Functional distributional clustering using spatio-temporal data

A. Venkatasubramaniam[a], L. Evers[b], P. Thakuriah[c], and K. Ampountolas[d,e]

[a]The Alan Turing Institute, The British Library, London, United Kingdom
[b]School of Mathematics and Statistics, University of Glasgow, United Kingdom
[c]E.J. Bloustein School of Planning & Public Policy, Rutgers University, NJ, USA
[d]James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom
[e]Department of Mechanical Engineering, University of Thessaly, Volos, Greece

ARTICLE HISTORY
Compiled November 25, 2021

ABSTRACT
This paper presents a new method called the *functional distributional clustering algorithm* (FDCA) that seeks to identify spatially contiguous clusters and incorporate changes in temporal patterns across overcrowded networks. This method is motivated by a graph-based network composed of sensors arranged over space where recorded observations for each sensor represent a multi-modal distribution. The proposed method is fully non-parametric and generates clusters within an agglomerative hierarchical clustering approach based on a measure of distance that defines a cumulative distribution function over temporal changes for different locations in space. Traditional hierarchical clustering algorithms that are spatially adapted do not typically accommodate the temporal characteristics of the underlying data. The effectiveness of the FDCA is illustrated using an application to both empirical and simulated data from about 400 sensors in a 2.5 square miles network area in downtown San Francisco, California. The results demonstrate the superior ability of the the FDCA in identifying *true clusters* compared to functional only and distributional only algorithms and similar performance to a model-based clustering algorithm.

KEYWORDS
Agglomerative hierarchical clustering; functional; distributional; spatial; non-parametric

1. Introduction

Clustering is an unsupervised learning method that maximises a measure of similarity between groups of objects to identify clusters with homogeneous characteristics [35, 41]. Heterogeneous datasets present diverse challenges for determining valuable insights and demand bespoke clustering algorithms that are able to accommodate multiple constraints (space, time, and network). Conventional methods of this exploratory approach include hierarchical [39] and partitional (e.g., k-means [29], Gaussian mixture models, etc.) techniques. Hierarchical methods (e.g., agglomerative process) generate a set of clusters in which smaller clusters are nested within larger clusters and a dendrogram illustrates the arrangement of clusters generated by the hierarchical clustering framework. On the other hand, k-means is a partitioning process which assigns objects to a pre-specified number of clusters.

CONTACT A. Venkatasubramaniam. Email: avenkatasubramaniam@turing.ac.uk

Numerous clustering methods have been developed for the purpose of clustering spatio-temporal datasets. The development of clustering methods are motivated by distinct characteristics such as the source of generated data (a static source (e.g., sensors at fixed locations across a road network) versus a dynamic source (e.g., trajectories corresponding to trips recorded across a network for a specific car) and choice of functional model. Hierarchical clustering algorithms have been modified to incorporate diverse constraints and generate spatially contiguous clusters. Several definitions of the distance measure to determine the notion of similarity for spatio-temporal data include the use of adapted Ward linkage [7, 8], coefficients of basis functions to smooth the observed data [16] and via a kernel estimator of the multivariate spatial data dependence structure [15]. Hierarchical clustering algorithms have also been used in combination with other models, as a two-stage process, to better incorporate space and time constraints (e.g., Bayesian spatio-temporal models [1, 23] and Kriging interpolation [6]). Other classical clustering algorithms have also been developed to identify differences in the shapes of curve patterns [11], to take advantage of diverse functional characteristics [9, 21, 22] or detect temporal patterns at a specific location of the network [10].

Dynamic clusters (e.g., formed by measuring similarities between trajectory pairs) correspond to partitions of space and their evolution over time and relevant clustering methods seek to accommodate changes in time and space simultaneously. Methods include the modified DBSCAN [2] approach, extended kernel density estimation [4, 40] and nearest neighbour methods [31]. More recently, an adaptive dynamic time warping method [26] using adaptive penalty functions was introduced to compute the distances between trajectories (e.g., to monitor moving behaviours and traffic patterns for a selected individual). An unsupervised learning method with a convolutional autoencoder (CAE) neural network, motivated by the use of deep learning, was recently proposed to compute (more robust) trajectory similarities [28].

This paper develops a novel functional distributional algorithm in a hierarchical agglomerative clustering framework to identify spatially contiguous clusters in a graph network and accommodate temporal characteristics at each vertex. This spatial clustering method for spatio-temporal data is specifically motivated by data generated from static sources. This non-parametric clustering method generates clusters that are distinguished by the nature and shape of curves rather than individual temporal observations at different vertices through the grid-style graph network. To the best of our knowledge, an algorithm that is both functional and distributional within a hierarchical clustering framework has not been introduced before. This simultaneous framework enables complex dependencies to be adequately modelled, which is not accommodated by existing methodologies. Further, by using the hierarchical clustering framework, this method retains its associated advantages (e.g., ease in interpretation and few assumptions).

The examples in this paper come from traffic modelling, where we assume that the urban road network is made up of junctions and road segments that link relevant junctions. For our analysis, we assume that the sensors are arranged in a network in a way that one can define a neighbourhood structure, or to be more precise, an adjacency matrix. More formally, we assume that the network of sensors can be represented as an undirected graph with sensors as vertices and edges linking neighbouring sensors. Occupancy is the percentage of time that a location on the road is occupied by vehicles and a measurement of occupancy that describes congestion is available for each junction and unit of time. Junctions which are joined directly by a road segment are considered to be adjacent and our objective is to identify contiguous areas of similar traffic patterns. For example, Figure 1 presents the distribution of occupancy observations aggregated over incoming links for an individual junction. This plot represents a bi-modal distribution for aggregated occupancy data, where levels of occupancy range between 0% and 100%. Successive jumps in occupancy levels over a period of time would be lost by clustering methods that fail to accommodate the distribution of occupancy levels and only include summary values. Instead, the distance measure incorporates multi-modal distributions by accounting for functions defined to accommodate temporal patterns.

The rest of the paper is organised as follows. Section 2 proposes the functional distributional clustering algorithm and describes methods to choose the optimal number of clusters and a measure of clustering similarity between identified clusters and a given set of 'true' clusters. The functional distributional clustering algorithm described in this paper is available for implementation in the R package *FdiClust*[1]. Section 3 presents an application of this algorithm to pre-defined data generated from an accurate micro-simulator for a 2.5 square miles network area in downtown San Francisco, CA. The simulation study evaluates the performance of the algorithm by comparing similarities between pairs of clusters. In Section 4, we illustrate the application of this algorithm to real data for the same traffic network and duration, but with no knowledge of underlying 'true' clusters. Finally, Section 5 summarises the algorithm and highlights its advantages and disadvantages.

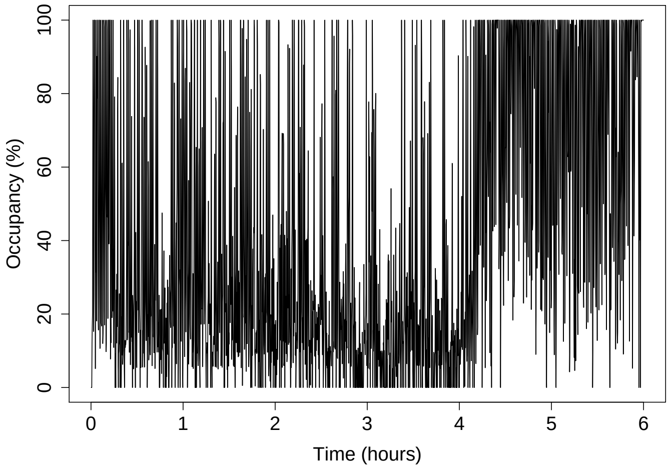


Figure 1. Occupancy measurements with bi-modal characteristics recorded over six hours for an individual junction.

2. Functional distributional clustering model

This section proceeds in two stages to develop the proposed clustering method that identifies spatially contiguous clusters across the network and incorporates temporal patterns of recorded observations. The first stage utilises a hierarchical agglomerative clustering algorithm and generates a series of cluster configurations. The clustering

[1] https://github.com/AshwiniKV/FdiClust

algorithm is built on a measure of distance that is defined using estimated conditional cumulative distribution functions (CDFs) for each cluster and determined utilising functions calculated over individual observations rather than aggregated observations. In the second stage, we use a clearly defined criterion to determine the optimal number of clusters and generate a distinct partition structure of the network. We also describe a measure of clustering similarity to examine the accuracy of identified clusters.

2.1. Hierarchical agglomerative clustering algorithm

Let $G = (V, E)$ be an undirected graph, where V is a set of vertices and E is a collection of edges linking neighbouring vertices. Assume that $V = \{v_1, \ldots, v_N\}$ and the adjacency matrix of graph G is a square matrix $\mathbf{W}$ with elements $W_{ij} = 1$ if $\{v_i, v_j\} \in E$ (i.e., if there is an edge between vertices v_i and v_j) and $W_{ij} = 0$ otherwise. Let observations for vertex j at time t_i be denoted by x_{ij}, where $i = 1, \ldots, n$ and $j = 1, \ldots, N$ and Table 1 describes the recorded observations. For example, at times $t_1, \ldots, t_n$, the observations at vertex $j = 1$ are recorded as $x_{11}, x_{21}, \ldots, x_{n1}$ and at vertex $j = N$ are recorded as $x_{1N}, \ldots x_{nN}$.

Table 1. Representation of the observations x_{ij} recorded over time $t_i = t_1 \ldots t_n$ for $j = 1 \ldots N$ vertices.

i	1	2	3	$\ldots$	n
times (t_i)	t_1	t_2	t_3	$\ldots$	t_n
Vertex $(j = 1)$	x_{11}	x_{21}	x_{31}	$\ldots$	x_{n1}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Vertex $(j = N)$	x_{1N}	x_{2N}	x_{3N}	$\ldots$	x_{nN}

The probability density function (PDF) for observations relevant to vertex j is defined by,

$$\hat{f}^{(j)}(x_0) = \frac{1}{nh_x} \sum_{i=1}^{n} \phi\left(\frac{x_{ij} - x_0}{h_x}\right),$$

while the estimated conditional probability density function (PDF) is defined as [17, 19, 27],

$$\hat{f}^{(j)}_{t_i}(x_0) = \frac{1}{h_x} \sum_{i=1}^{n} \phi\left(\frac{x_{ij} - x_0}{h_x}\right) w_{t_0}(t_i), \tag{1}$$

where

$$w_{t_0}(t_i) = \frac{\phi\left(\dfrac{t_0 - t_i}{h_t}\right)}{\displaystyle\sum_{\eta=1}^{n} \phi\left(\dfrac{t_\eta - t_i}{h_t}\right)},$$

$\phi(.)$ is a standard normal PDF, h_t is a bandwidth defined for time and h_x is a bandwidth which corresponds to recorded observations. Let a set of clusters $\mathcal{C}_{l=1}$ be initially represented by $\mathcal{C}_1 = \{C_1, \ldots, C_k\} = \{\{1\}, \{2\}, \ldots, \{N\}\}$, where each cluster is composed of a single vertex. At subsequent levels of the algorithm, clusters are consolidated to eventually form a single larger cluster composed of all N vertices in the network. The conditional probability density function for a cluster C is determined over observations recorded for relevant vertices and is defined as,

$$\hat{f}_{t_i}^{(C)}(x_0) = \frac{1}{|C|}\sum_{j\in C}\hat{f}_{t_i}^{(j)}(x_0). \tag{2}$$

The estimator of the conditional cumulative distribution function (CDF) is defined as

$$\hat{F}_{t_i}^{(j)}(x_0) = \sum_{i=1}^{n}\Phi\left(\frac{x_{ij}-x_0}{h_x}\right)w_{t_0}(t_i), \tag{3}$$

where $\Phi(\cdot)$ is a standard normal CDF, and

$$\hat{F}_{t_i}^{(C)}(x_0) = \frac{1}{|C|}\sum_{j\in C}\hat{F}_{t_i}^{(j)}(x_0). \tag{4}$$

A single observation provides less information about temporal patterns for each vertex compared to a single value in $\hat{F}_{t_i}^{(j)}(x_0)$. A pair of clusters C_1 and C_2 are merged if they have the lowest distance compared to distances calculated for all other pairs of clusters. The distance d is built using a L_1 norm, rather than the more commonly used L_2 norm or squared L_2 norm and distance d is determined over estimated conditional CDFs rather than individual observations. Let the distance d between cluster C_1 and cluster C_2 at time t_i be defined as the area between the two CDFs, i.e.,

$$d\Big(\hat{F}_{t_i}^{(C_1)}(\cdot),\hat{F}_{t_i}^{(C_2)}(\cdot)\Big) = \int\Big|\hat{F}_{t_i}^{(C_1)}(x_0)-\hat{F}_{t_i}^{(C_2)}(x_0)\Big|dx_0 \approx \Delta\sum_{s=1}^{S}\Big|\hat{F}_{t_i}^{(C_1)}(\xi_s)-\hat{F}_{t_i}^{(C_2)}(\xi_s)\Big| \tag{5}$$

for a regular grid $\xi_1,\ldots,\xi_S$ with $\xi_{s+1}-\xi_s=\Delta$.

Accordingly, let D be a distance matrix, where distance between cluster C_1 and C_2 in the matrix is defined as the sum of the above distance over time t_1, …, t_n,

$$D_{C_1,C_2} = \begin{cases}\displaystyle\sum_{i=1}^{n} d\big(\hat{F}_{t_i}^{(C_1)}(\cdot),\hat{F}_{t_i}^{(C_2)}(\cdot)\big), & \text{if } C_1\sim C_2,\\ \infty, & \text{otherwise,}\end{cases} \tag{6}$$

where $C_1\sim C_2$ indicates that calculating the distance between clusters is feasible only if an edge exists between any two vertices in the clusters. This condition helps enforce spatial contiguity in the formation of clusters and two clusters are merged at each iteration such that they correspond to the lowest computed distance d. The CDFs corresponding to the clusters C_1 and C_2 are also merged as,

$$\hat{F}_{t_i}^{(C_1\cup C_2)}(x_0) = \frac{|C_1|}{|C_1|+|C_2|}\hat{F}_{t_i}^{(C_1)}(x_0) + \frac{|C_1|}{|C_1|+|C_2|}\hat{F}_{t_i}^{(C_2)}(x_0). \tag{7}$$

Updated CDFs are then utilised to calculate the distance d at each subsequent iteration and this process continues until a single larger cluster containing every vertex in the network is obtained. In a hierarchical clustering approach, a partition then occurs at each iteration to determine non-overlapping clusters.

Algorithm 1: Functional distributional clustering

```
Input  : Initialize $\mathcal{C}_{l=1}$, where $\mathcal{C}_1 = \{C_1 \dots, C_k\} = \{\{1\}, \dots, \{N\}\}$.
Output: Hierarchical set of clusters, $\zeta$.
if $|\mathcal{C}_l| > 1$ then
    (1) For all pairs of clusters, compute distance $d$ as defined in Equation (6).
    (2) Set $\{C_1, C_2\} = \underset{C_1, C_2 \in \mathcal{C}_l}{\text{argmin}}(D_{C_1,C_2})$ to identify the pair of clusters that
        correspond to the minimum distance.
    (3) Merge the pair of clusters $C_1$ and $C_2$ as $C_1 \cup C_2$.
    (4) Update $\mathcal{C}_l$ to $\mathcal{C}_l \backslash \{C_1, C_2\} \cup \{C_1 \cup C_2\}$ and $\hat{F}_{t_i}^{(C_1)}(x_0)$ and $\hat{F}_{t_i}^{(C_2)}(x_0)$ using
        Equation (7).
else
    return $\zeta$;
end
```

2.2. Bandwidth selection

This section addresses the selection of smoothing parameters or bandwidths to estimate the conditional PDF $\hat{f}_{t_i}^{(j)}(x_0)$ defined in Equation (1). A data driven method such as cross-validation [3, 18, 37] selects the bandwidth that corresponds to the minimum of the expected loss function and avoids the arbitrary selection of bandwidths that can lead to under smoothing or over smoothing. We use an extended cross-validation method [14] to select optimal bandwidths h_x and h_t and denote an estimated conditional PDF for a cluster C dependent on the bandwidths as $\hat{f}_{t_i}^{(C)h}(x_0)$. The integrated squared error (ISE) is defined as,

$$\begin{aligned}
\text{ISE} &= \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \{\hat{f}_{t_i}^{(C)h}(x_0) - f_{t_i}^{(C)}(x_0)\}^2 \, dx_0 \right) \\
&= \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0)^2 \, dx_0 - \frac{2}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0) f_{t_i}^{(C)}(x_0) \, dx_0 \right. \\
&\quad \left. + \frac{1}{n} \sum_{i=1}^{n} \int f_{t_i}^{(C)}(x_0)^2 \, dx_0 \right). \qquad (8)
\end{aligned}$$

The last term is not dependent on bandwidth h and accordingly can be ignored in the bandwidth selection process. A reasonable estimator of the ISE is,

$$CV(h) = \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0)^2 \, dx_0 - \frac{2}{n|C|} \sum_{i=1}^{n} \sum_{j \in C} \hat{f}_{t_i,-ij}^{(C)h}(x_{ij}) \right). \qquad (9)$$

The optimal bandwidth parameter corresponds to the minimum cross validation error $\hat{h} = \arg\min_{h^*} CV(h^*)$. In practice, an initial estimate of $h_x = 10$ is utilised to determine optimal bandwidths, i.e., h_x and h_t through a grid search. One could argue that the bandwidth should be re-tuned for each update in the cluster structure; however, to reduce the computational footprint we determine the optimal bandwidth only at the beginning of the algorithm. Towards the end of the algorithm clusters are substantially bigger and there could be scope to further reduce the bandwidths. We have found that using the same bandwidth throughout the algorithm usually gives similar clusterings.

2.3. Optimal number of clusters

A major challenge in clustering is the identification of the optimal number of clusters. In hierarchical clustering algorithms, the assignment of parameters to determine clusters often relies on the number of 'true' clusters, which may not necessarily be available or easily defined. Methods of cluster validation to determine the 'true' number of clusters include the CH index [5], Dunn index [13], Davies-Bouldin index [12], and the Silhouette index [36] and these methods seek to identify compact and well separated clusters, where clusters are deemed to be more distinct for smaller values of the index. In comparison to other methods, the time complexity for computation of the Davies-Bouldin index was found to be far lower than for the Silhouette method [32]. Alternatively, the *gap statistic* [38] compares within-cluster errors in the observed data to within-cluster errors calculated for data from an appropriate null reference distribution and removes the need for calculating validation scores. However, the need to bootstrap samples in the gap statistic approach leads to the method being rather computationally expensive and inefficient for calculating the number of clusters.

We modify the *clustering balance criterion* [24], a method similar to the Davies-Bouldin index, to compare the inter-cluster distances and intra-cluster distances in a computationally efficient manner for larger datasets. Let the aggregated CDF over all sensors in a cluster C be defined as $F_{t_i}^{(C)}(\cdot) = \frac{1}{|C|}\sum_{j\in C} F_{t_i}^{(j)}(\cdot)$. Using this definition, let $\Lambda = \sum_{C\in\mathcal{C}_l}\sum_{j\in C} d\big(F_{t_i}^{(j)}(\cdot), F_{t_i}^{(C)}(\cdot)\big)$ be the intra-cluster distance sum calculated for all k identified clusters in $\mathcal{C}_l$. The inter-cluster distance sum is defined by $\Gamma = \sum_{C\in\mathcal{C}_l} d\big(F_{t_i}^{(C)}(\cdot), F_{t_i}^{(C_0)}(\cdot)\big)$, where $F_{t_i}^{(C_0)}(\cdot) = \frac{1}{|\mathcal{C}_l|}\sum_{C\in\mathcal{C}_l} F_{t_i}^{(C)}(\cdot)$. Within an agglomerative hierarchical clustering framework, the intra-cluster sum Λ has zero distance for singleton clusters and this value is maximised when all sensors in the network belong to a single cluster. On the other hand, the inter-cluster sum Γ is minimised when all sensors belong to a single cluster and maximised when each sensor is a singleton cluster. Accordingly, the clustering balance is defined as $\epsilon = \alpha\Lambda + (1-\alpha)\Gamma$, where weights α and $1-\alpha$ are assigned to Λ and Γ. In the examples, we used an α value of 0.5.

The hierarchical clustering algorithm described above yields a sequence of nested partitions. We then retain the partition minimising the above modification of the clustering balance criterion, which is deemed to have optimal number of clusters.

2.4. Measure of clustering similarity

The optimal number of clusters determines objects within each cluster by utilising the constructed hierarchy of clusters. This set of defined clusters and their elements are compared against external criteria such as a pre-defined cluster structure or known set of labels. Let a set of vertices in the network be defined as $\mathcal{J} = \{1, 2, 3, \ldots N\}$ and $\mathcal{U}$ and $\mathcal{V}$ are two partitions of $\mathcal{J}$, where $\mathcal{U} = \{U_1, \ldots, U_u\}$ is defined as the set of u true clusters and $\mathcal{V} = \{V_1, \ldots, V_v\}$ represents a clustering result composed of v clusters. Let a be the number of pairs of vertices in $\mathcal{J}$ that are in the same cluster within $\mathcal{U}$ and the same cluster within $\mathcal{V}$, b be the number of pairs of vertices in $\mathcal{J}$ that are in the same cluster in $\mathcal{U}$ but not the same cluster in $\mathcal{V}$, c be the number of pairs of vertices in $\mathcal{J}$ that are not in the same cluster in $\mathcal{U}$ but in the same cluster in $\mathcal{V}$, and d be the number of pairs of vertices in $\mathcal{J}$ that are in different clusters for both $\mathcal{U}$ and $\mathcal{V}$. Similarity measures between clustering results and 'true' clusters can be calculated using a method called the *Rand index* (RI) [34]. The Rand index is then defined as

$$\mathrm{RI} = \frac{a+d}{a+b+c+d}, \tag{10}$$

where $a+d$ refers to the number of agreements between the clustering output of the developed algorithm and the given truth and $a+b+c+d$ includes both agreements and disagreements. Values of the RI lie between 0 and 1, where 0 represents little agreement and 1 represents strong agreement. However, the expected value of the RI for two random partitions does not necessarily take a constant value and the RI approaches an upper limit of unity as the number of clusters increases.

A modified version of the RI was introduced by [20] to account for problems within the RI method and is called the *Adjusted Rand index* (ARI). In general, a larger ARI indicates a higher agreement between two partitions and the ARI has a maximum value of 1 but can also take negative values. This index is typically recommended as the choice for measuring agreement between any two clustering results even when the number of clusters are different [30] and is computed using:

$$\frac{(a+b+c+d)(a+d) - [(a+b)(a+c) + (c+d)(b+d)]}{(a+b+c+d)^2 - [(a+b)(a+c) + (c+d)(b+d)]}. \tag{11}$$

3. Simulated occupancy data

In this section, the group of sensors arranged as a network correspond to junctions within an urban road network, where adjacent junctions are linked by road segments. An urban road network constitutes a network which can be represented as an undirected graph with junctions as vertices and road segments that link relevant junctions as edges.

3.1. Data

We simulate occupancy data over a 2.5 square miles network area in downtown San Francisco, California composed of $N = 158$ junctions and 316 links to reflect a heterogeneous network composed of homogeneous clusters. Correlated occupancy data is generated in R version 3.4.2 [33] using a spatio-temporal precision matrix to define three distinct clusters in the network, where within each cluster in $\mathcal{C}_l$, a given state space model generates zero and one values corresponding to defined occupancy levels. We assume that each junction within an urban road network has a maximum of four links to adjacent junctions. The presence of a limited number of road segments between junctions in the network leads to a sparse spatial precision matrix modelled as a type of conditional auto-regressive (CAR) model [25]. The temporal precision structure is defined as a first order auto-regressive model (AR-1) and occupancy observations for each junction are recorded over a period of six hours (21600 seconds) with a sampling rate of 60 seconds.

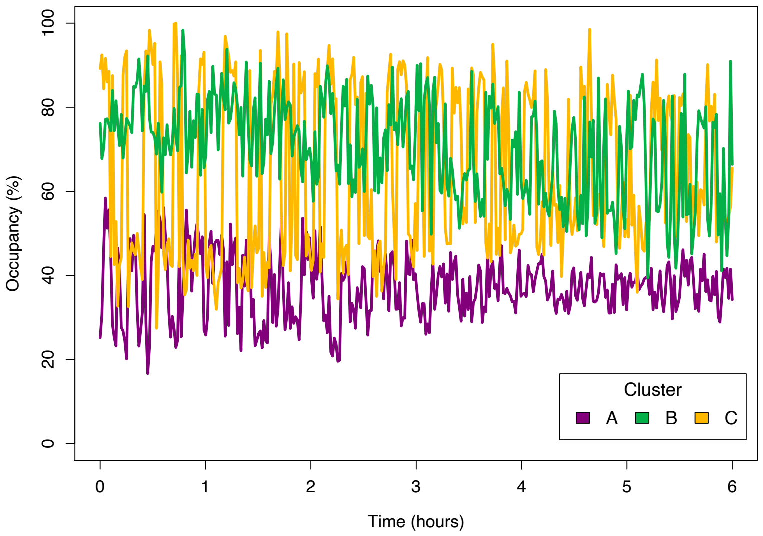


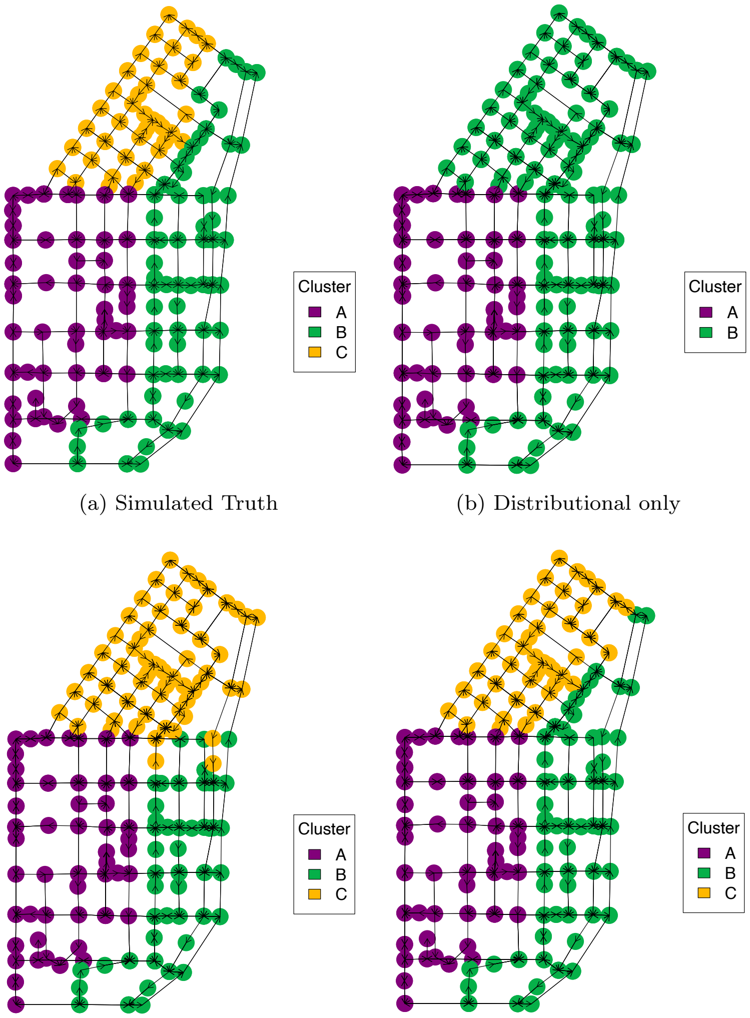
Figure 2. Occupancy measurements generated for three distinct clusters.

Figure 2 illustrates the simulated occupancy data to represent distinct clusters. Occupancy values (20 – 50%) displayed in purple for cluster A are typically lower and variations in jumps between successive observations reduce over time. The values (40 – 100%) plotted in yellow for cluster C are composed of both higher and lower values, with differences between successive observations reducing marginally over time. Occupancy values (70 – 100%) in green for cluster B are typically higher in the first three hours and display greater variation (50 – 90%) over the next three hours.

3.2. Results

The proposed algorithm introduced in Section 2.1 is applied to simulated occupancy observations generated within the urban network as described in Section 3.1. Each junction is initially treated as a singleton within the agglomerative clustering framework. The conditional CDF $F_{t_i}^{(C)}(x_0)$ for a cluster C is estimated over a sample of 360 observations, where bandwidths $h_x = 10$ (occupancies recorded in %) and $h_t = 6$ (time in seconds) are selected using the extended cross validation method described in Section 2.2. Conditional CDFs are estimated for each cluster and stored outside individual iterations of the algorithm to improve the proposed algorithm's computational efficiency. The distance d is calculated between adjacent clusters using Equation (5) and (6) and individual clusters are merged at each iteration of the algorithm corresponding to the minimum distance. This process stops when all junctions belong to a single larger cluster and we obtain a series of merged clusters from the hierarchical clustering algorithm.

Figure 3 displays networks with clusters identified by three different clustering algorithm scenarios and the defined 'true' clusters. These 'true' clusters in Figure 3(a) correspond to the simulated occupancy data in Figure 2. Figure 3(b) displays clusters identified when the distance measure uses (1) and (3) with only observations over time and without the functions ϕ and Φ. Cluster C is not identified as distinct from cluster B and the *distributional only* algorithm is unable to determine the 'true' clusters. In particular, the algorithm is unable to identify the cluster C which is composed of



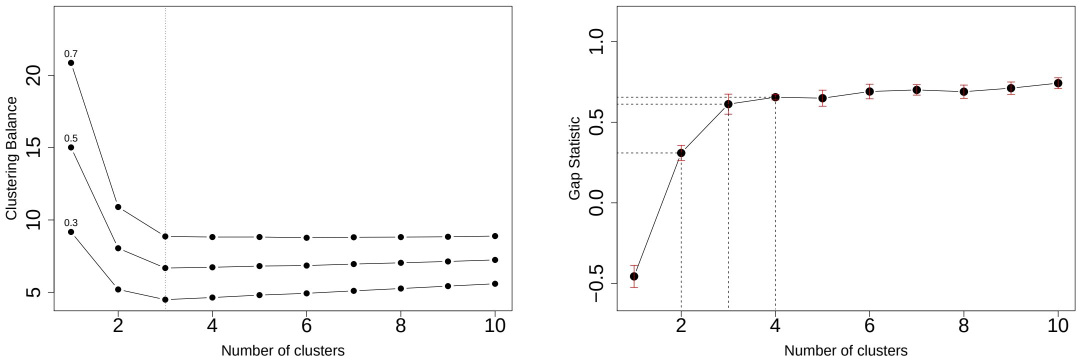
(a) Simulated Truth

(b) Distributional only

(c) Functional Only

(d) Functional and Distributional

Figure 3. The FDCA applied to data simulated in the network for a period of six hours.



(a) Clustering balance criterion

(b) Gap statistic

Figure 4. Methods to determine the optimal number of clusters.

occupancy observations that successively jump between high and low values. Figure 3(c) depicts three clusters identified by the *functional only* algorithm, where Equation (3) is determined using observations aggregated over time. In Figure 3(c), the identified clusters reflect the diminished ability of the algorithm to distinguish between cluster C and cluster B as compared to the clusters identified in Figure 3(d). The clustered network in Figure 3(d) displays results of the *functional distributional clustering* algorithm that calculates $F_{t_i}^{(C)}(x_0)$ using all components in (3). This algorithm is functional and distributional because distance measures are calculated using conditional CDFs for occupancy observations recorded over time. The clusters identified by the functional distributional algorithm are nearly equivalent to the three 'true' clusters displayed in Figure 3(a). This indicates the ability of the functional distributional algorithm to recover the true spatially contiguous clusters when each cluster corresponds to a distinct distribution of occupancy observations.

The optimal number of clusters within the network is determined using both the commonly used gap statistic and a clustering balance criterion defined in Section 2.3. For each clustering algorithm, the gap statistic and clustering balance criterion are calculated for scenarios ranging from when the network has ten clusters to a scenario when the all the sensors belong to a single cluster. Figure 4(a) and Figure 4(b) display the clustering balance criterion and gap statistic against the corresponding number of clusters for results determined by the functional distributional clustering algorithm. The clustering balance criterion selects $k = 3$ for $\alpha = 0.5$ and for higher and lower values of α. The gap statistic chooses minimum k such that $\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$ and this rule also determines that $k = 3$. However, determining bootstrap samples for the gap statistic is computationally expensive and we utilise the clustering balance criterion to determine the optimal number of clusters in Section 3.3 and Section 4.

To compare the clusters identified by the functional and distributional clustering algorithm to the 'true' clusters displayed in Figure 3(a), we calculate the Adjusted Rand index discussed in Section 2.4. ARI indicates agreement between a set of clusters $\mathcal{V}$ that is determined by the functional distributional clustering algorithm and a set of 'true' clusters $\mathcal{U}$ and is equivalent to 0.93. Similarly, $\mathcal{V}$ determined by the functional only algorithm results in an ARI of 0.68 for three clusters and $\mathcal{V}$ determined by the distributional only algorithm leads to an ARI of 0.57 for two identified clusters. The functional only algorithm is unable to correctly identify all the junctions belonging to

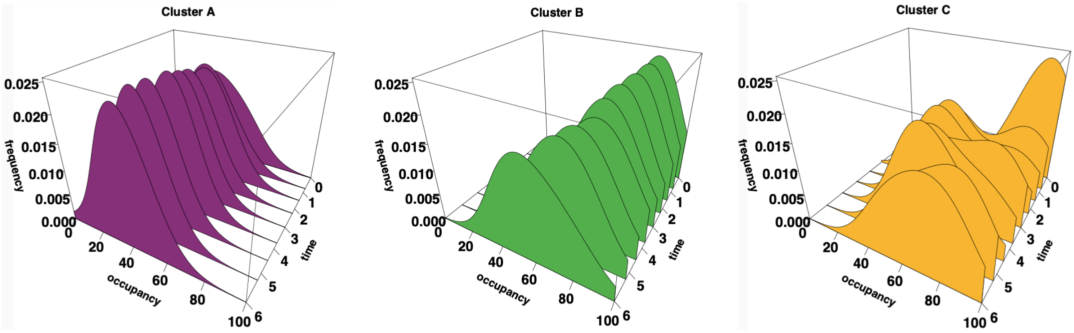


Figure 5. Three-dimensional density plots for distinct clusters determined using the functional distributional clustering algorithm.

cluster B and the distributional only algorithm is able to only identify two out of three distinct clusters.

Figure 5 displays three-dimensional density plots for occupancy observations that correspond to clusters identified by the functional distributional algorithm in Figure 3(d). These plots describe a relationship for each cluster between 100 occupancy observations (values between 0% and 100%), a time period of six hours (21600 seconds) with a sampling rate of sixty seconds and estimates for a Gaussian kernel density (over occupancy observations within the relevant cluster) with bandwidth equivalent to 15%. This value of bandwidth enables meaningful comparisons among curves within a cluster; lower values result in 'choppier' density curves that inhibit the ability to identify differences.

In Figure 5, the sub-plot for cluster A represents observations with density levels between 0.015 and 0.025 but are concentrated at lower occupancy levels between 10% to 40%. There is also a steady increase in density values over six hours. The sub-plot for cluster B displays observations with density levels reaching approximately 0.020 and occupancy levels concentrated between 30% to 75%. This sub-plot also reflects the concentration of occupancy data for cluster B in Figure 3(d) towards higher levels over the first few hours and a decrease in concentration reflected by lower density over the latter half of the time period. The sub-plot for cluster C represents varied density and occupancy levels through the observed time period. This corresponds to the variation identified within the cluster C in Figure 3(d) and reflects the ability of the clustering approach to adequately represent the differences in the shapes of curves and the spread of occupancy values over time described in Figure 2.

3.3. Simulation study

This section provides a quantitative analysis of the proposed functional distributional clustering algorithms to validate the clustering results in Section 3.2 for varied/various datasets. To this end, we simulated datasets as described in Section 3.1 with seeds from one to hundred to evaluate the developed algorithm's ability to identify clusters. The determined cluster structure is compared to the 'true' number of clusters as described in Figure 3(a). For a given seed, the optimal number of clusters is determined using the defined clustering balance criterion. At the selected number of clusters, the ARI measures its agreement to the 'true' number of clusters. We average the ARI over all simulation results and present a comparison between the functional distributional algorithm, the functional only algorithm, and the distributional only algorithm. The

Table 2. Results aggregated over 100 simulations with varied seeds for the functional distributional clustering algorithm, functional only algorithm, and distributional only algorithm.

	ARI		Number of Clusters		
Algorithm	Mean	SE	25th Q	50th Q	75th Q
Functional Distributional	0.85	0.174	3	3	4
Functional only	0.69	0.176	2	3	3
Distributional only	0.59	0.070	2	2	2

mean and corresponding standard error of the ARI for all three algorithms are presented in Table 2. In addition, the 25th quantile, the median, and the 75th quantile of the determined optimal number of clusters are described for different algorithms.

The functional distributional algorithm generates clusters that are reasonably similar to the defined 'true' clusters, as indicated by the aggregated ARI value equivalent to 0.85. The functional only algorithm has a lower mean ARI equivalent to 0.69 while the distributional only clustering algorithm struggles to identify three clusters with ARI equivalent to 0.59. This is reflected by the lower ARI and the suggested two optimal clusters.

In Table 3, the effectiveness of kernel density estimation is compared to coefficients from a b-spline basis function and principal component scores. Both these modifications are closer to the performance of the functional only and distributional only frameworks rather than the superior performance of the functional distributional clustering algorithm. These simulations help highlight the effectiveness of modified kernel density estimation within the functional distributional framework. A recently developed model-based clustering method [9] is also used for comparison in Table 3 and is referred to as the STM model. The STM mixture model (with three mixture components) is adapted for application to the simulated network. Each individual component (within the mixture model) is an autoregressive polynomial regression with logistic weights that are based on the space and time dimensions. Parameters are then estimated using an expectation maximisation algorithm within a maximum likelihood framework. As displayed in Table 3, the performance of the functional distribution clustering algorithm within the hierarchical framework is very similar to the STM method.

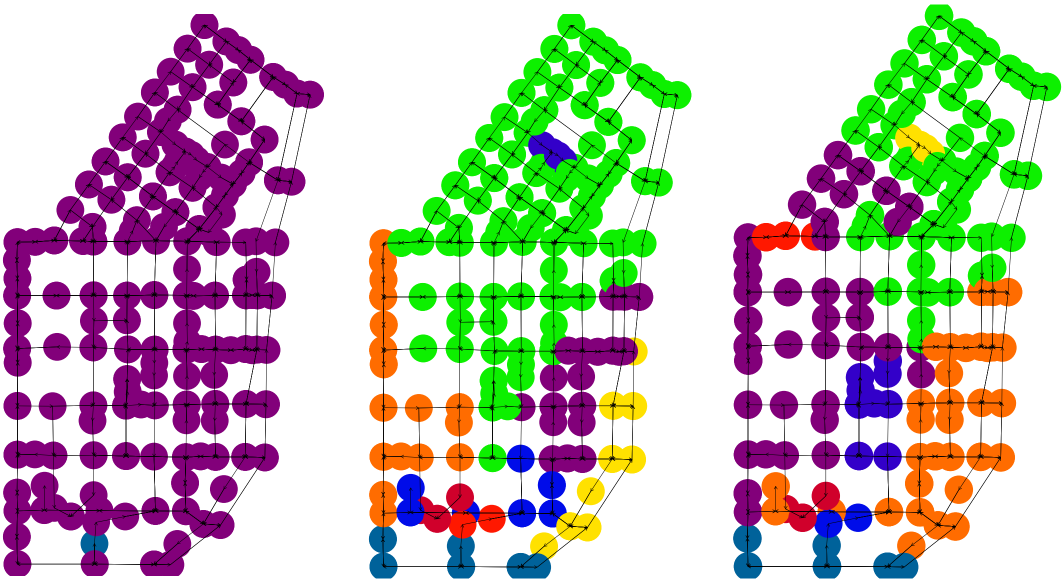
Table 3. Results aggregated over 100 simulations with varied seeds for variations of the functional distributional clustering algorithm and the SpaTimeClus method.

	ARI		Number of Clusters		
Algorithm	Mean	SE	25th Q	50th Q	75th Q
Coefficients of B-spline basis	0.63	0.143	2	2	2
Principal component scores	0.58	0.258	3	4	8
SpaTimeClus	0.87	0.128	3	3	3

4. Application

4.1. Occupancy data

To illustrate the functional distributional algorithm, we apply the developed clustering method to occupancy data generated for the 2.5 square miles network area in downtown San Francisco, CA. High resolution spatio-temporal data for urban road networks



(a) Distributional only

(b) Functional only

(c) Functional and Distributional

Figure 6. Clustering results using micro-simulated data over four hours.

are not readily available in open data sources and so we use an AIMSUN microscopic traffic simulator to mimic relevant origin-destination traffic demand scenarios. These scenarios are simulated to broadly represent three different clusters. 120 observations are recorded over six hours (21600 seconds) with a sampling rate of 180 seconds and we seek to identify the differences in occupancy levels that reflect the spread of congestion across the network. Since data within the first two hours is limited to very low levels of occupancy across the network, the functional distributional algorithm is applied to 80 occupancy observations recorded between 10 am to 2 pm (14400 seconds).

4.1.1. Results

In the described dataset, the underlying structure in the network for the 'true' number of clusters is unavailable and making assumptions of the partition structure is challenging. The functional distributional algorithm is implemented using the distance measure specified in Equation (6) and the bandwidths are calculated using the extended cross validation method described in Section 2.2. Selected bandwidths for h_x and h_t are equivalent to 15 (occupancy in %) and 7.5 (time in seconds) and conditional functions are estimated over the sample of 80 occupancy observations. The clustering balance criterion suggests optimal number of clusters for the functional and distributional algorithm, functional only algorithm and distributional only algorithm. In Figure 6(c), the functional distributional clustering algorithm partitions a network into nine clusters with three main clusters (green, purple, and orange). This is in contrast to the clusters obtained in Figures 6(a) and 6(b), where the clustering balance criterion suggests a single larger cluster for the distributional only clustering algorithm and a main larger cluster along with several smaller clusters for the functional only clustering algorithm.

Figure 7 displays the corresponding density distributions for the clusters determined by the functional distributional clustering algorithm. Within a sub-plot for an individual cluster, Gaussian density curves (bandwidth equivalent to 15%) over relevant

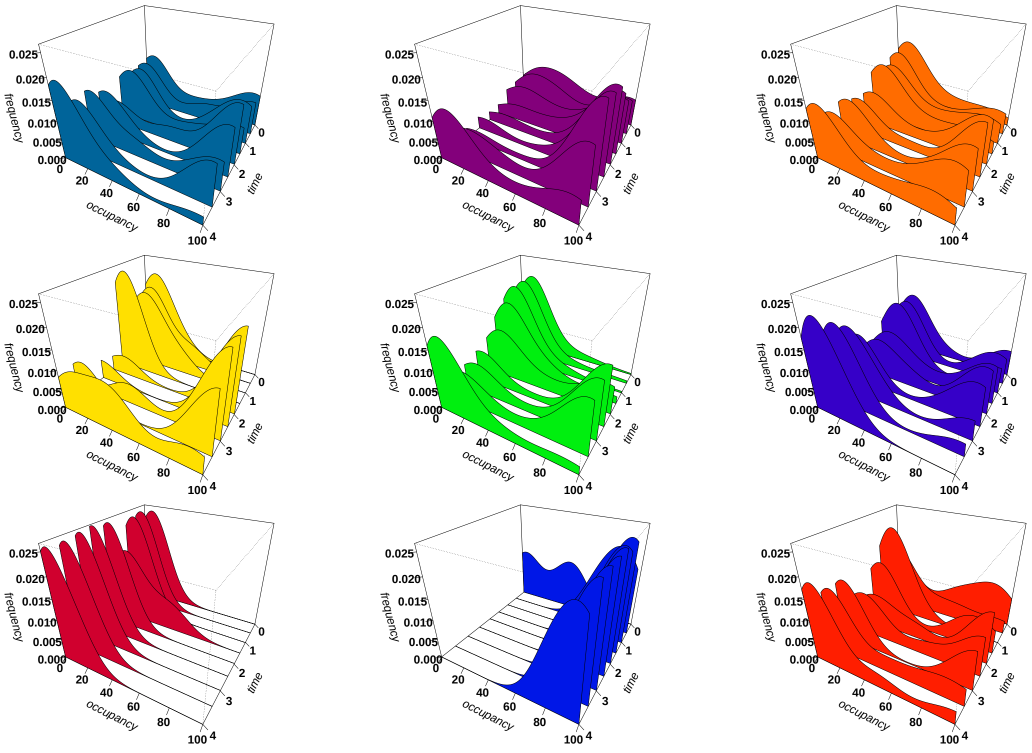


Figure 7. Three dimensional plots for the identified clusters in Figure 6(c).

occupancy observations are displayed at defined time points (at 30 minute intervals) over the period of four hours (14400 seconds). This value of the bandwidth enables density curves to retain differences within each curve and allows for comparisons between clusters. Individual curves also describe the concentration of occupancy and their corresponding values between 0% and 100% through the day. The curves in the three-dimensional sub-plot for the green cluster have a higher magnitude in density levels as compared to the sub-plots for the orange and purple cluster. The sub-plot for the green cluster also displays variations in the concentration of occupancies and the range of occupancy values over four hours. A similar set of variations can be viewed in the sub-plot for the yellow cluster but far more pronounced. The purple cluster has occupancy values that are concentrated at higher values in the middle of the day. On the other hand, the orange cluster has occupancy values that are concentrated far more equally at lower and higher values and with lower change in the distribution of density values through the period of four hours. The sub-plots on the third row display density curves and variations in occupancy levels that correspond to the smaller distinct clusters in the lower part of the network.

5. Discussion

This paper proposes a functional distributional clustering algorithm within an agglomerative hierarchical framework to identify spatially contiguous clusters in a connected grid style graph network. The algorithm seeks to identify homogeneous regions within a heterogeneous network such that individual clusters reflect differences corresponding to vertices. In a given network, these clusters correspond to distinct temporal patterns through the network. Within the framework of this clustering approach, the algorithm is both functional such that a distance measure is defined utilising cumulative distribution functions and distributional to account for temporal patterns present in the available data rather than aggregating over the relevant data. In this proposed non-parametric method, conditional CDFs are determined and stored outside individual iterations of the algorithm in order to improve the computational efficiency for larger datasets. The simulation study demonstrates the superior ability of the functional distributional clustering algorithm in identifying 'true' clusters compared to the functional only, distributional only algorithms and similar performance to a more complex (i.e., greater number of assumptions) model-based clustering method for spatio-temporal data. However, this algorithm is built within an agglomerative hierarchical clustering framework and inherits the associated disadvantages. For example, clusters identified in each iteration of the algorithm are dependent on the structure constructed in the previous steps and cannot be undone. This algorithm generates a hierarchy of clusters and the optimal number of clusters are then determined using well-defined methods (e.g., the gap statistic). In general, the agglomerative hierarchical clustering method is better suited to datasets over small graph networks; computing distances for a large number of cluster pairs can be computationally expensive. Further, the proposed method identifies spatially contiguous clusters that only accommodate temporal patterns. In future work, we seek to extend the functional distributional clustering algorithm to be capable of identifying clusters that also change over time (i.e., dynamic clusters).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

KA and PT would like to acknowledge support by the UK Economic and Social Research Council (ESRC) (grant number ES/S007105/1, ES/L011921/1).

References

[1] A. Adin, D. Lee, T. Goicoa, and M.D. Ugarte, *A two-stage approach to estimate spatial and spatio-temporal disease risks in the presence of local discontinuities and clusters*, Statistical methods in medical research 28 (2019), pp. 2595–2613.

[2] D. Birant and A. Kut, *St-dbscan: An algorithm for clustering spatial–temporal data*, Data & knowledge engineering 60 (2007), pp. 208–221.

[3] A.W. Bowman, *An alternative method of cross-validation for the smoothing of density estimates*, Biometrika 71 (1984), pp. 353–360.

[4] C. Brunsdon, J. Corcoran, and G. Higgs, *Visualising space and time in crime patterns: A comparison of methods*, Computers, environment and urban systems 31 (2007), pp. 52–75.

[5] T. Caliński and J. Harabasz, *A dendrite method for cluster analysis*, Communications in Statistics-theory and Methods 3 (1974), pp. 1–27.

[6] R. Cao, B. Li, Z. Wang, Z.R. Peng, S. Tao, and S. Lou, *Using a distributed air sensor net-*

work to investigate the spatiotemporal patterns of pm2. 5 concentrations, Environmental pollution 264 (2020), p. 114549.

[7] A.X.Y. Carvalho, P.H.M. Albuquerque, G.R. de Almeida Junior, and R.D. Guimaraes, *Spatial hierarchical clustering*, Revista Brasileira de Biometria 27 (2009), pp. 411–442.

[8] M. Chavent, V. Kuentz-Simonet, A. Labenne, and J. Saracco, *Clustgeo: an r package for hierarchical clustering with spatial constraints*, Computational Statistics 33 (2018), pp. 1799–1822.

[9] A. Cheam, M. Marbac, and P. McNicholas, *Model-based clustering for spatiotemporal data on air quality monitoring*, Environmetrics 28 (2017), p. e2437.

[10] J.M. Chiou and P.L. Li, *Functional clustering and identifying substructures of longitudinal data*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69 (2007), pp. 679–699.

[11] J.M. Chiou and P.L. Li, *Correlation-based functional clustering via subspace projection*, Journal of the American Statistical Association 103 (2008), pp. 1684–1692.

[12] D.L. Davies and D.W. Bouldin, *A cluster separation measure*, IEEE transactions on pattern analysis and machine intelligence 2 (1979), pp. 224–227.

[13] J.C. Dunn, *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters*, Journal of Cybernetics 3 (1973), pp. 32–57.

[14] J. Fan and T.H. Yim, *A crossvalidation method for estimating conditional densities*, Biometrika 91 (2004), pp. 819–834.

[15] F. Fouedjio, *A hierarchical clustering method for multivariate geostatistical data*, Spatial Statistics 18 (2016), pp. 333–351.

[16] R. Giraldo, P. Delicado, and J. Mateu, *Hierarchical clustering of spatially correlated functional data*, Statistica Neerlandica 66 (2012), pp. 403–421.

[17] P. Hall, J. Racine, and Q. Li, *Cross-validation and the estimation of conditional probability densities*, Journal of the American Statistical Association 99 (2004), pp. 1015–1026.

[18] J.D. Hart and P. Vieu, *Data-driven bandwidth choice for density estimation based on dependent data*, The Annals of Statistics (1990), pp. 873–890.

[19] A. Harvey and V. Oryshchenko, *Kernel density estimation for time series data*, International journal of forecasting 28 (2012), pp. 3–14.

[20] L. Hubert and P. Arabie, *Comparing partitions*, Journal of classification 2 (1985), pp. 193–218.

[21] R. Ignaccolo, S. Ghigo, and E. Giovenali, *Analysis of air quality monitoring networks by functional clustering*, Environmetrics 19 (2008), pp. 672–686.

[22] G.M. James and C.A. Sugar, *Clustering for sparsely sampled functional data*, Journal of the American Statistical Association 98 (2003), pp. 397–408.

[23] I.G.N.M. Jaya and H. Folmer, *Identifying spatiotemporal clusters by means of agglomerative hierarchical clustering and bayesian regression analysis with spatiotemporally varying coefficients: methodology and application to dengue disease in bandung, indonesia*, Geographical Analysis (2020).

[24] Y. Jung, H. Park, D.Z. Du, and B.L. Drake, *A decision criterion for the optimal number of clusters in hierarchical clustering*, Journal of Global Optimization 25 (2003), pp. 91–111.

[25] B.G. Leroux, X. Lei, and N. Breslow, *Estimation of disease rates in small areas: a new mixed model for spatial dependence*, in *Statistical models in epidemiology, the environment, and clinical trials*, Springer, 2000, pp. 179–191.

[26] H. Li, J. Liu, Z. Yang, R.W. Liu, K. Wu, and Y. Wan, *Adaptively constrained dynamic time warping for time series classification and clustering*, Information Sciences 534 (2020), pp. 97–116.

[27] Q. Li and J.S. Racine, *Nonparametric estimation of conditional cdf and quantile functions with mixed categorical and continuous data*, Journal of Business & Economic Statistics 26 (2008), pp. 423–434.

[28] M. Liang, R.W. Liu, S. Li, Z. Xiao, X. Liu, and F. Lu, *An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation*, Ocean Engineering 225 (2021), p. 108803.

[29] J. MacQueen, *et al.*, *Some methods for classification and analysis of multivariate observations*, in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA., 1967, pp. 281–297.

[30] G.W. Milligan and M.C. Cooper, *A study of the comparability of external criteria for hierarchical cluster analysis*, Multivariate Behavioral Research 21 (1986), pp. 441–458.

[31] T. Pei, C. Zhou, A.X. Zhu, B. Li, and C. Qin, *Windowed nearest neighbour method for mining spatio-temporal clusters in the presence of noise*, International Journal of Geographical Information Science 24 (2010), pp. 925–948.

[32] S. Petrovic, *A comparison between the silhouette index and the davies-bouldin index in labelling IDS clusters*, in *Proceedings of the 11th Nordic Workshop of Secure IT Systems*. 2006, pp. 53–64.

[33] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. Available at https://www.R-project.org.

[34] W.M. Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association 66 (1971), pp. 846–850.

[35] M.Z. Rodriguez, C.H. Comin, D. Casanova, O.M. Bruno, D.R. Amancio, L.d.F. Costa, and F.A. Rodrigues, *Clustering algorithms: A comparative approach*, PloS one 14 (2019), p. e0210236.

[36] P.J. Rousseeuw, *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*, Journal of computational and applied mathematics 20 (1987), pp. 53–65.

[37] M. Rudemo, *Empirical choice of histograms and kernel density estimators*, Scandinavian Journal of Statistics (1982), pp. 65–78.

[38] R. Tibshirani, G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2001), pp. 411–423.

[39] J.H. Ward Jr, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association 58 (1963), pp. 236–244.

[40] Q. Wei, J. She, S. Zhang, and J. Ma, *Using individual gps trajectories to explore foodscape exposure: A case study in beijing metropolitan area*, International journal of environmental research and public health 15 (2018), p. 405.

[41] D. Xu and Y. Tian, *A comprehensive survey of clustering algorithms*, Annals of Data Science 2 (2015), pp. 165–193.