Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP)

D. Guo [a]

[a] Department of Geography , University of South Carolina , 709
Bull Street, Room 127, Columbia, SC 29208, USA
Published online: 07 May 2008.

PLEASE SCROLL DOWN FOR ARTICLE

Research Article

Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP)

D. GUO*

Department of Geography, University of South Carolina, 709 Bull Street, Room 127, Columbia, SC 29208, USA

Regionalization is to divide a large set of spatial objects into a number of spatially contiguous regions while optimizing an objective function, which is normally a homogeneity (or heterogeneity) measure of the derived regions. This research proposes and evaluates a family of six hierarchical regionalization methods. The six methods are based on three agglomerative clustering approaches, including the single linkage, average linkage (ALK), and the complete linkage (CLK), each of which is constrained with spatial contiguity in two different ways (i.e. the first-order constraining and the full-order constraining). It is discovered that both the Full-Order-CLK and the Full-Order-ALK methods significantly outperform existing methods across four quality evaluations: the total heterogeneity, region size balance, internal variation, and the preservation of data distribution. Moreover, the proposed algorithms are efficient and can find the solution in $O(n^2\log n)$ time. With such data scalability, for the first time it is possible to effectively regionalize large data sets that have 10 000 or more spatial objects. A detailed comparison and evaluation of the six methods are carried out with the 2004 US presidential election data.

Keywords: Regionalization; Spatial data mining; Zoning; Segmentation; Hierarchical clustering; Constrained clustering

1. Introduction

Region building has been one of the most common problems encountered in spatial analysis (Haggett *et al.* 1977). Given a set of spatial objects (e.g. US counties) with univariate or multivariate information (e.g. incidence rates of different types of cancer), a regionalization method attempts to aggregate the spatial objects into a number of spatially contiguous regions while optimizing an objective function, which is normally a measure of the attribute similarity in each region. Regionalization has been an important and challenging problem for a large spectrum of research and application domains, for example, climatic zoning (Fovell and Fovell 1993), ecoregion analysis (Handcock and Csillag 2004), hazards and disasters management (Cutter 2001), map generalization (Tobler 1969), location optimization (Goodchild 1979), census reengineering (Openshaw and Rao 1995), and health-related analysis (Morris and Munasinghe 1993; Haining *et al.* 1994; Osnes 1999). Regionalization is essentially a special form of classification where spatial units are grouped together, based on a set of defined criteria and a set of contiguity or adjacency constraints (Haining 2003).

---

*Corresponding author. Email: guod@sc.edu

Recently, a new regionalization method (SKATER—Spatial Kluster Analysis by Tree Edge Removal) has been proposed, based on minimum spanning trees (Assunção *et al.* 2006). SKATER first constructs a spatially contiguous graph by removing edges that do not connect geographic neighbours and then builds a minimum spanning tree from the graph (note: any tree built from this graph is spatially contiguous). The tree is then recursively partitioned to generate a given number of regions. However, the SKATER method has three limitations. First, it uses contiguity constraints in a static way, where the contiguity matrix is not dynamically updated during the clustering process (i.e. it is ignored that spatial objects that are not spatial neighbours at the beginning may become neighbours if they later belong to two clusters that are next to each other in space). Second, a minimum spanning tree has a well-known 'chaining' problem and cannot guarantee that observations within a cluster are similar to each other (Jain and Dubes 1988; Hastie *et al.* 2001, p. 477). Third, the implementation of the SKATER method is inefficient and cannot process large datasets.

Inspired by SKATER and yet to address its limitations, this research proposes, implements, and compares six methods for regionalization, which are based on three agglomerative clustering methods (i.e. the *single linkage* clustering (SLK), *average linkage* clustering (ALK), and the *complete linkage* clustering (CLK)) and two different spatial constraining strategies (i.e. the First-Order constraining and the Full-Order constraining). Therefore, the six methods are *First-Order-SLK*, *First-Order-ALK*, *First-Order-CLK*, *Full-Order-SLK*, *Full-Order-ALK*, and *Full-Order-CLK*. This family of methods is named REDCAP, an acronym for Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning. ("Redcap" is a British nickname for a military policeman or in American English historically for a railway porter; it is also the name of a type of goldfinch bird.) The *First-Order-SLK* method is conceptually the same as the SKATER method, but the former is much more efficient than the latter due to implementation differences. It is discovered that *Full-Order-CLK* and *Full-Order-ALK* significantly outperform SKATER across four quality measures: total heterogeneity, size balance, internal variation, and the preservation of data distribution. Moreover, the proposed methods are efficient (with a time complexity of $O(n^2 \log n)$) and able to process large data sets.

The paper is organized as follows. Related work is reviewed in the next section. Section 3 includes details on the algorithm and its complexity analysis for each of the six contiguity constrained clustering methods. Section 4 presents the iterative partition algorithm that cuts a spatially contiguous tree into a desirable number of regions. Quality and efficiency evaluations of the six methods are presented in section 5. Section 6 includes a summary and discussions.

2. Related work

Regionalization is a combinatorial problem. Given a large set of spatial objects, it is not feasible to enumerate all possible partitions and find the best regionalization according to some objective function. Therefore, existing regionalization methods either take a heuristic-based approach to search for near-optimal solutions or attempt to find an 'optimal' solution within a confined or reduced search space. Existing methods can be classified into four groups: (1) non-spatial clustering followed by spatial processing; (2) non-spatial clustering with a spatially weighted dissimilarity measure; (3) trial-and-error search and optimization; and (4) spatially constrained clustering and partitioning.

The first group of methods uses existing non-spatial clustering methods to derive clusters based on attribute similarity only. The clusters are then divided or merged to form regions in the geographic space (Fovell and Fovell 1993; Haining *et al.* 1994). Although this type of approach is useful for examining the spatial distribution and dependence of multivariate patterns, the final number of regions is unpredictable and difficult to control. Very often, clusters in the attribute space are segmented into pieces in the geographic space.

The second type of regionalization methods uses non-spatial clustering methods in a different way. It modifies the similarity (or dissimilarity) measure to incorporate spatial information explicitly, for example, using a distance-weighted attribute similarity (Oliver and Webster 1989) or treating geographic coordinates as additional variables (Spence 1968). Haggett *et al.* (1977, p. 461) demonstrated a series of examples of a weighted sum of spatial and attribute similarity. In addition to the geographical distance, some research has also tried to incorporate more criteria into the objective function, for example, homogeneity, region size constraints, and compactness (Wise *et al.* 1997). However, the drawback of such a combined objection function is that those criteria are not comparable with each other and that the user has to specify weights without understanding the implications (Haining 2003).

The third group of methods takes a trial-and-error optimization approach, represented by the AZP method (Openshaw 1977; Openshaw and Rao 1995). AZP starts with an initial random regionalization and iteratively refines the solution by reassigning objects to neighbouring regions for better solutions. To avoid local minima, simulated annealing and tabu search heuristic have been integrated (Openshaw and Rao 1995). However, a major problem with the AZP method is its high computational cost and inability to process large datasets.

The fourth group of methods, represented by SKATER (Assunção *et al.* 2006), is different from the second group in that it considers spatial constraints (rather than spatial similarities) with a non-spatial clustering method. The SKATER method follows the single linkage clustering procedure and constrains it with spatial contiguity, to reach a reasonable regionalization solution. In other words, this type of method only explores a reduced search space, which is confined by the clustering method employed. As explained earlier, the performance of SKATER is limited by the clustering method (i.e. minimum spanning tree) on which it is based and the constraining strategy that it adopts.

Clustering methods can be classified into two main groups: partitioning clustering (e.g. K-Means) and hierarchical clustering. Detailed reviews of various clustering methods can be found in Jain and Dubes (1988), Jain *et al.* (1999), and Guo *et al.* (2003). Note that, although there are numerous spatial clustering methods (see a detailed review in Han *et al.* 2001), they are not applicable for regionalization. To perform hierarchical clustering, a pairwise dissimilarity matrix is often constructed. If we view such a dissimilarity matrix as a weighted graph, graph-partitioning methods (Karypis and Kumar 1998; Saab 2004; Felner 2005) may be used to partition the data into a number of parts while optimizing an objective function, e.g. minimizing the total weights of edges to be cut (Karypis and Kumar 1998). However, existing graph-partitioning methods cannot consider contiguity constraints and thus cannot produced spatially contiguous regions.

3. Contiguity-constrained hierarchical clustering

This research proposes a family of six new regionalization methods, which extend three commonly used hierarchical clustering methods (i.e. single linkage clustering, average

linkage clustering, and complete linkage clustering) with two different contiguity constraining strategies. Each method consists of two steps: (1) clustering data with contiguity constraints to produce a spatially contiguous tree (hierarchy) and (2) partitioning the tree to generate regions while optimizing an objective function. This section presents the six contiguity constrained clustering methods, each of which is a combination of a clustering method (i.e. the single-, average-, or complete-linkage clustering) and a constraining strategy (i.e. the first-order constraining or the full-order constraining). The tree-partitioning algorithm is presented in section 4.

3.1 *Three agglomerative clustering methods*

The three agglomerative clustering methods differ in their distance definition (i.e. dissimilarity) between clusters (Jain and Dubes 1988; Hastie *et al.* 2001). The single linkage clustering (SLK) defines the distance between two clusters as the dissimilarity between the closest pair of data points from each cluster:

$$d_{\text{SLK}}(L, M) = \min_{u \in L, \, v \in M}(d_{uv}) \tag{1}$$

where L and M are two clusters, $u \in L$ and $v \in M$ are two data points, and d_{uv} is the dissimilarity between u and v. The single linkage clustering has a tendency to group points linked by a series of close intermediate data points. This phenomenon is referred to as the 'chaining effect', which can result in clusters that contain dissimilar data points (Hastie *et al.* 2001, p. 477).

The average linkage clustering (ALK) defines the distance between two clusters as the average dissimilarity between all cross-cluster pairs of data points:

$$d_{\text{ALK}}(L, M) = \frac{1}{|L||M|} \sum_{u \in L} \sum_{v \in M} d_{uv} \tag{2}$$

where $|L|$ and $|M|$ are the number of data points in clusters L and M, respectively. The average linkage clustering depends on the numerical scale on which the dissimilarities are measured—a strictly increasing monotone transformation may change the result (Hastie *et al.* 2001).

The complete linkage clustering (CLK) uses the dissimilarity between the furthest pair of data points as the distance between two clusters:

$$d_{\text{CLK}}(L, M) = \max_{u \in L, \, v \in M}(d_{uv}). \tag{3}$$

For the complete linkage clustering, two clusters are considered similar only if all the observations in the two clusters are similar to each other.

Except for different definitions of cluster distance, the three methods use the same procedure to form hierarchical clusters. At the beginning, each individual data object is a cluster by itself. Then, the most similar (determined by the distance definition) pair of clusters are selected and merged into one cluster. The merging process repeats until all data points are in the same cluster.

To facilitate subsequent presentation of the proposed contiguity-constrained clustering methods, the above clustering process is reformulated from a graph-based perspective. A data set of n objects can be modelled as a graph $G(V, E)$, where V is a set of n vertices (each representing a data object), and E is a set of edges (each connecting a unique pair of data objects). The length (or weight) for an edge is the attribute dissimilarity between the two data objects that it connects. (Note: the edge length is not the geographic distance.) G is a complete graph if it consists of edges

for all unique pairs of objects (thus $|E|=n(n-1)/2$). A hierarchical clustering of this data set is to build a tree T from G. T contains exactly $n-1$ edges and connects all data objects. Each edge in T is added to merge two clusters. The order in which those edges are added to T represents the sequence in which clusters are merged. Therefore, T and the order of its edges together represent a cluster hierarchy. T by itself does not represent a unique cluster hierarchy—it can represent a very different hierarchy if its edges are selected in a different order. The three clustering methods introduced above differ in their criteria for selecting and adding edges to T (figure 8).

3.2 Two constraining strategies: first-order vs. full-order

Following the above graph-based notation, an edge is *first order* if it connects two spatial neighbours, according to a predefined spatial contiguity matrix. A graph (or a tree) is *spatially contiguous* if it is a connected graph and consists only of first-order edges. Two clusters (or trees) are *spatially contiguous* with each other if they can be connected by a first-order edge. Contiguity-constrained agglomerative clustering requires essentially that two clusters cannot be merged if they are not spatially contiguous. This research designs two different strategies to incorporate contiguity constraints in hierarchical clustering: the *first-order constraining* and the *full-order constraining*. The first-order constraining strategy uses only first-order edges during the clustering process. Other edges are removed at the beginning and not considered throughout the clustering process. The distance between two clusters is defined with
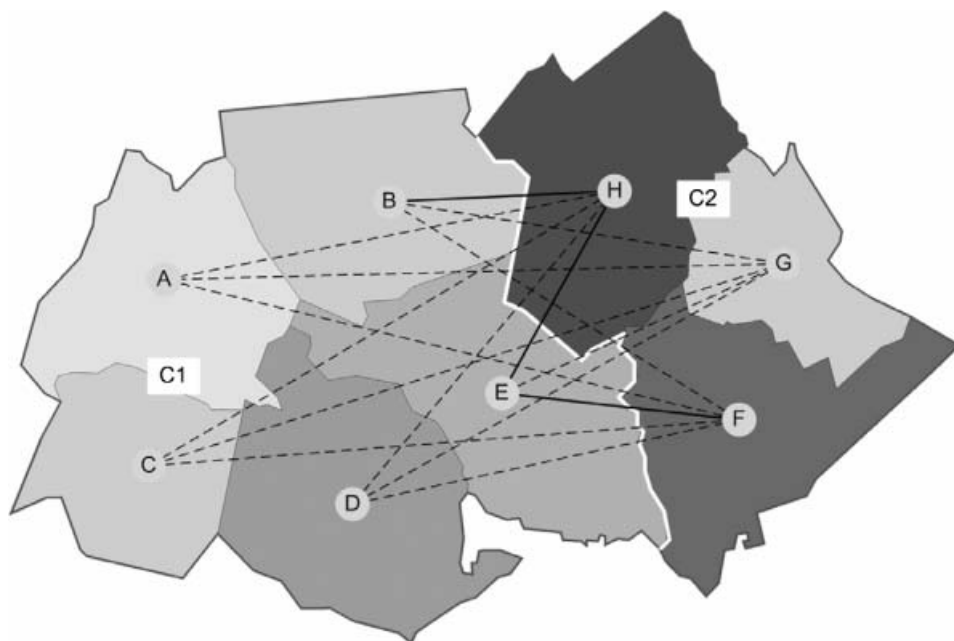

Figure 1. Demonstration of the difference between the *first-order* constraining strategy and the *full-order* constraining strategy. The eight counties are grouped to two clusters: C1={A, B, C, D, E} and C2={F, G, H} according to their attribute values (signified with greyscale tones). Among the 15 edges that connect C1 and C2, only BH, EH, and EF are first-order edges. The first-order constraining uses only the three first-order edges to determine the distance (or dissimilarity) between C1 and C2, while the full-order strategy uses all 15 edges (see text for details).

first-order edges only (figure 1). A full-order constraining strategy considers all edges in the clustering process, and the distance between two clusters is defined over all edges between them. Therefore, the full-order constraining strategy is dynamic in nature, since it updates the contiguity matrix after each merge to track all edges that connect two different clusters.

Figure 1 shows a simple demonstration of the difference between the first-order strategy and the full-order strategy. Suppose, at a certain level, eight counties (which are part of a larger data set) are grouped into two clusters (C1 and C2). Among all the 15 edges that connect C1 and C2, only BH, EH, and EF are first-order. The first-order strategy uses only BH, EH, and EF to determine the distance (or dissimilarity) between C1 and C2, while the full-order strategy uses all 15 edges. For example, the First-Order-SLK uses the length of edge EF (which is the shortest first-order edge *in terms of the attribute dissimilarity between its two nodes*) as the distance between C1 and C2, while the Full-Order-SLK uses the length of edge BG, which is the shortest among all 15 edges. Similarly, the First-Order-CLK uses BH (which is the longest first-order edge), the Full-Order-CLK uses AH (which is the longest among all edges), the First-Order-ALK uses the average of BH, EH, and EF (i.e. first-order edges), and the Full-Order-ALK uses the average of all 15 edges as the distance between C1 and C2.

The tree built by a first-order constrained clustering method is naturally a spatially contiguous tree because only first-order edges are considered and used in the clustering process. However, to build a spatially contiguous tree with the full-order constraining strategy, an extra step is needed after each merge. When an edge e (which is not necessarily first-order) is selected to merge two clusters, the shortest first-order edge $e*$ that connects the two clusters is added to the tree (instead of adding e to the tree). This does not change the tree structure or the cluster hierarchy, since either e or $e*$ makes the same merge. Thus, both first-order methods and full-order methods build a spatially contiguous tree.

3.3 *Six contiguity-constrained agglomerative clustering methods*

By combining the two constraining strategies and the three clustering methods, six different contiguity-constrained clustering methods are proposed. This section introduces the algorithmic detail and computational complexity for each method. Following the notation used in section 3.1, a data set of n spatial objects forms a complete graph G, which consists of edges E that connect all unique pairs of objects. A contiguity matrix C is given to specify the spatial contiguity relationship between each pair of objects.

3.3.1 First-order constrained single linkage clustering (First-Order-SLK). The First-Order-SLK first reduces the complete graph G to a spatially contiguous graph $G*$ by removing edges that are not first-order. A minimum spanning tree T is then constructed from $G*$ following three steps: (1) all edges in $G*$ are sorted in an ascending order, and each edge is evaluated following that order; (2) if an edge connects two different clusters, it is added to T, and the two clusters are merged; (3) repeat step 2 until all data objects are connected (i.e. in the same cluster). The algorithm is shown in figure 2. Its computational complexity is $O(n\log n)$.

3.3.2 First-order constrained average linkage clustering (First-Order-ALK). The First-Order-ALK method also starts with the spatially contiguous graph $G*$. However, after each merge, the distance between the new cluster and every other cluster is recalculated. Therefore, edges that connect the new cluster and every other cluster are updated with new length values. Edges in $G*$ are then re-sorted and re-

The First-Order Constrained Single-Linkage Clustering (First-Order-SLK)	Overall: $O(nlogn)$
Input: $E^* = \{e_i\}$: all first-order edges Output: $T \subset E^*$: a spatially contiguous tree Step 1: Sort all edges in E^* in an ascending order according to their lengths Set $T = \varnothing$ Stop until all objects are covered by T Step 2: For each edge e in the sorted list (shortest first) If e connects two separate clusters, add e to T Stop the loop if all objects are covered by T	$\|E^*\| = \alpha n$ for some constant α $\|T\| = n - 1$ Step 1: $O(nlogn)$ Step 2: $O(n)$

Figure 2. Tree construction algorithm for the first-order constrained single linkage clustering (First-Order-SLK). This algorithm builds a spatially contiguous minimum spanning tree.

evaluated from the beginning. The procedure stops when all objects are in one cluster. The algorithm is shown in figure 3. The complexity is $O(n^2 \log n)$ due to the sorting after each merge.

3.3.3 First-order constrained complete linkage clustering (First-Order-CLK). The First-Order-CLK also starts with G^*. The distance between two clusters is defined by the longest first-order edge that connects them. Edges are sorted only once. The algorithm is shown in figure 4, and its complexity is $O(n^2)$.

3.3.4 Full-order constrained single linkage clustering (Full-Order-SLK). Different from first-order constrained methods, a full-order constrained method starts with the complete graph G, the spatial contiguous graph G^*, and the contiguity matrix C. Edges in G and G^* are sorted in an ascending order, separately. Each edge e in G is evaluated following the ascending order. If e connects two spatially contiguous clusters, the shortest first-order edge e^* in G^* that connects the two clusters is added to T. The contiguity matrix C is then updated to reflect the effect of this new merge, i.e. some clusters become spatially contiguous with the new cluster. After each merge, the edges

The First-Order Constrained Average-Linkage Clustering (First-Order-ALK)	Overall: $O(n^2logn)$
Input: $E^* = \{e_i\}$: all first-order edges Output: $T \subset E^*$: a spatially contiguous tree Step 1: Sort all edges in E^* in an ascending order according to edge lengths Set $T = \varnothing$ Step 2: Construct an $n \times n$ matrix *avgDist* to store distances between clusters For each u ($u = 1..n$) and v ($v = 1..n$) $avgDist(u, v) = 0$ Step 3: For each edge e in the sorted list (shortest first) If e connects two separate clusters l, m and $e.length >= avgDist(l, m)$ (1) Find the *shortest* edge e^* in E^* that connect cluster l and m (2) Add e^* to T and merge cluster m to l (i.e., l is now the new cluster) (3) For each cluster c that is not l $avgDist(c, l) = $ average length of edges in E^* that connect c and l update the length of edge (c, l) in E^* with $avgDist(c, l)$ (4) Sort all edges in E^* and set $e = $ the shortest one in the list End the loop if all points are covered by T	$\|E^*\| = \alpha n$ for some constant α $\|T\| = n - 1$ Step 1: $O(nlogn)$ Step 2: $O(n^2)$ Step 3: $O(n^2logn)$ Steps (1) – (4) run exactly $n - 1$ times. Step (4) takes $O(nlog\ n)$ time

Figure 3. Tree construction algorithm for the first-order constrained average linkage clustering (First-Order-ALK).

The First-Order Constrained Complete-Linkage Clustering (First-Order-CLK)	Overall: $O(n^2)$
Input: $E^* = \{e_i\}$: all first-order edges Output: $T \subset E^*$: a spatially contiguous tree Step 1: Sort all edges in E^* in an ascending order according to edge lengths 　　　Set $T = \varnothing$ Step 2: Construct an $n \times n$ matrix *maxDist* to store the distance between clusters 　　　For each u ($u = 1..n$) and v ($v = 1..n$) 　　　　$maxDist(u, v) = 0$ Step 3: For each edge e in the sorted list (shortest first) 　　　　If e connects two separate clusters l, m and $e.length >= maxDist(l, m)$ 　　　　　Find the shortest edge e^* in E^* that connects cluster l and m 　　　　　Add e^* to T and merge cluster m to cluster l (i.e., l is the new cluster) 　　　　　for each cluster c that is not l 　　　　　　$maxDist(c, l) = \max (maxDist(c, l), maxDist(c, m))$ 　　　　End the loop if all points are covered by T	$\|E^*\| = \alpha n$ for some constant α $\|T\| = n - 1$ Step 1: $O(n\log n)$ Step 2: $O(n^2)$ Step 3: $O(n^2)$

Figure 4.   Tree construction algorithm with the first-order constrained complete linkage clustering (First-Order-CLK).

(sorted already) in G are re-evaluated from the beginning because some clusters that are previously non-contiguous to each other now become eligible for the next merge. The Full-Order-SLK algorithm is shown in figure 5. Its theoretical complexity is $O(n^3)$. However, since edges are sorted, and the final tree consists mostly of short edges, long edges in the list are seldom visited, and thus the actual computational cost is much less than what the theoretical complexity indicates.

3.3.5   Full-order constrained average linkage clustering (Full-Order-ALK). To speed up the algorithm and reduce memory usage, the Full-Order-ALK method starts with G^*, adds new edges, and removes some old edges after each merge. A binary search and sorting tree (B_T) is used (Wirth 1976; Cormen *et al.* 2001) to efficiently insert and remove edges while maintaining a sorted list of edges. Both the contiguity

The Full-Order Constrained Single-Linkage Clustering (Full-Order-SLK)	Overall: $O(n^3)$
Input: E: edges that connect all possible pairs of objects; 　　　$E^* \subset E$: all first-order edges; 　　　C: contiguity matrix, $C(u, v)=1$ if clusters u and v are spatially contiguous. Output: $T \subset E^*$: a spatially contiguous tree. Step 1: Sort E and E^* in an ascending order, respectively 　　　Set $T = \varnothing$, $i=0$; Step 2: $e =$ the ith edge in E 　　　If e connects two separate clusters l and m, and $C(l, m) = 1$ 　　　　(1) Find the shortest e^* in E^* that connect cluster l and m. 　　　　(2) Add e^* to T 　　　　(3) For each existing cluster c, $c \neq l$, $c \neq m$ 　　　　　Set $C(c, l) = 1$ if $C(c, l) =1$ or $C(c, m) =1$. 　　　　(4) Merge cluster m to cluster l (i.e. l is now the new cluster) 　　　　(5) Reset $i = 0$; 　　　Else 　　　　$i = i + 1$; Step 3: Repeat step 2 until all objects are in T.	$\|E\| = n(n - 1) / 2$ $\|E^*\| = \alpha n$ for some constant α C: $n \times n$ $\|T\| = n - 1$ Step 1: $O(n^2\log n)$ Step 2: (1)–(5) is repeated exactly $n - 1$ times. At worst, step 2 is repeated $O(n^3)$ times, among which, only $O(n)$ times for steps (1)–(5))

Figure 5.   Tree construction algorithm for the full-order constrained single linkage clustering (Full-Order-SLK).

matrix and the B_T tree are updated after each merge. Edges in the B_T tree are in an ascending order and evaluated from the beginning after each merge. The algorithm is shown in figure 6. Its complexity is $O(n^2 \log n)$.

3.3.6 Full-order constrained complete linkage clustering (Full-Order-CLK). The Full-Order-CLK algorithm is similar to the Full-Order-SLK except that it needs a matrix to keep the distances among existing clusters. The Full-Order-CLK algorithm can achieve a computational complexity of $O(n^2 \log n)$ without using a binary search and sorting tree because there is no need to re-evaluate previously visited edges after each merge. The correctness of this no-look-back rule can be proved as follows. Suppose that the latest merge is between clusters l and m, and $d(l, m)$ is the distance between them. For any other cluster c, we know that $d(c, l) > d(l, m)$ and $d(c, m) > d(l, m)$. Otherwise, the merge should be between c and l or between c and m instead of l and m. After l and m are merged, the distance between cluster c and the new cluster is $\max[d(c, l), d(c, m)]$, which is greater than $d(l, m)$ and therefore is greater than the length of any edge that has been processed (because edges are processed in an ascending order). Therefore, edges that have been evaluated do not qualify to merge existing clusters, even if the contiguity matrix is changed after each merge. The algorithm is presented in figure 7.

Figure 8 shows an illustrative example of different trees built with different methods. Due to limited space, it only shows trees produced by the First-Order-SLK, the Full-Order-SLK, and the Full-Order-CLK. For this simple data set, First-Order-SLK and

The Full-Order Constrained Average-Linkage Clustering (Full-Order-ALK)	Overall: $O(n^2 \log n)$
Input: E^*: all first-order edges; 　　　　C: contiguity matrix, $C(u, v) = 1$ if clusters u and v are spatially contiguous. **Output:** $T \subset E$: a spatially contiguous tree	$\|E^*\| = \alpha n$ for some constant α C: $n \times n$
Step 1: Sort E^* in an ascending order; 　　　　Organize edges in E^* with a binary search and sorting tree (B_T) 　　　　(Note: E^* remains intact while edges are copied and organized in B_T) 　　　　Set $T = \varnothing$, $i = 0$; 　　　　Set $avgDist(u, v) = 0$, for each u and v. 　　　　Set $numEdges(u, v) = 1$, for each u and v. 　　　　e = the left-most (i.e., shortest) edge in the sorted list of B_T	Step 1: $O(n^2)$
Step 2: If e connects two separate clusters l and m, $C(l, m)$ is 1, and $e.length >= avgDist(l, m)$ 　　　　(1) Find the shortest e^* in E^* that connect cluster l and m. 　　　　(2) Add e^* to T and merge cluster m to cluster l 　　　　(3) For each existing cluster c, $c \neq l$, $c \neq m$ 　　　　　　Set $avgDist(c, l) = [avgDist(c, l) \cdot numEdges(c, l) + avgDist(c, m) \cdot numEdges(c, m)] / [numEdges(c, l) + numEdges(c, m)]$; 　　　　　　Set $numEdges(c, l) = numEdges(c, l) + numEdges(c, m)$; 　　　　　　If $C(c, l)$ is 1 or $C(c, m)$ is 1 　　　　　　　　Set $C(c, l) = 1$; 　　　　　　　　Remove edges (c, l) and (c, m) from B_T 　　　　　　　　Insert a new edge (c, l) of length of $avgDist(c, l)$ to B_T 　　　　(4) e = the left-most (i.e., shortest) edge in the sorted list of B_T; 　　　　Else 　　　　　　e = the next edge after e in the sorted list of B_T **Step 3:** Repeat step 2 until all objects are in T	Step 2: $O(n^2 \log n)$ (1)–(4) are repeated exactly $n - 1$ times. Removal or insertion of an edge to B_T takes $O(\log n)$ time. Therefore Step (3) is of $O(n \log n)$ complexity, which is repeated $n - 1$ times.

Figure 6. Tree construction algorithm for the full-order constrained average linkage clustering (Full-Order-ALK). A binary search and sorting tree is used to speed up the algorithm.

The Full-Order Constrained Complete-Linkage Clustering (Full-Order-CLK)	Overall: $O(n^2 \log n)$

Input: E: edges that connect all possible pairs of objects;
               E^*: all first-order edges;
               C: contiguity matrix, $C(u, v)=1$ if clusters u and v are spatially contiguous
Output: $T \subset E$: a spatially contiguous tree
	$\|E\| = n(n-1)/2$
	$\|E^*\| = \alpha n$ for some constant α
	$C: n \times n$

Step 1: Sort E and E^* separately in an ascending order
               Set $T = \varnothing$, $i=0$;
               For each u ($u = 1..n$) and v ($v = 1..n$)
                   $maxDist(u, v) = 0$
	Step 1: $O(n^2 \log n)$

Step 2: $e =$ the ith edge in E
               If e connects two separate clusters u and v, $C(u, v)$ is 1, and $e.length >= maxDist(l, m)$
                   (1) Find the shortest e^* in E^* that connect cluster u and v.
                   (2) Add e^* to T and merge cluster m to cluster l
                   (3) For each existing cluster c, $c \neq l$, $c \neq m$
                       Set $maxDist(c, l) = \max(maxDist(c, l), maxDist(c, m))$
                       Set $C(c, l) = true$ if $C(c, m)$ is $true$ or $C(c, l)$ is $true$.
               $i = i + 1$;
	Step 2: $O(n^2)$
	
	Steps (1)–(3) take $O(n)$ time and are repeated exactly $n - 1$ times.

Step 3: Repeat step 2 until all objects are in T.	

Figure 7.   Tree construction algorithm for the full-order constrained complete linkage clustering (Full-Order-CLK).

Full-Order-SLK produce similar trees (whose edges, however, are added in different orders). The Full-Order-CLK produces a different tree. Different trees define different search spaces for the partitioning algorithm (see section 4). For this simple dataset, the optimal solution of two regions are {I, B, C, E} and {G, H, A, D, F}, according to the attribute values labelled in figure 8. This optimal solution can be derived with the Full-Order-CLK tree by removing edge CF. It is impossible to derive such two regions with either the First-Order-SLK tree or the Full-Order-SLK tree.

4.   Regionalization via partitioning a spatially contiguous tree

Deriving a spatially contiguous tree with one of the above contiguity-constrained clustering methods is the first of two steps for regionalization. The second step is to partition the spatially contiguous tree to obtain a number of subtrees, each of which corresponds to a spatially contiguous region. In other words, the first step builds a spatially contiguous tree from the bottom up, and the second step partitions the tree from the top down to obtain regions. The tree is built based on the distances between clusters while the partition is carried out to optimize an objective function, which is to minimize the total heterogeneity value of all regions. The heterogeneity of a region is a measure of attribute similarity among the spatial objects inside the region. The definition of heterogeneity may vary for different application problems.

4.1   *Heterogeneity measure and homogeneity gain*

To allow comparison with existing methods, this research uses the sum of squared deviations (SSD) (Assunção *et al.* 2006) as the heterogeneity measure, which is defined as:

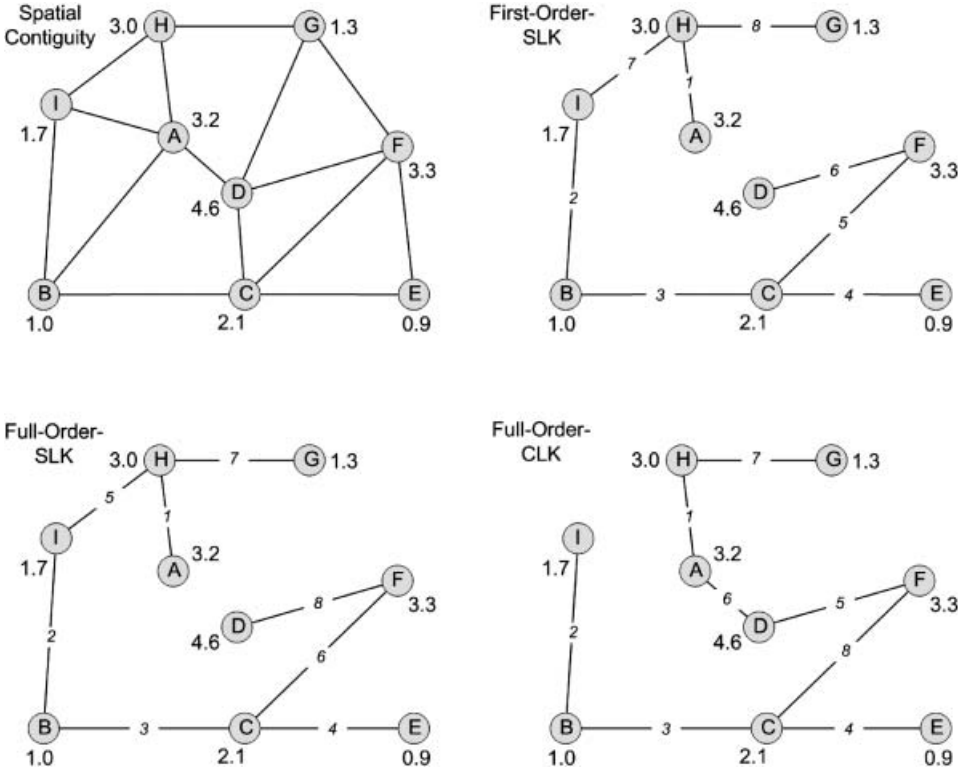$$H(R) = \sum_{j=1}^{d} \sum_{i=1}^{n_r} \left(x_{ij} - \overline{x}_j \right)^2 \tag{4}$$

Figure 8. Demonstration of the differences among trees derived with different constrained clustering methods. Top left: Graph showing the spatial contiguity relations among objects (nodes). Each edge connects two spatial neighbours. The numerical value associated with each node represents its attribute value. Top right: Tree derived with the First-Order-SLK method. The number on each edge indicates the order that the edge is added to the tree. Bottom left: Tree derived with the Full-Order-SLK method. Note that this tree is structurally the same as the First-Order-SLK tree (top-right) but its edges are added in a different order. Bottom-right: Tree constructed by the Full-Order-CLK method. The optimal solution for two regions is {I, B, C, E} and {F, D, A, H, G}, which can be derived with the Full-Order-CLK tree (by cutting edge CF) but cannot with the First-Order- or Full-Order-SLK tree.

where R is a region, $H(R)$ denotes its heterogeneity, d is the number of attributes, n_r is the number of objects in R, x_{ij} is the value for the jth attribute of the ith object, and $\bar{x}_j$ is the mean value of the jth attribute for all objects in R.

The overall heterogeneity (H_k) for a regionalization result with k regions is the total of the k heterogeneity values:

$$H_k = \sum_{j=1}^{k} H(R_j) \tag{5}$$

The partitioning algorithm (see section 4.2) iteratively partitions a spatially contiguous tree into k regions by cutting a subtree (i.e. a region) into two at each step. The best subtree to cut at each step is the one with the largest homogeneity gain h_g^*, which is calculated for each tree (or subtree) with the following equation:

$$h_g^*(R) = \max(H(R) - H(R_a) - H(R_b)) \tag{6}$$

for all possible cuts of tree R into two subtrees R_a and R_b. R_a and R_b are the two subtrees from a possible cut of R, and h_g^* is the homogeneity gain (i.e. heterogeneity reduction) for tree R after the best cut. Such a notation of heterogeneity and homogeneity gain is conceptually similar to the use of entropy and information gain in the decision-tree algorithm (Quinlan 1993).

4.2 *Tree partitioning and regionalization algorithm*

Given a spatially contiguous tree, removing any edge from the tree generates two subtrees and thus two regions. To obtain k regions, $k-1$ edges are to be removed. The tree-partitioning algorithm is presented in figure 9. Its complexity is $O(kn^2)$. Usually, k is much smaller than n, and thus sometimes can be ignored.

5. Evaluation and comparison

This section evaluates the *quality* and *efficiency* of the six regionalization methods (among which the First-Order-SLK method produces the same result as the SKATER method). The quality of a regionalization result is evaluated based on four criteria: (1) the overall heterogeneity, (2) the balance of region sizes, (3) the data variation within each region, and (4) the preservation of original data distribution. The efficiency of the regionalization methods is evaluated based on their computational complexities and actual time costs.

Figure 9.   Partitioning algorithm dividing a spatially contiguous tree into k regions.

Evaluations are carried out with the 2004 US presidential election data for 3111 US counties. Although all methods introduced in this paper are able to derive regions based on multivariate information, evaluations presented here only use one variable so that readers can visually examine and confirm regionalization results. The selected variable is the percentage of total votes for Bush in each county (figure 10). The spatial distribution of data values are rather complex—many areas have mixed values and exhibit no clear boundary. Prior to the regionalization procedure, the univariate data are normalized to a zero mean and unit variance. There are three counties, including Yellowstone National Park (Montana), Clifton Forge (Virginia), and South Boston (Virginia), that had no election data available and thus zero percentage (0%) for Bush. These three counties are included in the evaluation to see how each method is affected by outliers (or extreme values).

5.1 Quality comparison

5.1.1 Overall heterogeneity. Each of the six methods is used to divide the country into k regions according to values of the selected variable. The overall heterogeneity values for different regionalization methods and for different k values (ranging from 2 to 30) are plotted in figure 11. The Full-Order-SLK is the worst among all, for any k value. The Full-Order-ALK and Full-Order-CLK methods are significantly better than other methods for all k values. Moreover, if k is greater than 12, the Full-Order-CLK method is the best. In another evaluation (which is not shown) with a
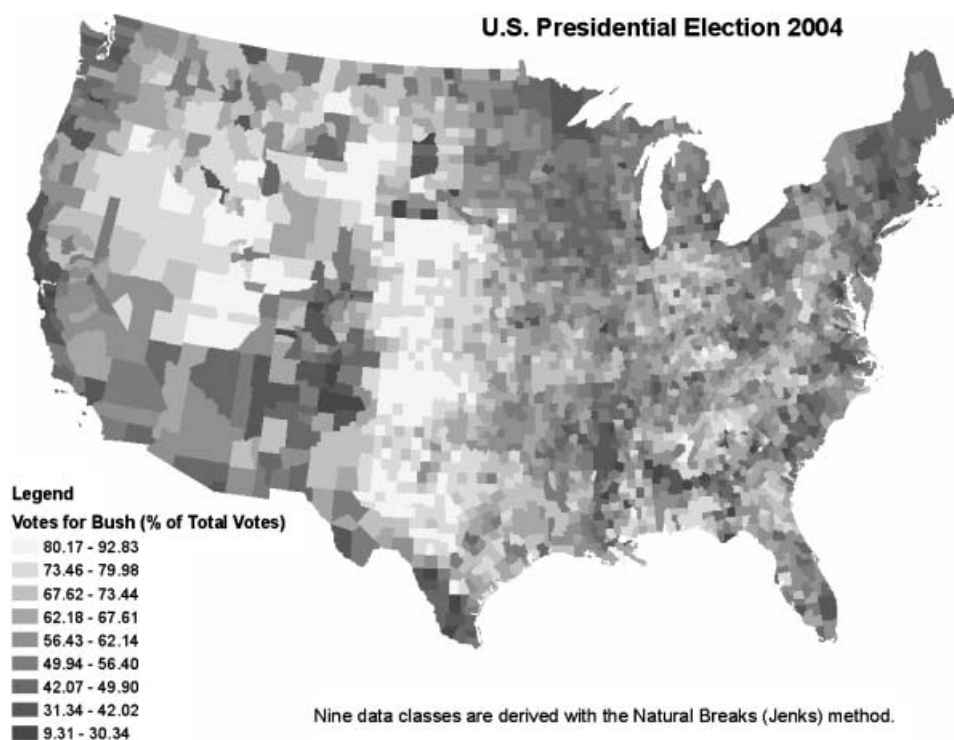

Figure 10. 2004 US presidential election data. A single variable (i.e. the percentage of total votes for Bush) is used for the simplicity of presentation and visual verification, although all regionalization methods proposed in this research can process multivariate data as well.

different variable, the Full-Order-CLK is the best for all k values. Due to space limitations, this paper does not intend to provide a comprehensive evaluation with various data sets. Given the complexity of the election data being used (figure 10), it is reasonable to conclude that: (1) the Full-Order-CLK and the Full-Order-ALK methods are significantly better than other methods in producing homogeneous regions; and (2) the Full-Order-CLK is the best if k is relatively large (e.g. greater than 12 for the data being used here). Since the proposed methods are efficient, one can apply them separately to the same data and find out which is the best.

Subsequent quality evaluations focus on three of the six methods: the *First-Order-SLK* (which produces the same result as the SKATER method does), the *Full-Order-ALK* and *Full-Order-CLK* methods (which are the best two methods according to the overall heterogeneity shown in figure 11). The Full-Order-ALK curve in figure 11 shows a relatively large decrease in heterogeneity when $k=7$, where it is also better than the Full-Order-CLK method. On the other hand, the Full-Order-CLK is better than the Full-Order-ALK when $k>12$, and the heterogeneity difference between the two methods reaches the largest when k is equal to or greater than 22. Therefore, the regionalization results for $k=7$ and $k=22$ are used for subsequent quality evaluations of the three selected methods.

Figures 12–14 show the regionalization results for seven regions, derived with the Full-Order-CLK, Full-Order-ALK, and the First-Order-SLK (i.e. SKATER), respectively. Figures 15–17 show the regionalization results for 22 regions with the three methods. From the six regionalization maps, it is apparent that the regions produced by the Full-Order-CLK and -ALK methods are more reasonable than the First-Order-SLK (i.e. SKATER) result. This is in agreement with the heterogeneity rankings shown in figure 11.
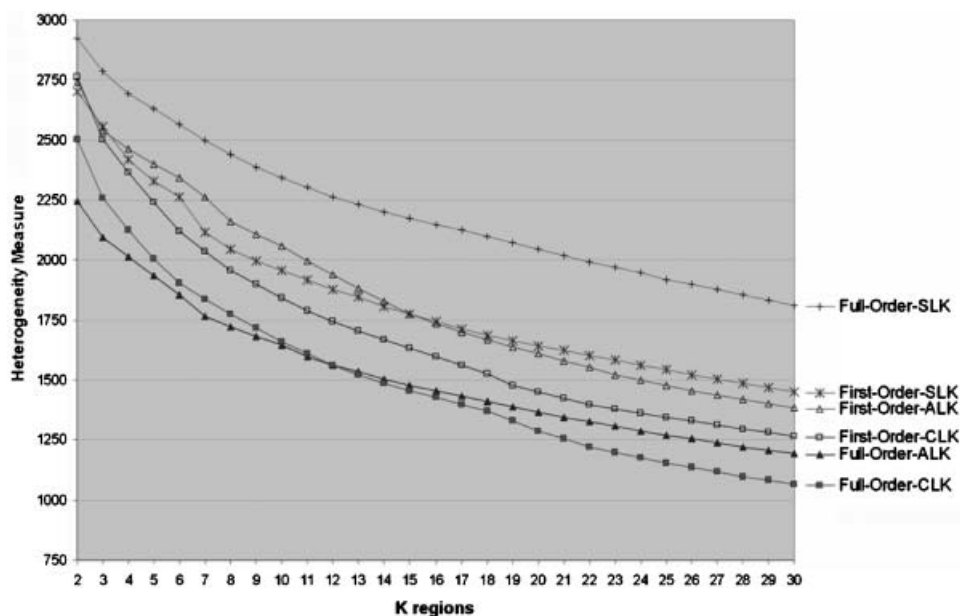

Figure 11. Comparison of the six methods based on the heterogeneity measure. Both Full-Order-ALK and Full-Order-CLK are consistently and significantly better than the First-Order-SLK method (i.e. SKATER) for any number of regions.

5.1.2 Region sizes and internal variations. Although the regionalization algorithms seek only to minimize the total heterogeneity, it is often desirable that region sizes are relatively even (unless natural groupings strongly indicate otherwise) and that the internal variation (measured with standard deviation) within each region is as small as possible. Note that the standard deviation of a region is different from the heterogeneity of a region in that the latter is a sum of squared deviations, which is not normalized by the region size (equation (4)).

Figure 18 shows region sizes and internal standard deviations for seven regions. The seven regions produced by the Full-Order-CLK have relatively small internal standard deviations (with a maximum around 11.0 and a minimum around 6.0), while the seven Full-Order-ALK regions are more balanced in region sizes. The First-Order-SLK result for seven regions is the worst in terms of either region sizes (with a region larger than 2000) or internal standard deviations (with two regions above 12.0).

Figure 19 shows the region sizes and internal standard deviations for the three 22-region results. The 22 regions produced by the Full-Order-CLK are better in terms of small internal standard deviations (all regions but one are below 10) and balanced region sizes (all are smaller than 500). The Full-Order-ALK regions are uneven in size (with a region larger than 1000) and have relatively larger internal standard deviations than the Full-Order-CLK regions. The First-Order-SLK result is again the worst among the three selected methods in terms of both region sizes (with a maximum around 1250) and internal standard deviations (with six regions above 10 and two above 12). Regarding the three outliers (i.e. zero percentages), both the
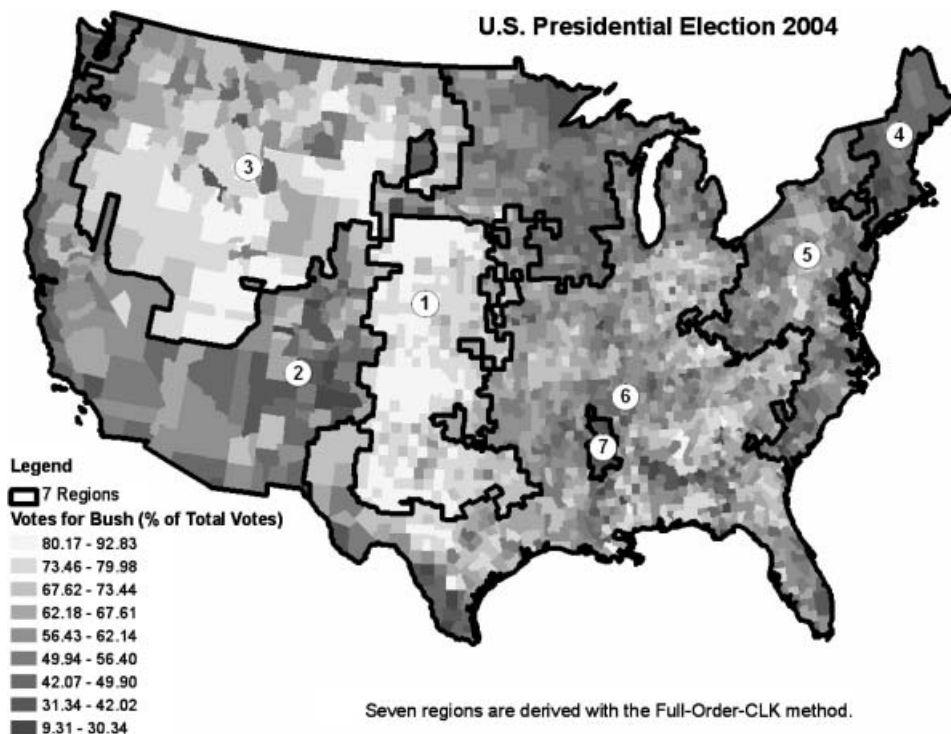

Figure 12.   Seven regions derived with the Full-Order-CLK method.
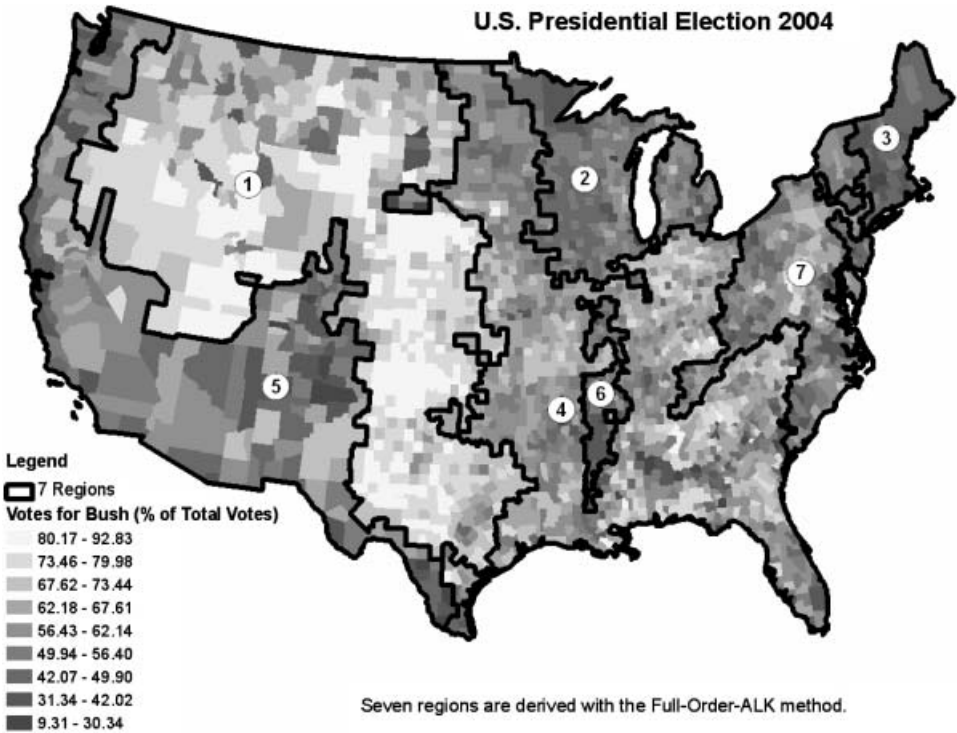
D. Guo

Figure 13.   Seven regions derived with the Full-Order-ALK method.
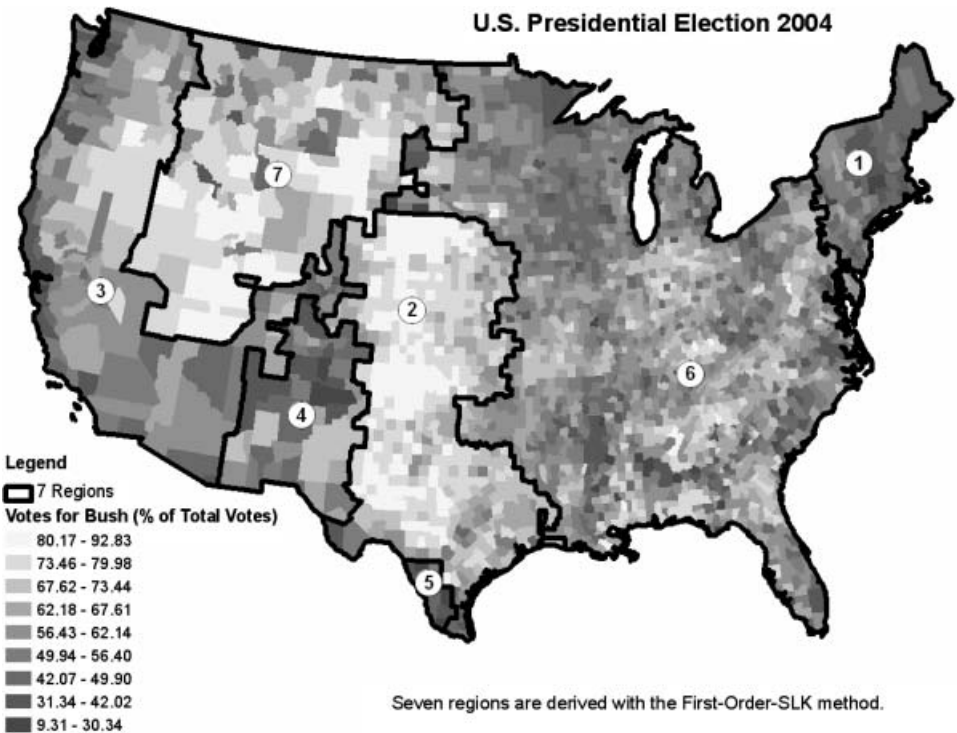

Figure 14.   Seven regions derived with the First-Order-SLK (i.e. SKATER) method.
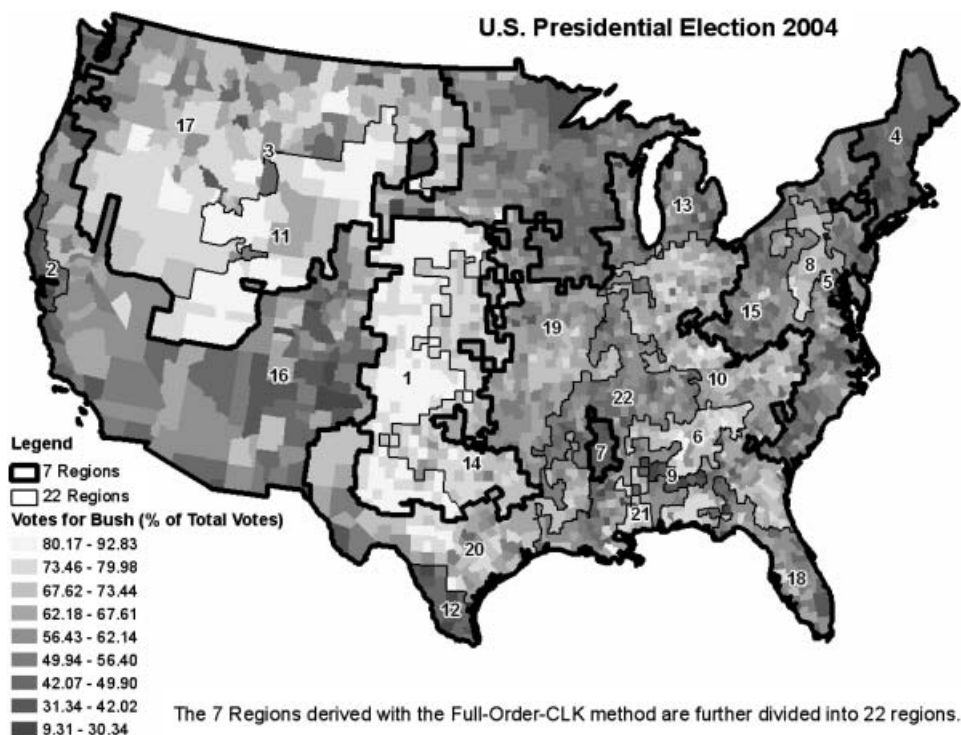
Figure 15. Twenty-two regions derived with the Full-Order-CLK method. Among the three outliers (i.e. zero percentage for Bush), only Yellowstone National Park stands out as a single-county region (i.e. region 3).

Full-Order-CLK and -ALK methods treat Yellowstone National Park as a separate region (see region 3 in figure 15 and region 1 in figure 16), while the First-Order-SLK has two single-county regions, i.e. Yellowstone National Park (Montana) and Clifton Forge (Virginia) (see region 7 and region 21 in figure 17). This indicates that First-Order-SLK is more sensitive to outliers.

5.1.3 Preservation of data distribution. The preservation of a data distribution is important when regionalization is used to summarize large data for subsequent analysis. In other words, if each region is treated as a single object (whose attribute values are defined as the average values of data objects contained in the region), it is desirable that the distribution of regional averages be similar to the original data distribution. The empirical distribution (i.e. cumulative probability) for each regionalization result is estimated with its 22 regional average values (one for each region). The original data distribution is estimated with the 3111 original values (one for each county). The four distributions are shown in figure 20.

The Kolmogorov–Smirnov test (Conover 1999, p. 428) is used to test if the difference (D) between a regionalization distribution and the original data distribution is statistically significant. The three test results are $D_{\text{CLK-ORG}}=0.245$, $D_{\text{ALK-ORG}}=0.326$, and $D_{\text{SLK-ORG}}=0.355$. The critical value for D is 0.281 (at the confidence level $\alpha=0.05$ with a sample size $n=22$; see Conover 1999, p. 547). Therefore, the distributions of both First-Order-SLK and Full-Order-ALK are significantly different from the original data distribution, while the distribution of Full-Order-CLK is statistically the same as the original distribution. Visually, one
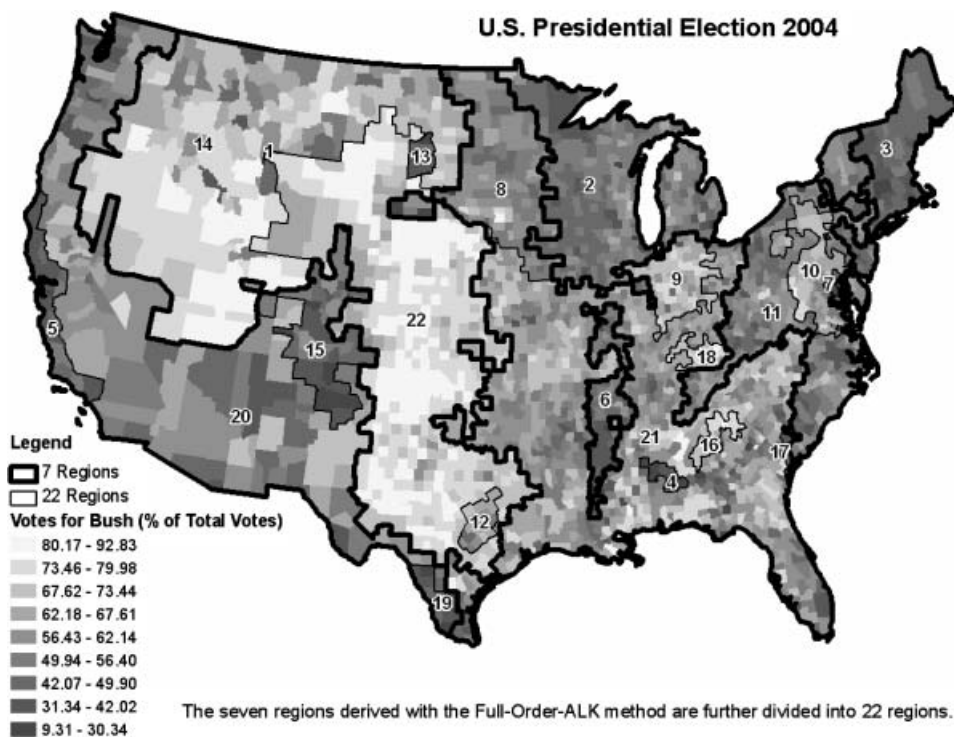
Figure 16.   Twenty-two regions derived with the Full-Order-ALK method. Among the three outliers (i.e. zero percentage for Bush), Yellowstone National Park stands out as a region on its own (i.e. region 1).

can also confirm that the Full-Order-CLK distribution is closer to the original distribution than the other two (figure 20).

   To summarize, seven quality comparisons were presented (table 1). The Full-Order-CLK method excels in five comparisons, while the Full-Order-ALK excels in two (both of which are for seven regions).

5.2   *Efficiency comparison*

The theoretical complexity analysis for each algorithm has been presented in sections 3.3 and 4.2. However, since both SKATER and AZP do not provide a complexity analysis, a detailed comparison based on data sizes and actual running time for each method is provided in table 2. Although the First-Order-SLK and the SKATER produce the same result, their implementations (and thus efficiency performances) are different. The Full-Order-CLK method, which is the best in quality comparisons, can find the solution in $O(n^2 \log n)$ time. For the country-level US presidential election data, it took 39 s to process 3111 counties on a desktop computer with 2 GB of RAM and a 3.60-GHz Pentium 4 CPU.

6.   **Conclusion and discussion**

This paper presents the REDCAP family of six regionalization methods, which are designed, based on three contiguity-constrained hierarchical clustering methods and two different constraining strategies. Overall, the Full-Order-CLK method produces
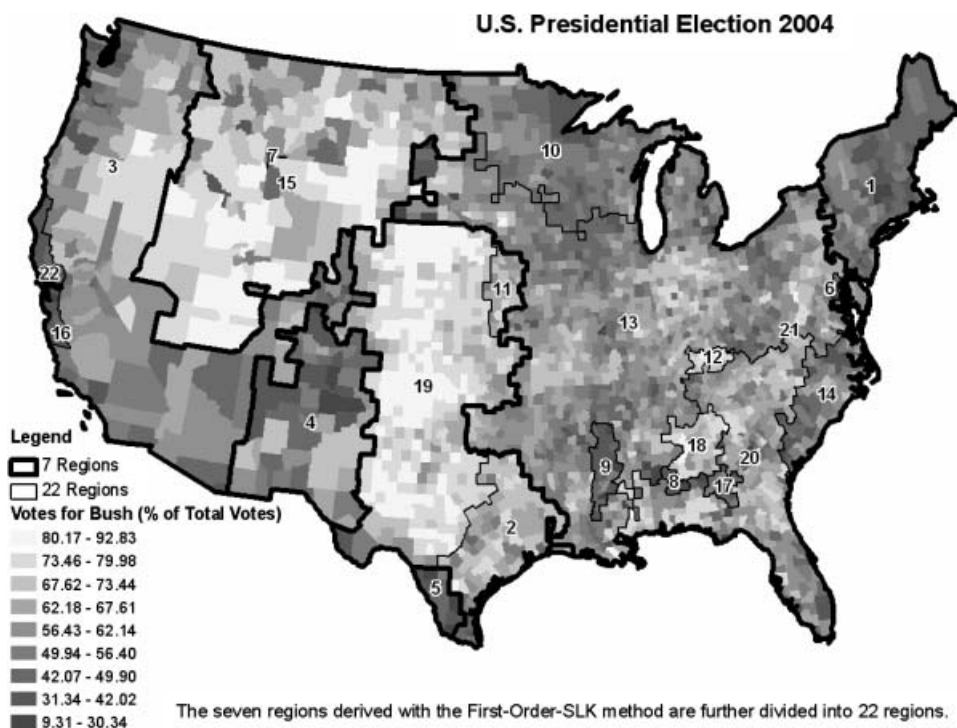
Figure 17.   Twenty-two regions derived with the First-Order-SLK (i.e. SKATER) method.
Among the three outliers (i.e. zero percentage for Bush), two stand out as single-county
regions, including region 7 for Yellowstone National Park and region 21 for Clifton Forge
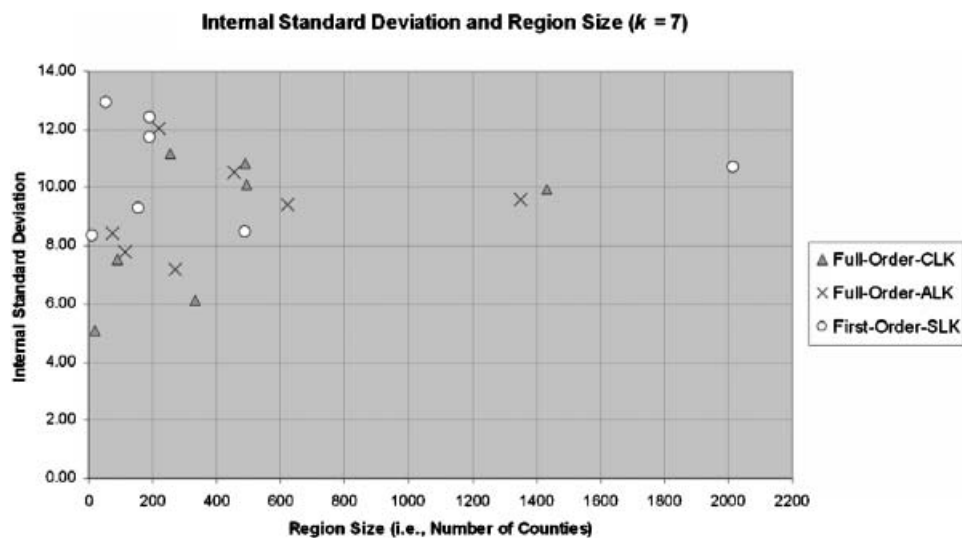(Virginia).

Figure 18.   Comparison of region sizes and internal variations for seven regions produced by
the Full-Order-CLK, Full-Order-ALK, and First-Order-SLK (i.e. SKATER) methods.

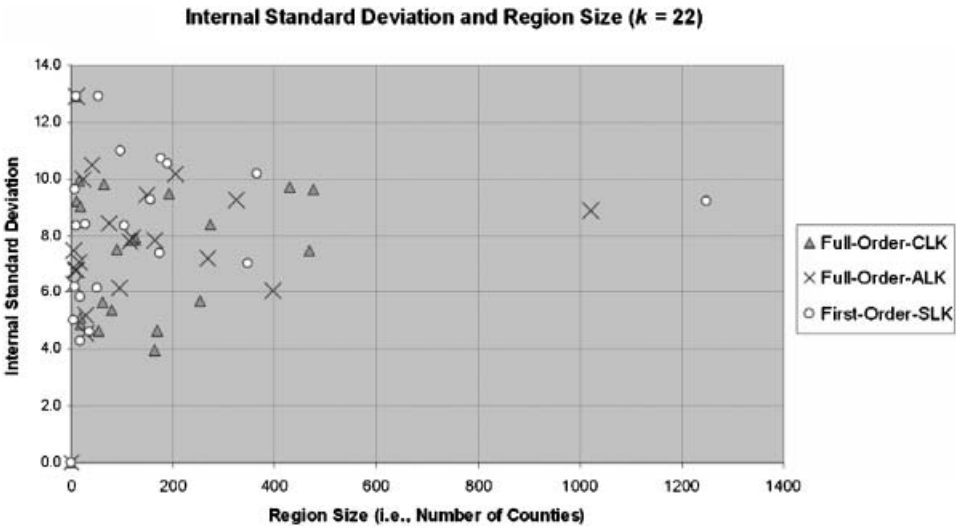**Internal Standard Deviation and Region Size (k = 22)**

Figure 19.   Comparison of region sizes and internal variations for 22 regions derived with the Full-Order-CLK, Full-Order-ALK, and First-Order-SLK (i.e. SKATER) methods.

significantly better results than other methods. Evaluations are carried out with the 2004 US presidential election data. Due to limited space, this paper does not provide a comprehensive evaluation with various data sets of different data distributions or spatial patterns. Given the complexity of the election data being used, the evaluation result reasonably reflects the differences among the six methods. An evaluation with
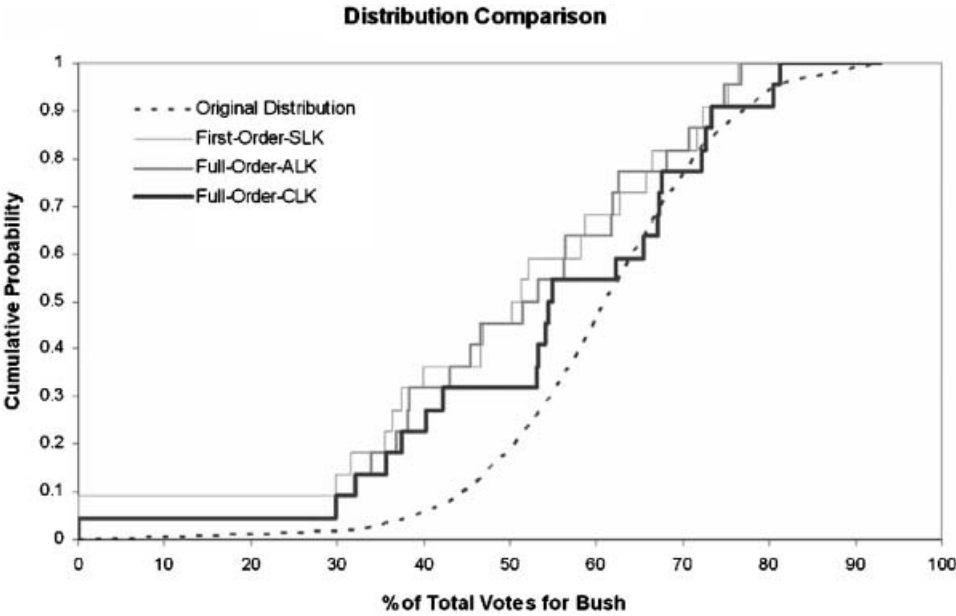
Distribution Comparison

Figure 20.   Original data distribution (with 3111 values) and three regionalization distributions (estimated empirically with 22 regional average values). The distribution of the 22 Full-Order-CLK regions is most similar to (and statistically the same as) the original data distribution (see text for details).

Table 1. Summary of the quality evaluation results for the three selected regionalization methods[a].

Regionalization methods	k=7			k=22			
	Heterogeneity	Region size	Internal variation	Heterogeneity	Region size	Internal variation	Distribution preservation
Full-order-CLK	2	2	1	1	1	1	1
Full-order-ALK	1	1	2	2	2	2	2
First-order-SLK (i.e. SKATER)	3	3	3	3	3	3	3

[a]Integer numbers indicate the ranking (with 1 being the best) of the three methods for each specific criterion. The Full-Order-CLK method excels in five comparisons (out of seven).

Table 2. Efficiency comparison of eight regionalization methods (including SKATER and AZP).

Regionalization methods	SLK		ALK		CLK		SKATER[a]	AZP[b]
	First-order	Full-order	First-order	Full-order	First-order	Full-order		
Theoretical complexity[c]	$O(n^2)$	$O(n^3)$	$O(n^2\log n)$	$O(n^2\log n)$	$O(n^2)$	$O(n^2\log n)$	NA	NA
Data set size (no. of objects)	3111	3111	3111	3111	3111	3111	853	2926
Time used[d] (s)	21	49	31	60	19	39	889	1671
Result quality rank[e] (k=30)	5	6	4	2	3	1 (best)	5	6

[a]Results are from Assunção *et al.* (2006).
[b]Results are from Openshaw and Rao (1995).
[c]All methods are linearly scaled to the number of regions (k) and the number of attributes (d), which are not included in the complexity notation. The complexity includes both the constrained clustering procedure and the tree partitioning procedure. The implementation of the SKATER method is not efficient according to its actual running time, although it produces the same result as the First-Order-SLK.
[d]This is the time cost for deriving 30 regions. The machine used in this research is a desktop computer with 2.0 GB of RAM and a 3.60-GHz Pentium 4 CPU. The machine used for the SKATER result was not specified in Assunção *et al.* (2006). It is assumed to be similar to the machine used in this research. The machine used for the AZP result was a Sun-Supersparc 10 model 40 workstation (Openshaw and Rao 1995).
[e]See figure 11 and table 1 for details on quality comparisons. According to Assunção *et al.* (2006), SKATER produces better results than AZP, which is the reason that AZP is ranked six. However, there is no comparison available between AZP and Full-Order-SLK. Therefore, both are ranked six.

another data set (not included in this paper) shows a similar ranking for the six proposed methods.

The Full-Order-CLK and the Full-Order-ALK methods, which produce regions of better qualities than other methods, are efficient and can find the solution in $O(n^2 \log n)$ time. With such data scalability, it is possible to regionalize large data sets of 10 000 or more spatial objects. However, full-order constrained regionalization methods require $O(n^2)$ memory usage. Therefore, processing large data sets might require memory management (or memory-efficient implementations of the algorithms). The first-order constrained methods are relatively more efficient in terms of both memory usage and computational time, at the cost of regionalization quality.

The regionalization methods proposed in this research can be applied to different application domains and allow different configurations for spatial contiguity, dissimilarity, and/or heterogeneity. The spatial contiguity matrix can be arbitrarily defined by the user and thus is transparent to the regionalization method. In this research, two counties are considered spatial neighbours if their boundaries touch each other (even by a single point), which may result in regions that are connected by a narrow bridge. A more strict contiguity definition can produce more geographically compact regions. Real applications often require a domain-specific definition of the dissimilarity between two data objects, which the user can control by providing a pairwise dissimilarity matrix. A different heterogeneity definition requires a minor modification of the program but does not affect the algorithm itself. However, if the calculation of the new heterogeneity measure is more complex than calculating the sum of squared deviations, it may increase the overall computational complexity. Related software that implements the proposed algorithms is available for download at www.spatialdatamining.org.

Acknowledgements

References

Assunção, R.M., Neves, M.C., Câmara, G. and Freitas, C.D.C., 2006, Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, **20**, pp. 797–811.

Conover, W.J., 1999, *Practical Nonparametric Statistics* (New York: Wiley).

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2001, *Introduction to Algorithms* (Cambridge, MA: MIT Press and McGraw-Hill).

Cutter, S.L. (Ed.), 2001, *American Hazardscapes: the Regionalization of Hazards and Disasters* (Washington, DC: Joseph Henry Press).

Felner, A., 2005, Finding optimal solutions to the graph partitioning problem with heuristic search. *Annals of Mathematics and Artificial Intelligence*, **45**, pp. 293–322.

Fovell, R.G. and Fovell, M.-Y.C., 1993, Climate zones of the conterminous United States defined using cluster analysis. *Journal of Climate*, **6**, pp. 2103–2135.

GOODCHILD, M.F., 1979, The aggregation problem in location allocation. *Geographical Analysis*, **11**, pp. 240–255.

GUO, D., PEUQUET, D. and GAHEGAN, M., 2003, ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata. *GeoInformatica*, **7**, pp. 229–253.

HAGGETT, P., CLIFF, A.D. and FREY, A., 1977, *Locational Analysis in Human Geography* (London: Arnold).

HAINING, R., 2003, *Spatial Data Analysis—Theory and Practice* (Cambridge: Cambridge University Press).

HAINING, R.P., WISE, S.M. and BLAKE, M., 1994, Constructing regions for small area analysis: material deprivation and colorectal cancer. *Journal of Public Health Medicine*, **16**, pp. 429–438.

HAN, J., KAMBER, M. and TUNG, A.K.H., 2001, Spatial clustering methods in data mining: a survey. In *Geographic Data Mining and Knowledge Discovery*, H.J. Miller and J. Han (Eds), pp. 33–50 (London: Taylor & Francis).

HANDCOCK, R. and CSILLAG, F., 2004, Spatio-temporal analysis using a multiscale hierarchical ecoregionalization. *Photogrammetric Engineering and Remote Sensing*, **70**, pp. 101–110.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J., 2001, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (New York: Springer).

JAIN, A.K. and DUBES, R.C., 1988, *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall).

JAIN, A.K., MURTY, M.N. and FLYNN, P.J., 1999, Data clustering: a review. *ACM Computing Surveys (CSUR)*, **31**, pp. 264–323.

KARYPIS, G. and KUMAR, V., 1998, Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, **48**, pp. 96–129.

MORRIS, R.D. and MUNASINGHE, R.L., 1993, Aggregation of existing geographic regions to diminish spurious variability of disease relative risks. *Statistics in Medicine*, **12**, pp. 1915–1929.

OLIVER, M.A. and WEBSTER, R., 1989, A geostatistical basis for spatial weighting in multivariate classification. *Mathematical Geology*, **21**, pp. 15–35.

OPENSHAW, S., 1977, A geographical solution to scale and aggregation problems in region-building, partitioning, and spatial modelling. *Transactions of the Institute of British Geographers*, **NS 2**, pp. 459–472.

OPENSHAW, S. and RAO, L., 1995, Algorithms for reengineering 1991 census geography. *Environment & Planning A*, **27**, pp. 425–446.

OSNES, K., 1999, Iterative random aggregation of small units using regional measures of spatial autocorrelation for cluster localization. *Statistics in Medicine*, **18**, pp. 707–725.

QUINLAN, J.R., 1993, *C4.5: Programs for Machine Learning* (San Francisco, CA: Morgan Kaufmann).

SAAB, Y.G., 2004, An effective multilevel algorithm for bisecting graphs and hypergraphs. *IEEE Transactions on Computers*, **53**, pp. 641–652.

SPENCE, N.A., 1968, A multivariate uniform regionalization of British counties on the basis of employment data for 1961. *Region Studies*, **2**, pp. 87–104.

TOBLER, W.R., 1969, Geographical filters and their inverses. *Geographical Analysis*, **1**, pp. 234–253.

WIRTH, N., 1976, *Algorithms + Data Structures = Programs* (Englewood Cliffs, NJ: Prentice-Hall).

WISE, S.M., HAINING, R.P. and MA, J., 1997, Regionalization tools for the exploratory spatial analysis of health data. In *Recent Developments in Spatial Analysis: Spatial Statistics, Behavioural Modelling and Neuro-Computing*, M. Fischer and A. Getis (Eds), pp. 83–100 (Berlin: Springer).