Multiscale Analysis Of Data: Clusters, Outliers and Noise - Preliminary Results

Chetan Gupta
Dept Of Mathematics Statistics and Computer Science, University of Illinois, Chicago

Robert Grosssman
National Center For Data Mining, University of Illinois, Chicago

Abstract

Large, complex geospatial data sets often have structure at different scales. Many common methods for identifying clusters and structures work at a single scale. In this note, we introduce a simple method for identifying clusters and outliers at different scales. The basic idea is to decompose feature space into a hierarchical set of cubes. This is done recursively. First the data set as a whole is placed in a cube, which is cube at scale 1. Two or more cubes at scale k are derived from a cube at scale $k-1$ by dividing the larger cube into smaller cubes obtained by bisecting one or more dimensions. Cubes at smaller and smaller scales are computed until a stopping criterion is satisfied to create a dyadic decomposition. From this dyadic decomposition, muliscale clusters, outliers, and related structure can be easily identified. In this paper, we introduce this approach for identifying clusters and outliers and provide some preliminary experimental results. This approach appears to be promising for complex, multidimensional geospatial data sets.

1 Introduction

Large geospatial data sets often have structure at different scales, which can be missed by many standard approaches for identifying clusters and outliers. In this work, we present a simple multiscale approach to identifying clusters and outliers and provide some preliminary experimental results.

Here is an outline of our algorithm. First, we enclose the n-dimensional data set in a cube. The *scale* of this initial cube is defined to be one. For $k > 1$, we divide a cube whose scale is k, into 2 or more cubes whose scale we define to be $k+1$, by bisecting one or more edges of the larger cube. This produces 2^c new cubes, where the number of bisections c satisfies: $1 \leq c \leq n$.

We continue this process until a stopping criterion is reached, such as a cube contains fewer than a specified number of points, say, N_{stop}, which may be scale dependent in the sense that $N_{\text{stop}} = N_{\text{stop}}(k)$. This produces a *dyadic decomposition*.

Given a dyadic decomposition, we divide cubes into three types. If we fix a threshold N_{noise}, then adjacent cubes with more than N_{noise} points can be linked together to define clusters. Adjacent cubes to these cluster cubes with fewer than N_{noise} are defined to be *noise cubes.* Finally, points in other cubes are defined to be *outliers.* More detail about this is provided below.

Our experimental studies are based on data sets described in Cure [10] and Chameleon [5]. We want to thank the authors of these papers for providing us with these data sets.

2 Related Work

There is a vast literature in clustering and [1] is a good reference.

Outlier detection is a well studied problem in both statistics and data mining. We look here at distance based approaches to data mining. In [6] a point p is defines as an outlier with respect to parameter k and λ, if no more than k points are a distance λ or less from p. Later on we will state a lemma, which shows a similarity between ours and Knorr's definition. In [11] an outlier is defined as: Given a k and a n, a point p is an outlier if the distance to its $k-th$ nearest neighbor is smaller than the corresponding value for no more than $n-1$ other points. In [8] every point is given a LOF (Local Outlier Factor), which measures how isolated an object is from its surrounding. They assign a degree to every point of being an outlier.

Wavelets are a common technique for multiresolution analysis. In our context, WaveCluster [4]is a grid based clustering method which uses wavelet transform to filter the data. Scale-based clustering has been presented before. In [3], Radial Basis Function Network(RBFN) are used for scale-based clustering. In [15], an approach inspired by statistical mechanics is described, where the temperature is the scaling parameter. Another interesting work along similar lines is [16], which attempts to provide a unified framework for various scale space approaches. [9] uses a scale-based smoothing function to estimate probability density function. Our algorithm shows similar properties to these scale-based algorithms.

An early grid based clustering algorithm is [14]. They address the issue of "neighbors" in a grid setting. Two more grid based techniques are Bang [13] and GRIDCLUST [12], which also create hierarchical clustering using grids. In fact [12] combines [14] and [2]. GRIDCLUST is also in a scale-based grid clustering algorithm. Their grid structure is based on a k-d tree. In GRIDCLUST, first the cells are arranged in order of their density and merged in that order and clusters at different scale can be merged too. It is a bottom up approach. Our grid structure is different and we go in top down approach where the idea is to study each scale and see if certain parts of data should be studied at that or a finer scale. But most importantly its the philosophy, GRIDCLUST is a clustering algorithm and they explicitly do not recognize the idea that difference parts of data should be looked at different scale. DBSCAN [7] is also a similar approach, but it is primarily a clustering algorithm and is not a multiscale dyadic cube approach.

3 Multiscale Analysis With Cubes as Units

As outlined above, our approach to the multiscale analysis of clusters and outliers is based upon bisecting cubes. We begin with some notation and definitions.

Definition 1 Dyadic decompositions *are defined recursively as follows. Let the data set be $D \subseteq R^n$. Let $|D| = N$. First, we enclose the n-dimensional data set D in a cube. The* scale *of this initial cube is defined to be one. We define additional cubes in our decomposition recursively as follows. For $k > 1$, we divide a cube whose scale is k, into 2 or more cubes whose scale we define to be $k+1$, by bisecting one or more edges of the larger cube. This produces 2^c new cubes, where the number of bisections c satisfies: $1 \leq c \leq n$.*

We continue this process until a stopping criterion is reached, such as a cube contains fewer than a specified number of points, say, ϵ, which may be scale dependent in the sense that $\epsilon = \epsilon(k)$. This produces a *dyadic decomposition.*

For identifying a set of points or clusters some notion of adjacency or neighborhood is required.

Remark 1 *We sometimes call a cube whose scale is k a k-cube.*

Definition 2 *Fix a threshold N_{noise}. A cube is a* noise cube *in case it contains fewer than N_{noise} points and is adjacent to some other cube with more than N_{noise} points.*

Remark 2 *A square in the plane $\mathbf{R^2}$ has at most 8 adjacent dyadic squares. A cube in $\mathbf{R^3}$ has at most 26 possible adjacent dyadic cubes. More generally, a hypercube in $\mathbf{R^n}$ has at most $3^n - 1$ adjacent hypercubes.*

Using this notion of adjacent cubes, we now define a neighborhood relationship.

Definition 3 *Two points are* neighbors *iff:*

1. *They belong to the same k-cube or*
2. *They belong to adjacent k-cubes and neither cube is a noise cube.*

Now that we have definitions for two points being neighbors, a cluster can be defined. A cluster is either a singleton or a set of points such that any two points in the set are either neighbors or have an neighborhood relationship through a series of points. We consider all points having a neighborhood relationship to lie in the same cluster.

Definition 4 *A* cluster *at scale k is precisely a set S of those points $x \in D$ such that:*

1. *Either $S = \{x\}$ is a singleton, i.e. there is no $y \in D$ such that x, y are neighbors OR*
2. *or $\forall y \in S, y \neq x$*
   - (a) *Either x and y are neighbors.*
   - (b) *or there exists a sequence of points $\{z_1, \ldots, z_n\} \in S$ such that,*
     x, z_1 are neighbors
     z_1, z_2 are neighbors
     $\vdots$
     z_{n-1}, z_n are neighbors
     z_n, y are neighbors

Points belonging to certain type of clusters are called outliers. Fix a threshold N_{outlier} and consider clusters defined by neighboring cubes at scale k.

Definition 5 *If a cluster consiting of neighboring cubes at scale k contains fewer than N_{outlier} points, then it is called an outlier cluster and its points are called outliers.*

4 The Algorithm

Clusters Derived from Dyadic Decompositions (CDDD)

1. Fix constants N_{stop}, N_{noise}, N_{outlier} and N_{merge}.
2. Fix a data set $D \subseteq \mathbf{R}^{\mathbf{n}}$.
3. Fit a cube around the data set and recusively bisect the sides in the cube as described above until the number of points in each cube is less than N_{stop}.
4. Identify those cells that have fewer than N_{noise} points. The points in these cells are noise points.
5. Compute the clusters by linking together neighboring cells containing more than N_{noise} points, as described above. The points in these clusters are cluster points.
6. Merge two or more clusters if the clusters are separated by noise cells and each cluster individually contains fewer than N_{merge} points.
7. Identify those clusters with fewer than N_{outlier} points. The points in these clusters are outliers.

4.1 Stopping Criteria and Merging Criteria

A variety of stopping criteria can be used, including:

- *A Threshold Min Size*: Specify a minimum size of a cluster. Once a cluster is smaller than the threshold size, do not further subdivide it.
- *A Max Scale Value*: Specify the smallest scale we are intersted in. This approach also limits the cost of the computation. computations. In our experiments we never had to go higher than a scale of 25.

When clustering with noise, the noise points are used to ensure that two clusters not be regarded as one when joined by a series of noise points. This also leads to legitimate clusters being broken into smaller clusters; hence, it is important to merge small clusters separated by noise points.

A variety of merging criteria can also be used, the most useful of which we found to be:

- Merge clusters separated by noise clusters that are smaller than a specified minimum size N_{merge}. When merging clusters, we recursively assign noise points to the nearest cluster.

5 Experimental Results

5.1 Synthetic Data Sets

To demonstrate properties of multiscale analysis, we created five artificial data sets in two dimensions. In all the data sets, a cluster is a set of points uniformly distributed and circular in shape. (i) This is a set with 1200 points. There were 5 clusters of 198 points each and of equal radius. There were 4 smaller clusters of 50 points each which lie on the 4 corners of a rhombus. There are 10 points which are distributed randomly. (ii) This is a set of 19000 points. 6 clusters in all. A set of 9000 points, with radius 100, a set of 100 points with radius 100, a set of 3000 points with radius 50, a set of 6000 points with radius 10, a set of 500 points with radius 20 and a set of 400 points with radius 70. (iii) This is a set of 600 points. There were 5 clusters of 98 points each and of equal radius. 9 smaller clusters of 10 points each lie on a rhombus. There are 10 points which are distributed randomly. (iv) A set of 1000 randomly distributed points. (v) A set of 6 clusters of 200 points each of equal width.

In the clustering experiments, the number of clusters rises slowly. The number of clusters somewhat stabilizes when the numbers of clusters are same (or in the ballpark) as the number of clusters originally in the data (by construction). The number of clusters rises rapidly then, as their is not much structure to be discovered. Data set 1 and 3 are good examples of how multi-scaling is relevant in more complex data sets. In the first data set, from scale 7 to scale 8 one large cluster breaks into the four 10 point clusters. Same in data set 3, where 9 clusters of size 10 are obtained from a single cluster. In data set 2, from scale 4 till 7, despite the varying sizes and densities we get the right number of clusters and then lesser dense clusters begin to disintegrate. In fact, the highly dense cluster of 6000 points breaks up only at scale 13. In data set 4, since there is no structure the graph clearly shows that number of clusters rises continuously. Data set 5, is a regular data set and behaves as expected. In data set 1 and 3 all the 9 points are identified as outliers. We have plotted the number of clusters as a function of scale in figure 1

We have also shown a sample multiscale clustering in figure 2. The data is borrowed from Cure [10] data set.

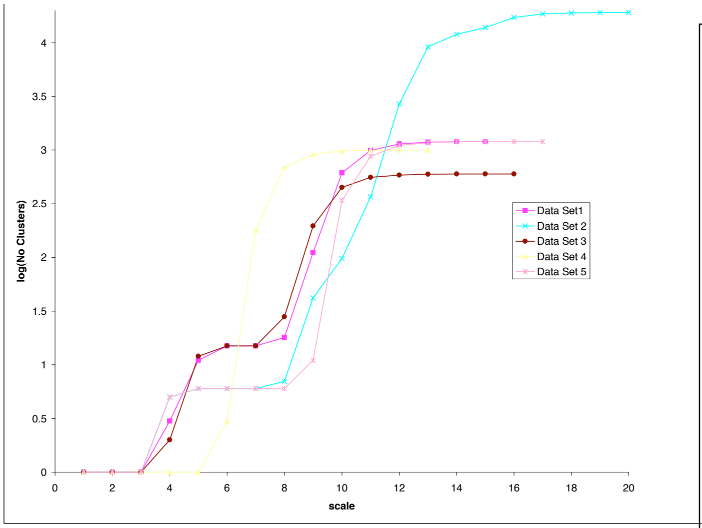

Figure 1: This is a plot showing the relationship between the scale and the number of clusters for several synthetic data sets from the publication [10].

5.2 Forest Cover Data

We next tried our algorithm on the forest cover data set from the UCI Data Mining Achive. This data contained forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. The original data set contained 54 attributes, out of which we picked the seven continuous ones, representing seven different types of forest types. The examples were labelled. We removed the labels and tried to see if different types of forest types would separate out into own clusters. Forest of type 3 and type 4 were well separated by the algorithm, with some overlap with forest type 6. Forest type 7 was also well separated by the algorithm. Forest Type 1 and type 2 were also separated, but their clusters overlapped with each other. Forest Type 1 (the predominant type) was separated, but had some overlap with other clusters.

5.3 NASA Shuttle Data

We also tried clustering data from the shuttle. This data has 7 classes and 14500 trained examples in 9 dimensions. There are 11478 examples of class 1, 12 examples of class 2, 38 examples of class 3, 2155 examples of class 4, 807 examples of class 5 and 4 examples of class 6 and 2 of class 7. This is an intersting data set for us, since several of the classes have very small representation and can be understood as outliers.

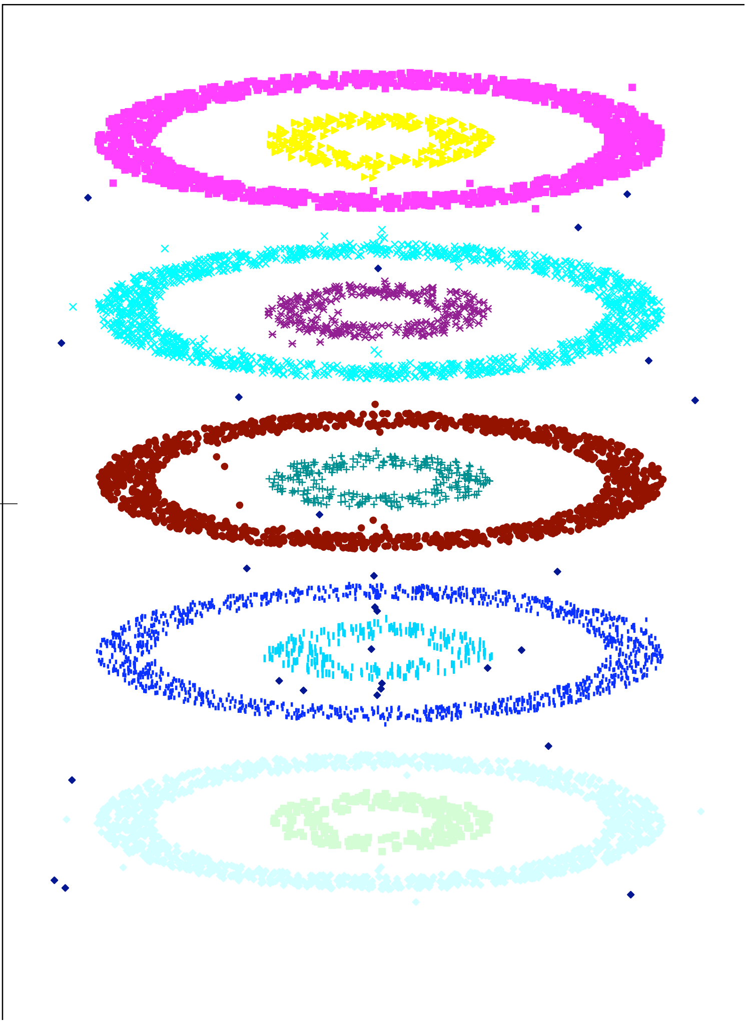

Figure 2: This figure shows the result of clustering data used in the study [10].

At scale 3, there are 3 clusters, one of 14998 points and two singletons. At scale 4, the cluster with 14998 points splits into a large cluster of 14887 and the rest are singletons. Three of the 4 class 6 points split out as singletons. At scale 5, we have 14 clusters. A cluster of size 590 splits off from the cluster with 14995 points. This cluster contains 588 points of class 5 and 2 points of class 3. A smaller cluster of 4 points only containing class 5 splits out. At scale 6 the large cluster breaks up into a cluster of size of 212 containing only points of class 5. All class 7 points also break out as outliers. By scale 6 all the points of 5, 6 and 7 are no longer part of the large cluster. At scale 7, members of class 2 and 3 split out from the large clusters. At 8 we start getting numerous clusters containing either elements of class 4 or class 1. The algorithm terminates at scale 12.

6 Conclusions

In this paper, we have shown how dyadic decompositions lead naturally to multiscale identification of clusters, noise points, and outliers. We presented experimental studies with synthetic data sets that have been used by previously by researchers studying clustering. We have also presented preliminary experimental studies showing that this approach might be well suited for geospatial data sets.

References

[1] A. K. Jain, M. N. M., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*, 264–323.

[2] Broder, A. J. (1990). Strategies for efficient incremental nearest neighbor search. *Pattern Recognition*, *23*, 171–178.

[3] Chakravarty, S., & Ghosh, J. (1996). Scale based clustering using the radial basis function network. *IEEE Transcations on Neural Networks.*

[4] G. Sheikholeslami, S. C., & Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach to very large databases. *Proceedings of the* 24^{th} *VLDB Conference, VLDB '98.*

[5] George Karypis, E.-H. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, *32(8)*, 68–75.

[6] Knorr, E., & Ng, R. (1998). Algorithms for mining distance based outliers in large datasets. *Proceedings of International Conference on Very Large Databases, VLDB '98*, 392–402.

[7] M. Ester, H.-P. Kriegel, J. S., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. of the Second Intl Conference on Knowledge Discovery and Data Mining, Portland, OR.*

[8] M.M. Breunig, H. Kriegel, R. N., & Sander, J. (2000). Lof: Identifying density based local outliers. *In Proceedings of the ACM International Conference Management Of Data*, 93–104.

[9] Roberts, J. S. (1997). Parametric and nonparametric unsupervised cluster analysis. *Pattern Recognition*, *30*, 261–272.

[10] S. Guha, N. Mishra, R. M., & O'Callaghan, L. (2000). Clustering data streams. *In the Annual Symposium on Foundations of Computer Science, IEEE.*

[11] S. Ramaswamy, R. R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *In Proceedings of the ACM International Conference Management Of Data,SIGMOD '00*, 427–438.

[12] Schikuta, E. (1996). Grid clustering: A fast hierarchical clustering method for very large data sets. *In Proceedings 13th International Conference on Pattern Recognition*, *2*, 101–105.

[13] Schikuta, E., & Erhart, M. (1997). The bang clustering system: A grid based data analysis. *In Proceedings Advances in Intelligent Data Analysis, Reasoning About Data*, 2^{nd} *International Symposium*, 513–524.

[14] Warnekar, C. S., & Krishna, G. (1979). A heuristic clustering algorithm using union of overlapping pattern cells. *Pattern Recognition*, *11*, 85–93.

[15] Won, Y. F. (1993). Clustering data by melting. *Neural Computation*, *5:1*, 89–104.

[16] Y. Leung, J.-S. Z., & Xu, Z.-B. (2000). Clustering by scale-space filtering. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, *22*.