Pattern Recognition Letters
journal homepage: www.elsevier.com

Nearest neighbour estimators of density derivatives, with application to mean shift clustering

Tarn Duong, Gaël Beck, Hanene Azzag, Mustapha Lebbah
Computer Science Laboratory (LIPN) UMR 7030
University Sorbonne Paris City (USPC), University of Paris 13, F-93430 Villetaneuse, France
Email: {beck, duong, hanane.azzag, mustapha.lebbah}@lipn.univ-paris13.fr

ABSTRACT

Nearest neighbour estimators of the general order derivatives of the probability density function are introduced. We establish their squared error consistency, and most importantly for data analysis, an automatic, single pass normal scale or 'rule of thumb' selector of the number of nearest neighbours. Density derivatives are crucial components for statistical unsupervised learning based on density gradient ascent known as mean shift clustering. The proposed automatic choice of the nearest neighbours for density gradients is applied to the mean shift clustering and is demonstrated to discover accurately the number, location and shape of non-ellipsoidal clusters in multivariate data analysis and image segmentation.

1. Introduction

Nearest neighbour estimators are well-suited to multivariate data analysis as they intuitively adapt to the local data density. Nearest neighbour estimators of the probability density function were introduced in the seminal papers [26, 27] and have been widely used since due to their ease of implementation and interpretation. Derivatives of the density function are important quantities to analyse as they provide supplementary information about the data set which is not revealed by the density function on its own. Estimators of the first derivative (gradient) have been considered [16], whereas higher order derivatives have not yet been considered.

We set up a framework for nearest neighbour estimators for the general r-th order derivatives of multivariate density functions. This is achieved by following recent work in kernel estimators of density derivatives [6] and by exploiting the connection between nearest neighbour and variable kernel estimators [21]. Whilst variable kernel estimators of the density gradient [8, 10] are mathematically similar to their nearest neighbour analogues above, the key difference is that the former suffer from the data sparsity in higher dimensions whereas the latter do not. Modifications of kernel estimators have been proposed to overcome the data sparsity problem via reduced set density estimators [11, 17], though these authors did not consider the extension to density derivatives. We do not pursue this extension as we focus on nearest neighbour estimators.

The local adaptivity of nearest neighbour estimators is controlled by a single scalar parameter, namely the number of the nearest neighbours k. Almost all of the current proposals for selecting k are based on cross-validation [1, 23, 25] or grid searches [23] of the density function, with a scarce focus on the density derivatives. Furthermore, these multiple pass approaches are computationally intensive as they evaluate the optimality criteria based on (almost) the entire data set for a sequence of candidate values of k. In contrast, we propose a single pass automatic selector for estimating a general derivative of order r of the density function.

Density derivatives, whilst being important quantities to estimate in their own right, are also crucial components in statistical machine learning methods, such as the mean shift clustering. This was introduced by [16] as a more flexible, nonparametric alternative to the classic k-means clustering. The k-means clustering is the most widely used method for clustering multivariate data, despite that its limitations are well-known. The main advantages of the mean shift over the k-means is that the former (a) can discover clusters of arbitrary shape and (b) exploits the gradient ascent directionality in addition to the inter-point distances to form more representative clusters.

The more widespread use of nearest neighbour methods for

mean shift clustering has been hampered by the lack of an efficient selector for the number of nearest neighbours. Analogous to above choices of k for density estimation, the choice of k for clustering follows the cross validation and grid- based searches with respect to minimising clustering quality indices e.g. the Silhouette index [22]. These are time consuming, even for moderately large data sets, as the (almost) entire data set must be clustered for each value of k in a sequence of candidate values, in order to find a global optimum of the clustering quality indices. Our proposed automatic nearest neighbour mean shift clustering (NNMS) uses the k which is optimal for the density gradient estimation, in conjunction with the nearest neighbour estimators of the density and density gradient. We demonstrate that this k is an efficient empirical choice for clustering.

In Section 2, we introduce the analysis of nearest neighbour estimators of density derivatives by drawing upon their connection to variable kernel estimators. Within this framework, we define an automatic, single pass normal scale selector of the optimal number of nearest neighbours. In Section 3, we apply this to the mean shift clustering algorithm. In Section 4, we examine the finite sample behaviour of our proposed nearest neighbour mean shift clustering NNMS in comparison to other modal clustering methods for simulated and experimental data. The appendix contains the proofs of the mathematical results in Section 2.

2. Density derivative estimation

The nearest neighbour estimator of a density function, as introduced by [26] and elaborated by [27], is

$$\hat{f}(\boldsymbol{x}; k) = 1/[n\delta_{(k)}(\boldsymbol{x})^d] \sum_{i=1}^{n} K((\boldsymbol{x} - \boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x})) \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d)$, $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{id})$, and $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are a d-variate random sample drawn from a common density f. Eq. (1) is the mathematically most general form of a nearest neighbour density estimator as the kernel K can be any symmetric multivariate density function. The mathematical analysis of nearest neighbour estimators is simplified if we recast them as variable kernel estimators [21]. A variable multivariate kernel estimator $\tilde{f}$ with a variable bandwidth matrix function $\mathbf{H}(\boldsymbol{x})$ of the density is

$$\tilde{f}(\boldsymbol{x}; \mathbf{H}(\boldsymbol{x})) = n^{-1}|\mathbf{H}(\boldsymbol{x})|^{-1/2} \sum_{i=1}^{n} K(\mathbf{H}(\boldsymbol{x})^{-1/2}(\boldsymbol{x} - \boldsymbol{X}_i)),$$

and of the r-th density derivative is

$$\begin{aligned}&\mathsf{D}^{\otimes r}\tilde{f}(\boldsymbol{x}; \mathbf{H}(\boldsymbol{x}))\\ &\quad = n^{-1}|\mathbf{H}(\boldsymbol{x})|^{-1/2}(\mathbf{H}(\boldsymbol{x})^{-1/2})^{\otimes r} \sum_{i=1}^{n} \mathsf{D}^{\otimes r} K(\mathbf{H}(\boldsymbol{x})^{-1/2}(\boldsymbol{x} - \boldsymbol{X}_i))\end{aligned}$$

where the differentiation of K with respect to $\boldsymbol{x}$ is carried out keeping $\mathbf{H}(\boldsymbol{x})$ constant, and that the dependence on $\boldsymbol{x}$ is only reinstated after differentiation, employing an approach similar to [16]. The $\otimes r$ superscript indicates an r-fold Kronecker product, thus the r-th derivative $\mathsf{D}^{\otimes r}$ is organised as a d^r-vector arising from an r-fold Kronecker product of the differential operator $\mathsf{D} = [(\partial/\partial x_1), \ldots, (\partial/\partial x_d)]$, see [19]. The connection between nearest neighbour and variable kernel estimators [26, 27] appears when $\mathbf{H}(\boldsymbol{x}) = \delta_{(k)}(\boldsymbol{x})^2\mathbf{I}_d$. This implies that the nearest neighbour estimator of the r-th derivative of f follows as

$$\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x}; k) = n^{-1}\delta_{(k)}(\boldsymbol{x})^{-d-r} \sum_{i=1}^{n} \mathsf{D}^{\otimes r} K(\delta_{(k)}(\boldsymbol{x})^{-1}(\boldsymbol{x} - \boldsymbol{X}_i)). \tag{2}$$

Writing nearest neighbour estimators in this form in Eq. (2) greatly facilitates the task for optimal selection of the number of nearest neighbours.

The most common criterion utilised for optimal smoothing is the asymptotic mean integrated squared error (AMISE), which is the leading asymptotic term of the integral of the mean squared error between the target quantity and the estimator. We start with the AMISE of the fixed bandwidth kernel estimator $\mathsf{D}^{\otimes r}\tilde{f}(\cdot; \mathbf{H})$, i.e., $\mathrm{AMISE}\,[\mathsf{D}^{\otimes r}\tilde{f}(\cdot; \mathbf{H})]\{1 + o(1)\} = \int_{\mathbb{R}^d} \mathbb{E}[\mathsf{D}^{\otimes r}\tilde{f}(\boldsymbol{x}; \mathbf{H}) - \mathsf{D}^{\otimes r} f(\boldsymbol{x})]^2\, d\boldsymbol{x}$, as established by [7, Theorem 2]. This can be rewritten as

$$\begin{aligned}\mathrm{AMISE}\,[\mathsf{D}^{\otimes r}\tilde{f}(\cdot; \mathbf{H})] &= n^{-1}|\mathbf{H}|^{-1/2}\,\mathrm{tr}((\mathbf{H}^{-1})^{\otimes r}\mathbf{R}(\mathsf{D}^{\otimes r}K))\\ &\quad + (-1)^r \tfrac{1}{4} m_2(K)^2 \boldsymbol{\psi}_{2r+4}^T(\mathrm{vec}\,\mathbf{I}_{d^r} \otimes \mathrm{vec}\,\mathbf{H}^{\otimes 2})\end{aligned}$$

where $\mathbf{R}(\mathsf{D}^{\otimes r}K) = \int_{\mathbb{R}^d} \mathsf{D}^{\otimes r}K(\boldsymbol{x})\mathsf{D}^{\otimes r}K(\boldsymbol{x})^T\, d\boldsymbol{x}$, and $m_2(K)\mathbf{I}_d = \int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^T K(\boldsymbol{x})\, d\boldsymbol{x}$ and $\boldsymbol{\psi}_{2r+4} = \int_{\mathbb{R}^d} \mathsf{D}^{\otimes(2r+4)} f(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x}$. Replacing $\mathbf{H}$ by $\mathbf{H}(\boldsymbol{x}) = \delta_{(k)}(\boldsymbol{x})^2\mathbf{I}_d$ results in a random quantity, so we compute its expectation to derive an AMISE-like quantity for the nearest neighbour density derivative estimator $\mathsf{D}^{\otimes r}\hat{f}(\cdot; k)$,

$$\begin{aligned}&\mathrm{A}[\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x}; k)]\\ &\quad = \mathbb{E}\{\mathrm{AMISE}\,[\mathsf{D}^{\otimes r}\tilde{f}(\cdot; \delta_{(k)}(\boldsymbol{x})^2\mathbf{I}_d)]\}\\ &\quad = \mathrm{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))[v_0 f(\boldsymbol{x})]^{(d+2r)/d} n^{2r/d} k^{-(d+2r)/d}\\ &\qquad + (-1)^r \tfrac{1}{4} m_2(K)^2 \boldsymbol{\psi}_{2r+4}^T(\mathrm{vec}\,\mathbf{I}_d)^{\otimes(r+2)}[v_0 f(\boldsymbol{x})]^{-4/d} n^{-4/d} k^{4/d}\end{aligned} \tag{3}$$

where $v_0 = \pi^{d/2}/\Gamma((d+2)/d)$ is the hyper-volume of the unit d-ball. The derivation of Eq. (3) is contained in the proof of the Theorem 1 in the Appendix. As the first term is the integrated variance and the second term is the integrated squared bias of $\mathsf{D}^{\otimes r}\hat{f}$, the role of k in a bias-variance trade-off is established in Eq. (3). So $\mathrm{A}[\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x}; k)]$ is a suitable basis for an optimality criterion. We obtain in Theorem 1 a closed form expression of

$$k_{\mathrm{A},r} = \int_{\mathbb{R}^d} \left\{\operatorname*{argmin}_{k>0} \mathrm{A}[\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x}; k)]\right\} d\boldsymbol{x}$$

which serves as an optimal number of the nearest neighbours.

Theorem 1. *Suppose that the conditions (A1–A3) in the Appendix hold. An optimal number of the nearest neighbours for* $\mathsf{D}^{\otimes r}\hat{f}$ *is* $k_{\mathrm{A},r} = C_r n^{(2r+4)/(d+2r+4)}$ *where*

$$C_r = v_0\left[\frac{(d+2r)\,\mathrm{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))}{(-1)^r m_2(K)^2 \boldsymbol{\psi}_{2r+4}^T(\mathrm{vec}\,\mathbf{I}_d)^{\otimes(r+2)}}\right]^{d/(d+2r+4)}.$$

When $K = f = \phi$*, where* ϕ *is the standard normal density, this yields the normal scale selector*

$$k_{\mathrm{NS},r} = v_0\left[\frac{4}{d+2r+2}\right]^{d/(d+2r+4)} n^{(2r+4)/(d+2r+4)}.$$

This $k_{\mathrm{NS},r}$ is closely related to the normal scale bandwidth selector $[4/(d+2r+2)]^{2/(d+2r+4)}\, n^{2/(d+2r+4)}\mathbf{I}_d$ for the kernel estimator $\mathsf{D}^{\otimes r}\tilde{f}$ in [7, Theorem 6].

Fig. 1 illustrates the importance of selecting a suitable value of k. The contours of the target density gradient ($r = 1$) of the standard normal bivariate density is in Fig. 1(a). The nearest neighbour estimate with the normal scale selector $k = k_{\mathrm{NS},1} = 505$ in Fig. 1(b) produces a similar structure as the target contours. If a much smaller $k = 50$ is utilised, the resulting estimate in Fig. 1(c) is considered to be undersmoothed as it is too noisy. If a much larger $k = 1000$ is utilised, the resulting estimate in Fig. 1(d) is considered to be oversmoothed as it displays insufficient detail. The kernel utilised is the Epanechnikov kernel $K(\boldsymbol{x}) = [(d+2)/(2v_0)](1-\boldsymbol{x}^T\boldsymbol{x})\mathbf{1}\{\|\boldsymbol{x}\| \le 1)\}$.

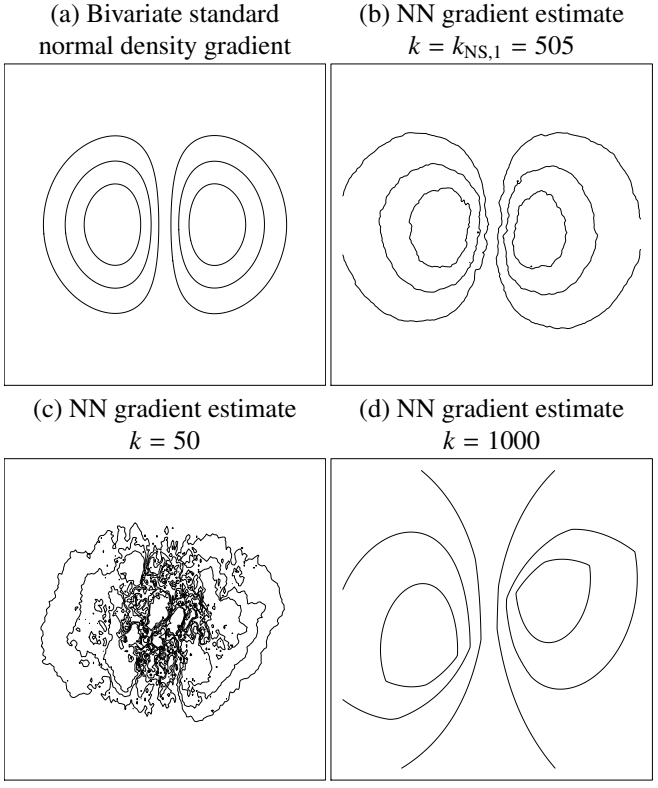

Fig. 1. Effect of the choice of k for nearest neighbour density gradient estimates for an $n = 1000$ random sample from the bivariate standard normal density. (a) Contours of the target bivariate standard normal density gradient. (b) Nearest neighbour density gradient estimate with $k = k_{\mathrm{NS},1} = 505$. (c) Nearest neighbour density gradient estimate with $k = 50$. (d) Nearest neighbour density gradient estimate with $k = 1000$.

The pioneering work of [15, 26] established the oracle local and global mean squared error optimal selectors for density estimators, though these authors did not consider data-based selectors. Perhaps [25] is the first to consider automatic data-based selection for nearest neighbour estimators, in the context of cross validation for regression. Authors who have proposed cross validation selectors for density estimation include [1, 23]. The latter authors [23] also suggest a grid based search for k. We observe that these are multiple pass methods. In contrast, we propose an efficient, single pass fully automatic selector for the nearest neighbour estimator of a general order r of the density derivative in Theorem 1.

3. Mean shift clustering

Whilst estimators of the density derivatives are important in their own right, they also serve as crucial components in other statistical analysis problems such as clustering. Modal clustering methods [4, 24], where the clusters are defined as basins of attractions to the modes in the density function [32], include the classic k-means and the more recent mean shift [16]. The k-means method aims directly at estimating the number and location of the ellipsoidal clusters, by minimising intra- and maximising inter-cluster distances, where each ellipsoidal cluster is identified to the mode of its normal mixture density component. Mean shift proceeds in an alternative, indirect manner based on local gradients, and without imposing an ellipsoidal shape to the clusters. From a candidate point $\boldsymbol{x}$, the mean shift method generates a sequence of points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots\}$ which follows the gradient density ascent using the recurrence relation

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \frac{\delta_{(k)}(\boldsymbol{x}_j)^2}{d+2}\frac{\mathsf{D}f(\boldsymbol{x}_j)}{f(\boldsymbol{x}_j)} \tag{4}$$

for $j \ge 1$ and $\boldsymbol{x}_0 = \boldsymbol{x}$. Eq. (4) implies that the gradient is normalised by the density. For regions of low density, this has the effect of increasing the step size, and is the basis of its fast convergence compared to unnormalised gradient methods [16].

It was established in [26] that the beta family kernels are computationally efficient for estimating f and $\mathsf{D}f$ in Eq. (4). The nearest neighbour density estimator in Eq. (1) becomes

$$\hat{f}(\boldsymbol{x};k) = n^{-1}\sum_{i=1}^{n}\mathbf{1}\{\boldsymbol{X}_i \in B_d(\boldsymbol{x}, \delta_{(k)}(\boldsymbol{x}))\} = k/[v_0 n \delta_{(k)}(\boldsymbol{x})^d]$$

when using the zeroth order beta kernel $K(\boldsymbol{x};0) = v_0^{-1}\mathbf{1}\{\boldsymbol{x} \in B_d(\mathbf{0},1)\}$, which is the uniform kernel on the unit d-ball $B_d(\mathbf{0},1)$. The summation counts the number of data points which fall inside $B_d(\boldsymbol{x}, \delta_{(k)}(\boldsymbol{x}))$, which is equal to k from the definition of $\delta_{(k)}(\boldsymbol{x})$ as the k-th nearest neighbour distance to $\boldsymbol{x}$. The nearest neighbour estimator in Eq. (2) for the density gradient becomes

$$\begin{aligned}\mathsf{D}\hat{f}(\boldsymbol{x};k) &= \hat{f}(\boldsymbol{x};k)\frac{d+2}{\delta_{(k)}(\boldsymbol{x})^2}\Big[\frac{1}{k}\sum_{i=1}^{n}\boldsymbol{X}_i\mathbf{1}\{\boldsymbol{X}_i \in B_d(\boldsymbol{x}, \delta_{(k)}(\boldsymbol{x}))\} - \boldsymbol{x}\Big] \\ &= \hat{f}(\boldsymbol{x};k)\frac{d+2}{\delta_{(k)}(\boldsymbol{x})^2}\Big[\frac{1}{k}\sum_{\boldsymbol{X}_i \in k\text{-nn}(\boldsymbol{x})}\boldsymbol{X}_i - \boldsymbol{x}\Big]\end{aligned}$$

where k-nn$(\boldsymbol{x}) = \{\boldsymbol{X}_i : \boldsymbol{X}_i \in B_d(\boldsymbol{x}, \delta_{(k)}(\boldsymbol{x}))\}$ is the set of the k nearest neighbours to $\boldsymbol{x}$, when using the first order beta kernel $K(\boldsymbol{x};1) = [(d+2)/(2v_0)](1-\boldsymbol{x}^T\boldsymbol{x})\mathbf{1}\{\boldsymbol{x} \in B_d(\mathbf{0},1)\}$ is the Epanechnikov (or quadratic) kernel, with derivative $\mathsf{D}K(\boldsymbol{x};1) = -[(d+2)/v_0]\boldsymbol{x}\,\mathbf{1}\{\boldsymbol{x} \in B_d(\mathbf{0},1)\}$.

Replacing $\mathsf{D}f(\boldsymbol{x})/f(\boldsymbol{x})$ by its estimator $\mathsf{D}\hat{f}(\boldsymbol{x};k)/\hat{f}(\boldsymbol{x};k)$ in Eq. (4) we have that

$$\boldsymbol{x}_{j+1} = \frac{1}{k}\sum_{\boldsymbol{X}_i \in k\text{-nn}(\boldsymbol{x}_j)}\boldsymbol{X}_i \tag{5}$$

This recurrence relation was introduced by [16], beginning from a different starting point to us. Eq. (5) gives the mean shift method its name since the current iterate $\boldsymbol{x}_j$ is shifted to

the sample mean of its k nearest neighbours in the next iterate $\boldsymbol{x}_{j+1}$. The convergence of the sequence $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots\}$ to a local mode for the kernel version of Eq. (5) for a wide class of kernels was established by [9, Theorem 1] for fixed bandwidths. Their proofs remain valid when the fixed bandwidth is replaced with a number of nearest neighbours that decreases as the iteration number increases.

Fig. 2 illustrates Eq. (5) for a random sample of $n = 1000$ points drawn from a bivariate crescent density [6], which has a mode at the halfway point in the crescent. The candidate point is $\boldsymbol{x} = (-1.27, 0.30)$, and the mean shift is initialised with $\boldsymbol{x}_0 = \boldsymbol{x}$. The number of nearest neighbours is $k = k_{\mathrm{NS},1} = 505$. Applying one iteration, we obtain $\boldsymbol{x}_1 = (-0.70, 1.10)$. The step size from $\boldsymbol{x}_0$ is large since $\boldsymbol{x}_0$ is located in a low density region. The algorithm terminates at $\boldsymbol{x}_{23} = (0.18, 1.34)$. The mean shift gradient ascent path is given by the blue arrows and black circles as the iterations converge. The step sizes decrease in size as they approach the mode located in a high density region.

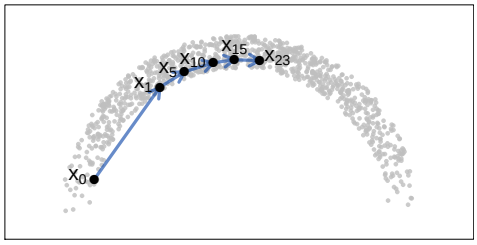

Fig. 2. Nearest neighbour mean shift NNMS for an $n = 1000$ random sample from the bivariate crescent density, with $k = k_{\mathrm{NS},1} = 505$. The data sample are the solid grey circles, the candidate point $\boldsymbol{x}_0$ is the solid black circle, and $\boldsymbol{x}_{23}$ is the final iterate.

The gradient ascent paths towards the local modes produced by Eq. (5) form the basis of Algorithm 1, our nearest neighbour mean shift clustering method NNMS. The inputs to NNMS are the data sample $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ and the candidate points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ which we wish to cluster (these can be the same as $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$, but this is not required); and the tuning parameters: the number of nearest neighbours k, the tolerance under which subsequent iterations in the mean shift update are considered convergent ε_1, the maximum number of iterations $j_{\max}$, the tolerance under which two cluster centres are considered to form a single cluster ε_2, and the minimum cluster size $s_{\min}$. The output are the cluster labels of the candidate points $\{\mathrm{c}(\boldsymbol{x}_1), \dots, \mathrm{c}(\boldsymbol{x}_m)\}$. There are three main sub-routines to Algorithm 1. Lines 1–6 correspond to gradient ascent paths in Eq. (5) which are iterated until subsequent iterates are less than ε_1 apart or the maximum number of iterations $j_{\max}$ is reached. The output from these lines are the final iterates $\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_m^*$. Lines 7–8 concern merging the final iterates within ε_2 distance of each other into a single cluster, thus creating an initial clustering of $\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_m^*$. In Lines 9–13, if the smallest cluster is less than the minimum cluster size $s_{\min}$, then it is iteratively merged into next nearest cluster, to produce cluster labels $\mathrm{c}(\boldsymbol{x}_1^*), \dots, \mathrm{c}(\boldsymbol{x}_m^*)$. Line 14 assigns these cluster labels to the original data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$.

Data-based bandwidth selection based on the density gradient has been demonstrated to be more suitable than that based on the density itself for kernel mean shift clustering in [6]. It follows that an optimal number of nearest neighbours mean shift can be

$$k_{\mathrm{NS},1} = v_0[4/(d+4)]^{d/(d+6)} n^{6/(d+6)}. \tag{6}$$

A grid based search for k which minimises clustering quality indices was proposed in [35], and which is $O(n)$. Our selector is $O(1)$ as it does not require this multiple pass approach.

Algorithm 1 NNMS Nearest neighbour mean shift

Input: $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_n\}, \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\}, k, \varepsilon_1, j_{\max}, \varepsilon_2, s_{\min}$
Output: $\{\mathrm{c}(\boldsymbol{x}_1), \dots, \mathrm{c}(\boldsymbol{x}_m)\}$
/* Compute gradient ascent paths */
1: **for** $\ell := 1$ to m **do**
2: $\quad j := 0$; $\boldsymbol{x}_{\ell,0} := \boldsymbol{x}_\ell$;
3: $\quad \boldsymbol{x}_{\ell,1} :=$ mean of k-nn of $\boldsymbol{x}_{\ell,0}$;
4: $\quad$ **while** $\|\boldsymbol{x}_{\ell,j+1}, \boldsymbol{x}_{\ell,j}\| > \varepsilon_1$ **or** $j < j_{\max}$ **do**
5: $\quad\quad j := j + 1$; $\boldsymbol{x}_{\ell,j+1} :=$ mean of k-nn of $\boldsymbol{x}_{\ell,j}$;
6: $\quad \boldsymbol{x}_\ell^* := \boldsymbol{x}_{\ell,j}$;
/* Create clusters by merging near final iterates */
7: **for** $\ell_1, \ell_2 := 1$ to m **do**
8: $\quad$ **if** $\|\boldsymbol{x}_{\ell_1}^* - \boldsymbol{x}_{\ell_2}^*\| \le \varepsilon_2$ **then** $\mathrm{c}(\boldsymbol{x}_{\ell_1}^*) := \mathrm{c}(\boldsymbol{x}_{\ell_2}^*)$;
/* Merge small clusters */
9: $C^* :=$ cluster with minimum cardinality;
10: **while** $\mathrm{card}(C^*) < s_{\min}$ **do**
11: $\quad C' :=$ nearest other cluster to C^*;
12: $\quad$ **for** $\boldsymbol{x}_\ell^* \in C^*$ **do** $\mathrm{c}(\boldsymbol{x}_\ell^*) := \mathrm{c}(C')$;
13: $\quad C^* :=$ cluster with minimum cardinality;
14: **for** $\ell := 1$ to m **do** $\mathrm{c}(\boldsymbol{x}_\ell) := \mathrm{c}(\boldsymbol{x}_\ell^*)$;

4. Data analysis

4.1. Simulated data

For mean shift clustering, we set the tuning parameters as follows: the mean shift iteration tolerance ε_1 is 0.005 times the maximum marginal data range, the maximum number of mean shift iterations is $j_{\max} = 100$, the cluster merging tolerance $\varepsilon_2 = 10\varepsilon_1$, and minimum cluster size is $s_{\min} = 50$. With the number of nearest neighbours $k = k_{\mathrm{NS},1}$ from Eq. (6), this is labelled as NNMS. An alternative nearest neighbour median shift clustering method, based on replacing the sample mean of the k nearest neighbours in Eq. (5) by a component-wise sample median and the choice of k based on a grid search to minimise the silhouette index [35] is labelled NNMS2. The kernel mean shift clustering with the plug-in selector [6] is labelled as KMS. The 'gold standard' parametric clustering method is the k-means, with a BIC method for selecting the number for normal mixture components [13] is labelled KM. We restrict ourselves to this small number of competing clustering methods as they are conveniently available as public `R` packages in order to be able to compare computation times: nearest neighbour median shift in `clues` [35], kernel mean shift in `ks` [12] and k-means in `mclust` [14].

The d-dimensional four-crescent density is $\frac{1}{4}\mathrm{Cres}_d([0.1\mathbf{1}_{d-2}, 0,0], \sqrt{2}, 0.05, 0.2, 1, 1) + \frac{1}{4}\mathrm{Cres}_d([-0.1\mathbf{1}_{d-2}, 1, -0.5], \sqrt{2},$

0.05, 0.2, 1, 0) + $\frac{1}{4}$Cres$_d$([$\mathbf{0}_{d-2}$, 1, −1], 0.5, 0.05, 0.4, 1, 0) + $\frac{1}{4}$Cres$_d$([−0.1$\mathbf{1}_{d-2}$, 1.5, −1.25], 0.5, 0.05, 0.4, 1, 1), where $\boldsymbol{X} \sim$ Cres$_d(\boldsymbol{\mu}, r, \alpha_1, \alpha_2, \alpha_3, s)$ is a crescent distributed random variable with components

$$
\begin{aligned}
X_1 &= \mu_1 + rR\cos(2\pi\Phi_1) \\
X_2 &= \mu_2 + rR\sin(2\pi\Phi_1)\cos(2\pi\Phi_2) \\
X_3 &= \mu_3 + rR\sin(2\pi\Phi_1)\sin(2\pi\Phi_2) \\
&\vdots \\
X_{d-1} &= \mu_{d-1} + rR\sin(2\pi\Phi_1)\cdots\sin(2\pi\Phi_{d-2})\cos(\pi\Phi_{d-1}) \\
X_d &= \mu_d + (-1)^s rR\sin(2\pi\Phi_1)\cdots\sin(2\pi\Phi_{d-2})\sin(\pi\Phi_{d-1},
\end{aligned}
$$

for $d \geq 2$, which generalises the bivariate version [6]. Here $R \sim$ Unif$[1-\alpha_1, 1+\alpha_1], \Phi_j \sim$ Beta$(2,2)$, $j = 1, \ldots, d-1$. The X_j are restricted so that $\Phi_j \leq \alpha_3\pi$. If $s \neq 0$, they are further restricted, $|X_j| \leq \alpha_2|X_d|$, $j = 1, \ldots, d-2$. Fig. 3 displays the scatter plots of $n = 1000$ samples for $d = 2, 3, 4, 5$. There are four crescent saddle-shaped clusters, with the two smaller clusters in the lower right posing particular difficulty to separate cleanly.

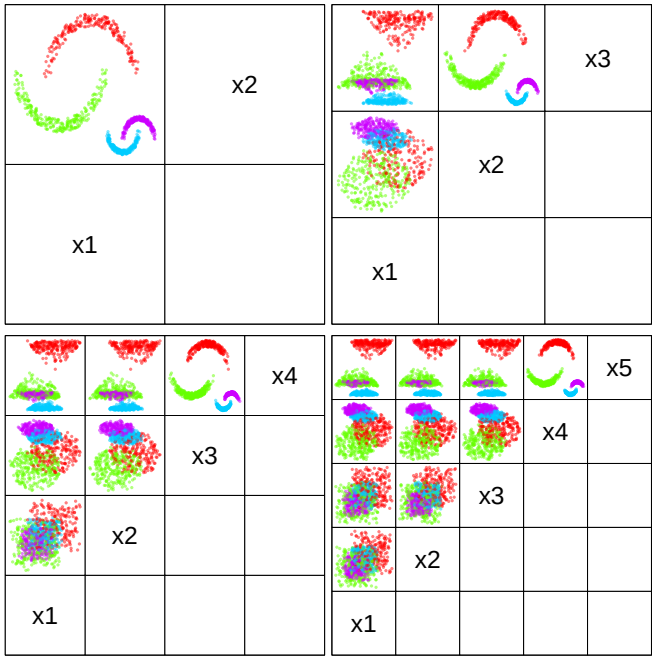

Fig. 3. Scatter plot matrices of the 4-crescent density data samples of size $n = 1000$, for $d = 2, 3, 4, 5$.

We examine 100 trials of $n = 1000$ random samples. Table 1 shows the numerical performance measures of the clustering methods for the different dimensions. Values of the Adjusted Rand Index (ARI) [20] and the normalised mutual information (NMI) [31] close to one indicate highly matched cluster labellings, and values close to (and less than for the ARI) zero indicate mismatched cluster labellings. The ARI and NMI values indicate that kernel mean shift KMS clustering is preferred for $d = 2$, whereas the nearest neighbour clusterings NNMS and NNMS2 are preferred for $d \geq 3$, confirming an analogous result for density estimation [33, Table 3].

As the execution times are highly dependent on the system utilised, we normalise them by the bivariate mean NNMS execution time in Table 1. As the dimension d increases, the times for the NNMS and NNMS2 decrease or remain stable, whereas the times for the kernel mean shift KMS and the k-means KM increase. For $d = 4, 5$, the NNMS is the most efficient, indicating that the grid-based search of the number of nearest neighbours for the NNMS2 and of the number of mixture components for the KM, and the dense nature of the kernel mean shift KMS leads to computational bottlenecks. From this simulation study, the NNMS efficiently computes the most accurate clusterings for higher dimensions.

		Clustering method			
	d	NNMS	NNMS2	KMS	KM
ARI	2	0.02±0.03	0.53±0.15	**0.78±0.04**	0.45±0.03
	3	0.56±0.07	**0.80±0.19**	0.78±0.05	0.45±0.04
	4	**0.92±0.02**	0.89±0.14	0.76±0.05	0.52±0.04
	5	**0.99±0.01**	0.87±0.14	0.70±0.09	0.62±0.06
NMI	2	0.19±0.04	0.74±0.10	**0.85±0.03**	0.74±0.02
	3	0.64±0.04	**0.89±0.12**	0.85±0.03	0.75±0.01
	4	0.90±0.02	**0.95±0.07**	0.85±0.03	0.77±0.02
	5	**0.98±0.01**	0.94±0.07	0.83±0.03	0.79±0.02
Time	2	1.00±0.32	**0.33±0.11**	0.36±0.04	0.52±0.05
	3	0.67±0.15	**0.40±0.15**	0.57±0.05	0.81±0.09
	4	**0.38±0.06**	0.43±0.14	1.19±0.05	0.89±0.08
	5	**0.43±0.10**	0.44±0.16	2.66±0.07	1.01±0.11

Table 1. 4-crescent densities for $d = 2, 3, 4, 5$, for size $n = 1000$. Clustering performance measures: Adjusted Rand Index (ARI), normalised mutual information (NMI), and relative computation times (Time). The optimal values are in bold. The clustering methods are the nearest neighbour mean shift with normal scale choice NNMS, the nearest neighbour median shift with Silhouette index choice NNMS2, the kernel mean shift KMS, and the k-means KM.

4.2. Vegetation cover data

An experimental data set which we consider is the vegetation cover in the Roosevelt National Forest, USA, available as `covertype` from the UCI Machine Learning repository `http://archive.ics.uci.edu/ml` and collected by [2]. We focus on the Comanche Peak wilderness area, with these six variables: elevation (m), azimuth aspect from true north (degrees), slope (degrees), horizontal and vertical distances to the nearest surface water feature (m), and a relative measure of incident sunlight (hillshade index) at 9 am on the summer solstice (0 to 255). The clusters correspond to the six different types of vegetation cover established by the US Forest Service, as illustrated in the scatter plot matrix in Fig. 4(a). The frequency counts of the $n = 4771$ observations are spruce/fir (698), lodgepole pine (706), ponderosa pine (644), aspen (976), Douglas fir (722) and krummholz (1025).

We compute the nearest neighbour mean shift NNMS with $k_{\text{NS},1} = 226$, the nearest neighbour median shift NNMS2 with k that minimises the Silhouette index, the kernel mean shift KMS with the plug-in selector, and the k-means KM with the BIC selector. For the mean shift methods, the tuning parameters are

$\varepsilon_1 = 7.74, j_{\max} = 100, \varepsilon_2 = 10\varepsilon_1 = 77.4, s_{\min} = 48$. The results of these clusterings are given in Fig. 4(b–e). Visually the true clusters stratified with respect to the elevation height are reproduced in the NNMS clusters, but not in the NNMS2, KMS or KM clusters.

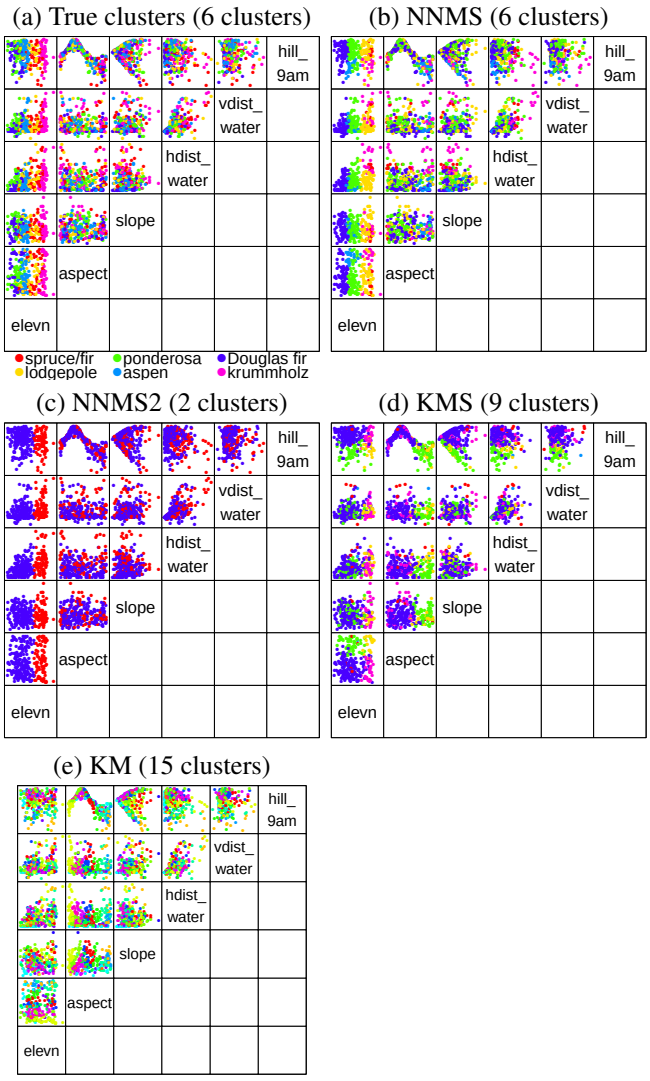

Fig. 4. Vegetation cover type training data ($n = 300$ subsets) for the Comanche Peak wilderness area. The six variables are: elevation, azimuth aspect, slope, horizontal and vertical distances to water features, and hill shade index at 9 am. (a) Six 'true' clusters corresponding to the six species. (b) NNMS clusters. (c) NNMS2 clusters. (d) KMS clusters. (e) KM clusters.

In Table 2 are the ARI, NMI and relative computation times for the different clustering methods. The NNMS and NNMS2 perform the best overall in terms of the ARI and NMI. The poor performance of the normal mixture clustering KM echoes the inadequacy of the normal mixture based linear discriminant analysis carried out by [2]. In terms of the relative computation times (divided by the NNMS computation time), the k-means KM is the most efficient but by far the lease accurate. The mean shift methods are uniformly preferred as they produce the most accurate clusterings, with the NNMS requiring less computation time than the NNMS2 or KMS.

	Clustering method			
	NNMS	NNMS2	KMS	KM
ARI	**0.293**	0.222	0.169	0.029
NMI	0.397	**0.429**	0.268	0.079
Time	**1.000**	1.103	4.287	0.188

Table 2. Vegetation cover type data for the Comanche Peak wilderness area. Clustering performance measures: Adjusted Rand Index (ARI), normalised mutual information (NMI) and relative computation times (Time). The optimal values are in bold.

4.3. Image segmentation

A recent resurgence in interest in the mean shift is due to its application to image segmentation where an image is transformed into a colour space in which clusters correspond to segmented regions in the original image. The 3-dimensional $L^*u^*v^*$ colour space [29, Eqs. 3.5-8a–f] is a common choice. Since an image is a 2-dimensional array of pixels, let (x, y) be the row and column index of a pixel. The spatial and colour (range) information of a pixel can be concatenated into a 5-dimensional vector (x, y, L^*, u^*, v^*) in the joint spatial-range domain. An image segmentation algorithm based on the kernel mean shift was introduced in [9] which we adapt to the NNMS.

The Berkeley Segmentation Dataset and Benchmark is an image database for testing image segmentation algorithms `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds`. We take image #7 from the colour test set. In Fig. 5(a) is the 481×321 pixels RGB image, and 5(b) the $n = 154401$ five-dimensional (x, y, L^*, u^*, v^*) spatial-range coordinates. Fig. 5(c) is the segmentation made by a human expert (User #1107) from the Berkeley project. Due to the computational limitations of the NNMS2, KMS and KM methods implemented in their `R` packages to treat data sets of size $n \sim 10^5$, we focus on the NNMS with $k_{\text{NS},1} = 2463, \varepsilon_1 = 0.005, j_{\max} = 100, \varepsilon_2 = 0.05, s_{\min} = 0.01n = 1544$, whose results are in Fig. 5(d). For comparison to standard image analysis methods, in Fig. 5(e–f), we perform the watershed [34] and Canny [3] segmentations in the `ImageJ` and `ImageMagick` software respectively, after an initial Gaussian blurring of 4 pixels on a grey scale version of the image. The human expert delimits the blue sky from brown rock formations and the sky from the green tree leaves. The automatic NNMS image segmentation performs visually similarly to this: it gives less well-defined segmentations in the tree roots, shrubs and soil, though it reveals a more detailed segmentation as the sky and rock formations. Of the two standard image analysis methods, the watershed segmentation produces more segments so the overall visual impression is fragmented and quite different to the human expert and NNMS segmentations. The Canny segmentation is more similar, but it is not always able to draw the complete edges, thus leaving a visual impression of an incomplete segmentation.

In Table 3 are the quantitative measures of the agreement between the human expert and computer edge detections: the figure of merit (FOM) [29, Eq. (15.5-1)] and the mean squared error distance (MSD) [28]. The FOM is calculated with the tuning constant = 0.1, and the FOM values close to 1 indicate a close agreement between the two edge sets. Values of the MSD

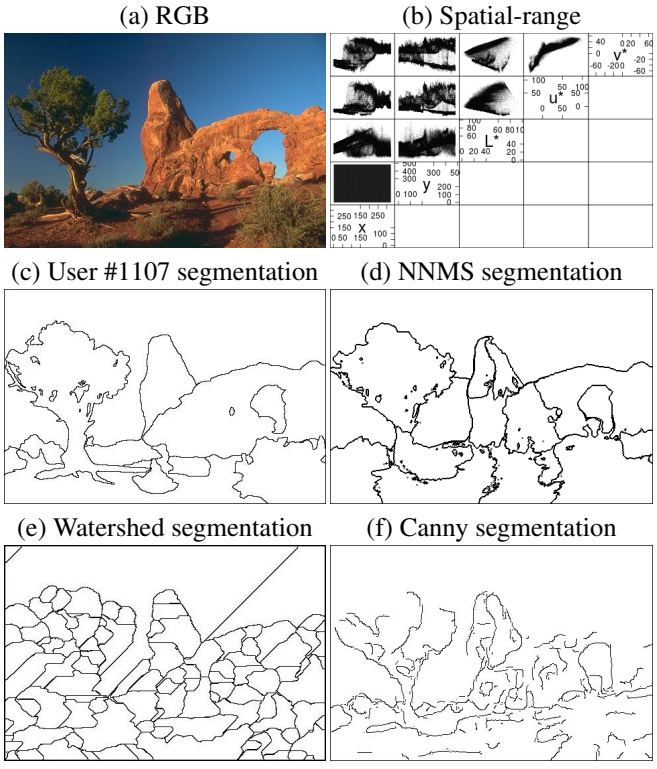

Fig. 5. Colour image segmentation. (a) RGB image 481×321 **pixels. (b) Scatter plot of** $n = 154401$ **transformed** (x, y, L^*, u^*, v^*) **spatial-range values. (c) User #1107 segmentation. (d) NNMS segmentation. (e) Watershed segmentation. (f) Canny segmentation.**

close to 0 indicate close agreement. The three computer segmentations perform similarly: in terms of the FOM, the Canny is the closest, though in terms of the MSD, the NNMS agrees most closely with the human expert segmentation.

	Segmentation method		
	NNMS	Canny	Watershed
FOM	0.967	**0.990**	0.961
MSD	**0.055**	0.148	0.063

Table 3. Colour image segmentation. Performance measures: Pratt's figure of merit (FOM) and mean squared error distance (MSD). The optimal values are in bold.

5. Conclusion

We have introduced a framework for the mathematical analysis of nearest neighbour estimators of the density function and its derivatives, allowing us to exhibit an automatic, single pass, normal scale selector for the optimal number of nearest neighbours. We apply these results for the density gradient to the mean shift for unsupervised learning. Our proposed automatic nearest neighbour mean shift clustering NNMS gave good empirical performance for discovering the number, location and shape of non-ellipsoidal clusters for multivariate data analysis and image segmentation. In the future, we anticipate to improve on this performance by developing more advanced data-based selectors, analogous to the improvements made with data-based bandwidth selectors for the kernel estimators. We also envisage to implement the NNMS in a distributed computing environment so that it becomes a readily available tool for Big Data Clustering.

Acknowledgements

This research has been partially supported by the Square Predict project within a 'Projet investissement d'avenir (PIA)' 2013–2016 grant provided by the French government.

Appendix A. Proof

Suppose that the following conditions hold:

(A1) f is a density function with all its partial derivatives up to order $r+2$ are bounded, continuous and square integrable.

(A2) $k = k_n$ is a sequence of the number of the nearest neighbours such that $k \to \infty$, $k/n \to \text{const}$ as $n \to \infty$.

(A3) K is a symmetric d-variate density such that $\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^T K(\boldsymbol{x})\, d\boldsymbol{x} = m_2(K)\mathbf{I}_d$, and all its r-th order partial derivatives are bounded, continuous and square integrable such that $\mathbf{R}(\mathsf{D}^{\otimes r}K) = \int_{\mathbb{R}^d} \mathsf{D}^{\otimes r}K(\boldsymbol{x})\mathsf{D}^{\otimes r}K(\boldsymbol{x})^T\, d\boldsymbol{x}$.

The assumptions (A1)–(A3) do not form a minimal set of assumptions, but they serve as a useful starting point.

Proof of Theorem 1. Substituting $\mathbf{H}(\boldsymbol{x}) = \delta_{(k)}(\boldsymbol{x})^2\mathbf{I}_d$ for $\mathbf{H}$ in $\operatorname{AMISE}[\mathsf{D}^{\otimes r}\tilde{f}(\cdot;\mathbf{H})] = n^{-1}|\mathbf{H}|^{-1/2}\operatorname{tr}((\mathbf{H}^{-1})^{\otimes r}\mathbf{R}(\mathsf{D}^{\otimes r}K)) + \frac{1}{4}(-1)^r m_2(K)^2\boldsymbol{\psi}_{2r+4}^T(\operatorname{vec}\mathbf{I}_{d^r}\otimes\operatorname{vec}\mathbf{H}^{\otimes 2})$ from [7, Theorem 2], we obtain

$$\begin{aligned}
&\operatorname{AMISE}[\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x};k)]\\
&\quad= \operatorname{AMISE}[\mathsf{D}^{\otimes r}\tilde{f}(\cdot;\delta_{(k)}(\boldsymbol{x})^2\mathbf{I}_d)]\\
&\quad= n^{-1}\operatorname{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))\delta_{(k)}(\boldsymbol{x})^{-d-2r}\\
&\qquad+ \tfrac{1}{4}(-1)^r m_2(K)^2\delta_{(k)}(\boldsymbol{x})^4\boldsymbol{\psi}_{2r+4}^T(\operatorname{vec}\mathbf{I}_d)^{\otimes(r+2)}
\end{aligned}$$

following similar reasoning to [7, Lemma 3(ii)] and [30, Theorem 1(iv)]. Taking its expected value yields our proposed optimality criterion

$$\begin{aligned}
&\mathrm{A}_r(\boldsymbol{x};k)\\
&\quad= \mathbb{E}\{\operatorname{AMISE}[\mathsf{D}^{\otimes r}\hat{f}(\boldsymbol{x};k)]\}\\
&\quad= n^{-1}\operatorname{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))[\mathbb{E}\,\delta_{(k)}(\boldsymbol{x})^{-d-2r}]\\
&\qquad+ (-1)^r\tfrac{m_2(K)^2}{4}\boldsymbol{\psi}_{2r+4}^T(\operatorname{vec}\mathbf{I}_d)^{\otimes(r+2)}[\mathbb{E}\,\delta_{(k)}(\boldsymbol{x})^4]\\
&\quad= \operatorname{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))[v_0 f(\boldsymbol{x})]^{(d+2r)/d}n^{2r/d}k^{-(d+2r)/d}\\
&\qquad+ (-1)^r\tfrac{1}{4}m_2(K)^2\boldsymbol{\psi}_{2r+4}^T(\operatorname{vec}\mathbf{I}_d^{\otimes(r+2)})[v_0 f(\boldsymbol{x})]^{-4/d}n^{-4/d}k^{4/d}
\end{aligned}$$

using $\mathbb{E}[\delta_{(k)}(\boldsymbol{x})^\alpha] = [k/(nv_0 f(\boldsymbol{x})]^{\alpha/d}\{1 + o(1)\}$ from [18, Eq. (2.2)]. The derivative of this with respect to k is

$$\begin{aligned}
&\frac{\partial}{\partial k}\mathrm{A}_r(\boldsymbol{x};k)\\
&\quad= -((d+2r)/d)\operatorname{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))[v_0 f(\boldsymbol{x})]^{(d+2r)/d}n^{2r/d}k^{-2r/d-2}\\
&\qquad+ (-1)^r\tfrac{1}{d}m_2(K)^2\boldsymbol{\psi}_{2r+4}^T(\operatorname{vec}\mathbf{I}_d^{\otimes(r+2)})[v_0 f(\boldsymbol{x})]^{-4/d}n^{-4/d}k^{4/d-1}.
\end{aligned}$$

Setting this derviative to zero, and noting that the exponent of $v_0 f(\boldsymbol{x})$ is $(d+2r)/d + 4/d = 1 + (2r+4)/d$ which is exactly the same as that of k, the solution is

$$k^*_{\mathrm{A},r}(\boldsymbol{x}) = \left[\frac{(d+2r)\,\mathrm{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}K))}{(-1)^r m_2(K)^2 \boldsymbol{\psi}^T_{2r+4}(\mathrm{vec}\,\mathbf{I}_d)^{\otimes(r+2)}}\right]^{d/(d+2r+4)} \times v_0 f(\boldsymbol{x}) n^{(2r+4)/(d+2r+4)}$$

and $k_{\mathrm{A},r} = \int_{\mathbb{R}^d} k^*_{\mathrm{A},r}(\boldsymbol{x})\,d\boldsymbol{x}$ follows immediately.

For $K = f = \phi$, as $m_2(\phi) = 1$ and $\mathrm{tr}(\mathbf{R}(\mathsf{D}^{\otimes r}\phi)) = 2^{-r}(4\pi)^{-d/2}\prod_{j=0}^{r-1}(d+2j)$ using [7, Lemma 3 and Corollary 7], and $\boldsymbol{\psi}^T_{2r+4}(\mathrm{vec}\,\mathbf{I}_d)^{\otimes(r+2)} = (-1)^{r+2}2^{-r-2}(4\pi)^{-d/2}\prod_{j=0}^{r+1}(d+2j)$ in conjunction with [5, Eq. (7)], the constant inside the brackets of $k_{\mathrm{A},r}$ reduces to

$$\frac{2^{-r}(4\pi)^{-d/2}\prod_{j=0}^{r}(d+2j)}{2^{-r-2}(4\pi)^{-d/2}\prod_{j=0}^{r+1}(d+2j)} = \frac{4}{d+2r+2}. \qquad \square$$

References

[1] Biau, G., F. Chazal, D. Cohen-Steiner, L. Devroye, and C. Rodríguez (2011). A weighted k-nearest neighbor density estimate for geometric inference. *Electron. J. Stat. 5*, 204–237.

[2] Blackard, J. A. and D. J. Dean (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. Electron. Agr. 24*, 131–151.

[3] Canny, J. (1986). A computational approach to edge detection. *IEEE T. Pattern Anal. 6*, 679–698.

[4] Chacon, J. E. (2015). A population background for nonparametric density-based clustering. *Stat. Sci. 30*, 518–532.

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

[6] Chacón, J. E. and T. Duong (2013). Data-driven density estimation, with applications to nonparametric clustering and bump hunting. *Electron. J. Stat. 7*, 499–532.

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

[8] Comaniciu, D. (2003). An algorithm for data-driven bandwidth selection. *IEEE T. Pattern Anal. 25*, 281–288.

[9] Comaniciu, D. and P. Meer (2002). Mean shift: a robust approach toward feature space analysis. *IEEE T. Pattern Anal. 24*, 603–619.

[10] Comaniciu, D., V. Ramesh, and P. Meer (2001). The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, Volume 1, pp. 438–445.

[11] Deng, Z. and S. Chung F.-L.and Wang (2008). FRSDE: Fast reduced set density estimator using minimal enclosing ball approximation. *Pattern Recogn. 41*, 1363–1372.

[12] Duong, T. (2007). ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. *J. Stat. Softw. 21(7)*, 1–16.

[13] Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc. 97*, 611–631.

[14] Fraley, C., A. E. Raftery, T. B. Murphy, and L. Scrucca (2012). mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report No. 597, Department of Statistics, University of Washington.

[15] Fukunaga, K. and L. Hostetler (1973). Optimization of k-nearest-neighbor density estimates. *IEEE T. Inform. Theory 19*, 320–326.

[16] Fukunaga, K. and L. Hostetler (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE T. Inform. Theory 21*, 32–40.

[17] Girolami, M. and C. He (2003). Probability density estimation from optimally condensed data samples. *IEEE T. Pattern Anal. 25*, 1253–1264.

[18] Hall, P. (1983). On near neighbour estimates of a multivariate density. *J. Multivariate Anal. 13*, 24–39.

[19] Holmquist, B. (1996). The d-variate vector Hermite polynomial of order k. *Linear Algebra Appl. 237/238*, 155–190.

[20] Hubert, L. and P. Arabie (1985). Comparing partitions. *J. Classif. 2*, 193–218.

[21] Jones, M. C. (1990). Variable kernel density estimates and variable kernel density estimates. *Aust. J. Stat. 32*, 361–371.

[22] Kaufman, L. and P. J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley and Sons.

[23] Kung, Y.-H., P.-S. Lin, and C.-H. Kao (2012). An optimal k-nearest neighbor for density estimation. *Stat. Probabil. Lett. 82*, 1786–1791.

[24] Li, J., S. Ray, and B. G. Lindsay (2007). A nonparametric statistical approach to clustering via mode identification. *J. Mach. Learn. Res. 8*, 1687–1723.

[25] Li, K. C. (1984). Consistency for cross-validated nearest neighbour estimates in nonparametric regression. *Ann. Stat. 12*, 230–240.

[26] Loftsgaarden, D. O. and C. P. Quesenberry (1965). A nonparametric estimate of a multivariate density function. *Ann. Math. Stat. 36*, 1049–1051.

[27] Mack, Y. P. and M. Rosenblatt (1979). Multivariate k-nearest neighbor density estimates. *J. Multivariate Anal. 9*, 1–15.

[28] Peli, T. and D. Malah (1982). A study of edge detection algorithms. *Comput. Vision Graph. 20*, 1–21.

[29] Pratt, W. K. (2001). *Digital Image Processing: PIKS Inside* (3rd ed.). New York: John Wiley and Sons.

[30] Schott, J. R. (2003). Kronecker product permutation matrices and their application to moment matrices of the normal distribution. *J. Multivariate Anal. 87*, 177–190.

[31] Strehl, A. and J. Ghosh (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3*, 583–617.

[32] Stuetzle, W. (2003). Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *J. Classif. 20*, 25–47.

[33] Terrell, G. R. and D. W. Scott (1992). Variable kernel density estimation. *Annals of Statistics 20*, 1236–1265.

[34] Vincent, L. and P. Soille (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE T. Pattern Anal. 13*, 583–598.

[35] Wang, X., W. Qiu, and R. H. Zamar (2007). CLUES: A non-parametric clustering method based on local shrinking. *Comput. Stat. Data Anal. 52*, 286–298.