

A Multiscale Algorithm for Dimension Reduction & Clustering

Lionel F. Lovett, II
Jackson State University

Research Alliance in Math and Science
Computer Science and Mathematics Division
Mentors: George Ostrouchov and Houssain Kettani

http://www.csm.ornl.gov/Internships/rams_06/abstracts/l_lovett.pdf

Abstract

RobustMap is a dimension reduction technique for large multidimensional data sets that includes identification and extraction of outlying subsets at multiple scales. It operates on distances between objects. While checking the distances against a given relative threshold, it searches for a split between clusters. If the distance distribution is bimodal and satisfies a statistical test, the clusters are separated and dimension reduction restarts within each new cluster. Because cluster separation can occur at any stage of the dimension reduction process, clusters at multiple scales can be accessed. When no additional clusters are found, a complete set of dimension reduction axes is computed. The process decomposes the data into a reduced hierarchical multiscale structure. Given data on n objects in p dimensions, reduction to k ($k \ll p$) dimensions is of time complexity $O(knp)$.

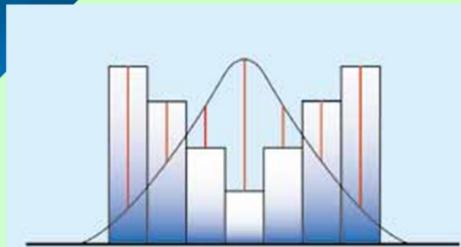


Fig 7. Bimodal Distribution:
Normal distribution vs. Bimodal

RobustMap's Processes

1. Compute n distances from the first object
 - Take point of largest distance
 - Repeat
2. Clustering using extreme points
 - Create list of difference of distance
 - Find threshold based on parameter
 - Find ratio of actual to expected distances for right and left of threshold exclusively (Figure 7)
 - Assign points to respected cluster (Figure 6)
3. Finish projection on remaining objects when no split of clusters occurs

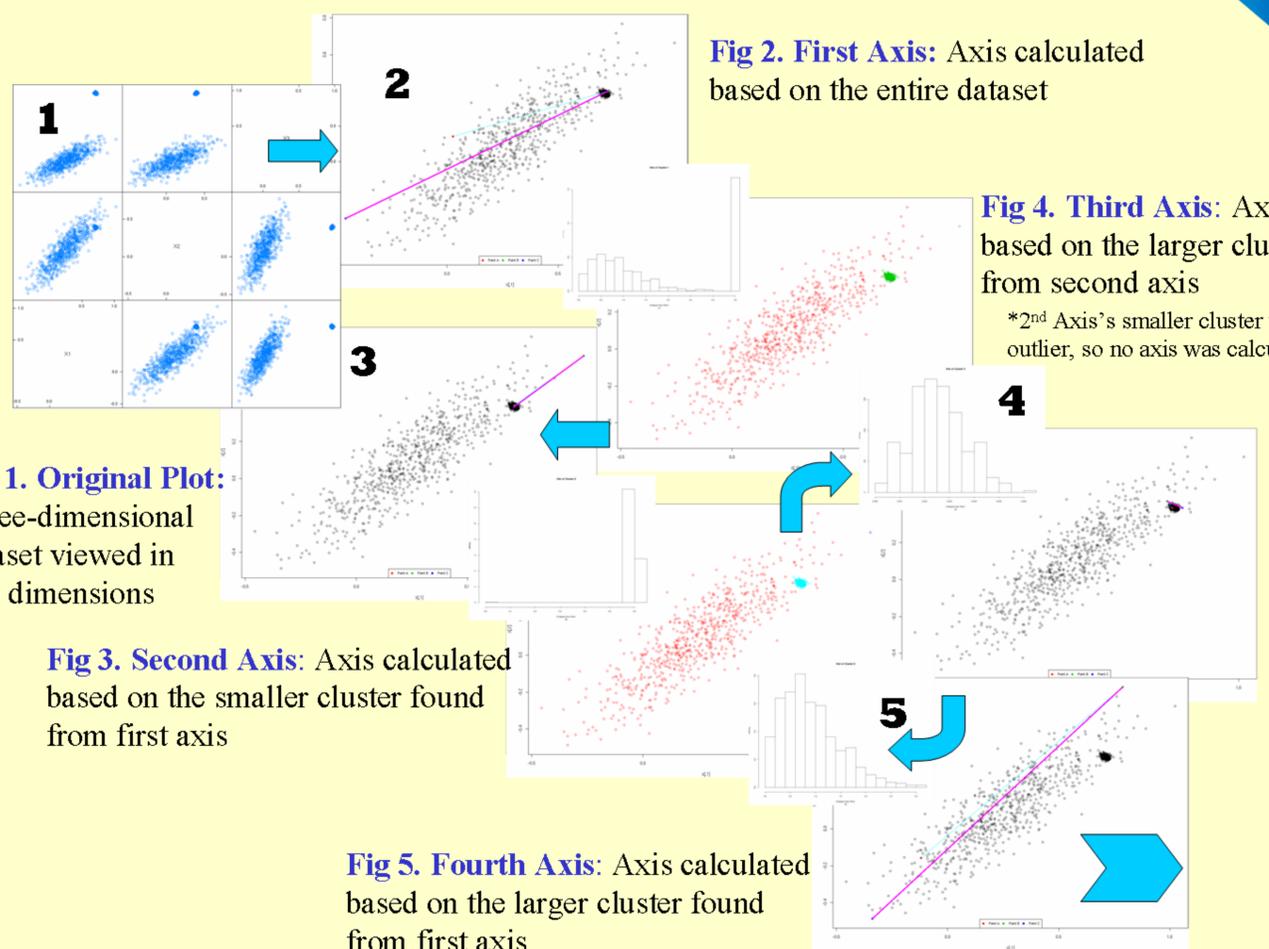


Fig 1. Original Plot:
Three-dimensional dataset viewed in two dimensions

Fig 2. First Axis: Axis calculated based on the entire dataset

Fig 3. Second Axis: Axis calculated based on the smaller cluster found from first axis

Fig 4. Third Axis: Axis calculated based on the larger cluster found from second axis
*2nd Axis's smaller cluster was an outlier, so no axis was calculated.

Fig 5. Fourth Axis: Axis calculated based on the larger cluster found from first axis

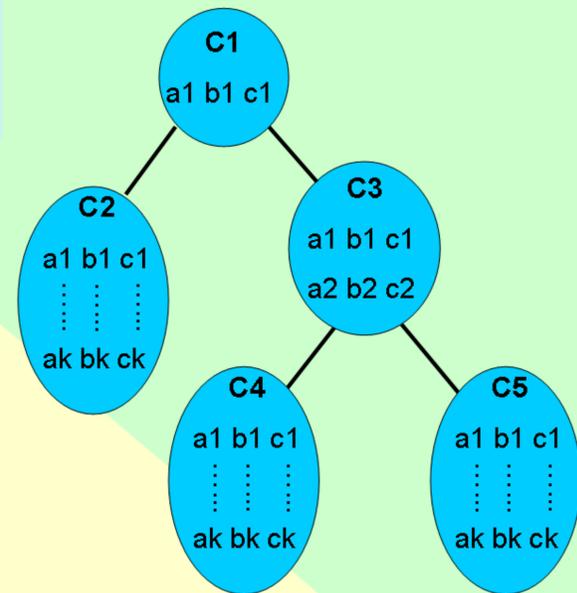


Fig 6. Cluster Distribution:
Leaves are final clusters

Cluster Index	Left	Right	Parent
1	2	3	0
2	0	0	1
3	4	5	1
4	0	0	3
5	0	0	3

Table Cluster Distribution: Record of how clusters split according to hierarchy

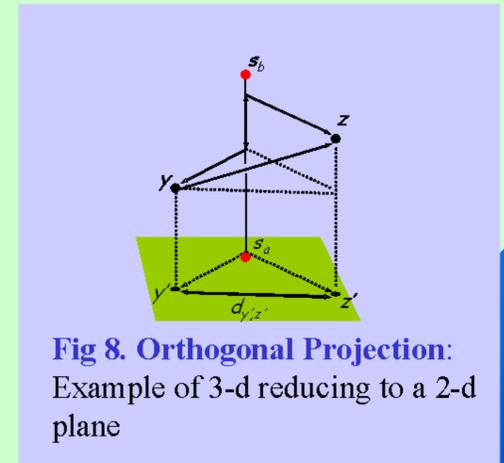


Fig 8. Orthogonal Projection:
Example of 3-d reducing to a 2-d plane