Discussion Paper
No. 89
Center for Spatial Information Science
University of Tokyo

A KERNEL DENSITY ESTIMATION METHOD FOR NETWORKS, ITS COMPUTATIONAL METHOD, AND A GIS-BASED TOOL

Atsuyuki Okabe, University of Tokyo

Toshiaki Satoh, PASCO Corp.

Kokichi Sugihara, University of Tokyo

June 9, 2008

Abstract

We develop a kernel density estimation method for estimating the density of points on a network and implement the method in the GIS environment. This method could be applied to, for instance, finding 'hot spots' of traffic accidents, street crimes or leakages in gas and oil pipe lines. We first show that the application of the ordinary two-dimensional kernel method to density estimation on a network produces biased estimates. Second, we formulate a 'natural' extension of the univariate kernel method to density estimation on a network, and prove that its estimator is biased; in particular, it overestimates the densities around nodes. Third, we formulate an unbiased discontinuous kernel function on a network, and fourth, an unbiased continuous kernel function on a network. Fifth, we develop computational methods for these kernels and derive their computational complexity. We also develop a plug-in tool for operating these methods in the GIS environment. Sixth, an application of the proposed methods to the density estimation of bag-snatches on streets is illustrated. Lastly, we summarize the major results and describe some suggestions for the practical use of the proposed methods.

1. Introduction

The objectives of this study are: to formulate a kernel density estimation method for estimating the density of points on a network; to develop its computational method; to construct a GIS-based tool for executing the method; and lastly, to show its actual application.

The potential demand for this method is large, because in the real world there are numerous events that can be analyzed in terms of the density of points on a network. Examples include car crashes on streets, animal roadkills on forest roads, urban crime sites, tree spacing along the roadside, seabirds located along a coastline, beaver lodges in a watercourse, levee crevasse distribution on river banks, leaks in gas and oil pipelines, breaks in a wiring network, and disconnections on the Internet. In addition to these events, another broad class of events that can be represented by the density of points on a network is the class of those events that occur alongside rather than directly on a network such as facilities located alongside street networks within densely inhabited areas. For instance, the entrances to almost all facilities in a city are adjacent to a street and users access amenities through these. Consequently, the distribution of almost all facilities within an urbanized area can be represented by the distribution of points on networks.

Although there is a great amount of potential demand for analyzing these events, few methods had been developed until recently. There were several reasons for this delay: detailed point-location data on networks, such as traffic accident spots on street networks, were not easily available; managing network data was difficult; and computation on a network was much harder than on a plane. In recent years, these technical difficulties have been resolved to a great extent by easily accessible detailed spatial data through the Internet and user-friendly Geographical Information Systems (GISs) that manage network data. However, theoretical methods for spatial analysis on networks are far behind these advances, although some initial studies are found in the literature (Miller, 1994; Okabe et al., 1995; Okabe and Yamada, 2001; Spooner et al., 2004; Decker, 2005; Yamada and Thill, 2004; Lu and Chen, 2006; Xu and Sui, 2007; Yamada and Thill, 2007; ABC). Because of the lack of theoretical methods, researchers analyze the above events with methods that may lead to misleading conclusions (Yamada and Thill, 2004; Lu and Chen, 2006). Practitioners, such as traffic managers and police officers, would also appreciate theory-based tools for precise detection of 'hot spots' or 'black spots' of traffic accidents and street crimes (Okabe et al., 2000a, b).

One of the most frequently demanded methods for analyzing the above types of event is a method for estimating the density of points on a network. A convenient method in practice is as follows. First, we apply the ordinary two-dimensional kernel density estimation method to points on a plane in which a network is embedded, and estimate the two-dimensional density. Second, we derive the density of points on the network from the density along the line segments forming the network, that is, the vertical intersection of the density surface along the line segments. However, Satoh and Okabe (2005) showed that this estimation produces a bias, because when points are distributed according to a uniform distribution on a network, the density estimated by the above method does not produce a uniform distribution. A few attempts at density estimation on a network can be found in the literature. Flahaut et al. (2003) developed a kernel density estimation method on a simple network, but the network is very simple, i.e., a line. Borruso (2004, 2005) dealt with density estimation on a network using GIS, but did not explicitly discuss the bias of the estimator. We formulate three types of kernel function on a network, and explicitly examine whether the estimators using those functions are biased or unbiased. The method can be used extensively for analyzing the events mentioned above, in particular, street crime analysis and traffic accident analysis.

This paper consists of seven sections including this introductory section. Section 2 fixes a general framework commonly used throughout this paper. Section 3 formulates a 'natural' extension of the ordinary kernel function on a plane to a network. Because this extension is likely to yield misleading conclusions, we develop two types of kernel functions on a network: a discontinuous kernel function in Section 4 and a continuous kernel function in Section 5. In Section 6 we derive the computational complexity of these functions and develop a GIS-based tool. Using this tool, we demonstrate in Section 7 an actual application to the density estimation of bag-snatches on streets in Kyoto. The last section summarizes the major results and refers to an unsolved problem.

2. General framework

We consider a connected finite network $N = (V, L)$ embedded in a plane, consisting of a set of nodes, V, and a set of links, L. The network N is assumed to be planar (i.e., links in L do not intersect each other except at nodes in V on a plane), and nondirected (i.e., links are two-way). Let $\tilde{L}$ be the union of all links in L including the nodes in V (note that $\tilde{L}$ is sometimes referred to as the set of points constituting $\tilde{L}$). For an arbitrary point y on $\tilde{L}$, we construct a subset, L_y, of $\tilde{L}$ in such a way that the shortest-path distance between y and any point on L_y is less than or equal to h (the heavy line segments in Figure 1). For an arbitrary y, we define a function, $K_y(x)$, satisfying

$$K_y(x) \geq 0 \text{ for } x \in L_y,$$
$$= 0 \text{ for } x \in \tilde{L} \setminus L_y, \quad (1)$$

$$\int_{x \in L_y} K_y(x)\mathrm{d}x = 1, \quad (2)$$

where $\tilde{L} \setminus L_y$ in equation (1) means the complement of L_y with respect to $\tilde{L}$, and the integral over $x \in L_y$ in equation (2) means the integral along the all line segments of L_y.

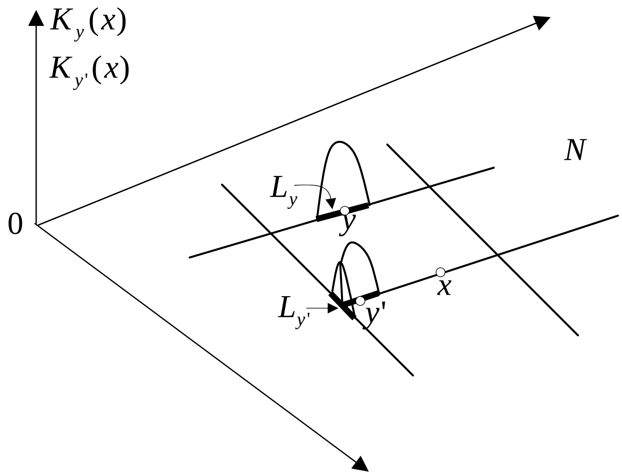


Figure 1: Kernel functions with centers at y and y' on $\tilde{L}$ of N.

We refer to the function $K_y(x)$ as a *kernel function* at y; y as the *kernel center* of $K_y(x)$; L_y as the *kernel support* of $K_y(x)$; and h as a *bandwidth*. The form of the function $K_y(x)$ may vary from location y to location y', or may not. The kernel support may be a line segment, a tree or it may include cycles (the heavy lines in Figures 1 and 2).

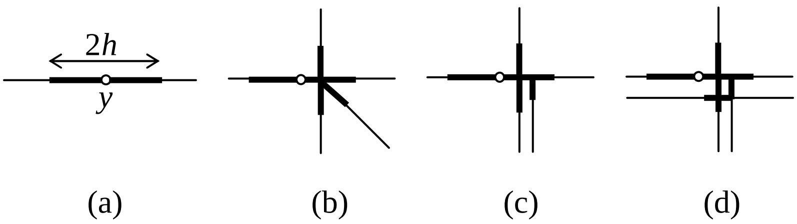


Figure 2: Various types of kernel support

Suppose that n points, $y_1, \ldots, y_n$, are independently distributed on $\tilde{L}$ according to an unknown probability density function $f(x)$ defined on $\tilde{L}$, and let

$$K(x) = \frac{1}{n}\sum_{i=1}^{n} K_{y_i}(x). \tag{3}$$

We call the value of $K(x)$ the *kernel estimator* for $f(x)$ at x, and the estimation of $f(x)$ by $K(x)$ the *kernel density estimation method* or the *kernel method* for short. The objective of this study is to develop a kernel method for estimating the density of points on a network.

One might notice from Figure 2a that the kernel method for univariate probability density functions (referred to as the *univariate kernel method*) can be applied to estimating part of the probability density function $f(x)$ on a network. This is true for Figure 2a, but it is not true for Figures 2b, c and d, where the *degree* of a node (i.e., the number of links meeting at the node) is three or more. To treat the cases in Figure 2 separately, we consider the subset $V_{(1)}$ of the nodes of V with degree one (the white circles in Figure 3) and the subset $V_{(\geq 3)}$ of the nodes of V with degree three or more (the black circles in Figure 3). We construct the *i*th subset of $\tilde{L}$, denoted by $L_{(1)i}$, in such a way that the shortest-path distance between the *i*th node of $V_{(1)}$ and any point on $L_{(1)i}$ is less than or equal to $2h$; similarly, we construct the *i*th subset of $\tilde{L}$, denoted by $L_{(\geq 3)i}$, in such a way that the shortest-path distance between the *i*th node of $V_{(\geq 3)}$ and any point on $L_{(\geq 3)i}$ is less than or equal to $2h$ (the heavy lines in Figure 3). Then the complement of $\bigcup_i L_{(1)i}$ and $\bigcup_i L_{(\geq 3)i}$ with respect to $\tilde{L}$, i.e., $L_S = \tilde{L} \setminus \{\bigcup_i L_{(1)i} \cup \bigcup_i L_{(\geq 3)i}\}$, consists of simple line segments (the hairlines in Figure 3).

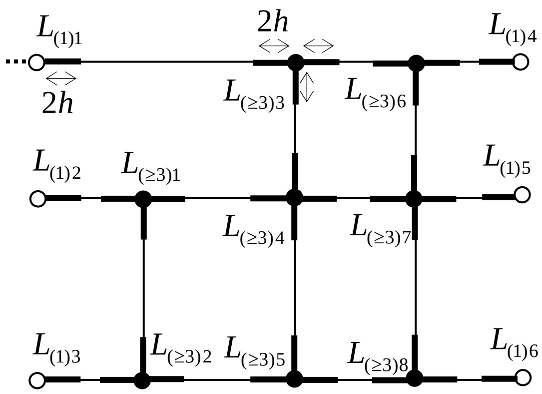
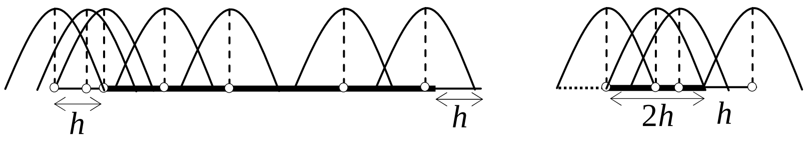


Figure 3: Three types of subsets of $\tilde{L}$: $L_{(1)i}$, $L_{(\geq 3)i}$ and L_S.

We can apply the univariate kernel method for density estimation to each line segment in L_S. Note that in this case, we can use as observed points not only those on these line segments but also those on the buffer zone line segments with width h (the hairlines in Figure 4a) that are included in $\tilde{L}$. Because the kernel functions with centers outside the extended line segments (the heavy line segment and the hairline segment in Figure 4a) do not affect the density estimation at a point on the line segment (the heavy line in Figure 4a), we do not encounter the edge problem for L_S.



(a) (b)

Figure 4: The edge problem does not exist for L_S but does for $L_{(1)i}$.

However, when we apply the univariate kernel method to line segments in $L_{(1)}$ (e.g., $L_{(1)1}$ in Figure 3), we do encounter the edge problem, because the broken line in Figure 4b does not exist in $\tilde{L}$. Many methods for treating the edge problem have been proposed in the literature of the univariate kernel method, and those methods can be applied to $L_{(1)}$ (see among others Devroye and Gyorfi (1985); Eggermont and LaRiccia (2001); Scott (1992); Silverman (1986); Tapia and Thompson (1978)).

The density estimation for $L_{(\geq 3)i}$ is completely different from the univariate kernel method and there are few studies of it in the literature. This estimation is particularly important for spatial analysis on networks, because many events, such as traffic accidents, tend to occur around intersections, and unbiased estimators are indispensable. Therefore, we focus on the density estimation on $L_{(\geq 3)i}$.

The kernel method must estimate an unknown probability density function $f(x)$ using the given observed points, $y_1, \ldots, y_n$, without parameters (i.e., nonparametric estimation); the function is not specified. However, in spatial analysis, we often start with the *complete spatial randomness* (*CSR*) hypothesis; that is, points are assumed to be independently distributed on $\tilde{L}$ according to the uniform probability density function, i.e.,

$$f(x) = \frac{1}{|\tilde{L}|}, \ x \in \tilde{L}, \tag{4}$$

where $|\tilde{L}|$ denotes the total length of the links forming the network N. If points are actually generated according to this distribution, kernel estimators are expected to produce the uniform probability density; otherwise, the kernel estimators produce a *bias*, which is defined by:

$$B(x) = \mathrm{E}[K(x)] - f(x), \tag{5}$$

where E is the expectation operator.

For a general function $f(x)$, the expected value of $K(x)$ is obtained from

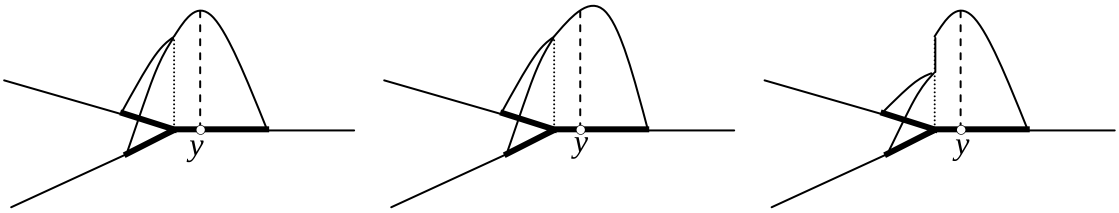
$$\mathrm{E}[K(x)] = \int_{y \in \tilde{L}} K_y(x) f(y) \mathrm{d}y. \tag{6}$$

In theory, we can generally examine whether or not an estimator is unbiased using this equation (i.e., E[$K(x)$] = $f(x)$), but in practice, the derivation of this integral is very hard without specifying the form of $f(x)$ for $L_{(\geq 3)i}$. Therefore, in this study, as a first step, we attempt to find unbiased kernel functions when $f(x)$ is given by the uniform probability density function (equation (4)), i.e., one of the most fundamental hypotheses in spatial analysis. This way of examination is similar to that of the ordinary krigging method, where the expected value (bias) and variance are derived under the assumption that an unknown density is specified as uniform over space (Cressie, 1991, Section 3.2).

Many possible kernel functions can be defined on a network. They may be characterized by the following properties.

Property 1 (unimodal): $K_y(x)$ is unimodal.
Property 2 (modal point = center): The modal point of $K_y(x)$ is at y.
Property 3 (continuous): $K_y(x)$ is continuous with respect to x.

One might consider that these properties (Figure 5a) are 'desirable'. In fact, frequently used univariate kernel functions satisfy these properties, perhaps because space on a line is isotropic (i.e., no difference between directions). However, space on $L_{(\geq 3)i}$ is not isotropic. In fact, as illustrated in Figure 5, the space on the right-hand side and that on the left-hand side of y are qualitatively different: on the left-hand side, a line splits into two lines at the intersection. If we consider this nonisotropic property, Properties 2 and 3 are not always 'desirable' for kernel functions on $L_{(\geq 3)i}$. Reflecting this nonisotropy around nodes, the modal point may not be at y (Figure 5b) or a kernel function may be discontinuous at a node (Figure 5c).



(a) (b) (c)

Figure 5: Three types of kernel function: (a) the kernel center coincides with the modal point; (b) the kernel center does not coincide with the modal point; (c) the kernel function is discontinuous at a node.

Most univariate kernel functions are symmetric at y, but we cannot use the same concept for kernel functions on the kernel support $L_{(\geq 3)i}$, because the kernel support is a tree or it may include cycles, which are not symmetric. However, we may consider these functions ‘symmetric’ in the sense that values at the same distance from the kernel center y are the same, i.e.:

Property 4 (equal-distance/equal-density): For x and x' ($x \neq x'$) on $L_{(\geq 3)i}$, if the shortest-path distance $d(x, y)$ between x and y is the same as that $d(x', y)$ between x' and y, i.e., $d(x, y)$ = $d(x', y)$, then $K_y(x)$ = $K_y(x')$ (Figure 6).

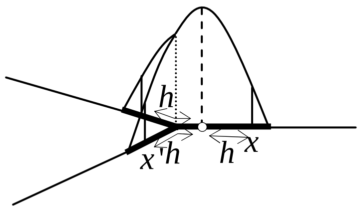


Figure 6: Property 4 (equal-distance/equal-density property).

In panels (a) and (b) in Figure 7, the shape of the kernel support L_y and that of $L_{y'}$ are the ‘same’ except for the angles between links ($\theta \neq \theta'$). In such a case, it seems to be appropriate to assume that the shape of the kernel function is invariant with respect to an angle. Stated a little more explicitly:

Property 5 (invariant with respect to a vertex angle): If the graph consisting of a node and links in L_y is isomorphic to the graph consisting of a node and links in $L_{y'}$, and if the length of each link in L_y and that of its corresponding link in $L_{y'}$ are the same, then the equation $K_y(x)$ = $K_{y'}(x')$ holds for a point x on L_y and its corresponding point x' on $L_{y'}$.

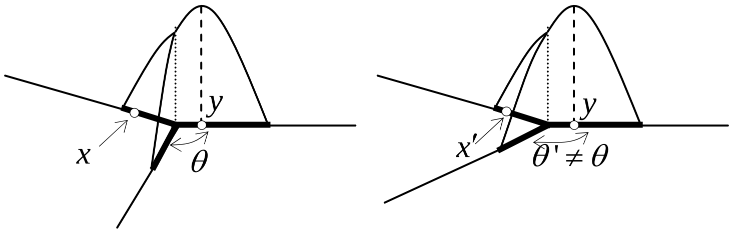


Figure 7: Property 5 (invariant with respect to an angle).

Property 6 (symmetric with respect to two kernel centers): The kernel function $K_y(x)$ is symmetric with respect to x and y, i.e.,

$$K_y(x) = K_x(y) \text{ for } x, y \in \tilde{L}. \tag{7}$$

This property means that the value of the kernel function $K_y(x)$ at x is the same as that of $K_x(y)$ at y (Figure 8).

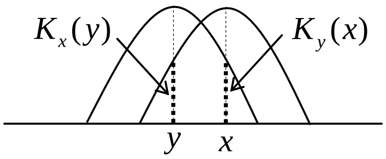


Figure 8: Property 6 (symmetric with respect to two kernel centers).

The above properties are all geometrical properties of kernel functions. Last, we add an important statistical property.

Property 7 (unbiased): The estimator $K_y(x)$ is unbiased, i.e., $B(x) = 0,\ x \in \tilde{L}$ for the uniform density function $f(x)$ given by equation (4).

In the subsequent three sections, we formulate three types of kernel function on a network in terms of a function, $k(x)$, defined on the real axis R that satisfies:

(i) $k(x) = k(-x)$ for any $x \in \mathrm{R}$;

(ii) $\int_{-\infty}^{\infty} k(x)\mathrm{d}x = 1$;

(iii) For a positive real h, $k(x) = 0$ if $|x| \geq h$, and $k(x) > 0$ if $|x| < h$;

(iv) $k(x)$ is nonincreasing for $0 \leq x < \infty$;

(v) $k(x)$ is continuous with respect to x.

We call this function the *base* kernel function. The base kernel function satisfies Properties 1, 2, 3, 4 and 7 (Properties 5 and 6 are not applicable).

3. The class of 'similar' shape kernel functions

In estimating a univariate probability density function, we usually use a unimodal, symmetric and continuous kernel function $K_y(x)$ (Properties 1, 2, and 3 in Section 2), and its shape does not change with respect to the location of its kernel center y. One might also wish to use this type of kernel function for density estimation on a network N, and such a kernel function can be applied to the line segments in L_S (the hairlines in Figure 3). However, it cannot be applied to the density estimation on $L_{(\geq 3)i}$, because the kernel support, usually a tree form, changes according to its kernel center y. Yet, one might wish to use the 'similar' shape kernel function $K'_y(x)$ as shown in Figure 9a, where the shape of $K'_y(x)$ along lines l_2 and l_1 and that along l_3 and l_1 are the same as the shape of the base kernel function $k(x)$ (Figure 9b). However, the function $K'_y(x)$ violates equation (2), because the integral of $K'_y(x)$ with respect to x over $L_{(\geq 3)i}$ is more than unity. To make it unity, $K'_y(x)$ is multiplied by a scaling factor $c(y)$ so that the integral of $c(y)\ K'_y(x)$ over its kernel support is unity (Figure 9c).

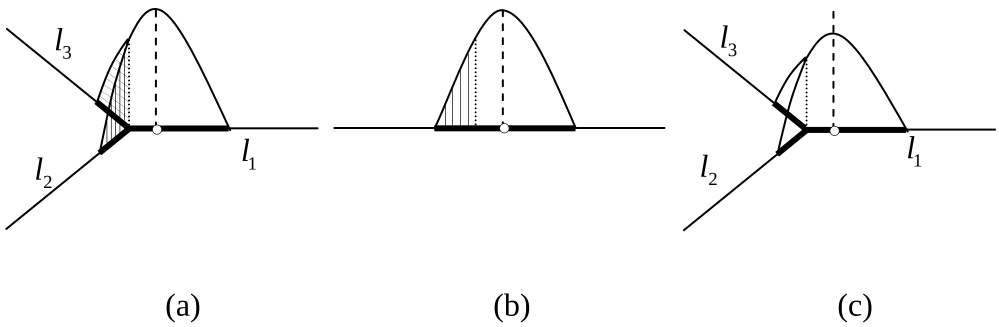


Figure 9: A function (a) with shapes along l_1 and l_2 (or l_3) that are the same as that of the base kernel function (b), and a function (c) with a shape along l_1 and l_2 (or l_3) that is similar to that of the base kernel function.

To formulate this 'similar' shape kernel function mathematically, we consider $L_{(\geq 3)i}$ consisting of n links, $l_1,...,l_n$ meeting at v as in Figure 10a, and the path consisting of l_i and l_1 through v (e.g., the heavy lines in Figure 10a). We indicate a point x (a location vector) on the path by the point x (a location scalar) on the real-value axis in Figure 10b (where the origin of the axis corresponds to v).

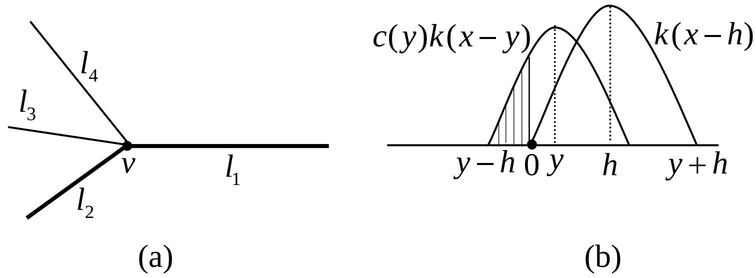


Figure 10: Links meeting at a node and the 'similar' shape kernel function shown on the real-value axis representing l_i and l_1 (e.g., the heavy line segments l_2 and l_1).

On the axis formed by l_i and l_1 ($i = 2,...,n$), we can define the '*similar*' shape kernel function $K_y(x)$ as

$$K_y(x) = k(x-y) \text{ for } |y| \geq h \ (l_1), \quad (8)$$

$$K_y(x) = c(y)\ k(x-y) \text{ for } 0 \leq |y| \leq h \ (l_2,...,l_n) \quad (9)$$

(Figure 10b), where

$$c(y) = \frac{1}{1+(n-2)\int_{y-h}^{0} k(x-y)\mathrm{d}x} \text{ for } 0 \leq y \leq h \ (l_1), \quad (10)$$

$$\frac{1}{1+(n-2)\int_{0}^{y+h} k(x-y)\mathrm{d}x} \text{ for } -h \leq y \leq 0 \ (l_2,...,l_n).$$

Because the denominator of equation (10) is larger than unity, we obtain the relation

$$0 < c(y) < 1 \text{ for } |y| < h. \quad (11)$$

As can be seen from equations (8) and (9), the kernel function $K_y(x)$ is 'similar' in the sense that the shape of $K_y(x)$ along l_i and l_1 ($i = 2,...,n$) is similar up to a scaling factor $c(y)$ to the base kernel function $k(y-x)$.

Since the integral on the domain $y-h \leq x \leq 0$ (or $0 \leq y \leq h$) in equation (10) (the shaded area in Figure 10) decreases (or increases) as y increases for $0 < y < h$ (or $-h < y < 0$), we obtain the relation

$$\frac{\mathrm{d}c(y)}{\mathrm{d}y} > 0 \text{ for } 0 < y < h\,;\ \frac{\mathrm{d}c(y)}{\mathrm{d}y} < 0 \text{ for } -h < y < 0\,. \tag{12}$$

The ‘similar’ shape kernel function is a natural extension of the univariate kernel function and many researchers are tempted to use it for density estimation on a network. If the ‘similar’ shape function is unbiased, that would be correct. However, we now show that this function is biased. To this end, we derive the expected value of $K_y(x)$ at x with respect to y on the network. Noticing that placing the kernel function at a point y in $|x-y|>h$ does not affect the expected value at x, we consider only the part of the network in which $|x-y|\le h$. Because the shape of $K_y(x)$ changes qualitatively at a node (the origin in Figure 11), we divide $0\le x\le 2h$ into two intervals: $h\le x\le 2h$ (Figure 11a) and $0\le x\le h$ (Figure 11b).

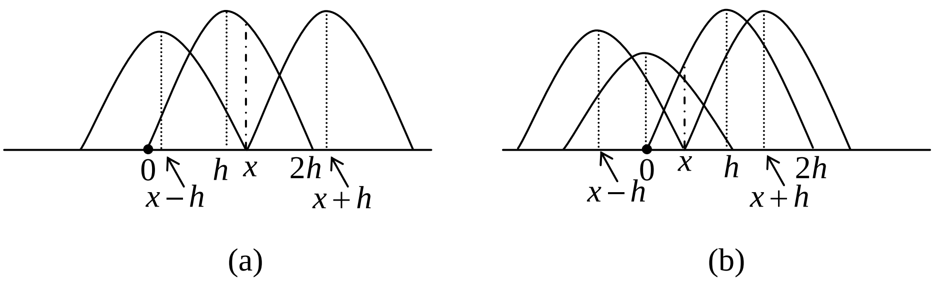


Figure 11: Derivation of the expected value of $K_y(x)$ with respect to y when (a) $h\le x\le 2h$ and (b) $0\le x\le h$.

In the former interval, the kernel function $K_y(x)$ is effective on x if $x-h\le y\le x+h$, and the shape of the kernel function qualitatively changes at h. Therefore, the expected value of $K_y(x)$ with respect to y in $x-h\le y\le x+h$ is obtained from

$$\mathrm{E}(K_y(x)) = \int_{x-h}^{h} c(y)k(x-y)\mathrm{d}y + \int_{h}^{x+h} k(x-y)\mathrm{d}y\,. \tag{13}$$

Because relations (11) and (12) hold, the expected value has the following properties for x in $h<x<2h$.

$$\mathrm{E}(K_y(x)) < 1, \tag{14}$$

$$\frac{\mathrm{dE}(K_y(x))}{\mathrm{d}y} > 0. \tag{15}$$

This implies that the kernel estimator is biased at x when $h<x<2h$.

In the interval $0 < x < h$, noticing that the form of the kernel function changes qualitatively at 0 and h, we obtain the expected value of $K_y(x)$ at x with respect y from

$$\mathrm{E}(K_y(x)) = \int_h^{x+h} k(x-y)\mathrm{d}y + \int_0^h c(y)k(x-y)\mathrm{d}y + (n-1)\int_{x-h}^0 c(y)k(x-y)\mathrm{d}y. \tag{16}$$

From this equation and equations (10)–(12), we obtain

$$\mathrm{E}(K_y(0)) = \int_0^h c(y)k(0-y)\mathrm{d}y + (n-1)\int_{-h}^0 c(y)k(0-y)\mathrm{d}y$$

$$> \int_0^h c(0)k(y)\mathrm{d}y + (n-1)\int_0^h c(0)k(y)\mathrm{d}y = \frac{2}{n}\frac{1}{2} + (n-1)\frac{2}{n}\frac{1}{2} = 1. \tag{17}$$

Therefore, the kernel function is biased not only when $h < x < 2h$ but also when $0 < x < h$ (except at a point). Figure 12 illustrates the case that the base kernel function $k(x)$ is given by the Epanechinikov (1969) function, i.e.,

$$k(x) = \frac{3}{4} - \frac{3}{4}x^2,\ 0 \le |x| \le 1,$$

$$0,\ |x| \ge 1 \tag{18}$$

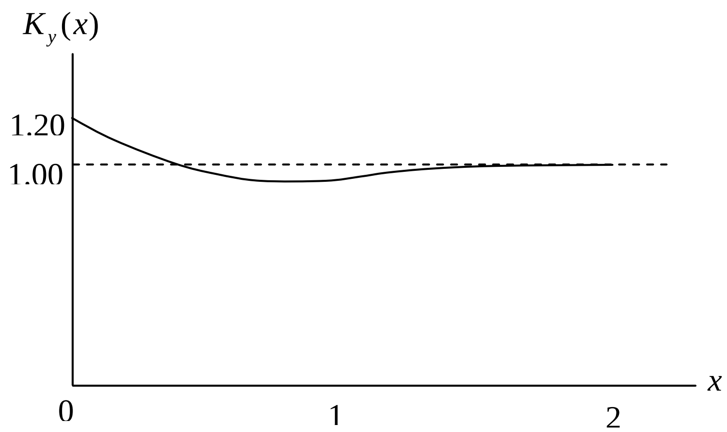


Figure 12: The estimator of the ‘similar shape’ kernel function with the base kernel function defined by equation (18).

This result warns us that if we use the ‘similar’ shape kernel function (which is a ‘natural’ extension of the univariate symmetric kernel function) in a network, then the estimator leads to a false conclusion that the points tend to gather around nodes, even though the true process is that the points are uniformly distributed on the network.

4. The class of equal-split kernel functions

To avoid false conclusions, we must use unbiased kernel functions. This section formulates such a kernel function.

We assume that the length of any cycle in a network is greater than $2h$ where h is the bandwidth of the base kernel function $k(x)$. Let $x, y \in \tilde{L}$ be two points on the links $\tilde{L}$ and let p be the shortest path from y to x (the heavy line in Figure 13). Because the length of any cycle in the network is greater than $2h$, the shortest path p is unique, and moreover, the length of any other path from y to x (if it exists) is longer than h.

First, we consider the case in which x does not coincide with a vertex of L, and second, the case in which it does. In the first case, as shown in Figure 13, let $v_1, v_2, ..., v_s$ be the nodes on the path p that one visits in this order when traveling from y to x along the shortest path p. Furthermore, let n_i denote the degree of the node v_i for $i = 1, 2, ..., s$, and let $d(y, x)$ be the shortest path distance from y to x. In these terms, we define a function as

$$K_y(x) = \frac{k(d(y,x))}{(n_1 - 1)(n_2 - 1)\cdots(n_s - 1)} \text{ for } 0 \le d(y,x) \le h.$$

$$0, \; d(y,x) \ge h. \qquad (19)$$

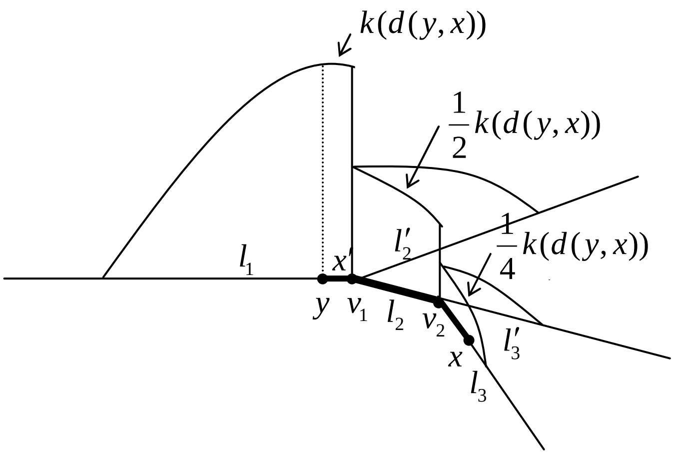


Figure 13: Equal-split kernel function

This definition can be understood intuitively in the following manner. Suppose that y is on a link l_1. If x' is also on the same link l_1 (Figure 13), then $K_y(x') = k(d(y, x'))$, that is, the kernel function $K_y(x')$ coincides with the base kernel function $k(d(y, x'))$. Next, suppose that there is exactly one node v_1 on the path p. Imagine that we travel along the path p from y to x. Then we encounter $n_1 - 1$ links, i.e., l_2 and l_2' at v_1 in Figure 13 (other than the link along which we have traveled, e.g., l_1), we chose the one forming the path p (e.g., l_2 in Figure 13). At node v_1, the value $k(d(y, v_1))$ is equally divided into $n_1 - 1$ values (e.g., $n_1 - 1$ = 2 in Figure 13), each of which is assigned to each link (l_2 and l_2' in Figure 13). If there are two or more nodes on the path p (e.g., v_2 in Figure 13), we divide the value again and assign them to each link equally (e.g., l_3 and l_3' in Figure 13). In this manner, the value of $K_y(x)$ is determined by delivering the value of $k(d(y, x))$ to the branch links equally at every node on the path p from y to x.

The function $K_y(x)$ defined by equation (19) admits Properties 1 (unimodal), 2 (modal point = center) and 5 (invariant w.r.t. an angle). However, as is seen in Figure 13, the function is not a continuous function of x: even if the base kernel function $k(d(y, x))$ is continuous, it is discontinuous at the nodes.

Now we wish to confirm that the function $K_y(x)$ defined by equation (19) is truly a kernel function (i.e., equations (1) and (2)). Because of the property (iii) of the base kernel function, equation (1) is satisfied.

We now consider the shortest path from x to y, i.e., the reverse path of the path p. By definition, we obtain $K_x(y)$ as

$$K_x(y) = \frac{k(d(x, y))}{(n_s - 1)(n_{s-1} - 1)\cdots(n_1 - 1)}. \tag{20}$$

Comparing equations (19) and (20), we obtain the equation

$$K_y(x) = K_x(y). \tag{21}$$

Therefore, the kernel function defined above satisfies Property 6 (symmetric with respect to two kernel centers, y and x).

Next consider the second case, where the point x is at a vertex, say v_0, with degree n_0. For another point y, let p be the shortest path from x to y, and let $v_0, v_1, ..., v_s$ be the sequence of vertices that we encounter as we move from x to y along p. We define

$$K_x(y) = \frac{1}{(n_s - 1)(n_{s-1} - 1)\cdots(n_1 - 1)} \frac{2}{n_0} k(d(x, y)) . \quad (22)$$

This definition differs from equation (19) by the term $2/n_0$. The intuitive meaning of this definition is as follows. If x is a general point on a link, it can be regarded as a degree-two vertex, and equation (19) implies that the total density is equally split in two directions at x and the value along one direction is represented by this expression.

Now, we suppose that x is at a vertex with degree n_0, and we want to split the total density to n_0 branches equally. For this purpose 2 should be replaced by n_0. This is why the term $2/n_0$ is included.

We should note that for this case equation (21) does not hold. However, x being at a vertex is a rare case; if x is uniformly distributed on N, it is of measure 0 for x to be at a vertex. Hence, the following discussion is valid.

Recall that the value of $k(d(y, x))$ to the branch links is assigned equally at every node; consequently, the total of the assigned values is the same as that of $k(d(y, x))$, and because $\int_{-\infty}^{\infty} k(x)\mathrm{d}x = 1$ holds for the base kernel function $k(d(y, x))$, we see that the equation

$$\int_{x \in \tilde{L}} K_y(x)\mathrm{d}x = 1 \quad (23)$$

holds. Therefore, the function defined by equation (19) satisfies the definition of the kernel function given by equations (1) and (2). We refer to it as the *equal-split* kernel function.

From equations (21) and (23), we obtain

$$\int_{y \in \tilde{L}} K_x(y)\mathrm{d}y = \int_{x \in \tilde{L}} K_y(x)\mathrm{d}x = 1 . \quad (24)$$

Therefore, the kernel estimator by the kernel function given by equation (19) is proved to be unbiased. This property holds for any base kernel function $k(x)$ forming the equal-split kernel function $K_y(x)$ of equation (19); consequently, the equal-split kernel function produces a broad class of kernel functions.

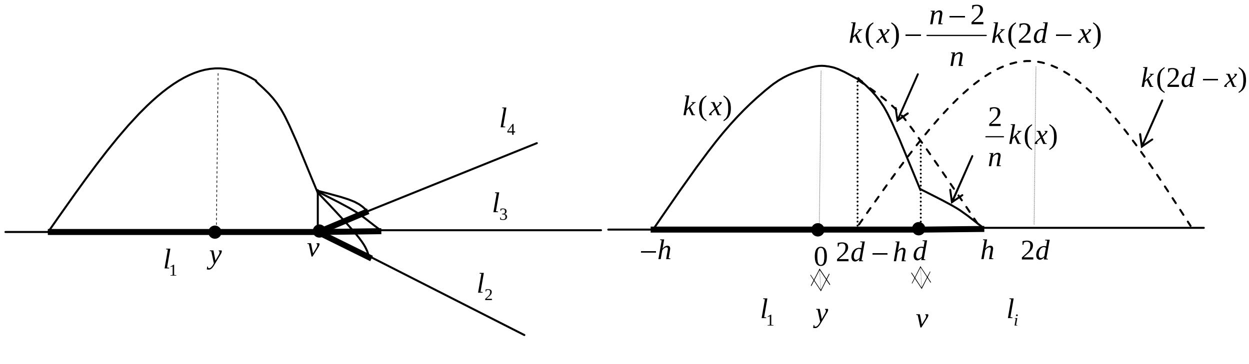
5. The class of equal-split continuous kernel functions

The equal-split kernel function defined in Section 4 is discontinuous at each node. This property looks peculiar on an isotropic line, but as Figure 14a shows, a network space is not isotropic around a point near a node (e.g., y near v in Figure 14a). Therefore, discontinuous kernel functions are not always peculiar. Yet it seems desirable to use a continuous kernel function, and so we develop such a function in this section. We first formulate a continuous kernel function in the simple case in which its kernel support consists of n links, $l_1,...,l_n$, meeting at one node, v, and the length of each link is not shorter than the bandwidth h (Figure 14a). The formulation in the general case in which links meet at more than one node on the kernel support is essentially similar to the simple case, but its derivation is lengthy. It is hence shown in the Appendix.

We regard two links, l_1 and l_i ($i \neq 1$), connected at v (Figure 14a) as a real-value axis (as in Figure 14b), and the origin on the axis corresponds to y. In a heuristic manner, we define a function as

$$\begin{aligned} K_y(x) &= k(x) \text{ for } -h \le x \le 2d-h, \\ &k(x) - \frac{n-2}{n}k(2d-x) \text{ for } 2d-h \le x \le d, \\ &\frac{2}{n}k(x) \text{ for } d \le x \le h. \end{aligned} \quad (25)$$

An example is illustrated in Figure 14.



(a) (b)

Figure 14: Equal-split continuous kernel function illustrated (a) on a network embedded in a plane; (b) on the axis representing l_1 and l_i.

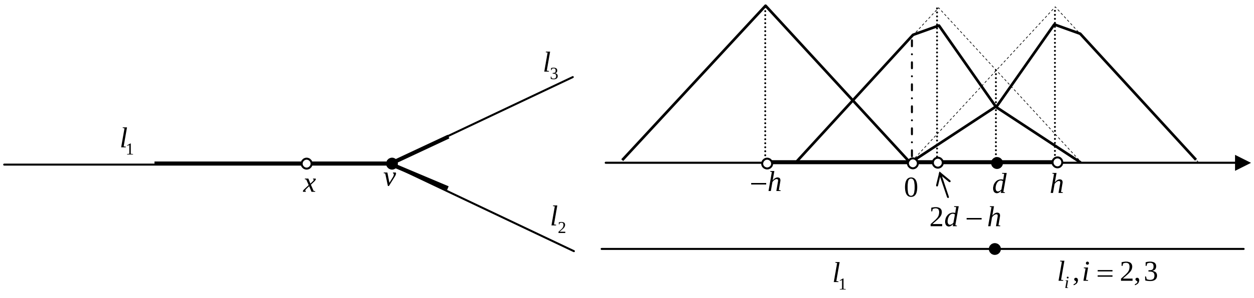
Because $k(d)-((n-2)/n)k(2d-d)=(2/n)k(d)$ holds, the above function is continuous at d. Moreover, noticing that $\int_{2d-h}^{d} k(2d-x)\mathrm{d}x=\int_{d}^{h} k(x)\mathrm{d}x$, we have

$$\int_{-h}^{2d-h} k(x)\mathrm{d}x+\int_{2d-h}^{d}[k(x)-\frac{n-2}{n}k(2d-x)]\mathrm{d}x+(n-1)\frac{2}{n}\int_{2d-h}^{h} k(x)\mathrm{d}x=1. \qquad (26)$$

Therefore, the function defined by equation (25) is a kernel function.

As Figure 14a shows, the shape of the kernel function is the same on l_2, l_3 and l_4, as with the equal-split kernel function shown in Figure 13 (the shape on l_2 and the corresponding line segment on l_2'). The equal-split kernel function in Section 4 and the kernel function in this section differ in the assignment of values to l_2, l_3 and l_4: the latter adjusts the values to make the function continuous in the 'local area' around the vertex v, i.e., the area in which the distance from the vertex v is within $h-d$. We refer to the kernel function defined by equation (25) as the *equal-split continuous* kernel function.

We now want to prove that the equal-split continuous kernel function is unbiased. We first consider the simple case, i.e., n links meeting at one node v as shown in Figure 14a (e.g., $n=4$) or Figure 15a (e.g., $n=3$) and prove that the kernel function given by equation (25) is unbiased. We consider an arbitrary point on a link, say x in Figure 15a. Because the bandwidth is h, we focus on the kernel function having its center within h from x (the heavy line segments in Figure 15a). We consider two links, e.g., l_1 and $l_i, i=2,3$, and represent them as a real-valued axis as shown in Figure 15b, where x and v correspond to the origin and the point at d, respectively, in Figure 15b (note that x and y are regarded as real values on the axis). In the figure, three kernel functions with centers placed at $-h$, $2d-h$ and h are depicted (for ease of illustration, the kernel functions are not curved).



(a) (b)

Figure 15: Links meeting at v and the equal-split continuous kernel functions with centers at $-h, 2d-h$ and h.

The density at the origin (corresponding to x in Figure 15a) is given by $k(0-y)$ for $-h \le y \le 2d-h$; it is given by $k(0-y)-((n-2)/n)k(0-2d+y)$ for $2d-h \le y \le d$; and it is given by $(2/n)k(0-y)$ for $d \le y \le h$ on $n-1$ links. Therefore, the density at the origin $D(0)$ is obtained from

$$D(0) = \int_{-h}^{2d-h} k(-y)\mathrm{d}y + \int_{2d-h}^{d} [k(-y) - \frac{n-2}{n} k(-2d+y)]\mathrm{d}y + \int_{d}^{h} (n-1)\frac{2}{n} k(-y)\mathrm{d}y. \qquad (27)$$

Because the equation

$$\int_{2d-h}^{d} k(-2d+y)\mathrm{d}y = \int_{d}^{h} k(-y)\mathrm{d}y \qquad (28)$$

holds, equation (27) is written as

$$D(0) = \int_{-h}^{2d-h} k(-y)\mathrm{d}y + \int_{2d-h}^{d} k(-y)\mathrm{d}y + \int_{d}^{h} k(-y)\mathrm{d}y = \int_{-h}^{h} k(y)\mathrm{d}y = 1. \qquad (29)$$

Therefore, the kernel estimator in equation (25) is unbiased.

The comparison between equation (27) and equation (29) shows a very nice property: the integral of the kernel function across the points on the n heavy line segments in Figure 16a ($n = 3$) is equivalent to that of the base kernel function across one heavy line segment in Figure 16b. We use this nice property to treat the general case.

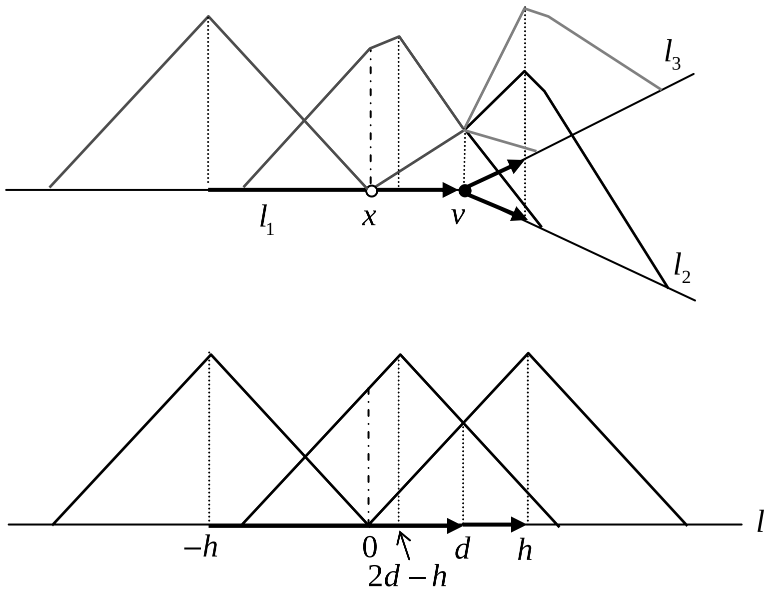


Figure 16: The density at x estimated by the kernel function moving along links l_1, l_2, l_3 (a) is equivalent to the kernel function moving along a line l (b).

We next consider the somewhat more general case shown in Figure 17a where a kernel support includes two nodes at which two or more links meet. Because the estimator at an arbitrary point x is obtained by adding the value of the kernel function placed at y and evaluated at x across any point y on the kernel support (the heavy lines in Figure 17a), we can decompose the integral domain. First we consider the integral domain shown in Figure 17b. As proved above (recall Figure 16a, b), the integral across line segments l_1, l_2, l_3 and l_4 is equivalent to the integral of the base kernel function across the line segment l_1' in Figure 17c. Then, in the same manner, the integral across the line segments l_5, l_6 and l_1' is equivalent to the integral of the base kernel function $k(x)$ across the line segment l_1'' in Figure 17d. Because the integral of the base function across the support line is unity, the kernel estimator of the kernel function $K_y(x)$ formulated above is unity. Therefore, the estimator is unbiased.

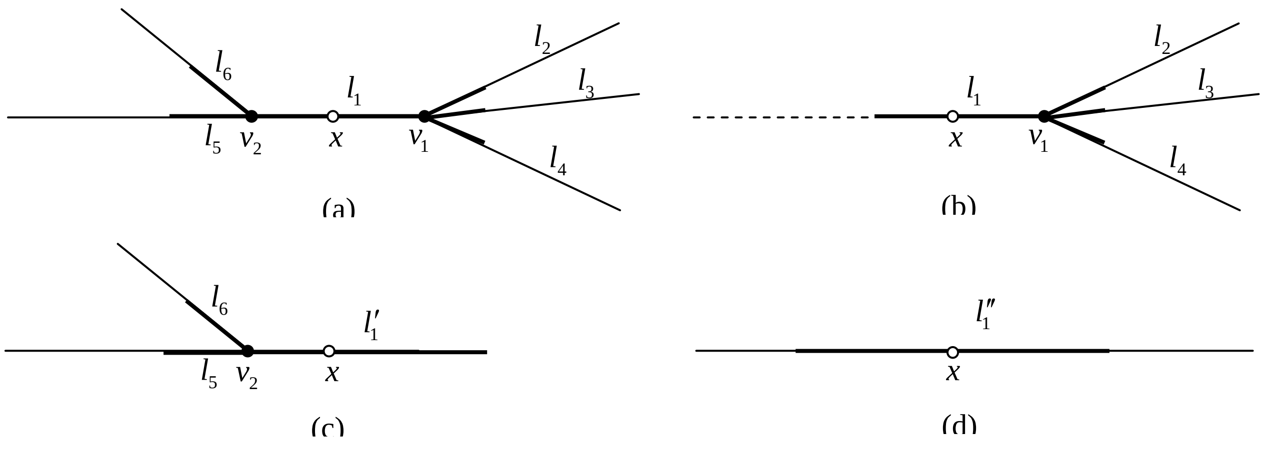


Figure 17: Equivalence of the density estimation at x by integrating over domains (a), (b), (c) and (d).

Last, we consider the most general case, in which a kernel support includes three or more nodes with degrees of three or more, as shown in Figure 18a. First we consider the integral along the line segments l_7, l_8 and l'_6 in Figure 18b. This value is equivalent to that for the line segment l_6 in Figure 17a. The integral along the line segments in Figure 17a is equivalent to that along the line segment in Figure 17d. Therefore, as shown above, the integral of the kernel function at x across y on the kernel support in Figure 18a is unity, showing that the estimator is unbiased. This completes the proof that the equal-split continuous kernel function is unbiased. This property holds for any base kernel function $k(x)$ forming the equal-split continuous kernel function $K_y(x)$ of equation (25); consequently, the equal-split continuous kernel function produces a broad class of kernel functions.

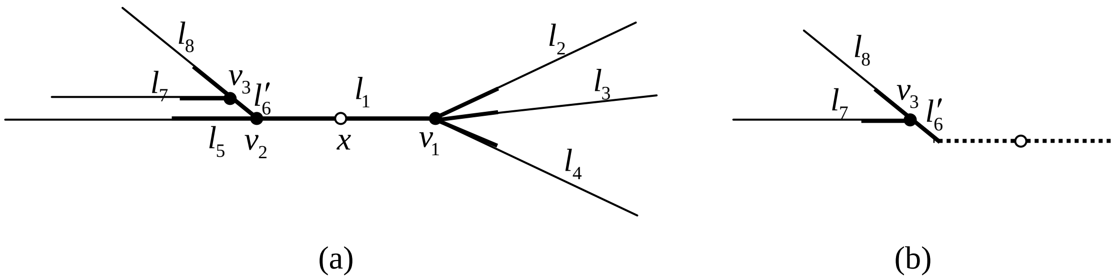


Figure 18: Equivalence of the density estimation at x in the general case.

In closing this section, we make two important remarks on the applicability of the equal-split function (Section 4) and the equal-split continuous function (Section 5).

First, we assumed that their kernel supports did not include cycles (Figure 2a–c). One reason for this was to avoid lengthy explanation. We should note that both kernel functions can be applied to cases in which their kernel supports include cycles (Figure 2d) and we can show that they give unbiased estimators (the proof is almost the same as in Section 5). However, the kernel functions have two modes, because their tails overlap on the cycles. Therefore, Property 1 (unimodal) is not satisfied.

Second, we obtained the unbiased estimators for $f(x)$ when the function $f(x)$ was given by equation (4), i.e., the uniform probability density function, because the CSR is one of the most fundamental hypotheses in spatial analysis. We usually start with this hypothesis, but if observed data do not support it we examine an alternative hypothesis. If $f(x)$ is explicitly given, we can obtain unbiased estimators for any function $f(x)$ through the following probability integral transformation (Okabe and Satoh, 2005).

Let $f_i(x)$ be part of $f(x)$ with its domain given by a link l_i of L; let t be the path distance from an end point of l_i to a point on l_i; let $f_i(t)$ be the density of points at t on l_i; $l_i' = f_i(l_i)$; and let z be given by $z(t)=\int_0^t f_i(t)\mathrm{d}t$. Then the probability density function $g_i(z)$ of z follows the uniform probability density function on l_i'. Let $L' = \{ l_1',...,l_n' \}$, $N'=(V,L')$, and $y_j' = f_i(y_j)$, for $y_j \in l_i$. Then the equal-split or equal-split continuous kernel functions generated at $y_1',...,y_n'$ on the network N' are unbiased estimators, as we have proved in the preceding sections (notice that $g_i(z)$, $i=1,..,n$ are the uniform probability density functions on N').

6. Computational implementation in the GIS environment

Having formulated two unbiased kernel functions in Sections 4 and 5, we now discuss how to implement these functions in the GIS environment and develop a user-friendly tool. Such a GIS-based tool is in great demand by nonmathematical researchers analyzing events occurring on networks in many fields, as discussed in the introduction, in particular for traffic accident analysis and street crime analysis.

The common task is to find the links that are within the shortest-path distance h from a given kernel center (where h is the bandwidth). This can be done using the shortest-path operation of the GIS. The underlying computation behind this operation is based on the computation of the shortest-path tree rooted at a given kernel center. The algorithms for this construction have been well developed in operations research, based on Dijkstra's (1959) algorithm (Aho et al., 1983). This has computational complexity of order O $(n_V \log n_V)$ where n_V is the number of nodes in a given planar network. Because the number of points having their density estimated on the network is n, the total order of computational complexity becomes O $(n n_V \log n_V)$.

Once the links within h are obtained, the next task is to compute the value of a kernel function on each link. When we use the equal-split kernel function in Section 4, the computation is done by equation (19), and it requires a constant time for each link. Because the number of links constituting the network is n_L, and the number of points is n, the total computational complexity is $\mathrm{O}(n n_L)$. This order is not affected by the complexity of a kernel support (compare with the next case).

When we use the equal-split continuous kernel function in Section 5, the computational complexity varies according to the complexity of the kernel support. In the simple case in which a kernel support consists of n links meeting at one node and the length of each link is not shorter than the bandwidth h (e.g., Figure 14a), the computation of equation (25) requires a constant time on each link. Therefore, the total computational complexity is $\mathrm{O}(n n_L)$, which is the same as that of the equal-split kernel function. In a complex case, in which a kernel support includes three or more nodes with degree three or more (e.g., Figures 17a and 18a), the computational complexity depends on the ratio of the bandwidth to the length of a link. We notice from the procedure in the Appendix that the computation requires a constant time operation executed $\prod_{i=1}^{i_{\max}} (h-d_i)/(d_{i+1}-d_i)$ times, where d_i is the distance from a kernel center to the ith-nearest node, and $i_{\max}$ is the maximum integer satisfying $d_i \le h$. This implies that if the length of a link approaches zero, the computation time increases to infinity. Therefore, if the length of a link is much smaller than that of the bandwidth, the computation becomes intractable in practice.

If visualization is required, we represent the network in terms of the many points forming the network (this may be called the raster representation in contrast to vector representation); we compute the values on all raster points (their number is denoted by m) with respect to n kernel functions having kernel centers placed at n points on the network. Therefore, the computation order for visualization is $\mathrm{O}(mn)$.

Considering that the users analyzing the density of points on networks are not only academic researchers but also practitioners, such as transportation managers, police officers and volunteers of environment NPOs (e.g., animal roadkills), we have developed a GIS plug-in tool for the kernel functions in Sections 4 and 5. The software was written in C++ and is independent of GIS software packages. For users' convenience, we have also developed an interface with ArcGIS, which will be included in SANET (Okabe et al., 2006a, b). The performance of this tool is discussed in the next section.

7. Application

This section shows an application of the kernel methods developed in the preceding sections to the density estimation of bag-snatches on streets in Kyoto, Japan. The street network consists of 5191 links and 6761 nodes, and the total length amounts to 305.241 km. The data source is a report by the Kyoto Police Office (http://www.pref.kyoto.jp/fukei/hanjou/). The number of locations where bag-snatches occurred on the street network was 146 (Figure 19) for the periods January–December in 2004 and December–February in 2005–2006. The digital street data were obtained from the Digital Map 2500 published from the Geographical Survey Institute, Japan.

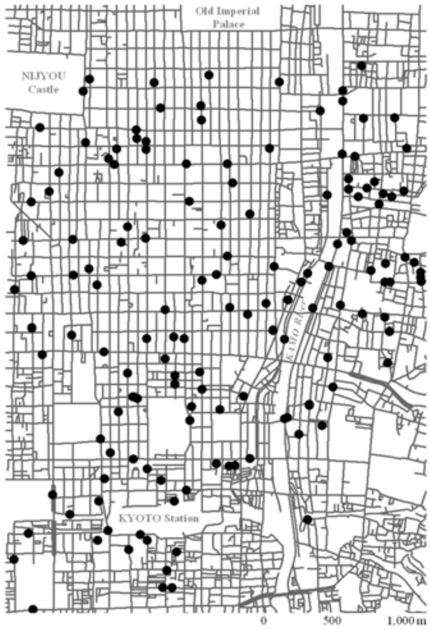


Figure 19: Locations of incidence of bag-snatches on the streets in Kyoto.

First, we applied the equal-split kernel function formulated in Section 4 to the data in Figure 19. Part of the estimated density is illustrated in Figure 20.

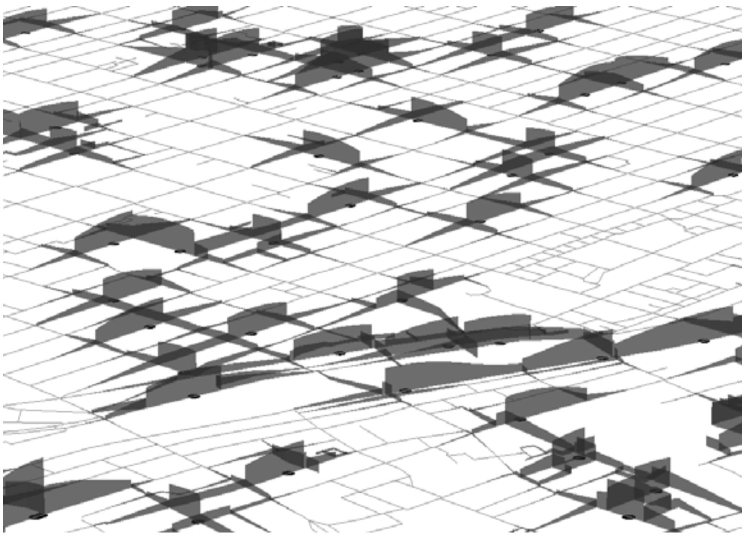


Figure 20: Density estimation of bag-snatches occurring on the streets in Kyoto by the equal-split kernel function.

Second, we applied the equal-split continuous kernel function formulated in Section 5 to the same data. The result corresponding to that in Figure 20 is depicted in Figure 21. The bandwidth was 230 m for both kernel functions.

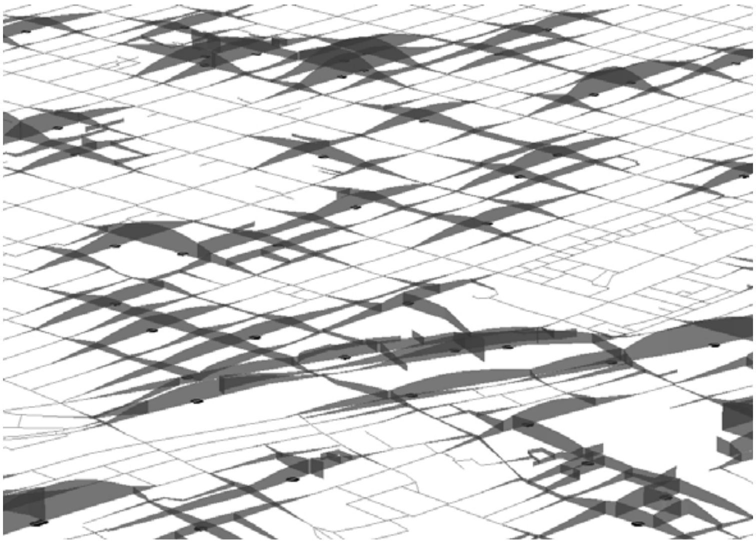


Figure 21: Density estimation of bag-snatches occurring on the streets in Kyoto by the equal-split continuous kernel function.

As Figures 20 and 21 show, the equal-split kernel function tends to show more concentrated densities than the equal-split continuous function. However, this difference would be weakened if the number of incidence locations increased. Actually, if bag-snatches occur according to the uniform distribution, as we proved in Sections 4 and 5, both functions eventually give a constant density over the network as the number of event locations increases.

As we noted in Section 6, in theory, one of the advantages of the equal-split kernel function is its short computation time. To confirm it in practice, we observed the computation times of both kernel functions. The computers used are DELL and HP. Their specifications were as follows. DELL: memory 2 GB, CPU Pentium 4 3.40 GHz; HP: memory 3.25 GB; CPU Intel Core (TM) 2 2.66 GHz. Both used Windows XP. The results are shown in Table 1.

Table 1: Computation times

Bandwidth (m)	DELL		HP	
	split kernel function (sec)	Equal-split continuous kernel function (sec)	split kernel function (sec)	Equal-split continuous kernel function (sec)
50	1.33	1.35	0.74	0.75
100	1.36	1.49	0.77	0.83
150	1.43	2.43	0.79	1.36
200	1.55	59.29	0.88	32.65
210	1.61	207.36	0.90	109.40
220	1.66	798.32	0.91	438.85
230			0.94	1777.49
240			0.99	
250	1.86		1.04	
300	2.74		1.50	

We notice from this table that the computation times were almost the same for the 50 m bandwidth. The computation time of the equal-split kernel function increases relatively slowly with bandwidth, but as we theoretically examined in Section 5, the computation time of the equal-split continuous kernel function increases rapidly when the bandwidth is larger than 200 m. Considering that the average length of a link is 59 m in Figure 19, we may say that if a kernel support includes more than four nodes, the estimation by the equal-split continuous kernel function requires much time, more than 30 minutes in the case of Figure 19. Therefore, if we deal with event-points on a network with links less than one quarter of the bandwidth, then the equal-split kernel function would be more practical than the equal-split continuous kernel function.

8. Conclusion

We have formulated three types of kernel function: the ‘similar’ shape, equal-split and equal-split continuous kernel functions in Sections 3, 4 and 5, respectively. Their properties are summarized in Table 2 under the assumption that the base kernel function satisfies Properties 1, 2 and 3.

Table 2: Properties satisfied by the 'similar' shape, equal-split and locally adjusted equal-split kernel functions.

Properties	1	2	3	4	5	6	7
Classes of kernel functions	unimodal	modal point =center	continuous	equal-distance/ equal-density	invariant w.r.t. angle	symmetric w.r.t. two kernel centers	unbiased
Similar shape (Section 3)	○	○	○	○	○	×	×
Equal-split (Section 4)	○	○	×	×	○	○	○
Equal-split continuous (Section 5)	○	×	○	×	○	×	○

○: satisfied ×: not satidfied

The 'similar' shape kernel function is a natural extension of the ordinary univariate kernel function, and many researchers would be tempted to use it. However, as was shown in Section 3, its estimator is biased. If we use this kernel function, we may conclude, for instance, that traffic accidents tend to cluster around nodes, contrary to the fact that they occur uniformly on a street network. The 'similar' shape kernel function is likely to give misleading conclusions and so we do not recommend its use for spatial analysis on a network.

The equal-spit kernel function satisfies five 'desirable' properties including the unbiased property that is not satisfied by the 'similar' shape kernel function. However, the function does not satisfy continuity at each node. This property looks peculiar in the univariate kernel functions, but as we discussed in Section 2, it does not always look peculiar on a network, because the topological property changes drastically at a node.

If we want to use a continuous kernel function at nodes, we can use the equal-split continuous kernel function of Section 5. However, that function does not satisfy Property 2 (the modal point and the kernel center do not coincide) or Property 6 (the kernel function is not symmetric with respect to two kernel centers).

From a practical viewpoint, the equal-split kernel function is more advantageous than the equal-split continuous kernel function in that the computational complexity of the former is much smaller than that of the latter when a network has short links compared with the bandwidth. We sometimes spent more than one hour running the tool for the equal-split continuous kernel function for actual street networks. Therefore, we suggest the use of the equal-split kernel function when a network includes many short links in relation to the bandwidth.

In closing this paper, we note two unsolved problems. First, having observed Table 2, one might wonder if any kernel function exists that satisfies all the properties listed there. We have searched for several years, but we have not yet found one. Nonetheless, we still feel that such a function exists.

Second, we obtained the unbiased estimators for the uniform probability density function on $N=(V,L)$ or for a given explicit function $f(x)$ on $N=(V,L)$, which is transformed into $N'=(V,L')$ by the probability integral transformation. In practice, however, the function $f(x)$ is unknown, and hence we would like to know under what conditions on $f(x)$, the equal-split kernel function or the equal-split continuous function are unbiased. This problem can be discussed to a certain extent in the case of L_S by applying the theorems obtained for the univariate kernel functions (for example, see the theorems in Section 6.2 of Scott (1992)). However, these theorems cannot be applied to the cases of $L_{(1)}$ and $L_{(\geq 3)}$, and we have not yet succeeded in deriving the corresponding theorems for these cases.

We hope that these problems will be solved by the reader.

Appendix

In a general case in which there is more than one node with degree three or more, such as the kernel support in Figure A1a, an explicit mathematical definition of the equal-split continuous kernel function would become very lengthy because there are many possible forms of kernel support. Therefore, we define it as a procedure.

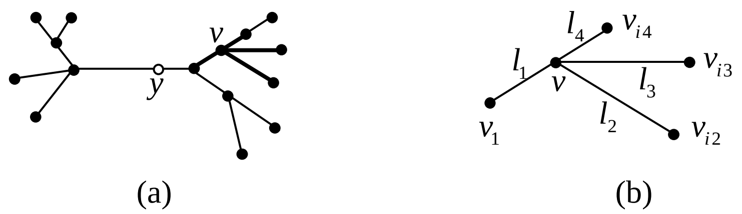


Figure A1: A kernel support (a) and an enlarged part of it (b)

We first construct a recursive function for determining the value of the kernel function $K_y(x)$ on the links incident to a node, v, of the kernel support with its center at y (e.g., the heavy line segments in Figure A1a, which are enlarged in Figure A1b). Let $n(v)$ be the degree of node v, and $L(v)=\{l_1,...,l_{n(v)}\}$ be the set of links incident to v, where l_1 is the link that is included in the shortest path from the kernel center y to v. Let v_i be a node of the link l_i other than the node v, and let α be a coefficient to be determined at every step, but at a node of degree one (i.e., an end node), we set α = 0.

Function $F(v,d,\alpha,l_1)$

Input: v

d : a distance variable (to be determined at every step)

α : a coefficient (to be determined at every step)

l_1

$n(v)$

$L(v)$

$k(x)$: a base kernel function

Output: $K_y(x)$, $x \in l_i, i = 1,...,n_v$

Step 1: For $x \in l_1$, $K_y(x) \leftarrow K_y(x) + \alpha k(x)$.

Step 2: $d \leftarrow d + d(v_1, v)$

If $h < d$, then stop.

Step 3: For all $l_i \in L(v)$.

Step 3.1: If $l_i = l_1$, $\alpha \leftarrow -\alpha \dfrac{n(v)-2}{n(v)}$;

otherwise, $\alpha \leftarrow \alpha \dfrac{2}{n(v)}$.

Step 3.2: If $\alpha \neq 0$, then $F(v,d,\alpha,l_i)$.

With this function, the equal-split continuous kernel function is defined by the following procedure. Note that for operational convenience, we divide the link on which the kernel center y is placed into two links and regard y as a node.

Procedure for defining the equal-split continuous kernel function

Step 1: For all points x on the kernel support L_y, $K_y(x) \leftarrow 0$,

$$\alpha \leftarrow \frac{2}{n(y)} .$$

$d \leftarrow 0$.

Step 2: For all links in $l_i \in L(y)$, compute $F(y,d,\alpha,l_i)$.

This procedure fixes the initial setting in Step 1 and the equal-split continuous kernel function is obtained through the function $F(v,d,\alpha,l_1)$ in Step 2. The value of the equal-split kernel function is successively determined along the paths starting from the kernel center y. Suppose that we are now in the jth phase and have reached a vertex v (e.g., v in Figure A1a) and obtained the equal-split continuous kernel function, $g^{(j)}(x)$ along the links forming the shortest path from the kernel center y to the vertex v (where x is the shortest-path distance from y to a point on the path, and $g^{(j)}(x)$ is written in terms of the degrees of nodes on the path and the base kernel function $k(x)$). After completing several steps in the above procedure, we can show that the kernel function in the ($j+1$)th phase for $l_i, i=2,...,n(v)$ is written as

$$g^{(j+1)}(x) = g^{(j)}(x) - \frac{n(v)-2}{n(v)} g^{(j)}(2d_v - x) \text{ for } x \in l_1, \tag{A1}$$

$$g^{(j+1)}(x) = \frac{2}{n(v)} g^{(j)}(x) \text{ for } x \in l_i, i = 2,...,n(v). \tag{A2}$$

From equation (A1), we obtain $g^{j+1}(d_v) = (2/n(v))g^{j}(d_v)$, and from equation (A2), we obtain $g^{j+1}(d_v) = (2/n(v))g^{j}(d_v)$. Therefore, the equal-split continuous kernel function is continuous at v. This property holds for any node of the kernel support, showing that the equal-split continuous kernel function is truly continuous.

We can show that the procedure in the function $F(v,d,\alpha,l_1)$ assigns the value $\int_{d_v}^{h} g^{(j)}(x)\mathrm{d}x$ in the jth phase to the values on $l_i, i=1,...,n(v)$ in the ($j+1$)th phase according to

$$\int_{2d_v-h}^{d_v} -\frac{n(v)-2}{n(v)} g^{(j)}(2d_v - x) g^{(j)}(x)\mathrm{d}x \text{ for } x \in l_1,$$

$$\int_{2d_v-h}^{d_v} \frac{2}{n(v)} g^{(j)}(x)\mathrm{d}x \text{ for } x \in l_i, i = 2,...,n(v).$$

The sum of these two values multiplied by $n(v)-1$ in the ($j+1$)th phase is equal to the value $\int_{d_v}^{h} g^{(j)}(x)\mathrm{d}x$ in the jth phase. Therefore, the value is retained at every phase. Because the initial value at y on the right-hand side is $\int_0^h k(x)\mathrm{d}x = 1/2$, and the same for the left-hand side, the integral of the equal-split continuous kernel function over the kernel support is unity. This proves that the equal-split continuous function is a kernel function.

Acknowledgements

This study was partly supported by the Grant-in-Aid for Scientific Research (B), No. 20300098, Japan Society for the Promotion of Science.

References

Aho, A.V., Hopcroft, J.E., and Ullman, J., (1983) *Data Structures and Algorithms*, Addison-Wesley.

Burruso, G. (2004) Network density and delimitation of urban arrears, *Transactions in GIS*, 7, 177-191.

Borruso, G. (2005) Network density estimation: analysis of point patterns over a network, Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, Chih Jeng Kenneth Tan (Eds.): *Computational Science and Its Applications* - ICCSA 2005, International Conference, Singapore, May 9-12, 2005, Proceedings, Part III. Lecture Notes in Computer Science 3482, 126-132.

Decker, B., Verheyen, K., Hermy, M. and Muys, B. (2005) Effects of landscape structure on the invasive spread pf black cherry *Prunus serotina* in an agricultural landscape in Flanders, Belgium, *Ecography*, 28, 99-109.

Cressie, N. (1991) *Statistics for Spatial Data*, New York: JohnWiley

Devroye, L. and Gyorfi, L. (1985) *Nonparametric Density Estimation The* L_1 *View*, New York: John Wiley.

Devroye, L. and Lugosi, G. (2000) *Combinatorial Methods in Density Estimation*, New York: Springer-Verlag.

Dijkstra, E.W. (1959) A note on two problems in connexion with graphs, *Numerische Mathematik*, 1, 269–271.

Eggermont, P.P.B. and LaRiccia, V.N. (2001) *Maximum Penalized Likelihood Estimation, Volume I: Density Estimation*, New York: Springer-Verlag.

Epanechinikov V.A. (1969) Nonparametric estimation of a multidimensional probability density, *Theory of Probability and its Applications*, 14, 153-158.

Flahaut, B., Mouchart, M., Martin, E.S., and Thomas, I. (2003), The local spatial autocorrelation and the kernel method for identifying black zones a comparative approach, *Accident Analysis & Prevention*, 35, 991-1004.

Lu, Y. and Chen, X. (2006) On the false alarm of planar *K*-function when analyzing urban crime distributed along streets, Social Science Research, 36, 611-632.

Miller, H.J., (2004), Market area delimitation with networks using geographic information systems, *Geographical Systems*, 1, 157-173.

Okabe, A., Okunuki, K. and Shiode, S. (2006a) The SANET Toolbox: New Methods for Network Spatial Analysis, *Transactions in GIS*, 10(4), 535-550.

Okabe, A., Okunuki, K. and Shiode, S. (2006b) SANET: A Toolbox for Spatial Analysis on a Network, *Geographical Analysis*, 38(1), 57-66.

Okabe, A. and Satoh, T. (2005) Uniform network transformation for points pattern analysis on a non-uniform network, *Journal of Geographical Systems*, 8, 25-37.

Okabe, A., and Yamada, I., (2001) The K-function method on a network and its computational implementation, *Geographical Analysis*, 33(3) , 271-290.

Okabe, A., Yomono, H., and Kitamura, M., (1995) Statistical analysis of the distribution of points on a network, *Geographical Analysis*, 27(2), 152-175.

Satoh, T. and Okabe, A. (2005) Methods for Finding Hot Spots on a Network, *Abstract of the 2005 Annual Meeting of the AAG*, April 5-9 2005, Denver, Colorado.

Scott, D.W. (1992) *Multivariate Density Estimation, Theory, Practice, and Visualization*, New York: John Wiley.

Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall.

Spooner, P.G., Lunt I.D., Okabe A. and Shiode S. (2004) Spatial analysis of roadside Acacia populations on a road network using the network K-function, *Landscape Ecology*, 19(5), 491-499.

Tapia, R.A., and Thompson, J.R. (1978) *Nonparametric Probability Density Estimation*, Baltimore: John Hopkins University Press.

Wand, M.P. and Jones, M.C. (1995) *Kernel Smoothing*, Chapman & Hall/CRC.

Yamada, I. and Thill, J-C. (2004) Comparison of Planar and Network K-functions in Traffic Accident Analysis. *Journal of Transport Geography*, 12, 149-158.

Yamada, I. and Thill, J-C. (2007) Local indicators of network-constrained clusters in spatial point patterns, *Geographical Analysis,* 39(3), 268-292.

Xu, Z. and Sui, D.Z. (2007) Small-world characteristics on transportation networks: a perspective from network autocorrelation, Journal of Geographical Systems, 9, 189-205.