

Greedy Optimization for Contiguity-Constrained Hierarchical Clustering

Diansheng Guo

Department of Geography
University of South Carolina
Columbia, SC 29229
E-mail: guod@sc.edu

Abstract—The discovery and construction of inherent regions in large spatial datasets is an important task for many research domains such as climate zoning, eco-region analysis, public health mapping, and political redistricting. From the perspective of cluster analysis, it requires that each cluster is geographically contiguous. This paper presents a contiguity constrained hierarchical clustering and optimization method that can partition a set of spatial objects into a hierarchy of contiguous regions while optimizing an objective function. The method consists of two steps: contiguity constrained hierarchical clustering and two-way fine-tuning. The above two steps are repeated to create a hierarchy of regions. Evaluations and comparison show that the proposed method consistently and significantly outperforms existing methods by a large margin in terms of optimizing the objective function. Moreover, the method is flexible to accommodate different objective functions and additional constraints (such as the minimum size of each region), which are useful to for various application domains.

Keywords—clustering; contiguity constraint; optimization

I. INTRODUCTION

Finding inherent regions in multivariate geographic datasets is an important task for many research problems such as climate zoning [3], eco-region analysis [6], map generalization [16], census reengineering [10], public health mapping [5, 11], and political districting [1].

Given a set of spatial objects, each of which having one or more attribute values, a regionalization method attempts to find an optimal grouping of the objects into a number of regions (which must be spatially contiguous) and meanwhile optimizes an objective function (e.g., a measure of multivariate similarities within regions). This is a combinatorial problem—it is not practical to enumerate all possible groupings to find the global optimal. Therefore, a regionalization method usually adopts heuristic-based approaches to reach a near-optimal solution.

A number of regionalization methods have been developed in the literature, which will be briefly reviewed in Section II. This research develops an iterative clustering and optimization approach to regionalization, which significantly outperforms existing methods by a large margin. The new approach consists of two steps:

1. Contiguity constrained hierarchical clustering, which optimizes an objective function at each merge,

enforces spatial contiguity, and eventually produces two clusters (regions);

2. Greedy optimization with a fine-tuning procedure, which iteratively modifies the boundaries between the two clusters (regions) and significantly improves the region quality (i.e., a much better objective value).

The above two steps are then repeated for each of the two newly generated regions to construct a hierarchy of regions. Both steps enforce the spatial contiguity constraint and therefore clusters at any hierarchical level are guaranteed to be spatially contiguous.

There are two main contributions of this research. First, the same objective function is used for both the clustering and the greedy optimization while previous research used different criteria in the two steps [4]. This improvement enables the integration of different objective functions with the proposed approach to accommodate different requirements in different applications.

Second, the greedy optimization (fine-tuning) step developed in this research significantly improves the quality of regionalization, which will be shown with evaluation results in Section IV. This fine-tuning procedure can be combined with any existing regionalization method to improve its result.

II. BACKGROUND

General-purpose clustering methods do not consider spatial contiguity and thus data items in a cluster are not necessarily contiguous in the geographic space. Existing regionalization methods that are based on the clustering concept often take three different strategies: (1) general-purpose clustering followed by spatial processing; (2) general-purpose clustering with a spatially weighted dissimilarity measure; and (3) enforcing contiguity constraints during the clustering process.

The first group of methods use a general-purpose clustering method to derive clusters based on multivariate similarity and then divide or merge the clusters to form contiguous regions [3, 5]. The drawback of this type of methods is that the number and quality of regions is very difficult to control.

The second type of methods incorporates spatial distance explicitly in the similarity measure for a general clustering method (e.g., K-Means) [18] and thus data items in the same cluster tend to be spatially close to each other. However, the spatial contiguity of a cluster is not guaranteed. Moreover,

the incorporation of spatial distance in the similarity measure reduces the importance of multivariate information and may also not be able to find clusters of arbitrary shapes.

The third approach, represented by REDCAP [4], explicitly incorporates spatial contiguity constraints (rather than spatial similarities) in a hierarchical clustering process. Particularly, the REDCAP approach can optimize an objective function during the construction and partitioning of a cluster hierarchy to obtain a given number of regions. REDCAP is a family of six regionalization methods, which respectively extend the single-linkage (SLK), average-linkage (ALK), and complete-linkage (CLK) hierarchical clustering methods to enforce spatial contiguity constraints during the clustering process [4]. These six methods are similar in that they all iteratively merge clusters (which must be spatial neighbors) into a hierarchy and then partition the hierarchy to obtain regions. They differ in their definitions of “similarity” between two clusters.

As shown in the results of this research, although REDCAP methods are better than other methods and can produce reasonably good regions, there is much room to improve in terms of optimizing the objective function.

Graph-partitioning methods [2, 7, 12] may also 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 [7]. However, most graph partitioning methods cannot consider spatial contiguity constraint, except several graph-based image segmentation methods [13, 14], which focus on detecting objects in images.

III. METHODOLOGY

The proposed new regionalization method is an iterative procedure that partitions a data set into a hierarchy of clusters under contiguity constraint. The method can be conceptualized as the following three steps:

- Generate a hierarchy of clusters with a contiguity-constrained Ward clustering method, which is a new method developed in this research;
- Enhance the top two regions in the hierarchy with a greedy optimization (fine-tuning) procedure, which is also a new development in this research;
- Repeat the above two steps to temporarily partition each of existing regions into two, and accept the best among all partitions as the next level of the hierarchy.

The above steps form a top-down procedure, which starts with the entire data as one region and generates one more region at a time. The iteration stops when a given condition is met, such as a maximum number of regions, a minimum size of a region, or a threshold of a quality measure. Below, the first two steps are explained in detail.

A. Contiguity Constrained Clustering

The Ward method [17] for hierarchical clustering seeks to partition a set of data items into a number of clusters while minimizing the information loss associated with each grouping. This information loss can be defined in terms of

the sum of squared differences (SSD), which is defined in Equations (1) and (2).

$$SSD(R) = \sum_{j=1}^d \sum_{i=1}^{n_r} (x_{ij} - \bar{x}_j)^2 \quad (1)$$

$$SSD = \sum_{j=1}^k SSD(R_j) \quad (2)$$

In equation (1), R is a region, $SSD(R)$ denotes its SSD value, d is the number of attributes, n_r is the number of objects in R , x_{ij} is the value for the j th attribute of the i th object, and \bar{x}_j is the mean value of the j th attribute for all objects in R . The SSD value for a regionalization result is the sum of the SSD values of all regions (k is the total number of regions).

At each step in the Ward clustering, the union of every possible cluster pair is considered and the two clusters whose fusion results in minimum increase in SSD are combined. Below is the proposed contiguity constrained Ward clustering algorithm.

Algorithm 1: Contiguity Constrained Ward Method

Input: V : multivariate spatial data points, $|V| = n$;

C : $C(u, v) = 1$ if $u, v \in V$ are contiguous

- (1) Set $R = \{R_1, R_2, \dots, R_n\}$, i.e., each data point is a cluster (region) by itself;
- (2) Set edges $E = \emptyset$
- (3) For each R_u and $R_v \in R$, If $C(u, v) = 1$
Add an edge $e = \langle R_u, R_v \rangle$ to E
 $|e| = SSD(R_u \cup R_v) - SSD(R_u) - SSD(R_v)$
- (4) Repeat the following steps until $|R| = 2$
 - a) Find the shortest edge e^* in E
 - b) Let R_u, R_v be the two clusters that e^* connects
 - c) Remove e^* from E , and remove R_v from R
 - d) Update $R_u = R_u \cup R_v$
 - e) Redirect edges incident on R_v to R_u (remove duplicate edges if a cluster connects to both R_v and R_u)
 - f) Update the length of edges related to R_u

The time complexity of the above contiguity constrained Ward clustering is $O(n^2d)$, where n is the number of data points and d is the number of variables. Since E only contains edges that connect spatial neighbors, $|E|$ is proportional to n . Updating the length of an edge (i.e., the SSD difference before and after the merge) only takes constant time if we store the multivariate mean vector and size for each region (see Equation 3). In other words, there is no need to visit each data point to calculate the SSD value of the newly merged cluster since we only care about the difference in SSD.

Step (4) in the algorithm is iterated exactly $n - 1$ times, and each iteration takes $O(nd)$ time to find the shortest edge, make the merge, and update related edges in E . Therefore the overall time complexity is $O(n^2d)$.

$$SSD(R) - SSD(R_u) - SSD(R_v) = \sum_{j=1}^d (|R_u| \cdot \bar{x}_{uj}^2 + |R_v| \cdot \bar{x}_{vj}^2 - |R| \cdot \bar{x}_j^2) \quad (3)$$

where $R = R_u \cup R_v$; \bar{x}_{uj} , \bar{x}_{vj} , \bar{x}_j are mean values for the j th attribute in R_u , R_v , and R , respectively.

The memory complexity of the algorithm as presented above is $O(n^2)$ since it involves a contiguity matrix. However, since the contiguity matrix is sparse (due to the fact that an object only has a small number of spatial neighbors), the memory complexity can be easily improved to $O(nd)$ if we only keep pairs of spatial neighbors in memory.

B. Greedy Optimization with Fine Tuning

The above contiguity-constrained Ward clustering process generates two regions by minimizing the increase of SSD at each merge. Although the two regions generated by this bottom-up clustering procedure are already very good in terms of the SSD value, there is still much room to improve the objective function value. To further enhance the two-region partition, a fine-tuning greedy optimization procedure is developed to modify the boundaries between the two regions by moving data points from one region to the other while maintaining the contiguity of each region. This fine tuning procedure is independent of the clustering procedure. In other words, it can improve the quality of a given two-region partition, regardless of what method has been used to construct the two regions.

Suppose the Ward clustering step divides the data into regions A and B (each of which is spatially contiguous). The fine-tuning algorithm will find the best data point (among all the data points in A and B) that, when moved to the other region, decreases the overall SSD measure the most. If no object can be moved to decrease the overall measure, the one that causes the least increase in SSD will be moved. While moving an object from one region to the other, the spatial contiguity of both regions must be enforced. In other words, moving an object from a region should not break the contiguity of that region.

Above moves are made repeatedly but each location can only be moved once. When all the possible locations have been moved, the entire sequence of moves will be analyzed and the sub-sequence (i.e., the first m moves) that gives the maximum decrease in SSD will be accepted (i.e., a new partition is generated using this sub-sequence of moves).

Then, take this new partition as the starting point, the above procedure is repeated again until there is no further improvement in SSD.

Algorithm 2: Greedy Optimization (Fine-Tuning):

Inputs: $\{R_a, R_b\}$: two regions by the Ward method

- (1) Set Candidates = \emptyset , Moves = \emptyset , $R_1 = R_a$, $R_2 = R_b$
- (2) Find out which objects can move between R_1 and R_2 , add them to Candidates (see Section III. C for details on identifying candidates)

- (3) From all candidates, find the best object obj that, if moved, decreases SSD the most (or increases SSD the least if no one can decrease)
- (4) Modifying R_1 and R_2 by moving obj to the other region, add obj to Moves, and mark obj as “moved” (it won’t be candidate again in step 2)
- (5) Repeat steps (2) – (4) until no candidate
- (6) Analyze Moves (see Figure 1) and find the best sequence of moves that improves (decreases) SSD the most
- (7) If step (6) does not find any sequence that can improve SSD, stop the fine-tuning procedure
- (8) If step (6) find an improvement, modify R_a and R_b by making the best sequence of moves; remove all “moved” marks (so that those previously moved objects can move again in the next round)
- (9) Repeat steps (1) – (8).

Figure 1 shows an illustrative example of the fine-tuning procedure. Let steps (1) – (8) be a round of fine-tuning. For round 1, Figure 1 shows the sequence of best moves and the SSD value after each move. After analyzing the sequence, the first 4 moves are accepted since they together achieve the lowest SSD value (2600). Then round 2 starts with this new partition (after moving the four objects) and again find a sequence of moves, which consists of the first three moves (achieving an SSD value of 2450). (Note: the first three objects in round 2 are not the same as the first 3 in round 1.) Round 3 gives no further improvement and the fine-tuning procedure stops.

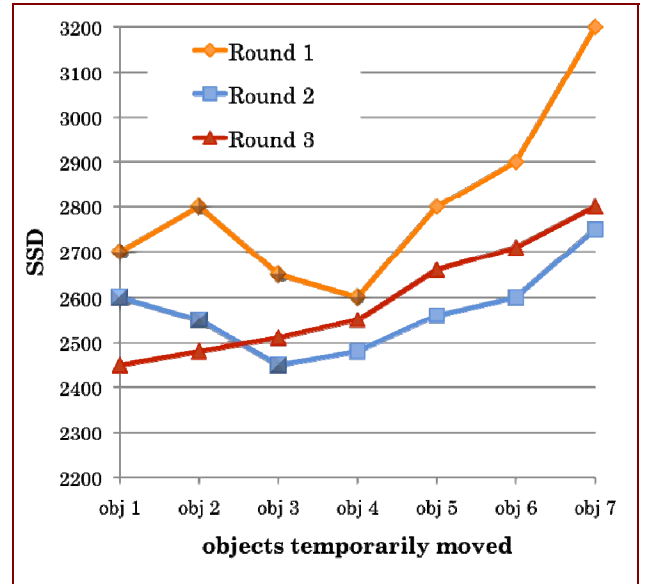


Figure 1. An illustrative example of the iterative fine-tuning procedure.

Please note that, the fine-tuning procedure allows objects to move even if they cause temporary increase in SSD, as long as subsequent moves can make up the loss and eventually achieve a better SSD. As such, the fine-tuning procedure has a chance to escape local optima and reach (or get close to) the global optima. Figures 3 and 4 shows the

results of the Ward clustering method alone and the Ward clustering coupled with the fine-tuning procedure. It is evident that the fine-tuning procedure can significantly improve the regionalization quality (in terms of minimizing the total SSD).

The complexity of this fine-tuning procedure is $O(n^2rd)$, where r is the number of rounds that the fine tuning procedure takes to converge and d is the number of variables. Based on the experiments conducted in this research, r is very small, typically ranging between 2 and 5. As seen in the conceptual outline of Algorithm 2, two steps are potentially time consuming: step (2) to find candidates to move and step (3) to find the SSD difference that each candidate move may cause. For each round, step (2) is repeated about n times and step (3) is repeated n^2 times. Section III.C will show that step (2) can be done with $O(n)$ time. Equation 3 shows that step (3) only takes $O(d)$ time. Thus, the overall complexity of the fine-tuning procedure is $O(n^2rd)$. However, this complexity depends on the efficiency of the specific measure or objective function being used. If a more time-consuming measure (instead of SSD) is used, the overall complexity will increase.

The method proposed in this paper may not be able to process very large datasets, such as images with millions of pixels. However, it is efficient enough to process most socio-economic data sets that commonly have about thousands of spatial objects. Most importantly, it delivers the high quality result that those applications demand. For much larger datasets, an approximate but more efficient version of this fine-tuning procedure is needed.

Although in spirit similar to the procedure introduced in [8, 9], the fine-tuning procedure developed in this research has two contributions. First, it ensures spatial contiguity, which makes the algorithm much more complex (see next subsection). Second, it starts with the Ward regionalization result (note: the fine-tuning procedure needs a reasonably good regionalization to start with).

C. Enforcing Contiguity During the Fine Tuning

The contiguity between two spatial objects may be defined in different ways. First, it depends on the type of spatial objects (e.g., areas, lines, or points). Second, for area data (e.g., states or counties), the contiguity between two

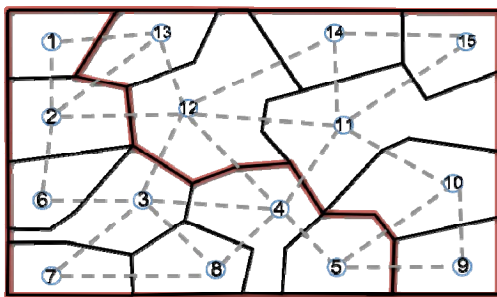
objects may be defined as (1) touching each other (even by a single point) or (2) sharing a boundary of certain length. Which definition to use is up to the user or application and is independent from the regionalization method, which only needs a contiguity matrix. In this research, the focus is on area data and two areas are contiguous if they share at least a line segment (not just a point) on the boundary.

Figure 2 shows an illustrative example data, which has 15 data objects (areas). If two objects are spatial neighbors (i.e., sharing a segment of boundary), they are connected by an edge. The data is divided into two regions, each being contiguous, i.e., the data objects in each region are connected. Suppose these two regions are the result of the contiguity constrained Ward clustering, then the fine-tuning procedure will attempt to modify the boundary between the two regions by moving objects from one to the other. During this process, the contiguity of each region must be maintained, which raises two issues.

First, given a partition, not all objects can move. We can only move those objects on the boundary between the two regions (i.e., there is an edge connecting the object to another object in the other region).

Second, some objects on the boundary, when moved, can break the contiguity of the origin region (which it belongs to before the move). From a graph-based perspective, such an object is an articulation point in the graph. For example, in Figure 2, if we move object 3 from the left region to the right region, the former will be broken into two components $\{1, 2, 6\}$ and $\{4, 5, 7, 8\}$. There are two options for this situation: do not allow object 3 to move; or move object 3 and its “associated component” together. The latter option is used because it provides more opportunity for improvement. We refer object 3 as the “primary object”.

The table on the right in Figure 2 shows the candidate list for the given partition (shown in the map on the left). This list will be updated in step (2) in the fine-tuning algorithm. An object is included in the list if it has not been moved before in the current round and it is on the boundary between the two regions. Each move may involve more than one object as explained above. However, after each move, we only mark the primary object as “moved” so that the objects in the associated component can still be candidates for the next move.



Primary object	Associated Component	Primary object	Associated Component
1	-	9	-
2	1	10	9
3	6, 2, 1	11	10, 9
4	5	12	13
5	-	13	-

Figure 2. Left: Graph-based representation of spatial contiguity. Spatial neighbors, which share a segment of boundary (solid thin line), are connected by a gray dash line. Right: Candidate moves for the given two regions (delineated by thick lines in the map)

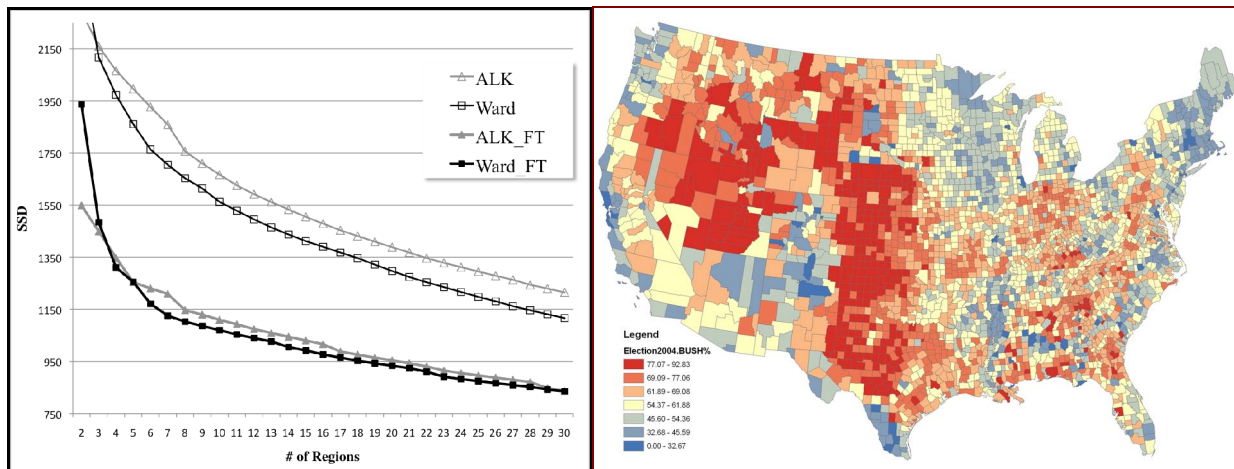


Figure 3. Left: Comparison of the contiguity constrained Average-Linkage clustering (ALK), Ward clustering (Ward), ALK with fine tuning (ALK_FT), and Ward with fine tuning (Ward_FT). Right: The evaluation data—percentage of votes for Bush for the 2004 Presidential Election.

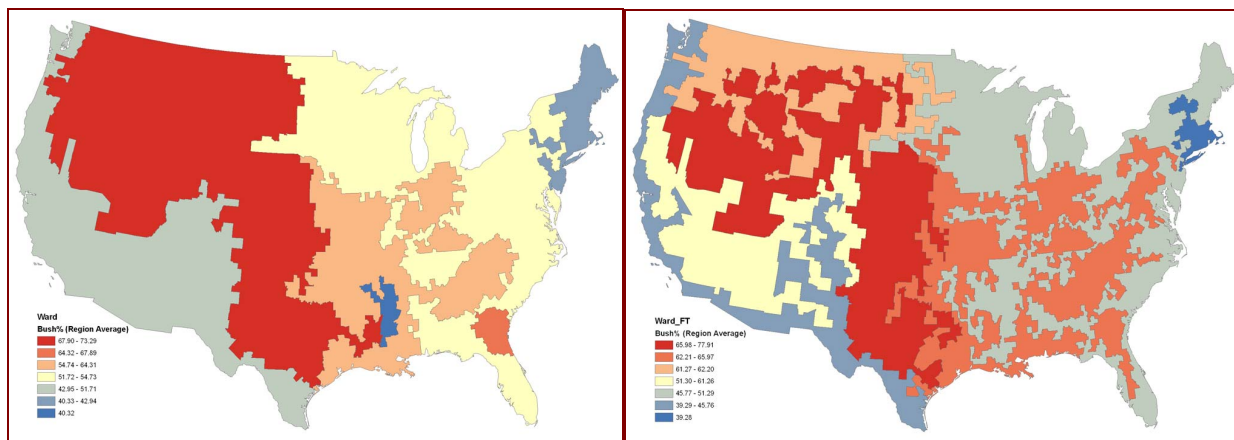


Figure 4. Left: Seven regions derived with Ward. Right: Seven regions derived with Ward_FT. Colors represent region averages. (A color version of this figure and Figure 3 are available at <http://www.spatialdatamining.org>)

Updating the candidate list (see step 2 in Algorithm 2) can be time consuming, since for each candidate it needs to traverse all objects in the region to see if it is an articulation point (i.e., its move will break contiguity). A linear complexity algorithm based on dept-first search has been devised in [15] to detect articulation points in a graph. We extended this algorithm to find all articulation points and their associated components by traversing the neighborhood graph (see Figure 2) only once, i.e., it takes $O(n)$ time. This procedure is repeated n time for each round in the fine-tuning procedure (see Algorithm 2).

IV. EVALUATION

The approach presented in this paper has two major contributions. First, it develops a contiguity constrained Ward clustering method. Second, it develops a fine-tuning greedy optimization method that can significantly improve the region quality. To evaluate the performance of both contributions, two comparisons are included:

- Comparing one of the best existing regionalization method in the REDCAP family, the contiguity

constrained average-linkage method (ALK), with the newly developed contiguity-constrained Ward clustering method; and

- Comparing the regionalization without the fine-tuning optimization and the regionalization with the fine-tuning optimization.

Since the fine-tuning procedure is independent of the clustering method, it is combined with both the ALK and the Ward method. Therefore, four methods are compared, i.e., ALK, Ward, ALK with fine-tuning (ALK_FT), and Ward with fine-tuning (Ward_FT).

The evaluation data being used is the 2004 election data. Although the proposed method can process any number of variables, for the ease of visual inspection, only one variable is used, i.e., the percentage of votes for Bush (Figure 3). Note that this experiment is only for the evaluation of the regionalization methods rather than a serious analysis of elections. For the latter purpose, a political scientist may want to include more variables in the process to better define political regions. The overall complexity of the approach is

linear to the number of variables and therefore it can easily handle more variables if needed.

Figure 3 presents the regionalization quality comparison of the four methods, with the SSD values for each hierarchical level (i.e., for different number of regions). Without the fine-tuning optimization, Ward is better than ALK except for 2 regions (the first partition). The most exciting part of the result is that the fine-tuning procedure can significantly improve the regionalization results of both ALK and Ward, by a large margin.

Figure 4 shows the seven regions derived with the Ward method alone and the seven regions by the Ward method combined with the fine-tuning procedure. Comparing both with the original data in Figure 3, it is obvious that the regions generated by the fine-tuning procedure are much better than those without the fine-tuning in terms of internal homogeneity (i.e., a smaller SSD value).

In Figure 4, the two large regions in the Eastern part interlocks with each other but each of them still maintains spatial contiguity! If for some applications such region shapes are not desirable, one can modify the objective function or add constraints to consider shapes.

Due to space limitation, we only show the evaluation result with the above one dataset. Similar performance results have also been achieved with other data sets.

V. DISCUSSION AND CONCLUSION

The paper presents a new regionalization method that is based on contiguity constrained clustering and greedy optimization. The evaluation results show that the new method outperforms existing methods by a large margin. Moreover, the fine-tuning optimization procedure can be combined with existing regionalization methods to significantly improve their performance.

By using Ward instead of ALK as the base clustering method, the proposed approach is flexible to integrate with different objective functions for different applications. With minor modification, it can also process non-metric data such as graph partitioning, under contiguity constraint. Other non-spatial constraint may also be incorporated such as the minimum size of each region or region shape requirements. However, adding more constraints may affect the efficiency of the method, if new constraints are time consuming to compute.

The overall complexity of the approach is $O(n^2rd)$ for computation time and $O(nd)$ for memory use. It is efficient to process 10,000 or more objects. However, a more efficient and approximate version is in need for processing very large data sets, such as high-resolution images. As to optimality, although the fine-tuning result in general appears to be very close to the global optima, it is still possible to improve.

A software package for the proposed method is available at <http://www.SpatialDataMining.org>.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant No. 0748813.

REFERENCES

- [1] M. Altman, and M.P. McDonald, "Bard: Better Automated Redistricting," *Journal of Statistical Software*, vol. 31, no. 3, pp. (in press), 2009.
- [2] A. Felner, "Finding Optimal Solutions to the Graph Partitioning Problem with Heuristic Search," *Annals of Mathematics and Artificial Intelligence*, vol. 45, no. 3-4, pp. 293-322, DEC, 2005.
- [3] R.G. Fovell, and M.-Y.C. Fovell, "Climate Zones of the Conterminous United States Defined Using Cluster Analysis," *Journal of Climate*, vol. 6, no. 11, pp. 2103-2135, 1993.
- [4] D. Guo, "Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning (REDCAP)," *International Journal of Geographical Information Science*, vol. 22, no. 7, pp. 801-823, 2008.
- [5] R.P. Haining, S.M. Wise, and M. Blake, "Constructing Regions for Small Area Analysis: Material Deprivation and Colorectal Cancer," *Journal of Public Health Medicine*, vol. 16, pp. 429-438, 1994.
- [6] R. Handcock, and F. Csillag, "Spatio-Temporal Analysis Using a Multiscale Hierarchical Ecoregionalization," *Photogrammetric Engineering and Remote Sensing*, vol. 70, pp. 101-110, 2004.
- [7] G. Karypis, and V. Kumar, "Multilevel K-Way Partitioning Scheme for Irregular Graphs," *Journal of Parallel and Distributed Computing*, vol. 48, no. 1, pp. 96-129, JAN 10, 1998.
- [8] M.E. Newman, "Modularity and Community Structure in Networks," *Proc Natl Acad Sci U S A*, vol. 103, no. 23, pp. 8577-82, Jun 6, 2006.
- [9] M.E.J. Newman, "Modularity and Community Structure in Networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577-8582, Jun, 2006.
- [10] S. Openshaw, and L. Rao, "Algorithms for Reengineering 1991 Census Geography," *Environment & Planning A*, vol. 27, no. 3, pp. 425-446, 1995.
- [11] K. Osnes, "Iterative Random Aggregation of Small Units Using Regional Measures of Spatial Autocorrelation for Cluster Localization," *Statistics in Medicine*, vol. 18, pp. 707-725, 1999.
- [12] Y.G. Saab, "An Effective Multilevel Algorithm for Bisecting Graphs and Hypergraphs," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 641-652, JUN, 2004.
- [13] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and Adaptivity in Segmenting Visual Scenes," *Nature*, vol. 442, no. 7104, pp. 810-813, AUG 17, 2006.
- [14] J. Shi, and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 22, pp. 888-905, 2000//, 2000.
- [15] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal of Computing*, vol. 1, pp. 146-60, 1972.
- [16] W.R. Tobler, "Geographical Filters and Their Inverses," *Geographical Analysis*, vol. 1, no. 3, pp. 234-253, 1969.
- [17] J.H. Ward, "Hierarchical Grouping to Optimise an Objective Function," *Journal of the American Statistic Association*, vol. 58, pp. 236-244, 1963.
- [18] S.M. Wise, R.P. Haining, and J. Ma, "Regionalization Tools for the Exploratory Spatial Analysis of Health Data," *Recent Developments in Spatial Analysis: Spatial Statistics, Behavioural Modelling and Neuro-Computing*, M. Fischer and A. Getis, eds., Berlin: Springer-Verlag, 1997