the riemannian metric could be calculated as $g(A(s,t),A(s,t)), g(A(s,t),B(s,t)), g(B(s,t),B(s,t))$. Therefore, we obtain that the volume density on the sphere is

$$\theta_p(q) = \left|\det g_q\left(\frac{\partial}{\partial \Psi_i}\bigg|_q, \frac{\partial}{\partial \Psi_j}\bigg|_q\right)\right|^{1/2} = \frac{|\sin(d_g(p,q))|}{d_g(p,g)},$$

for $q \neq p, -p$, and $\theta_p(p) = 1$.

A.7.1.2 S^1 Similarly in S^1, we could let $\{v,p\}$ be the orthonormal basis of $\mathbb{R}^2$. Then the exponential chart is induced by the parametrization

$$\Psi^{-1}(s) = \cos(r)p + \frac{\sin(r)}{r}sv,$$

and $\Psi^{-1}(0) = p$ where $r = s$. Again $r = d_g(p,q)$ if $q = \exp_p(sv)$. Then we have

$$A(s,t) = \frac{\partial}{\partial \Psi_1}\bigg|_{\Psi^{-1}(s,t)} = -\sin(r)p + \cos(r)v.$$

So the riemannian metric is $g_q = \cos(r)$ and further,

$$\theta_p(q) = \sqrt{|\cos(r)|} = \sqrt{|\cos(d_g(p,q))|},$$

where the geodesic distance is $d_g(p,q) = \arccos(p) - \arccos(q)$, and $\theta_p(p) = 1$.

References

Ahmed, M, AN Mahmood & MR Islam (2016). A survey of anomaly detection techniques in financial domain. *Future generations computer systems: FGCS* **55**, 278–288.

Ahmed, M, A Naser Mahmood & J Hu (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications* **60**, 19–31.

Bhuyan, MH, DK Bhattacharyya & JK Kalita (2013). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials* **16**(1), 303–336.

Breiman, L, W Meisel & E Purcell (1977). Variable Kernel Estimates of Multivariate Densities. *Technometrics: a journal of statistics for the physical, chemical, and engineering sciences* **19**(2), 135–144. https://www.tandfonline.com/doi/abs/10.1080/00401706.1977.10489521.

Cao, R, A Cuevas & W González Manteiga (1994). A comparative study of several smoothing methods in density estimation. *Computational statistics & data analysis* **17**(2), 153–176. https://www.sciencedirect.com/science/article/pii/016794739200066Z.

Chacón, JE & T Duong (2010). Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *Test* **19**(2), 375–398. https://doi.org/10.1007/s11749-009-0168-4.

Chavel, I (2006). *Riemannian Geometry: A Modern Introduction*. en. Cambridge University Press, p. 108.

Chen, YC (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology* **1**(1), 161–187.

Cheng, F, RJ Hyndman & A Panagiotelis (2021). Manifold Learning with Approximate Nearest Neighbors. (3/21).

Commission for Energy Regulation (CER) (2012). *CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]*. SN: 0012-00.

Duong, T (2004). Bandwidth selectors for multivariate kernel density estimation. https://www.mvstat.net/tduong/research/publications/duong-2005-thesis.pdf.

Duong, T (2007). ks: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R. en. *Journal of statistical software* **21**, 1–16.

Duong, T & M Hazelton (2003). Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of nonparametric statistics* **15**(1), 17–30.

Fernando, T, H Gammulle, S Denman, S Sridharan & C Fookes (2022). Deep learning for medical anomaly detection – A survey. en. *ACM computing surveys* **54**(7), 1–37.

Goldberg, Y, A Zakai, D Kushnir & Y Ritov (2008). Manifold Learning: The Price of Normalization. *J. Mach. Learn. Res.* **9**(Aug), 1909–1939.

Heidenreich, NB, A Schindler & S Sperlich (2013). Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA. Advances in Statistical Analysis. A Journal of the German Statistical Society* **97**(4), 403–433. https://doi.org/10.1007/s10182-013-0216-y.

Henry, G & D Rodriguez (2009a). Kernel Density Estimation on Riemannian Manifolds: Asymptotic Results. *Journal of mathematical imaging and vision* **34**(3), 235–239. https://doi.org/10.1007/s10851-009-0145-2.

Henry, G & D Rodriguez (2009b). Robust nonparametric regression on Riemannian manifolds. en. *Journal of nonparametric statistics* **21**(5), 611–628. http://www.tandfonline.com/doi/abs/10.1080/10485250902846439.

Hyndman, RJ, X Liu & P Pinson (2018). Visualizing big energy data: Solutions for this crucial component of data analysis. *IEEE Power Energ. Mag.*

Hyndman, RJ (1996). Computing and Graphing Highest Density Regions. *Am. Stat.* **50**(2), 120–126.

Jones, MC (1990). Variable kernel density estimates and variable kernel density estimates. en. *The Australian journal of statistics* **32**(3), 361–371.

Jones, MC, JS Marron & SJ Sheather (1992). *Progress in data-based bandwidth selection for kernel density estimation*. http://www.springer.com/statistics/journal/180.

Jones, MC, JS Marron & SJ Sheather (1996). A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American Statistical Association* **91**(433), 401–407. https://www.tandfonline.com/doi/abs/10.1080/01621459.1996.10476701.

Jones, MC & R. F. Kappenman (1992). On a Class of Kernel Density Estimate Bandwidth Selectors. *Scandinavian journal of statistics, theory and applications* **19**(4), 337–349.

McQueen, J, M Meilă, J VanderPlas & Z Zhang (2016). Megaman: Scalable Manifold Learning in Python. *J. Mach. Learn. Res.* **17**(148), 1–5.

Omar, S, A Ngadi & HH Jebur (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications in Technology* **79**(2).

Parzen, E (1962). On Estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics* **33**(3), 1065–1076.

Pelletier, B (2005). Kernel density estimation on Riemannian manifolds. *Statistics & probability letters* **73**(3), 297–304.

Perrault-Joncas, D & M Meila (2013). Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. arXiv: 1305.7255 [stat.ML].

Sain, SR, KA Baggerly & DW Scott (1994). Cross-Validation of Multivariate Densities. *Journal of the American Statistical Association* **89**(427), 807–817.

Scott, DW (1992). *Multivariate density estimation: theory, practice, and visualization*. New York: Wiley. https://openlibrary.org/books/OL1562497M/Multivariate_density_estimation.

Scott, DW (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. en. John Wiley & Sons. https://play.google.com/store/books/details?id=pIAZBwAAQBAJ.

Ten, CW, J Hong & CC Liu (2011). Anomaly detection for cybersecurity of the substations. *IEEE transactions on smart grid* **2**(4), 865–873.

Terrell, GR & DW Scott (1992). Variable Kernel Density Estimation. *Annals of statistics* **20**(3), 1236–1265.

Wand, MP & MC Jones (1994). *Kernel Smoothing*. en. CRC Press. https://play.google.com/store/books/details?id=GTOOi5yE008C.

Wand, MP, MC Jones, et al. (1994). Multivariate plug-in bandwidth selection. *Computational statistics* **9**(2), 97–116.

Zhou, X & M Belkin (2011). Semi-supervised Learning by Higher Order Regularization. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. Proceedings of Machine Learning Research. JMLR Workshop and Conference Proceedings, pp.892–900.