

Anomaly detection with kernel density estimation on manifolds

Abstract

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 3-D swiss roll or twin peaks data and 5-D semi-hypersphere meta data 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 highest densities and the most anomalous households with lowest densities. Both methods could identify the typical households with certain usage patterns, while the outliers are anomalous in different ways.

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

1 Introduction

In manifold learning, the underlying idea is that the data lies on a low-dimensional smooth manifold which 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 Metric Learning algorithm to augment any existing embedding output with geometric information in the Riemannian metric of the manifold itself. By applying the Metric Learning 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 defined at each point of the manifold is used to compute the geometric quantities, including angle, length, and volume, of the low-dimensional manifold embedding in any coordinate system, and be further applied to correct the distortion caused by the manifold learning algorithms. In variable kernel density estimate, the bandwidth matrix H is also defined to control the amount of smoothing for each data point. Therefore, if we could replace the bandwidth matrix with the Riemannian metric, 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 anomaly detection.

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 $\mathcal{L}_{\epsilon,n}$ (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) . The highlighted two steps in Figure 1 are the main contributions of this main chapter, replacing the bandwidth matrix 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, $\hat{f}()$, for anomaly detection.

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 Riemannian metric as the bandwidth of variable kernel density estimation, including the comparison with fixed bandwidth. Section 3 is composed with two simulations with the proposed algorithm; the first deals with a 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 4 contains the application to visualize and identify anomalies in the Irish smart meter dataset. Conclusions and discussions are presented in Section 5.

2 Variable kernel density estimation with manifold embeddings

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. For a high-dimensional data set, various manifold learning algorithms including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, are applied to get a low-dimensional embedding. The manifold learning algorithms map the points through nonlinear functions that stretches some regions of the space while shrinks others. Perrault-Joncas & Meila (2013) gives us an idea of how to measure the direction and angle of the distortion using the Riemannian metric and the Riemannian metric is a positive semi-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 lowest density estimates. The proposed schematic is shown in Figure 1.

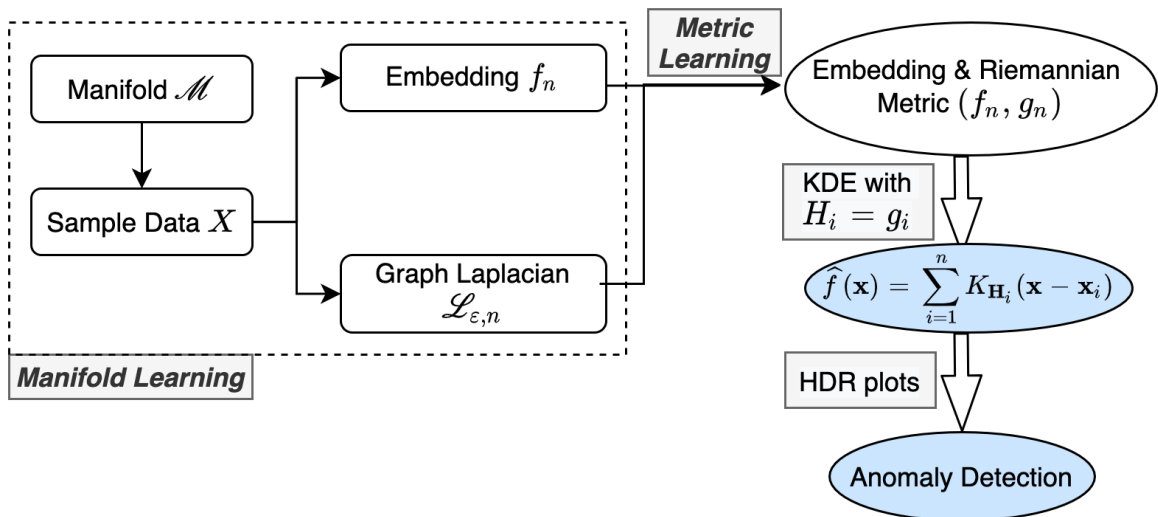


Figure 1: The proposed schematic for variable kernel density estimation with recovered geometry.

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 Notations

Denote the s -dimensional input data with observations $x_i, i = 1, 2, \dots, N$, where $x_i \in \mathbb{R}^s$. The underlying manifold \mathcal{M} has the true density f . Manifold learning takes the input data and finds its d -dimensional representation of data that lie on the manifold \mathcal{M} , with output data points $y_i, i = 1, 2, \dots, N$, where $y_i \in \mathbb{R}^d$ and $d \ll s$. The Riemannian metric of the embedding is denoted as $g_i, i = 1, 2, \dots, N$. In kernel density estimate, the pointwise variable bandwidth matrix H_i is used to estimate the density of the embedding, \hat{f} .

2.2 Kernel density estimator on Riemannian manifolds

In the case of non-Euclidean sample space, the observations often lie on a manifold with the differentiable structure called the Riemannian manifold. Consider a compact Riemannian manifold (\mathcal{M}, g) of dimension d without boundary and a probability distribution with density f on the manifold. Assume (\mathcal{M}, d_g) is a complete metric space, where d_g is the Riemannian distance induced by g and the strictly positive injectivity radius (Chavel 2006) of the manifold. Denote $A_i, i = 1, 2, \dots, N$, where $A_i \in \mathcal{M}$, as i.i.d. random objects on \mathcal{M} with density f . For each $p \in \mathcal{M}$, Pelletier (2005) proposed the kernel density estimator of f to be

$$f_N(p) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^d \theta_p(A_i)} K\left(\frac{d_g(p, A_i)}{h}\right), \quad (1)$$

where $\theta_p(x)$ is the volume density function on the manifold, $K : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a non-negative function, and h is the bandwidth. Pelletier (2005) discussed the volume density function $\theta_p(x)$ defined for p in a neighborhood of x on the manifold, with the geodesic normal coordinates at x , $\theta_p(x)$ is equal to the determinant of the Riemannian metric g at $\exp_p^{-1}(x)$. The expression in (1) is also proved to be consistent with the kernel density estimators in the Euclidean case as

$$f_N(p) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^d} K\left(\frac{\|p - A_i\|}{h}\right),$$

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

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 \mathcal{M} . Therefore, based on the estimator in (1), we

propose to a kernel density estimator of f with the manifold embedding $y_i, i = 1, 2, \dots, N$ to be

$$f_N(p) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_i^d |g(\exp_p^{-1}(y_i))|} K\left(\frac{d_g(p, y_i)}{h_i}\right), \quad (2)$$

where $h_i, i = 1, 2, \dots, N$ 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 (2). In this way, we present a consistent kernel density estimator with pointwise variable bandwidth as the Riemannian metric.

2.3 Multivariate kernel density estimation

In general, a multivariate variable kernel density estimate of f at point $\mathbf{x} \in \mathbb{R}^d$ is defined as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N K_{h_i}(\mathbf{x} - \mathbf{a}_i),$$

where $\mathbf{a}_i, i = 1, 2, \dots, N$ is the random sample drawn from a density f , \mathbf{h}_i is a $d \times d$ symmetric positive definite matrix that is variable at i , and K_{h_i} is the kernel function

$$K_{h_i}(\mathbf{x} - \mathbf{a}_i) = (2\pi)^{-d/2} |\mathbf{h}_i|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{a}_i)' \mathbf{h}_i^{-1} (\mathbf{x} - \mathbf{a}_i) \right],$$

with $|\mathbf{h}_i|$ being the determinant of matrix \mathbf{h}_i . The bandwidth matrix is important in controlling the smoothing across all data points and the shape is determined by the kernel function $K(\mathbf{x})$. An adaptive bandwidth could be used to control the amount of smoothing on the location \mathbf{x} [balloon estimator; Terrell & Scott (1992)] or the data point \mathbf{a}_i [sample smoothing estimator; Terrell & Scott (1992)]. We use the sample smooth estimator for each data point, which make a kernel density estimate with local smoothing. Duong (2007) present an R package *ks* that implement kernel density estimates with fixed bandwidth and variable bandwidth, but it is limited to bandwidth vector or diagonal matrix and is not applicable to full bandwidth matrix.

2.4 Riemannian metric as variable bandwidth

The Riemannian metric g is a symmetric and positive definite tensor field which defines an inner product $\langle \cdot, \cdot \rangle_g$ on the tangent space $T_p\mathcal{M}$ for every point $p \in \mathcal{M}$. 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 manifold. The Metric Learning algorithm (Perrault-Joncas & Meila 2013) then augment the embedded manifold with the Riemannian metric and produce a Riemannian manifold (\mathcal{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 $u, v \in T_p\mathcal{M}$, $\langle u, v \rangle_g = g_{ij}u^i v^j$ ¹, can be used to define some geometric quantities, such as the vector norm $\|u\| = \sqrt{\langle u, u \rangle_g}$ and the angle between two vectors $\cos \theta = \frac{\langle u, v \rangle_g}{\|u\| \|v\|}$ in the tangent space. Therefore, for each point $p \in \mathcal{M}$ 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 $c \in \mathcal{M}$ is defined as

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

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

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

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

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. 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.

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

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

megaman (McQueen et al. 2016). It is designed to apply the manifold learning methods to large-scale data sets, as well as computing the Riemannian metric of the manifold.

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

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

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

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

where $\tilde{D} = \text{diag}\tilde{W}\mathbf{1}$, $\tilde{W} = D^{-1}WD^{-1}$, and $D = \text{diag}W\mathbf{1}$;

- 3: Embed each data point $x_i \in \mathbb{R}^s$ to embedding coordinates $\mathbf{y} = (\mathbf{y}^1, \dots, \mathbf{y}^d)'$ by any existing manifold learning algorithm;
- 4: Obtain the matrix $\tilde{\mathbf{h}}$ of all data point by applying the graph Laplacian $\tilde{\mathcal{L}}_{\sqrt{\epsilon},N}$ to the embedding coordinates matrix \mathbf{y} with each element vector in $\tilde{\mathbf{h}}$ being

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

where $i, j = 1, \dots, d$ and the \cdot calculation is the elementwise product between two vectors;

- 5: Calculate the Riemannian metric \mathbf{h} as the rank d pseudo inverse of $\tilde{\mathbf{h}}$ with

$$\mathbf{h} = U \text{diag} 1/(\Lambda[1:d])U',$$

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

2.5 Algorithm for 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 $x_i, i = 1, \dots, N$ where $x_i \in \mathbb{R}$ to get the low-dimensional embedding y_i , where $y_i \in \mathbb{R}$ and the Riemannian metric \mathbf{H}_i with dimension $d \times d$ for each observation. Two parameters $\sqrt{\epsilon} = 0.4$ 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 (2). The top outliers of size n_{outliers} are obtained by ordering the embedding points y_i according to their density estimates $f_N(\mathbf{y}_i)$.

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 a 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 $i = 1, \dots, N$ **Output** : outliers embedding coordinates $y_1, \dots, y_{n_outliers}$ with their estimated densities $f_1, \dots, f_{n_outliers}$ **parameter:** number of outliers $n_outliers$, embedding dimension d

- 1: For all $i = 1, \dots, N$, compute the d -dimensional embeddings y_i with any existing manifold learning algorithms and the corresponding Riemannian metric \mathbf{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 $\mathbf{h}_i = \mathbf{g}_i$;
- 3: Compute the kernel density estimates $f_N(\mathbf{y}_i)$ for all $i = 1, \dots, N$ using Equation [refequ:denestimator](#);
- 4: Reorder the embedding coordinates \mathbf{y} according to the density estimates $f_N(\mathbf{y})$ and subset the top $n_outliers$ as the outliers.

3 Simulation

In this section, we exam two scenarios for both low and high dimensions to test our proposed algorithm. For visualization purpose, [Section 3.1](#) 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 $d = 2$, 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 3.2](#) 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 $d = 5$.

3.1 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 $\boldsymbol{\Sigma}_i = \text{diag}(0.02, 0.02)$, $i = 1, 2, 3, 4$. 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

$$P(\mathbf{X} = \mathbf{x}) = \sum_{i=1}^4 \pi_i \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_i|^{-1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right\}. \quad (3)$$

[Figure 2](#) shows the 2-dimensional meta data and the colors indicate the true density of all data points calculated from (3), 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.

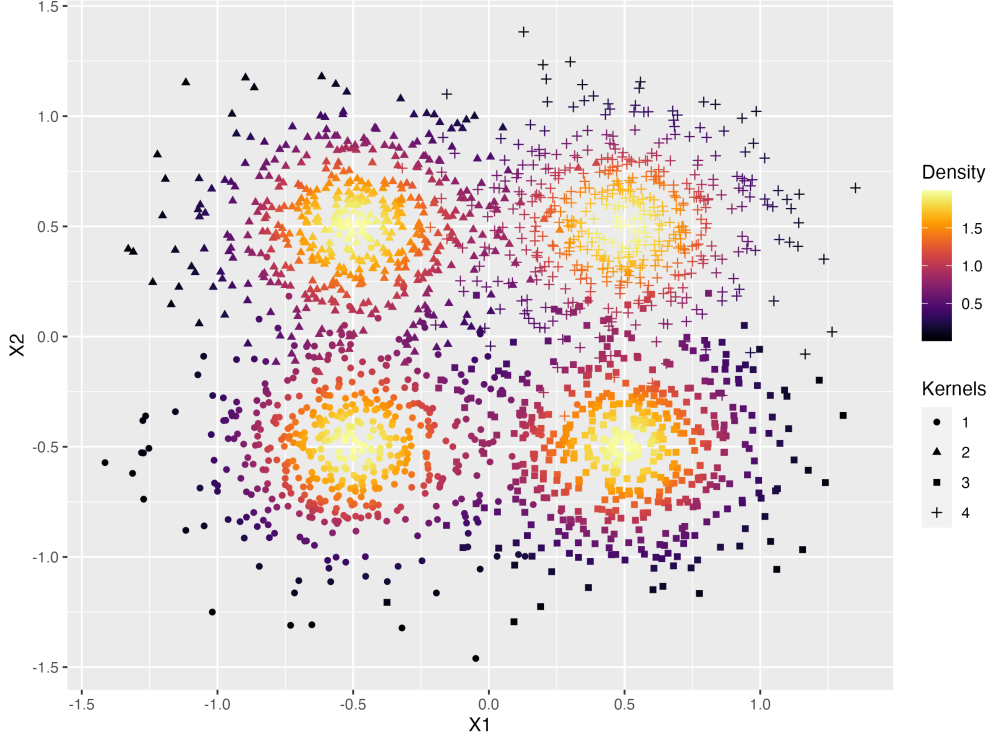


Figure 2: True density of the Gaussian mixture model of four kernels with means $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$, $(0.75, 0.75)$ and the same variance-covariance matrix $\text{diag}(0.02, 0.02)$. The colors indicate the density of the data and lower density points in darker colors are scattered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

3.1.1 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 example in manifold learning is the swiss roll data, with the mapping function in (4). The two-dimensional meta data $(\mathbf{X}_1, \mathbf{X}_2)'$ is transformed into the three-dimensional data $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})'$, shown in the left plot of Figure 3. The four colors in the mappings represents the four Gaussian kernels used to generate the meta data $(\mathbf{X}_1, \mathbf{X}_2)'$.

$$\begin{cases} X = X_1 \cos X_1, \\ Y = X_2, \\ Z = X_1 \sin X_1. \end{cases} \quad (4)$$

Now we are able to apply different manifold learning algorithms to $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})'$ and reduce the dimension back to $d = 2$, and further estimate the density of the 2-D embedding. According to

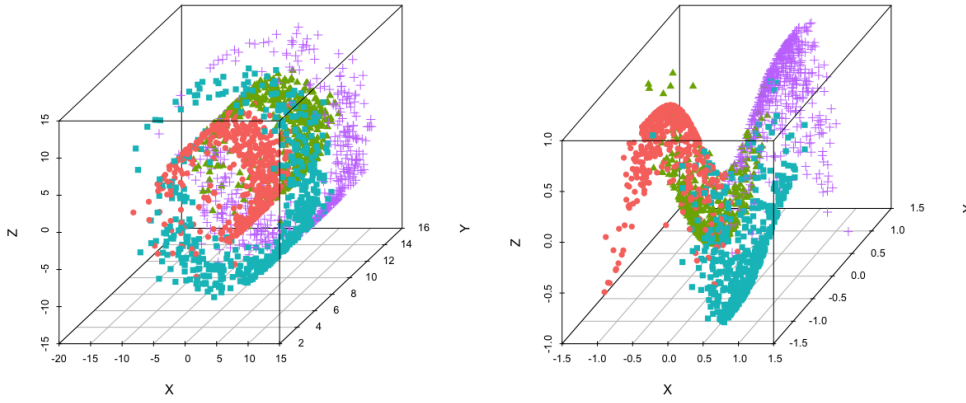


Figure 3: 3-D Mappings of the meta data with colors and shapes indicating the four kernels. Left: swiss roll mapping. Right: twin peak mapping.

the density estimates, we could rank the data points and then identify which observations lie in a 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 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 detect 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 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

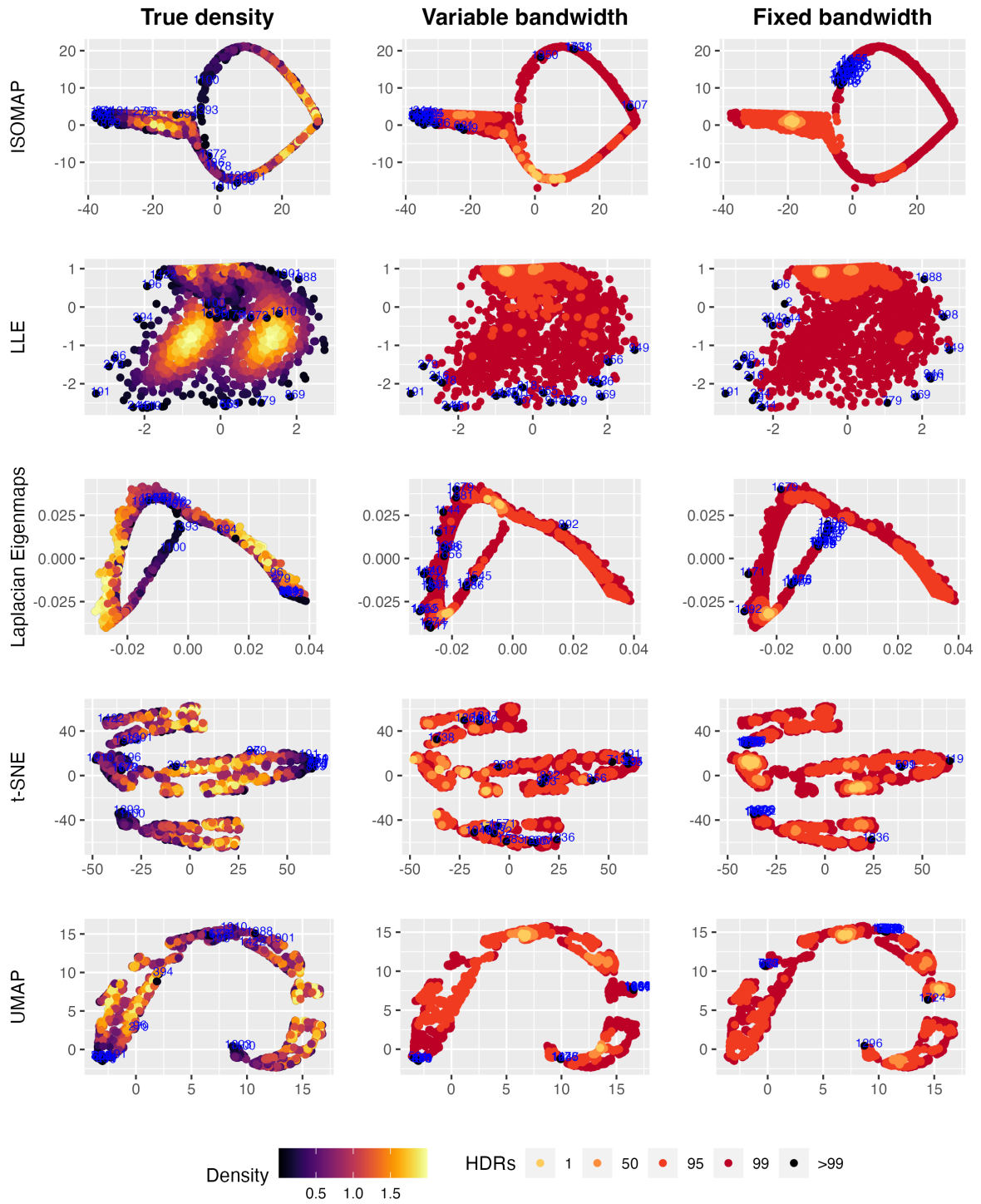


Figure 4: Highest density region plots of five manifold learning embeddings of the swiss roll data. Colors are indicating densities from left: true densities from the Gaussian mixture model; middle: KDE with Riemannian matrix as variable bandwidth; and right: KDE with fixed bandwidth. Variable KDE preforms 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 outperforms 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

comes from our method in LLE embedding, which is mainly due to it is closest to 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 the 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.

3.1.2 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 (5), with the corresponding 3-D structure shown in the right plot of [Figure 3](#).

$$\begin{cases} X = X_1, \\ Y = X_2, \\ Z = \sin(\pi X_1) \tanh(3X_2). \end{cases} \quad (5)$$

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 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 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

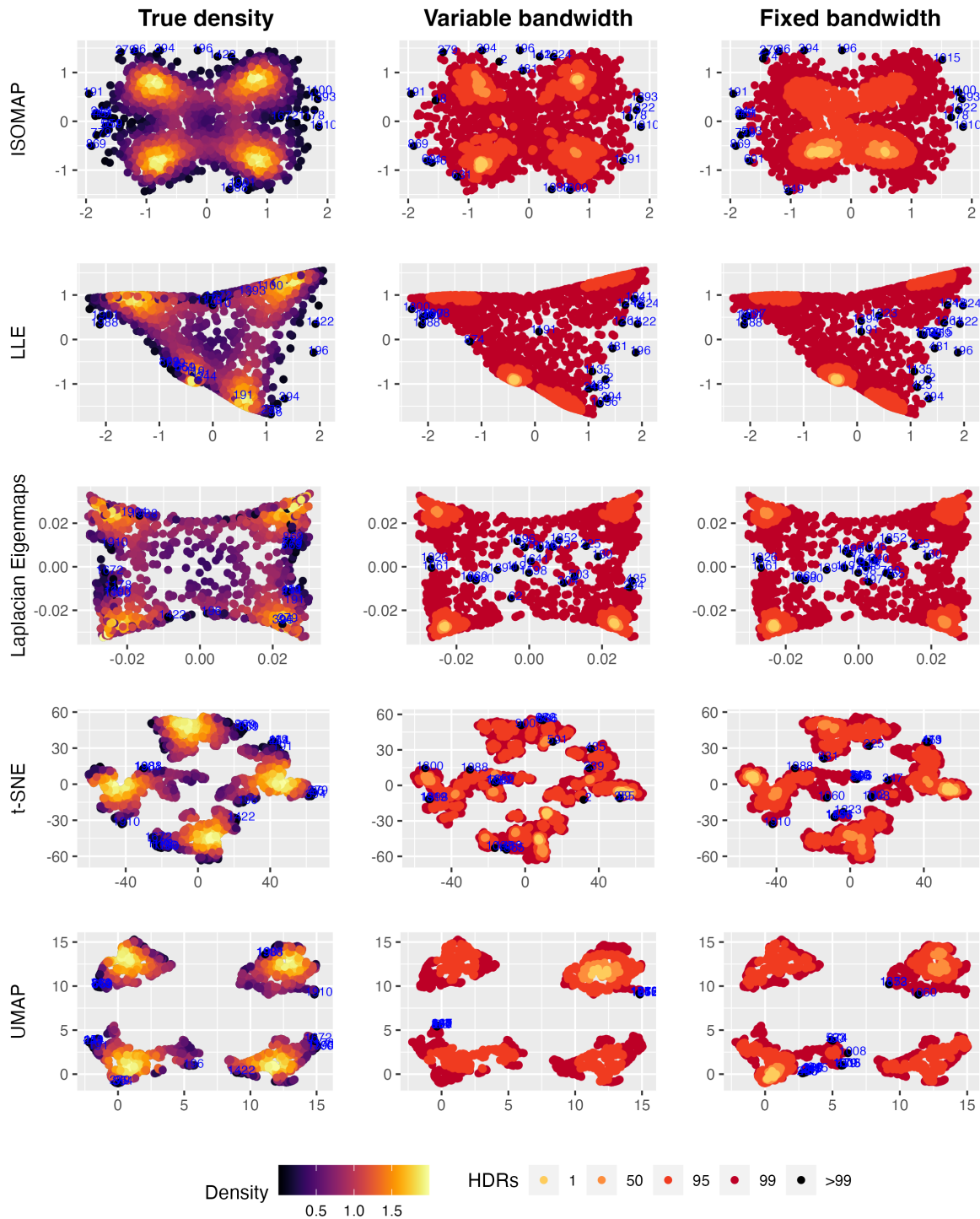


Figure 5: Highest density region plots of four manifold learning embeddings of the twin peak data. Variable KDE preforms better in finding kernel structures with ISOMAP and LLE, and in locating outliers with t-SNE and UMAP.

Table 2: Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the twin peak data. Variable bandwidth KDE outperforms 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

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 also 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 appears from embedding with higher quality, including ISOMAP, Laplacian Eigenmaps, and UMAP. The rank correlations between variable and fixed bandwidth is 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.

3.2 100-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, $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4)'$, from a 4-dimensional Gaussian mixture model with two mixture components, $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0, 0, 0, 0)'$, $\boldsymbol{\Sigma}_1 = \text{diag}(1, 1, 1, 1)$, and $\boldsymbol{\Sigma}_2 = \text{diag}(2, 2, 2, 2)$. 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

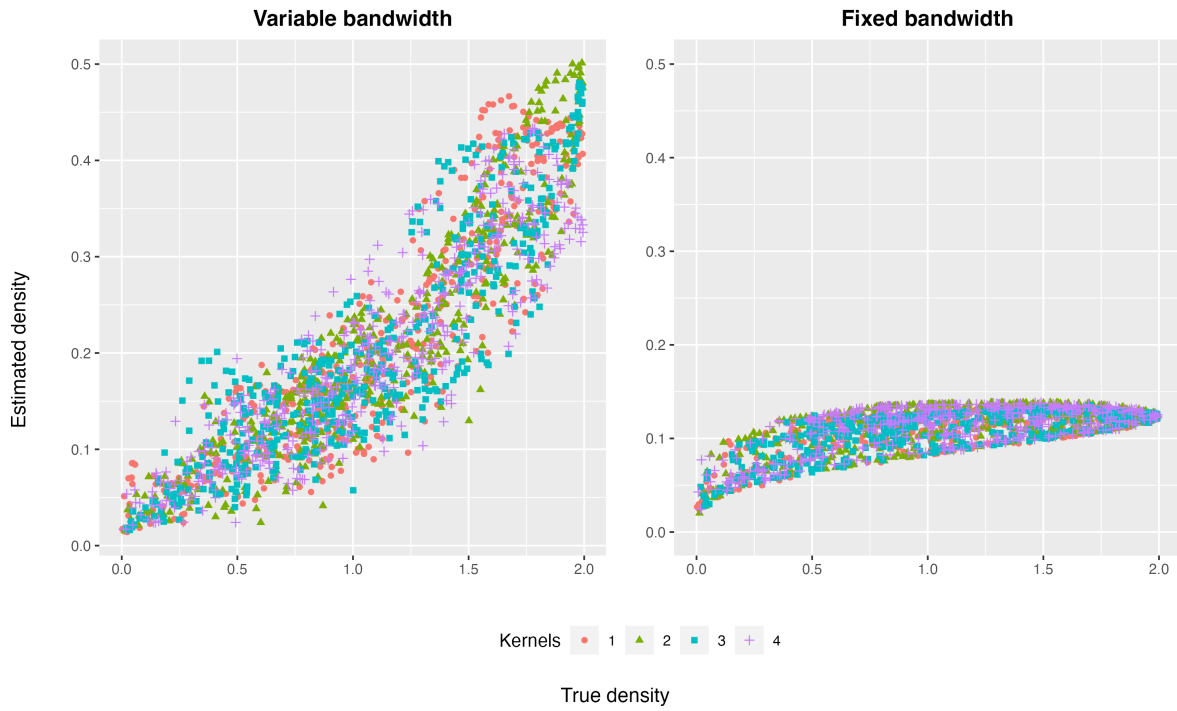


Figure 6: Scatterplot of true density and estimated density of ISOMAP embedding for KDE with both variable and fixed bandwidth. The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Variable bandwidth shows a strong linear positive relationship.

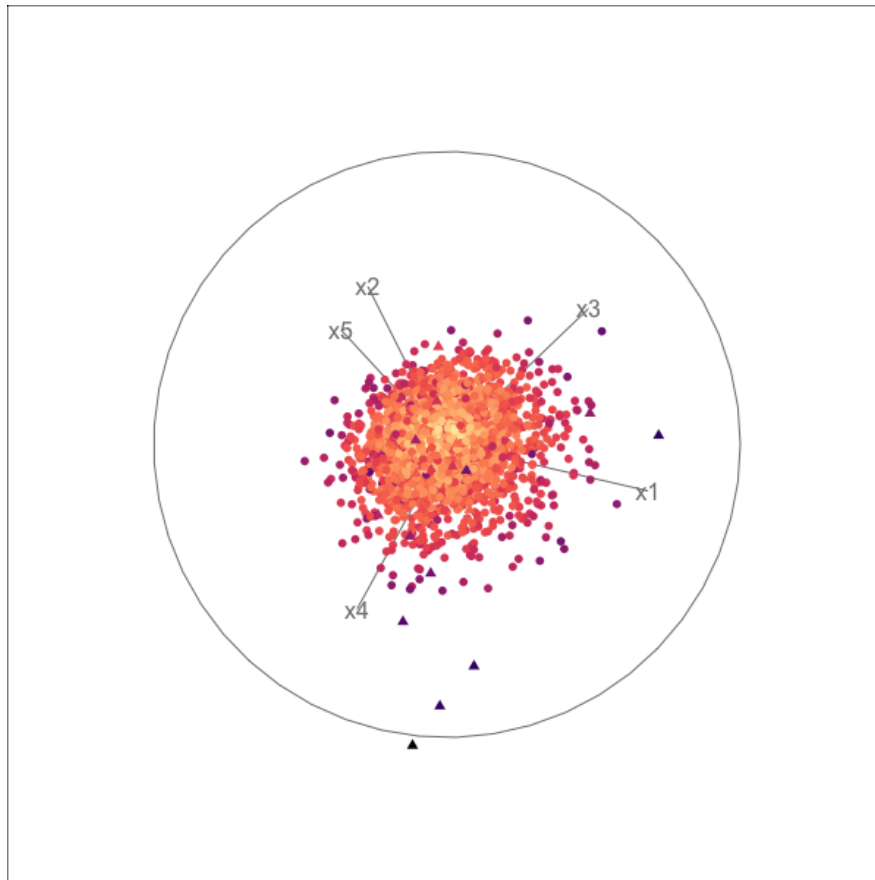


Figure 7: Scatterplot display of the animation of a 5-D tour path with shapes indexing the Gaussian mixture component and the colors showing the distance to the kernel cores.

$X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 = r^2$ where $X_5 > 0$ and r is set as 7. The Gaussian mixture densities could be calculated using Equation (3) 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 $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$, 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 triangular shape, meaning that they are simulated from $N(\mu_2, \Sigma_2)$. 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 \mathbf{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 \mathbf{A} with all components randomly generated from a uniform distribution $\mathcal{U}(0, 1)$. For any real matrix \mathbf{A} of dimension $p \times q$, the QR decomposition could decompose the matrix into the multiplication of two matrix \mathbf{Q} and \mathbf{R} so that $\mathbf{A} = \mathbf{QR}$, where the dimension of \mathbf{Q} is a matrix with unit norm orthogonal vectors, $\mathbf{Q}'\mathbf{Q} = \mathbf{I}$, and \mathbf{R} is an upper triangular matrix. Matrix \mathbf{Q} satisfies $\mathbf{X}_0'\mathbf{X} = (\mathbf{QX}_0)'(\mathbf{QX}_0)$, meaning that the pairwise Euclidean distances between data points in \mathbf{X}_0' is equivalent to that of $(\mathbf{QX}_0)'$. Therefore, we use matrix \mathbf{Q} as the rotation matrix for where the rotated data matrix $\mathbf{X} = (\mathbf{QX}_0)'$ 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 application to embedding dimension 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 suggest 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, 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 a fixed bandwidth, the bandwidth is often too large to

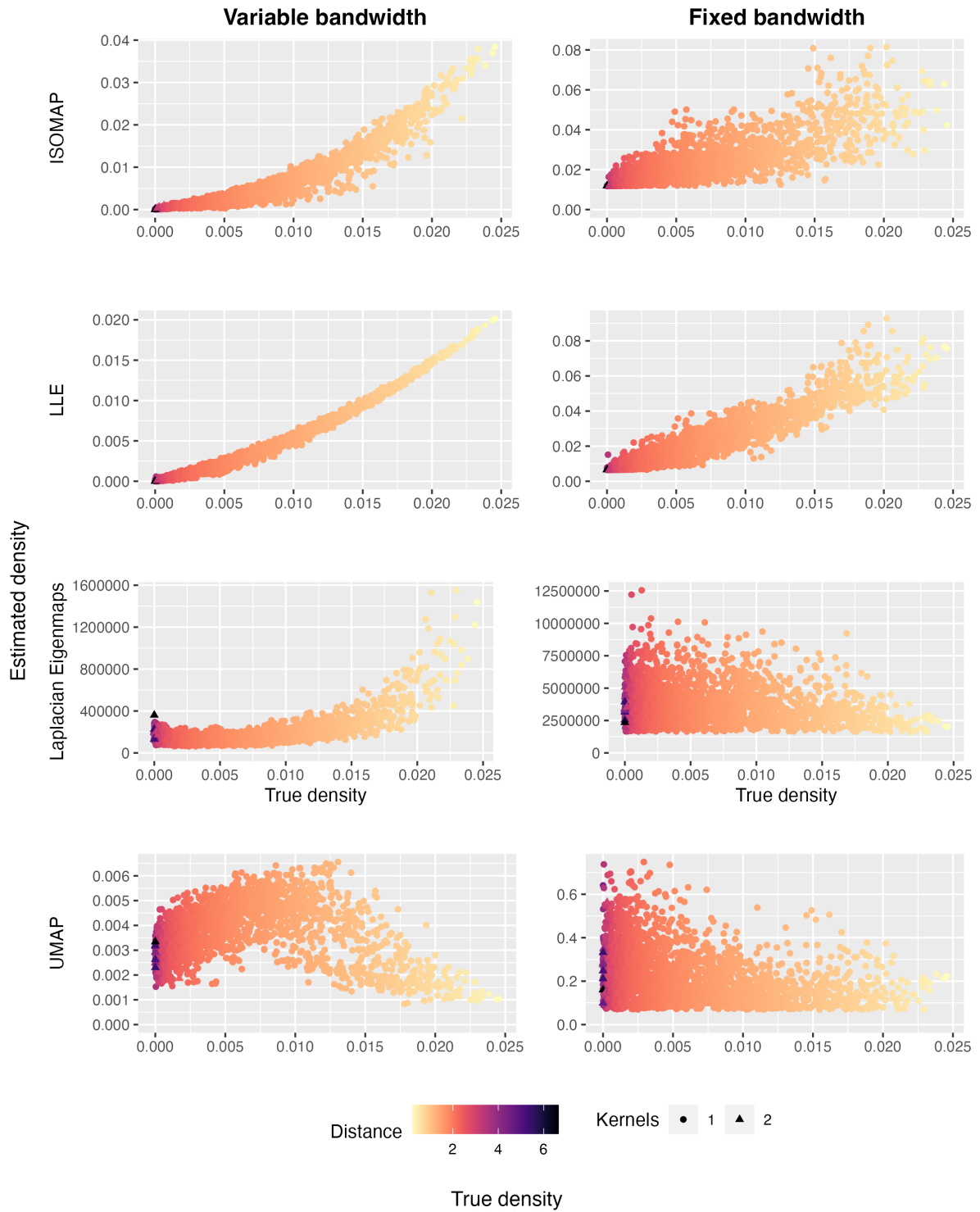


Figure 8: Scatterplot of true density and estimated density of different embeddings for KDE with both variable and fixed bandwidth. The point shapes indicates the two Gaussian mixture components and the colors shows the distance to the kernel cores.

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

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.

4 Application

4.1 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 dataset² 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 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 has been clearly stated in the application section of Cheng, Hyndman & Panagiotelis (2021) and they are

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

skipped here. Unlike the simulations in [Section 3](#), we know nothing about the true density of the electricity distributions for all periods of 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.

5 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 3-D swiss roll or twin peaks data and 5-D semi-hypersphere meta data 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 highest densities and the most anomalous households with 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.

References

- Chavel, I (2006). *Riemannian Geometry: A Modern Introduction*. en. Cambridge University Press, p. 108.
- 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 (2007). ks: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R. en. *Journal of statistical software* **21**, 1–16.
- Goldberg, Y, A Zakai, D Kushnir & Y Ritov (2008). Manifold Learning: The Price of Normalization. *J. Mach. Learn. Res.* **9**(Aug), 1909–1939.
- Henry, G & D Rodriguez (2009). Kernel Density Estimation on Riemannian Manifolds: Asymptotic Results. *Journal of mathematical imaging and vision* **34**(3), 235–239.
- 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.
- McQueen, J, M Meilă, J VanderPlas & Z Zhang (2016). Megaman: Scalable Manifold Learning in Python. *J. Mach. Learn. Res.* **17**(148), 1–5.
- 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 \[stat.ML\]](https://arxiv.org/abs/1305.7255).
- Terrell, GR & DW Scott (1992). Variable Kernel Density Estimation. en. *The Annals of Statistics* **20**(3), 1236–1265.

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.