

Opening the Black Box: Interactive Hierarchical Clustering for Multivariate Spatial Patterns

Diansheng Guo

Donna Peuquet

Mark Gahegan

GeoVISTA Center & Department of Geography, Pennsylvania State University

302 Walker Building, University Park, PA 16802, USA

1-814-865-3433, 1-814-863-7943 (fax), <http://www.geovista.psu.edu>

dguo@psu.edu

peuquet@geog.psu.edu

mark@geog.psu.edu

ABSTRACT

Clustering is one of the most important tasks for geographic knowledge discovery. However, existing clustering methods have two severe drawbacks for this purpose. First, spatial clustering methods have so far been mainly focused on searching for patterns within the spatial dimensions (usually 2D or 3D space), while more general-purpose high-dimensional (multivariate) clustering methods have very limited power in recognizing spatial patterns that involve neighbors. Secondly, existing clustering methods tend to be ‘closed’ and are not geared toward allowing the interaction needed to effectively support a human-led exploratory analysis. The contribution of the research includes three parts. (1) Develop an effective and efficient hierarchical spatial clustering method, which can generate a 1-D spatial cluster ordering that preserves all the hierarchical clusters. (2) Develop a density- and grid-based hierarchical subspace clustering method to effectively identify high-dimensional clusters. The spatial cluster ordering is then integrated with this subspace clustering method to effectively search multivariate spatial patterns. (3) The above two methods are implemented in a fully open and interactive manner and supported by various visualization techniques. This opens up the “black box” of the clustering process for easy understanding, steering, focusing and interpretation. At the end a working demo with US census data is presented.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*data mining, spatial databases and GIS*; I.5.3 [Pattern Recognition]: Clustering—*algorithms, similarity measures*; I.5.3 [Pattern Recognition]: Implementation—*interactive systems*

General Terms: Algorithms, Design, Human Factors

Keywords

Geographic Knowledge Discovery, Spatial Ordering, Hierarchical Subspace Clustering, Visualization and Interaction

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GIS'02, November 8-9, 2002, McLean, Virginia, USA.

Copyright 2002 ACM 1-58113-591-2/02/0011...\$5.00.

1. INTRODUCTION

The unprecedented *large size* and *high dimensionality* of existing geographic datasets make the complex patterns that potentially lurk in the data hard to find. It is critical to develop new techniques to efficiently and effectively assist in deriving information from these large and heterogeneous spatial databases. Towards this goal, spatial data mining and knowledge discovery has been gaining momentum [14].

Clustering is to organize a set of objects into groups such that objects in the same group are similar to each other and different from those in other groups. While clustering is one of the most important tasks in data mining literature, spatial clustering has also long been used as an important process in geographic analysis. To identify clusters over *geographical* space, various approaches have been developed, based on Delaunay triangulation [7], density-based notion [6], random walks [10], and even a brute-force exhaustive searching method [15].

However, on one hand, existing spatial clustering methods can only deal with low-dimensional spaces (usually 2-D or 3-D space: 2 spatial dimensions and a non-spatial dimension). On the other hand, general-purpose clustering methods mainly deal with non-spatial feature spaces and have very limited power in recognizing spatial patterns that involve neighbors. Spatial dimensions cannot simply be treated as 2 additional non-spatial dimensions in general clustering methods because of two important reasons. First, the combination of spatial dimensions bears unique and real-world meanings, which can cause difficulties for general clustering methods [8]. Second, spatial clustering often adopts real-world dissimilarity measures, e.g. road distance, and considers complex situations, e.g. geographic obstacles [18], which are hard to integrate within high-dimensional clustering methods.

Moreover, to achieve both the efficiency and effectiveness for exploring very large spatial databases, a knowledge discovery system should have automated computational methods integrated with interactive visualization techniques that leverage the human expert’s knowledge and inference.

The objective of the research is to develop a novel approach to integrate spatial clustering information within the non-spatial attribute or feature space, and then to use this combined space for discovering high-dimensional spatial clusters with effective and efficient *computational clustering methods* and highly *interactive visualization techniques*. To achieve this objective, the research includes three parts of work. (1) Develop an interactive hierarchical

spatial clustering method that can identify arbitrary-shaped hierarchical 2-D clusters. The method can generate a spatial cluster ordering that fully preserves all hierarchical clusters, i.e., any set of points that constitute a cluster at some hierarchical level, will be contiguous in the 1-D ordering. By transforming hierarchical spatial clusters into a linear ordering, the integration of spatial and non-spatial information is made simpler since the spatial cluster structure is reduced to a single axis (a “common” attribute) in the feature space. (2) Develop a density- and grid-based hierarchical *subspace* clustering method that can identify multivariate clusters within a very large and high-dimensional data set. It is efficient because it first generalizes data into a small set of hyper-cells and then performs clustering with those cells. The spatial cluster ordering is then integrated with this subspace clustering method to identify multivariate spatial patterns. (3) Implement the above two methods in a fully open and interactive manner with support of various visualization techniques. The user can interactively control parameters of the clustering methods and see the immediate result. Several visualization techniques are developed to facilitate the human interaction and interpretation.

The rest of the paper is organized as follows. Section 2 gives a review on related research. Section 3 presents the interactive hierarchical spatial clustering. Section 4 introduces the hierarchical subspace clustering method. Section 5 briefly introduces the integrated approach for searching multivariate spatial patterns, with a working demo on census data. An expanded version of this paper with color figures and more details is available at: www.geovista.psu.edu/members/dguo/clustering.

2. RELATED RESEARCH

2.1 General-purpose clustering methods

Clustering methods can be divided into two groups: *partitioning* and *hierarchical* approaches (figure 1). The partitioning approach aims to divide the data set into several clusters, which may not overlap each other but together cover the whole data space. A data item will be assigned to the “closest” cluster based on a type of proximity or dissimilarity measure. Hierarchical clustering approaches decompose the data set with a sequence of nested partitions, from fine to coarse resolution.

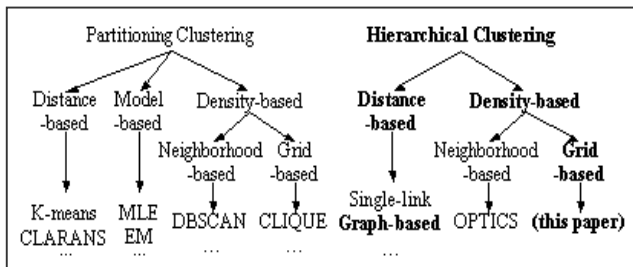


Figure 1: clustering methods (methods of bold font are used in this paper)

Within each group, according to their definitions of a cluster, clustering methods may also be classified into three sub-groups: *distance-based*, *model-based* (or *distribution-based*), and *density-based* methods (figure 1). Distance-based clustering methods need a distance or dissimilarity measurement, based on which they try to group those most similar objects into one cluster. *K-means* is a distance-based partitioning method, while graph-based methods [5]

can perform distance-based hierarchical clustering. *Model-based* clustering methods assume the data of each cluster conforms to a specific statistical distribution (e.g. Gaussian distribution) and the whole dataset is a mixture of several distribution models [5]. Maximum Likelihood Estimation (MLE) is an example. *Density-based approaches* regard a cluster as a dense region of data objects [13]. Density-based clustering can adopt two different strategies: grid-based or neighborhood-based. A grid-based approach divides the data space into a finite set of multidimensional grid cells and then groups those neighboring dense cells into a cluster. Such methods include CLIQUE [1], OptiGrid [11], etc. The key idea of neighborhood-based approaches is that, given a radius (ϵ), the neighborhood of an object has to contain at least a minimum number of objects (*MinPts*) to form a cluster around this object. Two representative methods are DBSCAN [6] and OPTICS [2]. Among density-based methods, only OPTICS can perform hierarchical clustering.

2.2 Hierarchical spatial clustering

Two groups of methods have been developed to detect hierarchical clusters with spatial data. The first group consists of those traditional hierarchical clustering methods, e.g., graph-based methods. AMOEBA is a Delaunay-based hierarchical clustering method for 2-D spatial data. It automatically derives a criterion function $F(p)$ as the threshold to cut off long edges and then recursively processes each sub-graph to construct a hierarchy of clusters. AMOEBA tries to avoid the single-link effect by detecting noise points and excluding them in any cluster. However, its criterion function is hard to justify/customize for different application data sets and tasks. With such a subjectively defined function, the result clusters are actually predetermined.

The second alternative for hierarchical spatial clustering is to use a density-based partitioning algorithm with different parameter settings. Extended from DBSCAN [6], OPTICS [2] is a neighborhood-based hierarchical clustering method (see figure 1). Given a “generating distance” (ϵ) and *MinPts*, OPTICS first identifies core objects and non-core objects. Core objects can be reached via other core objects, while non-core objects can only be reached via core objects (no connection allowed between non-core objects). After removing the connection between non-core objects, OPTICS works like a single-link method. In other words, it can only avoid the single-link effect at high levels (depending on the “generating distance”). OPTICS also relies heavily on index structures to speed up the *k*-nearest-neighbor query.

Both OPTICS and AMOEBA can only work well with low-dimensional data (the latter only works for 2-D points).

2.3 Subspace clustering methods

Subspace clustering (or projective clustering) is very important for effective identification of patterns in a high-dimensional data space because it is often not meaningful to look for clusters using all input dimensions. Some dimensions can be noisy and irrelevant, which may blur or even hide strong clusters residing in lower-dimensional subspaces. Traditional multi-dimensional scaling methods, e.g. principal component analysis (PCA) or self-organizing map (SOM) [5], have two major drawbacks [1,16]: 1) new dimensions (as linear combinations of the original dimensions) and result clusters are hard to interpret; 2) they cannot preserve clusters existing in different subspaces.

A subspace is formed by a subset of dimensions from the original high-dimensional data space. *Subspace clustering* is to identify subspaces within a high dimensional data space that allow better clustering of the data objects than the original space [1]. As a density- and grid-based subspace clustering method (figure 1), CLIQUE [1] partitions a subspace into multidimensional grid cells. The *selectivity* of a grid cell is the percentage of total data points contained in the cell. A cell is dense if its selectivity is greater than a density threshold τ . The sum of the selectivity of all dense cells in a subspace is the coverage of the subspace. CLIQUE first prunes candidate subspaces based on their coverage values. Then it tries to find clusters in each interesting subspace.

However, there are three severe drawbacks of existing subspace clustering methods. First, global parameters such as cell size or interval (ξ) and density threshold (τ), are very difficult to configure but can significantly influence the result. Secondly, existing methods cannot perform hierarchical clustering in each subspace and cannot adapt well to different application data sets and patterns of different scales. Third, they are open for interactive steering and exploration.

3. HIERARCHICAL SPATIAL CLUSTERING & ORDERING

Our method to hierarchical spatial clustering is efficient, achieving $O(n \log n)$ complexity without using any index structure, and fully supports interactive exploration of hierarchical clusters. It has the advantages of both AMEOBA and OPTICS. It is based on Delaunay triangulation (DT) and Minimum Spanning Tree (MST) and overcomes the single-link effect via singling out boundary points at various hierarchical levels for special treatment. Our method can also generate a 1-D spatial cluster ordering to preserve all hierarchical clusters and to encode spatial proximity information as much as possible. To simplify the description of the method, we first introduce the method without considering boundary points. Then a method is introduced for singling out boundary points and treating them differently.

3.1 Description of the Method

The input is a set of 2-D points $V = \{v_1, v_2, \dots, v_n\}$, where $v_i = \langle x, y \rangle$ is a location in the geographic space. Our clustering method takes 3 steps: 1) construct a DT and an MST from the DT, 2) derive an optimal clustering ordering; 3) visualize the cluster ordering and interactively explore the hierarchical structure.

3.1.1 Construct DT and MST

A Delaunay triangulation is constructed from the input points using the Guibas-Stolfi algorithm, which is of $O(n \log n)$ complexity [9]. The triangulation result (figure 2 and 3) is stored in memory with efficient access for: each point, each edge, end points of an edge and incident edges on a point. Each edge has a length, which is the dissimilarity between its two end points. Kruskal's algorithm [3], which is also of $O(n \log n)$ complexity, is used to construct an MST from the DT. Basically an MST is a subset of those edges in a DT.

3.1.2 Derive a cluster ordering

From the MST, an optimal ordering of all points can be derived to completely preserve the hierarchical cluster structure and other spatial proximity information. A cluster (connected graph) can be

viewed as a chain of points [19]. At the lowest level, each cluster (chain) contains a single point. Each chain has two end points (at the very beginning they are the same point). When two clusters are merged into one with an edge, the closest two ends (each from a cluster) are connected in the new chain (figure 2). All hierarchical clusters (points underscored by a line) are preserved (i.e., contiguous) in the 1-D ordering. Moreover, the ordering also preserves other spatial proximity information. For example, when D is merged to cluster $\{E, C, B, A\}$ with edge DC, it can be placed next to A or next to E in the ordering—either choice will equally preserve the cluster $\{D, E, C, B, A\}$. It is placed next to E rather than A in the ordering because $DE < DA$. Thus the proximity among D, E, and C is also preserved although they do not form a hierarchical cluster at any level.

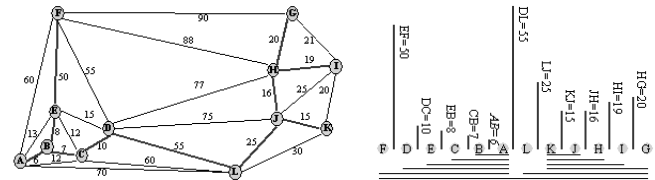


Figure 2: (Left) an MST $\{AB, BC, BE, CD, JK, HJ, HI, HG, JL, EF, DL\}$. Numbers indicate the length of each edge. (Right) The cluster ordering derived from the MST.

3.1.3 Visualization and interaction

Now we consider a larger data set (altogether 74 points). Its cluster ordering is shown in figure 3 (right top). The horizontal axis represents points (it is labeled *instances* because this visualization tool can also be used for non-spatial data set). The vertical axis represents the length of each edge. Each vertical line segment is an edge in the MST. Between two neighboring point in the ordering there is an MST edge. A cluster appears as a valley in the graph. Distinct clusters are separated by long edges (high ridges). Thus one can easily recognize the overall hierarchical structure. The second horizontal line (other than the bottom axis) is the threshold value for cutting off long edges. By interactively dragging this threshold line (bar), one can explore clusters at different hierarchical level.

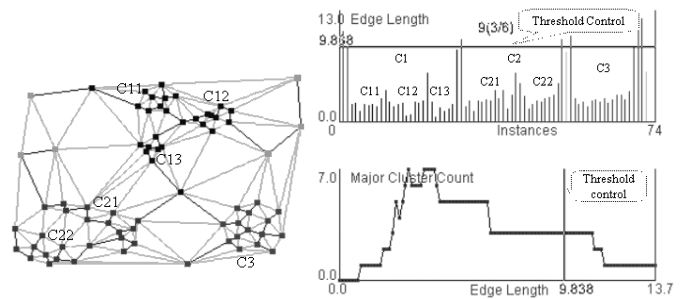


Figure 3: points, DT, MST (left), spatial cluster ordering (right top) and the trend plot (right bottom) ($MinClusSize = 3$).

A trend plot (figure 3—right bottom) is developed to visualize the relationship between a distance threshold and the total number of clusters in the data set. The horizontal axis represents values of length threshold and the vertical axis indicates the number of clusters. $MinClusSize$ is the minimal number of points that a cluster should have. The threshold can be interactively set via dragging the

vertical bar, which is linked with the horizontal bar in the cluster ordering (right top in figure 3): when you drag one, the other will move accordingly. Upon users’ interaction, the clustering result is shown immediately in the ordering, trend plot, and the map with a unique color for each cluster.

3.2 Tackling the Single-Link Effect

An MST-based clustering method can suffer from the single-link effect. As reviewed, AMOEBA and OPTICS both tried to avoid the single-link effect but the former cannot support a flexible hierarchical clustering while the latter can only avoid the single-link effect at high levels and rely heavily on an index structure.

We propose a Deviation-to-Minimum-Length (DML) measure to detect boundary points, which locate either on cluster boundaries (at various hierarchical levels) or on a line in a sparse area. For a point p , its DML value is calculated with the following equation.

$$DML(p) = \sqrt{\frac{\sum_{e=1}^N (L_e - \min)^2}{N}}$$

N is the number of edges incident to point p in the DT, L_e is the length of an edge incident to p , and \min is the length of the shortest edge incident to p . A high DML value indicates that the point locates on a boundary or a line—some neighbors are very close while others are far away. We also develop an interface for the user to interactively configure the threshold DML value and visualize those boundary points on a map.

We now name those non-boundary points as *core* points. In the improved MST, core points can only be connected through core points, i.e., on the path from one core point to another core point in the MST, no boundary point is allowed. Boundary points can connect to other boundary points or core points (figure 4). The new MST construction procedure remains an $O(\log n)$ time complexity (see expanded version for details).

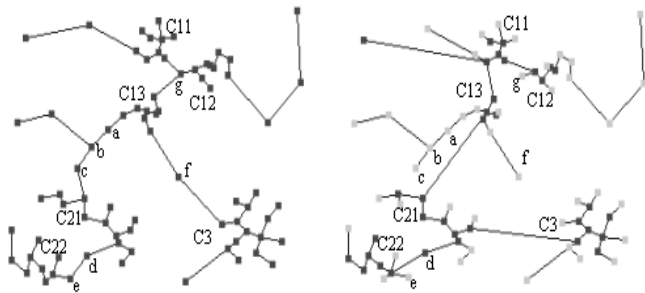


Figure 4: an MST (left) without consideration of boundary points and an MST (right) considering boundary points (gray). The single-link effect is avoided at various levels.

4. HIERARCHICAL SUBSPACE CLUSTERING

We develop a grid- and density-based approach for hierarchical subspace clustering, which is similar to CLIQUE but improved in several aspects. First, our approach uses a nested-mean discretization method instead of the equal-interval method used in CLIQUE. Second, an entropy-based evaluation method is adopted to rank subspaces before searching clusters in each of them. Third, by treating each multidimensional grid cell as a “point” and

calculating a synthetic distance measure between two “points”, the hierarchical spatial clustering method introduced above is extended to perform hierarchical subspace clustering. Fourth, our approach supports a fully interactive exploration of clusters.

4.1 Discretization of Each Dimension

There are many existing discretization (classification) methods for single-dimension data [17]. CLIQUE adopted the Equal-Interval (EI) method. We choose the Nested-Mean (NM) method (figure 5) to improve the effectiveness.

The EI approach cuts a dimension into a number of equal-length intervals. It often results in an extremely uneven assignment of data items to cells and fails to examine detailed patterns within a dense area. Extreme outlier values can severely affect the effectiveness of the EI approach. The NM approach calculates the mean value of a

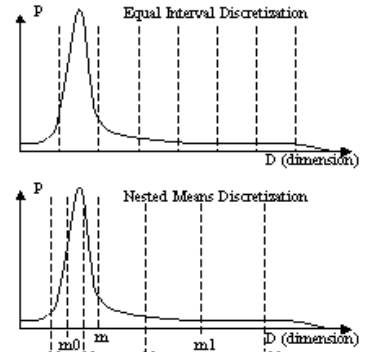


Figure 5: discretization methods

dimension data and cut the data into two halves with the mean value. Recursively, each half will be cut into halves with its own mean value. The process stops when the required number of intervals obtained. The NM discretization can examine detailed structures within a dense region and, at the same time, can capture coarse patterns in a comparatively sparse region.

The number of intervals (ni) needed for each dimension depends on the subspace size (d —the number of dimensions) and the data set size (n). A general rule adopted here is that $(ni)^d$ should roughly equal to n , i.e., ni should be around the value of $n^{1/d}$. For the nested-mean discretization, ni should also equal to 2^k (k is a positive integer). For example, if $d=4$ and $n=3800$, since $2^3 = 8$ and $8^4 = 4096$ (close to 3800), then ni should be 8.

4.2 Entropy-based Subspace Evaluation

A subspace clustering method needs an approach to evaluate subspaces and rank them according to their “interestingness”. We adopt an entropy-based evaluation criterion developed by Cheng and others for pruning subspaces [4]. The entropy of a grid-based subspace is calculated with the following equation.

$$H(x) = - \sum_{x \in \chi} d(x) \log d(x)$$

$H(x)$ is the entropy value of subspace X , which is a collection of grid cells. The density of a cell, $d(x)$, is the fraction of total data items contained in the cell. Based on the entropy value, all subspaces are ordered and listed. Thus the user can start from the top of this ranking to quickly locate important subspaces and hence significant patterns. Although this entropy-based approach can be used to prune “uninteresting” subspaces [4], in future we plan to implement a more flexible and robust pruning strategy with user interactions.

4.3 Synthetic Distance Between Two Cells

To find hierarchical clusters with a set of dense cells, a distance measure is proposed. We calculate a synthetic value (*SynVal*) for each interval within each cell based on its nominal position (*i*) among all intervals for the dimension, the interval bounding values— $[Min_i, Max_i]$, and the mean dimension value ($Mean_i$) of those data items contained by the cell.

$$SynVal = [(Mean_i - (Max_i + Min_i) / 2) / (Max_i - Min_i)] + i.$$

The *SynVal* of the same interval in different cells can be different due to different data items they cover. For easy explanation, let's consider a 1-D space, where each cell is defined by a single interval. The dimension is of range $[0, 100]$ and divided with the NM discretization into 4 intervals. Thus there are 4 cells, each of which is defined by a single interval. The synthetic value of each interval in each cell is shown in figure 6.

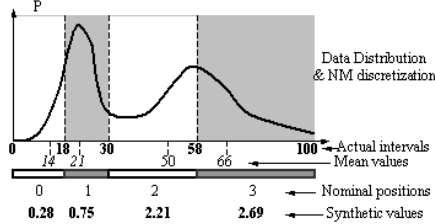


Figure 6: synthetic value for each interval

These synthetic interval values, which integrate both the global nominal ordering and the local numerical variance, can preserve the data distribution characteristics and be tolerant to extreme outlier values. With a vector of synthetic values of its constitutional intervals, each hyper-cell is defined as a high-dimensional "point". Then a synthetic distance is calculated for each pair of "points". The hierarchical spatial clustering method introduced above is extended here to perform hierarchical subspace clustering. In future, we will provide several more distance measures to let users choose and compare the results.

4.4 Interactive Subspace Clustering

To facilitate an interactive exploration and interpretation of the hierarchical subspace clustering process, a subspace chooser, a density plot, an HD cluster ordering and an HD cluster viewer (figure 7) are developed to cooperatively support a human-led exploration of hierarchical clusters in different subspaces.

A *Subspace Chooser* (right bottom in figure 7) is a visualization component that lists all subspaces ordered by their entropy values. The user needs to define a subspace size for the system to enumerate and evaluate all subspaces of that size. The constituent dimensions of each subspace and its entropy value are shown in the subspace chooser.

A *Density Plot* (right middle in figure 7) is a visualization component that helps the user understand the overall distribution of cell densities and

interactively control the density threshold via dragging the threshold bar (the horizontal line in the middle of the density plot) or typing in the number. The number right above the threshold bar is the coverage of the selected subspace according to current density threshold. For example, the current threshold is 1.015%, 16 dense cells (out of 199) have a density higher than 1.015% and altogether they contain 47.97% of all data items (coverage). The plot can be zoomed in or out for better views. The density plot can facilitate a proper configuration of the density threshold. Once the user set a new threshold, a new set of dense cells are passed to the HD cluster ordering component.

The *HD Cluster Ordering* (right top in figure 7) is similar to and extended from the spatial cluster ordering and visualization. The construction of an HD cluster ordering from those dense cells takes 4 steps: 1) construct a pair-wise distance (dissimilarity) matrix (since the number of dense cells is much smaller than the data set size *n*, this step will not cause a time complexity problem); 2) construct a hyper-MST from the distance matrix, 3) derive the HD cluster ordering, and visualize it for interactive control and exploration. This ordering can clearly show the hierarchical structure within the data and conveniently support dynamic browse of clusters at different hierarchical levels. During users' interaction, the immediate clustering result is visualized in the HD cluster viewer with each cluster of a different color.

The *HD Cluster Viewer* (left in figure 7) is based around the PCP (Parallel Coordinate Plot) that allows investigation of high dimensional spaces [12]. It is different from PCP in that it visualizes hyper-cells rather than the actual data. Each string (consisting of a series of line segments) represents a hyper-cell. The color and the width of the string represent the current cluster label and the density of the cell respectively. When the user interacts with the subspace chooser, HD density plot, or the HD cluster ordering, the HD cluster

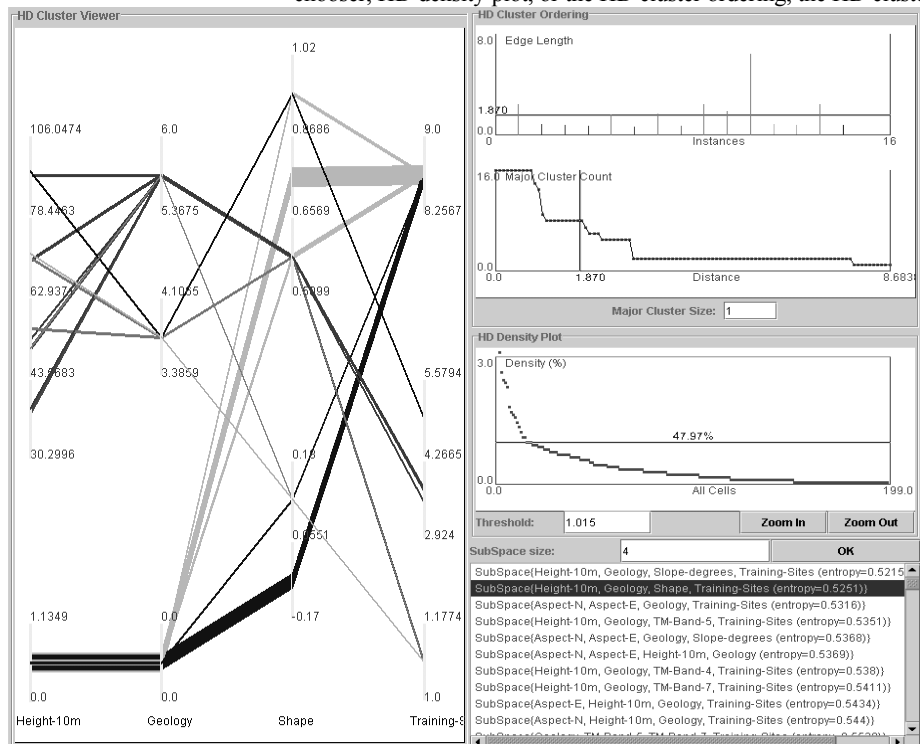


Figure 7: The ordered subspace list, HD density plot, HD cluster ordering, and the HD cluster viewer.

viewer will automatically be updated. Thus the clustering result associated with different input parameters can be immediately seen during interactions.

5. INTERACTIVE HIGH-DIMENSIONAL SPATIAL CLUSTERING

The spatial cluster ordering (derived in section 3) can be treated as a “common” attribute (we name it *SpaOrdering*) in the subspace clustering method introduced above for searching multivariate spatial clusters. If a subspace involves *SpaOrdering* as one of its dimensions and has a low entropy value, then this subspace has significant multivariate spatial clusters. Thus the spatial clustering information is integrated in but transparent to the subspace clustering process.

All related visualization components, including the map, boundary point identification interface, spatial cluster ordering, subspace list, density plot, HD cluster ordering and HD cluster viewer, are integrated together for coordinated visualization and exploration. When a subspace is selected and the density threshold is set, all points contained in dense cells are visualized in the map. During user interaction, clustering results are shown in both the map and the HD cluster viewer immediately and colored in the same way.

We have successfully applied the system to analyze census data of 2850 USA cities and a cancer data set from NCI (National Cancer Institute). Interesting patterns (both expected and unexpected) were found easily. See the expanded version for more details.

6. CONCLUSION & FUTURE WORK

This paper reports a framework and implementation of a hierarchical clustering approach to interactively and iteratively explore high-dimensional spatial data for multivariate patterns.

As an on-going development, the reported research will incorporate several desirable features in the future. First, currently our approach can process numerical data types (nominal data are treated as numerical data). It is not difficult to extend the system to address nominal data as well because the numerical data is actually first discretized into nominal intervals in the method and nominal data types are easier to discretize, but require a different distance measure. Second, a more efficient subspace pruning strategy is needed to eliminate “uninteresting” subspaces before constructing them. Third, more interactive controls are needed in the HD cluster viewer to let users interactively define clusters, focus/select a subset of data, and dynamically change the discretization of each dimension.

7. ACKNOWLEDGMENTS

This paper is partly based upon work funded by NSF Digital Government grant (No. 9983445) and grant CA95949 from the National Cancer Institute.

8. REFERENCES

- [1] Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P., Automatic subspace clustering of high dimensional data for data mining applications., *ACM SIGMOD'98*, Seattle, WA USA, 1998, pp. 94-105.
- [2] Ankerst, M., Breunig, M.M., Kriegel, H.-P. and Sander, J., OPTICS: Ordering Points To Identify the Clustering Structure., *Proc. ACM SIGMOD'99*, Philadelphia PA, 1999.
- [3] Baase, S. and Gelder, A.V., *Computer Algorithms*, 3rd edn., Addison-Wesley, 2000.
- [4] Cheng, C., Fu, A. and Zhang, Y., Entropy-based subspace clustering for mining numerical data., *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.
- [5] Duda, R.O., Hart, P.E. and Stork, D.G., *Pattern classification and scene analysis*, 2nd edn., Wiley, New York, 2000.
- [6] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., A density-based algorithm for discovering clusters in large spatial databases with noise., *the 2nd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Portland, Oregon, 1996.
- [7] Estivill-Castro, V. and Lee, I., Amoeba: Hierarchical Clustering Based On Spatial Proximity Using Delaunaty Diagram., *9th International Symposium on Spatial Data Handling*, Beijing, China, 2000, pp. 7a.26 - 7a.41.
- [8] Gahegan, M., On the application of inductive machine learning tools to geographical analysis, *Geographical Analysis*, 32 (2000) 113-139.
- [9] Guibas, L. and Stolfi, J., Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, *ACT TOG*, 4 (1985).
- [10] Harel, D. and Koren, Y., Clustering spatial data using random walks., *Proc. of the seventh ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, California, 2001.
- [11] Hinneburg, A. and Keim, D.A., Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering., *Proceedings of the 25th VLDB Conference*, Edingburgh, Scotland, 1999.
- [12] Inselberg, A., The plane with parallel coordinates, *The Visual Computer*, 1 (1985) 69-97.
- [13] Jain, A.K. and Dubes, R.C., *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs, NJ, 1988, 320 pp.
- [14] Miller, H.J. and Han, J., Geographic Data Mining and Knowledge Discovery: an overview. In H.J. Miller and J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery*, Taylor & Francis, London and New York, 2001, pp. 3-32.
- [15] Openshaw, S., Charlton, M., Wymer, C. and Craft, A., A Mark 1 Geographical Analysis Machine for the automated analysis of point data sets, *International Journal of Geographical Information Science*, 1 (1987) 335-358.
- [16] Procopiuc, C.M., Jones, M., Agarwal, P.K. and Murali, T.M., A Monte Carlo Algorithm for Fast Projective Clustering., *ACM SIGMOD'2002*, Madison, Wisconsin, USA, 2002, pp. 418-427.
- [17] Slocum, T.A., *Thematic cartography and visualization*, Upper Saddle River, N.J. : Prentice Hall, 1999, 293 pp.
- [18] Tung, A.K.H., Hou, J. and Han, J., Spatial clustering in the presence of obstacles., *The 17th International Conference on Data Engineering (ICDE'01)*, 2001.
- [19] Vandev, D. and G.Tsvetanova, Y., Perfect chains and Single Linkage Clustering Algorithm., *Statistical Data Analysis, Proceedings SDA-95*, 1995, pp. 99-107.