The INCLude (InterNodal Complete Linkage) Hierarchical Clustering Method

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

David Allen Olsen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

February, 2015

Dedication

This thesis is dedicated to people who genuinely care to do their jobs well and then some. It is dedicated to the people who gave me valuable counsel or helped me with my research or in other ways that kept my program moving forward, without expecting undue recognition or extra remuneration, and without an agenda or attaching strings. It is dedicated to those who seek the truth, even at the expense of learning that their original beliefs were wrong. It is especially dedicated to those whose character and integrity so shown out that they became role models along the way. To you, I am forever grateful, and promise to pass along to others what you did for and gave to me.

Abstract

This thesis presents each part of a three-part research project. The goal of this project was to develop a general, complete linkage hierarchical clustering method that 1) substantially improves upon the accuracy of the standard complete linkage method and 2) can be fully automated or used with minimal operator supervision. This project differs from main stream machine learning projects in that it elevates accuracy to equal importance with complexity and generalization.

For the first part of the project, a new, complete linkage hierarchical clustering method was developed. The INCLude (InterNodal Complete Linkage) hierarchical clustering method was designed with small-n, large-m data sets in mind, where n is the number of data points, m is the number of dimensions, and "large" means thousands and upwards. It unwinds the assumptions that underlie the standard complete linkage method. Instead of intercluster distances, interpoint distances are used to construct clusters; clusters can overlap; and data points can migrate between clusters. Further, evaluating pairs of data points for linkage is decoupled from constructing cluster sets, and cluster sets are constructed *de novo* instead of updating previously constructed cluster sets. Thus, it is possible to construct only the cluster sets for select, possibly noncontiguous levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence. The new clustering method has simplicity. Yet, it is consonant with the model for a measured value that scientists and engineers commonly use.

However, by unwinding the assumptions that underlie the standard complete linkage method, the size of a hierarchical sequence reverts back from n levels to $\frac{n\cdot(n-1)}{2}+1$ levels, and the time complexity to construct cluster sets becomes $O(n^4)$. This is large even for small-n, large-m data sets. Moreover, the *post hoc* heuristics for cutting dendrograms are not suitable for finding meaningful levels or meaningful cluster sets of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence. Thus, with today's technology, the project went back more than 60 years to solve a problem that could not be solved then. For the second part of the project, a means was developed for finding meaningful levels of an $\frac{n\cdot(n-1)}{2}+1$-level (complete linkage) hierarchical sequence *prior* to performing a cluster analysis. The means includes constructing at least one distance graph, which is

visually examined for features that correlate with meaningful levels of the corresponding hierarchical sequence. By finding meaningful levels of an $\frac{n\cdot(n-1)}{2}$ + 1-level hierarchical sequence *prior* to performing a cluster analysis, it is possible to know which cluster sets to construct and construct only these cluster sets. This reduces the time complexity to construct cluster sets from $O(n^4)$ to $O(ln^2)$, where l is the number of meaningful levels. *These are the cluster sets that can have real world meaning.* It is notable that the means does not rely on dendrograms or *post hoc* heuristics to find meaningful cluster sets. The second part also looked at how increasing the dimensionality of the data points helps reveal inherent structure in noisy data, which is necessary for finding meaningful levels.

The third part of the project resolved how to mathematically capture the graphical relationships that underlie the above-described features and integrate the means into the new clustering method. By doing so, the new method becomes self-contained and incurs almost no extra cost to determine which cluster sets should be constructed and which should not.

During the exposition of this thesis, four common misconceptions about the standard complete linkage method are uncovered. First, clusters constructed by the standard complete linkage method are not always maximally complete. Second, if a data set has inherent structure, the number of clusters in a meaningful cluster set is ascertainable as an artifact of cluster set construction. Third, it *is* possible to determine whether a data set has inherent structure and whether the data points can be grouped into clusters. Fourth, clusters do not have to be isolated in order to construct accurate hierarchical sequences of cluster sets for Euclidean distance or city block distance.

Empirical results from nine experiments are included. The first two experiments corroborate the theoretical work described in Chapter 5. The next three experiments show that the new clustering method performs well with respect to different kinds of data sets, including a data set that has no inherent structure, a data set comprised of nonmetric data, and a data set comprised of multiple attributes. The last four experiments show that the new clustering method compares favorably with the standard complete linkage method and k-means clustering with respect to accuracy and time. Further, they show that cluster sets constructed for meaningful levels can have real world meaning.

Contents

Dedication **i**

Abstract **ii**

List of Figures **vii**

1 Introduction **1**

1.1 Background 2

1.2 Problem Definition 6

2 Related Work **10**

2.1 Flat and Hierarchical Clustering Methods 10

2.2 The Standard Complete Linkage Method 12

2.3 Clique Detection Methods 19

2.4 Schemes for Cutting Dendrograms 19

2.5 Other Related Work 20

3 Contributions **21**

3.1 INCLude Hierarchical Clustering: A Hierarchical Clustering Method Based Solely on Interpoint Distances 21

3.2 Means for Finding Meaningful Levels of a Hierarchical Sequence *Prior* to Performing a Cluster Analysis 22

3.3 Closing the Loop on a Complete Linkage Hierarchical Clustering Method 23

3.4 Usefulness 24

4 INCLude Hierarchical Clustering: A Hierarchical Clustering Method Based Solely on Interpoint Distances 25

4.1 Design . . . 26

4.1.1 Design Decisions . . . 26

4.1.2 Overview of the INCLude Hierarchical Clustering Method . . . 28

4.1.3 Data Structures . . . 32

4.2 Pseudocode . . . 36

4.2.1 Pseudocode 1: Evaluating Ordered Triples for Linkage . . . 36

4.2.2 Pseudocode 2-4: Cluster Set Construction . . . 39

4.2.3 Cluster Set Construction Subproblems . . . 47

4.3 Theorems and Proofs for the INCLude Hierarchical Clustering Method and Algorithm . . . 52

4.3.1 Transformational Invariance . . . 52

4.3.2 Optimality . . . 53

5 Means for Finding Meaningful Levels of a Hierarchical Sequence Prior to Performing a Cluster Analysis 55

5.1 Noise Attenuation . . . 56

5.2 Finding Meaningful Levels and Cluster Sets . . . 61

5.3 Theorems and Proofs for Finding Meaningful Levels of a Hierarchical Sequence . . . 68

5.3.1 Calculating the Variance of $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ When $Y_k \sim N(0, \sigma_k^2)$ 68

5.3.2 Calculating the Variance of $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ When $Y_k \sim N(\mu_k, \sigma_k^2)$ . . . 72

5.3.3 Calculating the Variance of $Z_m = \sum_{k=1}^{m} |Y_k|$ When $Y_k \sim N(0, \sigma_k^2)$ 76

5.3.4 Calculating the Variance of $Z_m = \sum_{k=1}^{m} |Y_k|$ When $Y_k \sim N(\mu_k, \sigma_k^2)$ 78

6 Closing the Loop on a Complete Linkage Hierarchical Clustering Method 83

7 Empirical Experiments 88

7.1 Metrics for Evaluating the INCLude Algorithm . . . 88

7.2 Experiments . . . 90

7.2.1 Sensitivity Analysis 90
7.2.2 Monte Carlo Simulation 101
7.2.3 Structureless Data Sets 109
7.2.4 Ethnic Marriages Data Set 112
7.2.5 Residential Heat Pump Data Set 115
7.2.6 17-Points Geometric Pattern Data Sets 117
7.2.7 Synthetic Gene Expression Data Sets 126
7.2.8 Motes Sensing Luminescence Data Set 132
7.2.9 Sinus Rhythm Data Sets 137

8 Conclusion **140**

References **142**

List of Figures

1.1 Schematic of an automated process. Sequential, feedback, and/or feedforward control are included in the lowest level of control while monitoring, planning, prediction, decision making, and adaptation are included in higher levels of control. Adapted from [1]. 4

2.1 Survey of flat clustering methods. 13

2.2 Survey of hierarchical clustering methods. 14

2.3 Dendrogram, proximity vector, and state matrices for Example 1. 16

4.1 How the classical migration problem is resolved by INCLude hierarchical clustering. The different colored (gray scaled) numerals in the state matrices indicate which data points belong to the same cluster. 28

4.2 Activity diagram for agglomerative INCLude hierarchical clustering. . . 29

4.3 Cluster patterns encountered by the new clustering method. The dots represent data points, and the numbers represent the degrees of the respective data points. 31

4.4 Seven of the nine data structures that are used in the new clustering method. The other two data structures are *proxVector* and *clusterTree*. 34

4.5 Illustration of a cluster set comprised of six clusters and their representation as a *clusterTree*. The dots represent data points, and the numbers represent the global indices of the respective data points. 35

4.6 Ordered triples and data structures from a sensor system experiment similar to that described in Chapter 7. Euclidean distance was used to calculate the distances. 40

4.7 Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 6. Euclidean distance was used to calculate the distances. 45
4.8 Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 27. Euclidean distance was used to calculate the distances. 46
4.9 Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 12. Euclidean distance was used to calculate the distances. 49
4.10 Continuation of Fig. 4.9. 50
4.11 Trace for rank order index = 54 of the 17-points geometric pattern experiment described in Chapter 7. Euclidean distance was used to calculate the distances. The blue (gray scaled) subsets of data points were recognized as clusters. 51
5.1 Graph for Equation 5.5. The blue curve (highest curve) describes Equation 5.5, the decreasing red curve describes the first term in Equation 5.5, and the increasing red curve describes the second term in Equation 5.5. The value 50 was used for all $\sigma^2_{k,(i,j)}$. 59
5.2 Exemplary results from the sensitivity analysis described in Chapter 7. Using Euclidean distance, the minimum distances and the maximum distances (not shown) between data points from two different classes were calculated. Limits calculated using Equation 5.5 are very consistent with empirical results for STDDIST Normal. When noise is assumed to be uniformly distributed, the results are analogous to those when noise is assumed to be normally distributed, indicating that the Gaussian random variables assumption is reasonable. 60

5.3 Exemplary results from the same sensitivity analysis described in Fig. 5.2. Using city block distance, the minimum distances and the maximum distances (not shown) between data points from two different classes were calculated. As the dimensionality of the data points increases, the ratio DMIN/STDDIST decreases. When noise is assumed to be uniformly distributed, the results are analogous to those when noise is assumed to be normally distributed, indicating that the Gaussian random variables assumption is reasonable. 62
5.4 Illustrations that show how two classes of data points link as the classes grow farther apart. 64
5.5 Schematic for finding meaningful levels of a hierarchical sequence. Inherent structure is revealed through features along the curve of the distance graph. These features correlate with those levels of the corresponding hierarchical sequence at which multiple classes of data points have finished linking to form new configurations of clusters. 65
5.6 Illustration that shows how rank order indices align with levels of the corresponding hierarchical sequence and distance elements align with the respective threshold distances d'. The data come from the sensor system experiment described in Chapter 7. Euclidean distance was used to calculate the distances. The arrow in the column for the threshold distances signifies that threshold distance d' is a continuous variable. In the last column, the meaningful cluster sets are indicated by asterisks. 66
6.1 Proximity vector and state matrices from a sensor system experiment similar to that described in Chapter 7. The different colored (gray scaled) numerals in the state matrices indicate which data points (sensor nodes) belong to the same cluster. For a more detailed explanation about how these data structures are used, see Chapter 4. 84

6.2 Lower left portion of a distance graph from the sensor system experiment described in Chapter 7. The enlargement shows one of the angles used to find meaningful levels of the corresponding hierarchical sequence. The dashed arrow represents $DISTROI_{i+1} - DISTROI_i$. Here, $DISTROI_{i+1}$ is the distance element of the 7th ordered triple and $DISTROI_i$ is the distance element of the 6th ordered triple. 85
6.3 Proximity vectors, distance graphs, and test results for Example 6. . . . 87
7.1 Exemplary results from the first part of the sensitivity analysis. 100
7.2 Results from the Monte Carlo method where $m = 1$ and $m = 10$. 103
7.3 Results from the Monte Carlo method where $m = 100$ and $m = 1000$. . 104
7.4 Results from the Monte Carlo method where $m = 1$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$. 105
7.5 Results from the Monte Carlo method where $m = 10$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$. 106
7.6 Results from the Monte Carlo method where $m = 100$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$. 107
7.7 Results from the Monte Carlo method where $m = 1000$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$. 108
7.8 Distance graphs for the structureless data set. Inherent structure does not emerge as the dimensionality of the data points increases. 110
7.9 Deemed meaningful levels for the structureless data set where $m = 10{,}000$. 111
7.10 Disparity matrix for the ethnic marriages data set. 112
7.11 Cluster sets for levels 1 to 14 of the ethnic marriages data set. 113
7.12 Cluster sets for levels 15 to 28 of the ethnic marriages data set. 114
7.13 Means and standard deviations for the seven kinds of measurements that were excerpted from the NIST data sets, distance graphs for the 33-point data set, and meaningful cluster sets. 116
7.14 17-points geometric pattern (left), sub-pattern A (center), and sub-pattern B (right). 117

7.15 Distance graphs for the noiseless and the noisy 17-points geometric pattern data sets. The locations of sub-patterns A ($level = 470$ for Euclidean distance) and B ($level = 2820$ for Euclidean distance) are indicated by the arrows. 118
7.16 Meaningful levels for the noiseless and the noisy 17-points geometric pattern data sets. 120
7.17 Cluster sets for the noiseless 17-points geometric pattern data set. Euclidean distance was used to calculate the distances. Only results for the first 17 data points are provided, since they are representative of the entire data set. The letters A and B denote sub-patterns A and B, respectively. 121
7.18 Continuation of Fig. 7.17 122
7.19 Cluster sets for the noiseless 17-points geometric pattern data set. City block distance was used to calculate the distances. Only results for the first 17 data points are provided, since they are representative of the entire data set. 123
7.20 Number of deemed meaningful levels that were false positives or false negatives for the 17-points geometric pattern data sets where $m = 20{,}000$ to 80,000 dimensions. 125
7.21 Heat map for the synthetic gene expression data sets. 126
7.22 Distance graphs for the noiseless and three noisy synthetic gene expression data sets. 127
7.23 Meaningful cluster sets for the noiseless and the noisy synthetic gene expression data sets. 128
7.24 Number of deemed meaningful levels that were false positives or false negatives for the synthetic gene expression data sets having from 5000 to 25,000 dimensions. 131
7.25 Configuration, dendrograms, and means and standard deviations for the motes sensing luminescence data set. The motes are classified according to the data sequences that were collected. The different colors (gray scales) represent the different clusters for $level = 6$. 134

7.26 Proximity vectors, distance graphs, and exemplary cluster sets for the motes sensing luminescence data set. The meaningful cluster sets are indicated by asterisks. 135
7.27 Deemed meaningful levels for the motes sensing luminescence data set. . 136
7.28 ECG and distance graphs for the sinus rhythm data sets. 137
7.29 Results from clustering the sinus rhythm data sets, showing the ranges over which $cutoffAngle$ could vary without incurring any false positives or false negatives. 138

Chapter 1

Introduction

"Discovery consists of seeing what everyone has seen and thinking what nobody has thought."
–Albert Szent-Györgyi, Nobel Laureate

"A model is not a straightforward reflection of external reality, and to propose a model is not to assert or to believe that Nature behaves in a particular way (Nature is surely utterly indifferent to our attempts to ensnare her in our theories). Rather, a model is a construct within the mental universe, by means of which we attempt somehow to describe certain, more or less restricted, aspects of the empirical universe. ... So long as a model appears to describe the relevant aspects of the world satisfactorily, we may continue, cautiously, to use it; when it fails to do so, we need to search for a better one."
–A. Philip Dawid, Causal Inference without Counterfactuals

"The methods for learning the models that [are] currently in effect are as yet pretty far over on the computational side of things. They haven't been brought over to the system [identification]/model [identification] kind of thinking."
– Helen Gill, National Science Foundation Program Director

"[T]he key to top performance is creating algorithms that exploit the structure of the problem and pay careful attention to algorithmic and numeric issues."
–Kristin Bennett and Emilio Parrado-Hernández, The Interplay of Optimization and Machine Learning Research

1.1 Background

This thesis presents a research project that was undertaken to bring a machine learning method over from the "computational side of things ... to the system ... kind of thinking." In particular, this thesis presents each part of a three-part research project. The goal of this project was to develop a general, complete linkage hierarchical clustering method that 1) substantially improves upon the accuracy of the standard complete linkage method and 2) can be fully automated or used with minimal operator supervision. Referred to as the INCLude (InterNodal Complete Linkage) hierarchical clustering method, the new clustering method was designed with small-n, large-m data sets in mind, where n is the number of data points, m is the number of dimensions, and "large" means thousands and upwards [2]. This project differs significantly from main stream machine learning projects. It elevates accuracy to equal importance with complexity and generalization, so the new method can be used inside a wide variety of cyber-physical (embedded real-time) systems.

Many researchers expect that cyber-physical systems will be one of the important growth technologies during the next ten to twenty years [3]. Cyber-physical systems are comprised of closely configured, seamlessly integrated computational (discrete time) and physical (continuous time) components [4]. In particular, "cyber-physical systems are physical, biological, and[/or] engineered systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is 'deeply embedded' into every physical component, possibly even into materials. The computational core is an embedded system, [which must respond to] real-time [demands], and is...often distributed. The behavior of a cyber-physical system is a fully-integrated hybridization of computation[al] (logical) and physical action" [5]. Important research problems in this area concern the "adaptability, autonomy, efficiency, functionality, reliability, safety, and usability" of such systems [5].

Cyber-physical systems are characterized by three inherent properties. First, they are heterogeneous. They are hybrid systems that include closely configured, seamlessly integrated computational and physical components. These systems may be distributed,

i.e., they may be systems of systems, each of which may have a different purpose. Even when two such subsystems have similar purposes, their operating parameters may be different, due to different local environmental or other factors.

Second, cyber-physical systems must monitor and respond to concurrently occurring environmental and internal phenomena. Sequential computation creates an illusion of concurrency by using multiple processes or threads that share processing resources. Thermal and power consumption problems caused by running processors at higher and higher clock speeds are constraints on this paradigm. Distributed computation uses multiple, independent processors, but communication problems arise when the processors are networked. To some extent, these problems are resolvable with multicore and/or heterogeneous processors, if software can be multithreaded to take advantage of these new chips.

Third, whether the behavior of a cyber-physical system is correct depends on whether the results of its computations are logically correct, or give approximations that are good enough to be useful [3], and on when the results are available [6]. A real-time computation is hard if a missed deadline jeopardizes correct system behavior or has catastrophic consequences [6], such as loss of life, extensive property damage, or material harm to the environment in which the system operates [3]. A real-time computation is soft if a missed deadline affects system performance but does not jeopardize correct system behavior or have catastrophic consequences [6].

The drive to realize new cyber-physical systems and the ever present demand for better product performance and product quality, greater product versatility, and improved cost efficiencies [1] are motivating a shift away from human operator control towards fully automated control or control with minimal operator supervision. Judgment that once was exercised by human operators is being taken over by computational components of cyber-physical systems, which typically reside at the supervisory control and data acquisition levels and/or higher, more global levels of digital controllers. See Fig. 1.1. The functionality of these controllers can be very sophisticated. Moreover, automated control over real world processes can depend on sensed data that is unavailable and environmental conditions that cannot be fully replicated during the testing phase of product development [6]. Consequently, the growing demand for automated control includes a growing demand for automated, intelligent control systems and data

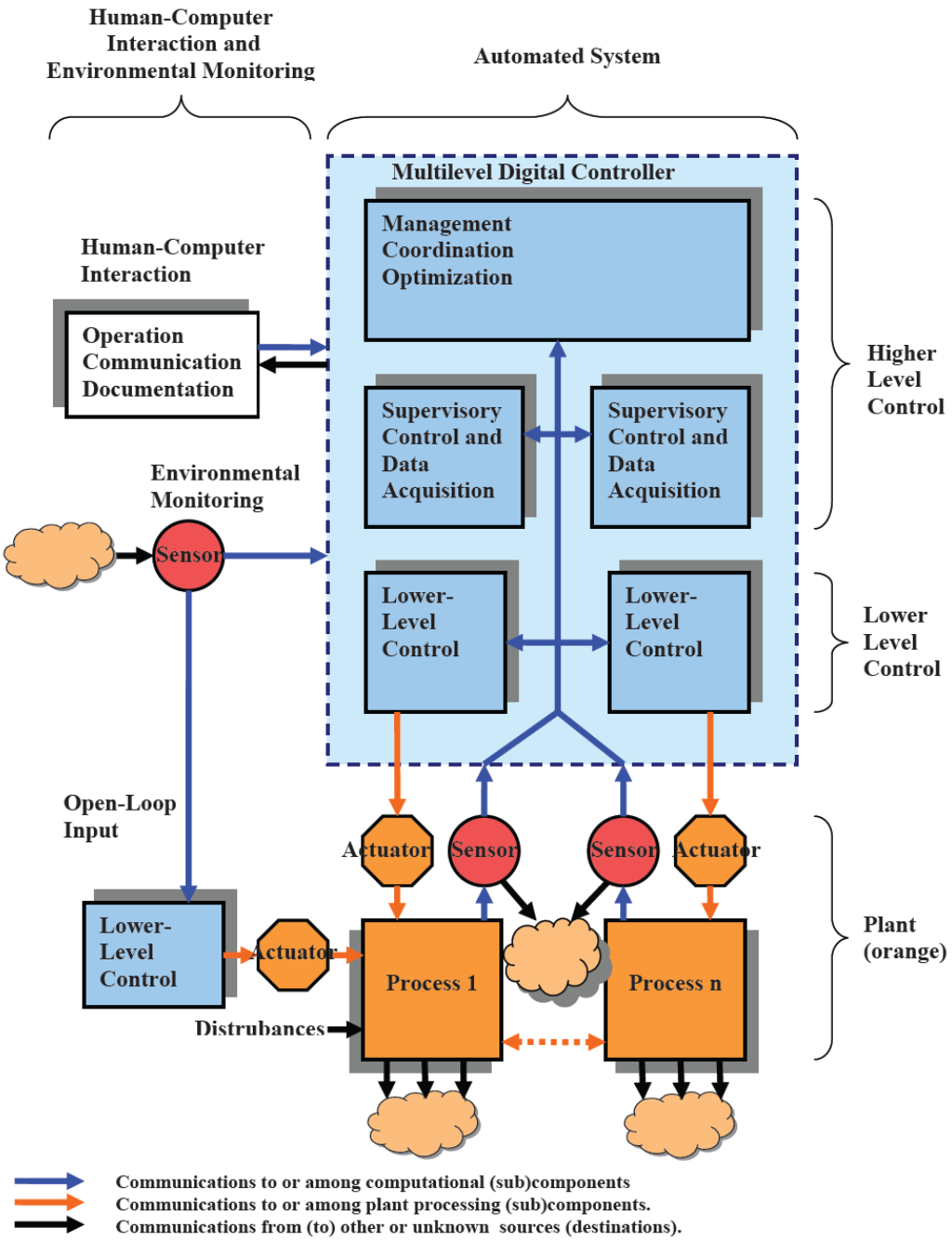


Figure 1.1: Schematic of an automated process. Sequential, feedback, and/or feed-forward control are included in the lowest level of control while monitoring, planning, prediction, decision making, and adaptation are included in higher levels of control. Adapted from [1].

analysis.

For example, the original vision of the Internet of Things did not include work in machine learning or automated control. More recently, however, a shift has occurred to integrate work from these areas [7]. Another example is ambient intelligence. AmI is closely related to a longer-term vision for creating an intelligent services system. This system will be a fully automated platform comprised of adaptive, embedded devices that can anticipate the needs of people and provide context aware, personalized services [8]. Yet another example is machine condition monitoring. One well-known European researcher, Hermann Kopetz, believes that automatic fault detection and diagnostics and system maintainability by untrained users will become important attributes for many future products to have [3]. Unlike consumer goods, it is less costly to repair than to replace many durable goods.

Developing software for sophisticated automated control has been a bigger challenge than many researchers envisioned. *See, e.g.*, [9]. Often, real world problems have been more involved than anticipated, and the linchpins for developing broadly applicable methods have eluded researchers. Moreover, the hardware that supports this software needed to be developed before the software could move beyond conceptualization to reduction to practice and testing. Computational hardware problems include memory size and latency, computational speed, parallelization capabilities, and cost and availability. Sensor systems hardware problems include sensor node miniaturization and sensing capabilities; wireless communication having low latency, adaptive connectivity, and improved energy efficiency; resource constrained processors and memory for operating systems and in-network information processing; and cost and availability.

Until recently, machine learning researchers, like data mining researchers, have focused mostly on large-scale data sets. Consequently, low complexity and good generalization have been *the* desirable properties for machine learning algorithms to have. Assumptions and approximations that are used to manage complexity often sacrifice accuracy for efficiency. When software is embedded in systems, scalability with respect to the number of data points is often viewed as less important. Many systems collect and analyze data sets having fewer than thousands of data points.[1] Accuracy, on the

[1] For example, a typical automobile has about 500 sensors, a small specialty brewery has about 600 sensors, and a small power plant has about 1100 sensors. An environmental control system for a building may have as many as 22,600 sensors, of which the data from no more than about 14,000 sensors

other hand, is viewed as more important.

Some researchers are beginning to characterize data sets and the kinds of problems from which they arise in terms of their numbers of data points n and their dimensionalities (samples) m, i.e., large-n, small-m data sets, large-n, large-m data sets, and small-n, large-m data sets, where "large" means thousands and upwards. *See, e.g.* [2]. Small-n, large-m data sets are used by many cyber-physical systems and include time series. The work described in this thesis was completed with small-n, large-m data sets in mind. It may accommodate large-n, large-m data sets as well, especially as enhancements are made to the code, parallel programming techniques are incorporated, and further advances in computer hardware are introduced, but these are not the focus of this work.

1.2 Problem Definition

As already mentioned above, the goal of this project was to develop a general, complete linkage[2] hierarchical clustering method that 1) substantially improves upon the accuracy of the standard complete linkage method and 2) can be fully automated or used with minimal operator supervision. The application domain for the project was cyber-physical systems, and the work is important because there are no known clustering methods that can be automated for use inside such systems. The new clustering method satisfies the following broad requirements:

1. The only input that the new clustering method may require is a data set.

2. The new clustering method should be capable of constructing a cluster set for every level of an $\frac{n \cdot (n-1)}{2} + 1$-level hierarchical sequence. Here, a cluster set is defined as the minimum number of complete linkage clusters that are exhaustive.

3. The new clustering method should construct clusters that are globular or compact, and preferably comprised of maximally complete subsets of data points.

---

are analyzed together. An automobile assembly plant may have more than 100,000 sensors, although the data from no more than about 10,000 sensors are analyzed together. The number of sensors in an assembly plant is much larger than the numbers of sensors in the various sensor subsystems because many sensors are used for gates or for safety.

[2] Here, "complete linkage" refers to where every data point in a cluster is linked to every other data point in the cluster.

4. The new clustering method should construct hierarchical sequences of cluster sets that are transformationally invariant.

5. The new clustering method should not construct hierarchical sequences with inversions.[3]

Over the course of the project, three technological problems were resolved.

Problem 1 The standard complete linkage method 1) cannot resolve ties between intercluster distances, 2) constructs inaccurate cluster sets where the inherent structure in a data set is not taxonomic (misses meaningful cluster sets[4] , cannot construct clusters as compactly as is possible, does not recognize overlapping clusters, and cannot resolve the (cluster membership) migration problem), 3) sometimes produces different results for agglomerative hierarchical clustering and divisive hierarchical clustering, 4) cannot be conveniently used for divisive hierarchical clustering, and 5) uses *post hoc* heuristics to find meaningful cluster sets. Often, the cluster sets are hard to interpret.

Solution to Problem 1 The new clustering method unwinds the assumptions that the standard complete linkage method makes, decouples evaluating pairs of data points for linkage from constructing cluster sets, and constructs cluster sets *de novo*. Consequently, the new method can resolve ties and constructs accurate cluster sets. Here, the Rand Index is used to measure accuracy. Agglomerative hierarchical clustering and divisive hierarchical clustering are unified.

Problem 2 The standard complete linkage method and Peay's hierarchical clique detection method [10] are updating methods. Because updating methods construct

[3] An "inversion" refers to cluster sets of a hierarchical sequence that are out of order with respect to the threshold distances d'.

[4] A "meaningful cluster set" refers to a cluster set that can have real world meaning. Where a data set has inherent structure, a "meaningful level" refers to a level of a hierarchical sequence at which a new configuration of clusters has finished forming. These definitions appear to be synonymous for $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequences. The cluster set that is constructed for a meaningful level is a meaningful cluster set, so these terms can be used interchangeably.

cluster (clique) sets by updating previously constructed cluster sets, they must construct a cluster set for every level of a hierarchical sequence. The time complexity to construct a cluster set for every level of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence is $O(n^4)$.

Solution to Problem 2 A means was discovered and developed for finding meaningful levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence *prior* to performing a cluster analysis. The means includes constructing at least one distance graph, which is visually examined for features that correlate with meaningful levels of the corresponding hierarchical sequence. Because the new clustering method can construct only the cluster sets for select, possibly noncontiguous levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence, the new method can construct only the cluster sets for meaningful levels of such a hierarchical sequence. This reduces the time complexity to construct cluster sets from $O(n^4)$ to $O(ln^2)$, where l is the number of meaningful levels. *These are the cluster sets that can have real world meaning.*

Problem 3 The solution to Problem 2 is performed manually.

Solution to Problem 3 The graphical relationships that underlie the above-described features of a distance graph are mathematically captured in a single equation. This equation is used as a test to determine which levels of a hierarchical sequence are deemed meaningful. The test is integrated into the new clustering method, where it is performed after each pair of data points is evaluated for linkage, at almost no extra cost to the new method.

This thesis comprises eight chapters, including this Introduction. Chapter 2 describes related work. Chapter 3 describes contributions that the project makes. Chapter 4 provides a detailed description of the INCLude hierarchical clustering method. Chapter 5 describes the means for finding meaningful levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence. Chapter 6 describes how to mathematically capture the graphical relationships that underlie the above-described features and integrate the means into the new

clustering method. Chapter 7 describes empirical results from nine experiments. The first two experiments corroborate the theoretical work described in Chapter 5. The next three experiments show that the new clustering method performs well with respect to different kinds of data sets, including a data set that has no inherent structure, a data set comprised of nonmetric data, and a data set comprised of multiple attributes. The last four experiments show that the new clustering method compares favorably with the standard complete linkage method and k-means clustering with respect to accuracy and time. Further, they show that cluster sets constructed for meaningful levels can have real world meaning. The thesis is concluded in Chapter 8.

Chapter 2

Related Work

2.1 Flat and Hierarchical Clustering Methods

Cluster analysis is used to group finite sets of data points into clusters, which clusters have meaning within the context of a particular problem [11]. Historically, researchers have distinguished between clustering methods that are based on strategies for grouping data points into clusters and clustering methods that are based on strategies for traversing hierarchies. *See, e.g.*, [12], [13]. Grouping strategies seek clusters whose data points are, in accordance with some predetermined measure of "likeness", as like each other as is possible. In other words, they try to optimize intragroup homogeneity [12], [13]. The grouping strategies include partitional clustering methods that partition a set of data points into a single or flat set of clusters. *See, e.g.*, [11].

In contrast, hierarchical strategies seek the most efficient way to progressively combine or subdivide a set of data points. "They aim...to find the best route from population to individuals [or vice versa]; but this route may be found at some degree of sacrifice of the homogeneity of the groups through which the process passes" [12]. Consequently, users have had to decide "whether [to] optimize the clusters or the route" [12]. The hierarchical strategies include those methods, such as the standard hierarchical clustering methods, that find "a special sequence of partitional classifications [cluster sets]" by "transforming a proximity matrix into a sequence of nested partitions" [12].

A survey of flat clustering methods and hierarchical clustering methods was conducted as part of a search for designs that could at least partially fulfill the goal of this

project. A consensus among researchers does not exist regarding what constitutes a cluster. Optimizing a clustering method with respect to one performance measure often causes a drop in performance with respect to another measure [14]. Therefore, designing an optimal clustering method cannot be achieved in isolation. It must be done in view of a particular problem.

The flat clustering methods were examined for the following attributes:

- **Required Input** Required input refers to what parameters must be input into an algorithm in order for it to perform its function.
- **Observations About Complexity** Observations about complexity are observations about the design or operation of an algorithm.
- **Partitional** Partitional refers to whether a clustering method constructs partitional cluster sets.
- **Cluster Shape** Cluster shape refers to the shape of the clusters.
- **Transformational Invariance** Transformational invariance refers to whether the cluster sets are invariant to monotonic transformations of the distances between the data points.
- **Control of Cluster Diameter** Control of cluster diameter refers to whether a user can control the diameter of the clusters.
- **Other** Other is used for miscellaneous comments.

Likewise, the hierarchical clustering methods were examined for the following attributes:

- **Required Input** Required input refers to what parameters must be input into an algorithm in order for it to perform its function.
- **Linkage Metric** Linkage metric refers to the intercluster distance that is used to calculate distance between two clusters.
- **Partitional** Partitional refers to whether a clustering method constructs partitional cluster sets.
- **Cluster Shape** Cluster shape refers to the shape of the clusters.
- **Transformational Invariance** Transformational invariance refers to whether the cluster sets are invariant to monotonic transformations of the distances between the data points.
- **Subject to Inversions** Subject to inversions refers to whether the cluster sets of a hierarchical sequence can be out of order with respect to the threshold distances d'.

- **Other** Other is used for miscellaneous comments.

As the chart in Fig. 2.1 shows, the flat clustering methods require input in addition to a data set. Usually, this input is used to find a threshold index at which to cut a dendrogram or find the number of "meaningful" clusters in a cluster set. These methods group data points into a single or flat, partitional cluster set, i.e., the clusters cannot overlap. Consequently, a flat clustering method was not a suitable starting point for the project.

As the chart in Fig. 2.2 shows, none of the hierarchical clustering methods finds meaningful levels or meaningful cluster sets of a hierarchical sequence. Further, CURE uses random sampling, which trades accuracy for efficiency, and CHAMELEON requires input in addition to a data set. Although the standard hierarchical clustering methods impose taxonomic structure on data sets, a data set is the only input that they require. Of the standard hierarchical clustering methods, the standard complete linkage method is the only method that is invariant to monotonic transformations of the distances between the data points, that can cluster any kind of attribute, that is not prone to inversions, and that constructs globular or compact clusters. Thus, the standard complete linkage method became the starting point for the project.

2.2 The Standard Complete Linkage Method

The standard complete linkage method (Sorenson 1948) was the first of seven standard hierarchical clustering methods to be developed during the late 1940s to the mid-1960s [15]. At that time, clustering problems having about 150 data points were viewed as moderately sized problems while problems having about 500 data points were viewed as large. *Cf.* [16]. Reasoning about hardware limitations while an application is being developed is a key aspect of computational thinking [17]. To accommodate the hardware limitations of that time and solve these "large-scale" clustering problems, those who developed the standard complete linkage method made several assumptions. They assumed that cluster sets are nested partitions. In other words, they assumed that clusters are mutually exclusive and indivisible [11]. Making these assumptions reduces the size of a hierarchical sequence from $\frac{n \cdot (n-1)}{2} + 1$ levels to n levels [18], where n is

Name	Required Input	Observations About Complexity	Partitional	Cluster Shape	Transform -ational Invariance	Control of Cluster Diameter	Other
k-Means Clustering	Takes numerical data only; number of clusters *k*	Pair-wise computations are too expensive, so cluster representatives are used	Yes	Convex clusters having similar diameters		No	Only locally optimal; sensitive to outliers; the basic algorithm is not scalable
k-Medoids Clustering	Data set; number of clusters *k*	Pair-wise computations are too expensive, so cluster representatives are used	Yes			No	Only locally optimal; insensitive to outliers
Graph Cut Methods	Data set; number of clusters *k* or maximum cut value	Optimizing any min-cut modification is NP-hard	Yes	Ratio cuts and normalized cuts balance cluster sizes; tends to group denser regions together earlier.		No	Cannot resolve ties
Spectral Graph Methods	Data set; number of eigenvalues *k*		Yes			No	Must be tuned; slow
Probability Methods (Mixture Models)	Data set; number of clusters *k* and distribution parameters		Yes			No	Good interpretability
Density-Based Methods (e.g., DBSCAN and DENCLUE)	Data set; parameters €, σ, and MinPts (DBCLASD is nonparametric)		Yes	Can discover irregular cluster shapes		No	Lacks interpretability; requires a metric space; merges clusters that overlap; insensitive to outliers;
Grid-Based Methods	Data set; threshold values		Yes			No	Fast

Figure 2.1: Survey of flat clustering methods.

Name	Required Input	Linkage Metric	Partitional	Cluster Shape	Transform -ational Invariance	Subject to Inversions	Other
Complete Linkage (Farthest Neighbor)	Data set	Maximum distance (minimum similarity) between two data points	Taxonomic cluster sets	Tends to find globular or compact clusters having equal diameters	Yes	No	Does not find meaningful cluster sets
Single Linkage (Nearest Neighbor)	Data set	Minimum distance (maximum similarity) between two data points	Taxonomic cluster sets	Clusters can be unbalanced and straggly chains	Yes	No	Does not find meaningful cluster sets
(Group) Average Linkage (UPGMA)	Data set	Average distance (average similarity) between two data points	Taxonomic cluster sets		No	No	Does not find meaningful cluster sets
Unweighted Centroid Linkage (UPGMC)	Data set	Squared Euclidean distance between mean vectors (centroids)	Taxonomic cluster sets		No	Yes	Does not find meaningful cluster sets
Weighted Average Linkage (WPGMA)	Data set	Average distance (average similarity) between two data points	Taxonomic cluster sets		No	No	Does not find meaningful cluster sets
Median or Weighted Centroid Linkage (WPGMC)	Data set	Squared Euclidean distance between weighted centroids	Taxonomic cluster sets		No	Yes	Does not find meaningful cluster sets
Ward's Method (Min. Variance or Min. Sum of Squares)	Data set	Increase in sum of squares within clusters, after fusion, summed over all variables	Taxonomic cluster sets	Tends to find spherical clusters that are the same size	No	No	Does not find meaningful cluster sets; sensitive to outliers
CURE (Clustering Using REpresentatives)	Data set; number of clusters *k*	Minimum of distances between two scattered representatives	Taxonomic cluster sets	Can find clusters having different shapes and sizes			Does not find meaningful cluster sets; random sampling trades accuracy for efficiency
KNN-Based (e.g., CHAMELEON)	Data set; parameter *k*; user provided thresholds	Distances between a data point and its neighbors; CHAMELEON uses measures of relative interconnectivity and relative closeness	Taxonomic cluster sets	Uses majority voting to classify data points rather than grouping data points into clusters			Does not find meaningful cluster sets; locally normalized measures depend on local data

Figure 2.2: Survey of hierarchical clustering methods.

the number of data points in a data set.[1] Further, the number of combinations that need to be examined at each level of the hierarchical sequence becomes much smaller than complete enumeration [16]. The developers also assumed that notions of distance between data points ("interpoint" distances) can be generalized to notions of distance between clusters of data points ("intercluster" distances). By making this assumption, proximity measures known as linkage metrics could be devised. Linkage metrics are used to combine clusters of data points or subdivide a cluster of data points at a time [18].[2]

When inherent (hierarchical) structure in a data set is not taxonomic, these assumptions sacrifice accuracy for efficiency in at least five ways:

First, when clusters are being combined or a cluster is being subdivided, the standard complete linkage method cannot resolve ties between intercluster distances [11]. Treating ties simultaneously does not resolve the problem. Either one of the distances must be selected arbitrarily or alternative hierarchical sequences are constructed. Consequently, the results are no longer deterministic. Multiple occurrences of ties can lead

[1] Because the standard hierarchical clustering methods were the only viable hierarchical clustering methods for a long time, some data mining researchers use the term "hierarchy" synonymously with "taxonomic hierarchy". Other researchers use the term as it is commonly understood. *Compare* [11] and [15] (narrow meaning) *with* [19], [10], and [18] (common meaning). The Oxford American Dictionary [20] defines a "hierarchy" as a "system or organization in which people or groups are ranked one above the other according to status or authority" and "hierarchical" as "arranged in order of rank". A "taxonomic hierarchy" is provided as a particular example. Other hierarchies such as Maslow's hierarchy are not taxonomic.

Because of the assumptions that underlie the standard complete linkage method, it constructs n-level hierarchical sequences that are taxonomic. The new clustering method unwinds these assumptions. Consequently, it can construct $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequences that are not taxonomic. Cluster set construction for both n-level hierarchical sequences and $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequences is based on linkage between the data points in a data set, which linkage depends on a continuous variable, threshold distance d'.

[2] Linkage metrics are used to define distance between two clusters. The linkage metric for the standard complete linkage method uses the distance between those two data points, one from each cluster, that are the farthest apart [21], [15]. The order in which clusters are combined by agglomerative hierarchical clustering is determined by the minimum such distance. Agglomerative hierarchical clustering would assign each data point in a data set to its own cluster, calculate the intercluster distance between each pair of clusters and store these distances in a proximity matrix, find the minimum such distance to determine which two clusters should be combined next, calculate the intercluster distances between the new cluster and the other clusters, reconstitute the proximity matrix, and reiterate until all the data points belong to one cluster. Divisive hierarchical clustering would start at the other end of the hierarchical sequence and subdivide clusters until each data point is assigned to its own cluster. Complete linkage ensures that data points from the same cluster are within some maximum distance of each other [11], [21].

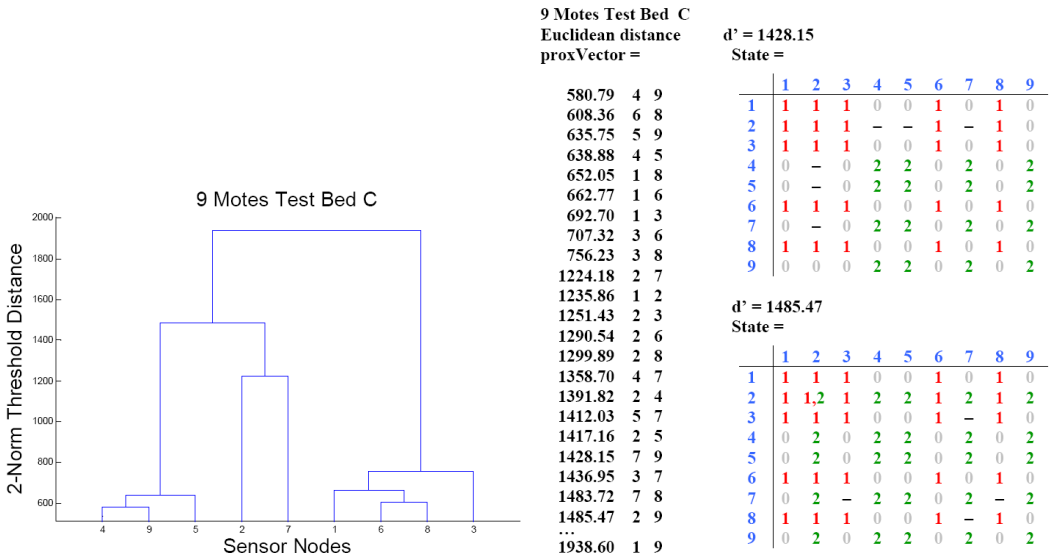


Figure 2.3: Dendrogram, proximity vector, and state matrices for Example 1.

to tree-like structures of dendrograms.

Second, cluster sets are often constructed inaccurately. As Example 1 illustrates, one reason for this is the (cluster membership) migration problem.[3] The migration problem is commonly characterized as an error correction problem [22], [21], [15] or a revisitation problem [18]. These views, however, implicitly accept the assumptions that underlie the standard complete linkage method. Consequently, the migration problem is addressed indirectly. *See, e.g.,* [22], which presents a multilevel refinement scheme.

Example 1 *This example illustrates how the standard complete linkage method constructs cluster sets that are inaccurate. The data in Fig. 2.3 come from a sensor system experiment similar to that described in Chapter 7. The different colored (gray scaled) numerals in the state matrices indicate which data points (sensor nodes) belong to the same cluster. The proximity vector contains distances between the data points (col. 1) and the indices of the respective data points (cols. 2 and 3).*

When the standard complete linkage method is used to construct clusters, data points 2 and 7 combine to form a cluster at threshold distance $d' = 1224.18$. *Because the*

[3] In other words, once two data points are assigned to the same cluster during agglomerative hierarchical clustering, they cannot be split-up; and once two data points are assigned to separate clusters during divisive hierarchical clustering, they cannot be recombined. *Cf.* [12].

cluster is indivisible, it cannot combine with another cluster until $d' = 1485.47$, when it combines with the cluster comprised of data points 4, 5, and 9. Data point 2 is closer to data points 1, 3, 6, and 8 than it is to data points 4, 5, or 9. However, it cannot combine with data points 1, 3, 6, and 8 because it has already combined with data point 7. Moreover, contrary to what many believe, see, e.g., [11], [23], some clusters do not maintain their maximal completeness. The cluster comprised of data points 1, 3, 6, and 8 is maximally complete when it is constructed at $d' = 756.23$, but not after $d' = 1299.89$.

When the assumptions that underlie the standard complete linkage method are unwound, data points 2 and 7 migrate to different clusters at $d' = 1428.15$. Consequently, two mutually exclusive clusters emerge. These clusters do not appear in the dendrogram. At $d' = 1485.47$, the two clusters overlap at data point 2. Neither does this information appear in the dendrogram.

Third, results from the standard compete linkage method can depend on which end of a hierarchical sequence is treated as the beginning. When the dendrograms for agglomerative hierarchical clustering and divisive hierarchical clustering are different, finding the cause(s) for their difference is both inconvenient and time-consuming.

Fourth, it is harder to subdivide clusters than it is to combine clusters, so divisive hierarchical clustering is used less often than agglomerative hierarchical clustering. The bipartitioning problem, i.e., determining which cluster should be subdivided and how to subdivide a cluster, can be an involved enumeration problem. The variance and the intercluster sum-of-dissimilarities criteria are NP-hard [24]. The diameter and the sum-of-diameters criteria are not [24], but they still suffer from the other above-described weaknesses.

Fifth, the standard complete linkage method does not find which levels or which cluster sets of a hierarchical sequence are meaningful. Once the cluster sets of an n-level hierarchical sequence are constructed, it still is necessary to construct a dendrogram and use a *post hoc* heuristic to determine where and how many times to "cut" the dendrogram. *See, e.g.,* [11], [21], [15]. *Post hoc* heuristics are computationally expensive to run. Moreover, the number of meaningful levels can be significantly smaller than n or, in some cases, larger than n. *See* the empirical experiments described in Chapter 7.

Because of these weaknesses, it is difficult to interpret results obtained from the

standard complete linkage method. Consequently, it is underutilized in automation and by intelligent control systems, including supervisory functions such as fault detection and diagnosis and adaptation. *Cf.* [1]. When the standard complete linkage method *is* used, stopping criteria are often used in place of *post hoc* heuristics. Usually, stopping criteria are predetermined. If the model upon which they are based is inadequate or changes, the stopping criteria lose their usefulness. Moreover, because the standard complete linkage method is an updating method, it uses information from previously constructed cluster sets to construct subsequent cluster sets. *See, e.g.,* [11], [21]. Until the stopping criteria are met, it must construct the cluster set for every level of an n-level hierarchical sequence. These cluster sets must be either materially accurate or, if possible, amendable for material inaccuracies. *See, e.g.,* U.S. Patent No. 8,312,395 (defect identification in semiconductor production; operators must ensure that the results are 80 to 90 percent accurate). As much as 90 percent of the effort that goes into implementing the standard complete linkage method is used to develop stopping criteria or interpret results.

Notwithstanding these weaknesses, the standard complete linkage method is an important clustering method. The distributions of many real world measurements are bell-shaped, so the standard complete linkage method has broad applicability. Its simplicity makes it relatively easy to mathematically capture its properties. Of the standard hierarchical clustering methods, the standard complete linkage method is the only method that is invariant to monotonic transformations of the distances between the data points, that can cluster any kind of attribute, that is not prone to inversions, and that constructs globular or compact clusters [21], [15]. Moreover, more sophisticated methods show no clear advantage for many purposes. *See, e.g.,* U.S. Patent No. 8,352,607 (load balancing), U.S. Patent No. 8,312,395 (defect identification), U.S. Patent No. 8,265,955 (assessing clinical outcomes), and U.S. Patent No. 7,769,561 (machine condition monitoring). Thus, a need exists to bring complete linkage hierarchical clustering over from the "computational side of things ... to the system ID/model ID kind of thinking" [25] as part of closing the loop on cyber-physical systems.

2.3 Clique Detection Methods

Because researchers could not come up with a hierarchical clique detection method that is tractable, they have avoided developing these methods. The number of possible cliques becomes "huge" as the number of data points n increases, and at least one researcher, D.W. Matula, explicitly teaches away from using these methods. "[C]lustering based on cliques is practical only for small n" [11] (citing [26]), where "small" means tens.

For example, in [19] and [10], E.R. Peay presents a linkage-based hierarchical clique detection method. For every level of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence for which a clique set is constructed, Peay's clique detection method recognizes every maximally complete subset of data points as a clique, including those from which the data points can migrate. Because Peay's clique detection method is an updating method, it must construct a clique set for every level of such a hierarchical sequence. It cannot construct only the clique sets for meaningful levels.

A similar problem exists for flat clique detection methods. Without knowing which levels of a hierarchical sequence are meaningful, flat methods are ineffective. Moreover, in general, flat clique detection methods are designed to find the maximum clique in a graph, the maximum weighted clique in a weighted graph, all maximal cliques, or one or more cliques whose size is equal to or greater than a given size k. *See, e.g.*, [27]. They are not designed to find cliques other than those from which all the data points can migrate.

2.4 Schemes for Cutting Dendrograms

Post hoc heuristics for cutting dendrograms include statistical methods, *see, e.g.*, [28] (gap statistic), methods that rely on machine learning, *see, e.g.*, [29] (using semi-supervised learning to learn a threshold index), and methods that depend on domain knowledge, *see, e.g.*, [30] (domain specific knowledge provided by an analyst). *Post hoc* heuristics are not suitable for finding meaningful levels or meaningful cluster sets of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence. To use these heuristics, every level of a hierarchical sequence must be constructed, so *post hoc* heuristics do not reduce the time complexity to construct cluster sets. Most *post hoc* heuristics find only one "optimal"

or maybe a few cluster sets. Moreover, because they use sophisticated statistical or machine learning methods, they are computationally expensive to run, and considerable operator supervision is needed in one way or another. Many have difficulty identifying clusters that are not well-separated[4] or overlap. *See, e.g.*, [28].

2.5 Other Related Work

Within a framework based on ultrametric topology and ultrametricity, F. Murtagh [2] observes that it is easier to find clusters in sparse or high dimensional spaces. This work does not describe how to find meaningful levels of a hierarchical sequence. Also, it assumes that the means and the standard deviations of all the dimensions of a data point are the same.

[4] A data point is "well-separated" if the distance between the data point and every other data point that does not belong to the same cluster is greater than threshold distance d'. Where all the data points of a cluster are well-separated, the cluster is well-separated.

Chapter 3

Contributions

The INCLude (InterNodal Complete Linkage) hierarchical clustering method is a hierarchical, hard, exhaustive, connectivity-based clustering method. The method is hierarchical because it constructs hierarchical sequences of cluster sets that evolve as a function of a continuous variable, threshold distance (index) $d' \in R$. The method is hard because a data point either belongs to a cluster or it does not. The method is exhaustive because, for each cluster set, every data point is assigned to at least one cluster. The method is connectivity-based because threshold distance d' is used to determine which data points are linked. Cluster set construction is based on this linkage. The new clustering method exploits the structure of the clustering problems for which it was designed and learns from new information.

3.1 INCLude Hierarchical Clustering: A Hierarchical Clustering Method Based Solely on Interpoint Distances

The work described in Chapter 4, INCLude Hierarchical Clustering: A Hierarchical Clustering Method Based Solely on Interpoint Distances, assumes that clusters are globular or compact, and preferably comprised of maximally complete subsets of data points. Thus, the new clustering method is consonant with the model for a measured

value that scientists and engineers commonly use.[1] This model has substantially broader applicability than the taxonomic model that is the basis for the standard complete linkage method. Chapter 4 makes the following contributions:

- INCLude hierarchical clustering can construct a cluster set for every level of an $\frac{n\cdot(n-1)}{2}$ + 1-level hierarchical sequence. This capability is needed for constructing meaningful cluster sets and when the number of meaningful levels of a hierarchical sequence is greater than n, the number of data points.
- Ties between interpoint distances are resolved within a single hierarchical sequence.
- Interpoint distances are used to construct clusters, clusters can overlap, and data points can migrate between clusters. Consequently, when inherent structure in a data set is not taxonomic, the new clustering method is substantially more accurate than the standard complete linkage method.
- Evaluating pairs of data points for linkage is decoupled from constructing cluster sets, so linkage between the data points can be updated without constructing cluster sets. Also, agglomerative hierarchical clustering and divisive hierarchical clustering are unified.
- INCLude hierarchical clustering constructs cluster sets *de novo* instead of updating previously constructed cluster sets. Because linkage between data points can be updated without constructing cluster sets, and cluster sets are constructed *de novo*, it is possible to construct only the cluster sets for select, possibly noncontiguous levels of an $\frac{n\cdot(n-1)}{2}$ + 1-level hierarchical sequence. Also, by constructing cluster sets *de novo*, data points can migrate between clusters.

3.2 Means for Finding Meaningful Levels of a Hierarchical Sequence *Prior* to Performing a Cluster Analysis

The expressiveness gained from using interpoint distances is both a strength and a weakness, because it trades computational efficiency for accuracy. The time complexity to construct a cluster set is $O(n^2)$. If a cluster set is constructed for every level of

[1] The model for a measured value provides that a measured value equals its true value plus a bias (accuracy) plus random error (statistical uncertainty or precision) [31].

an $\frac{n\cdot(n-1)}{2}+1$ level hierarchical sequence, the time complexity becomes $O(n^4)$. This is large even for small-n, large-m data sets. Thus, a means was developed for finding meaningful levels of such a hierarchical sequence *prior* to performing a cluster analysis.

The work described in Chapter 5, Means for Finding Meaningful Levels of a Hierarchical Sequence Prior to Performing a Cluster Analysis, makes four more assumptions: It assumes that the 2-norms and the 1-norms of the data points are calculable. It further assumes that noise (random error) is the only random component in a measured value, that noise can be modeled as Gaussian random variables, and that the noise that is embedded in each dimension (sample) of each data point is statistically independent. Chapter 5 makes the following contributions:

- The 2-norm and the 1-norm are used to calculate distances between the data points in a data set, which distances are used to construct distance graphs. Distance graphs can exhibit features that correlate with meaningful levels of the corresponding hierarchical sequences.
- By finding meaningful levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence *prior* to performing a cluster analysis and using a clustering method that can construct only the cluster sets for select, possibly noncontiguous levels of such a hierarchical sequence, it is possible to construct only the cluster sets for meaningful levels and reduce the time complexity to construct cluster sets from $O(n^4)$ to $O(ln^2)$, where l is the number of meaningful levels, $2 \leq l \leq \frac{n\cdot(n-1)}{2}+1$. *These are the cluster sets that can have real world meaning.*
- Theoretical and empirical support are provided to show how increasing the dimensionality of the data points helps reveal inherent structure in noisy data, which is necessary for finding meaningful levels.

3.3 Closing the Loop on a Complete Linkage Hierarchical Clustering Method

As described in Chapter 5, the means for finding meaningful levels includes constructing at least one distance graph, which is visually examined for features that correlate with meaningful levels of the corresponding hierarchical sequence. The work described in Chapter 6, Closing the Loop on a Complete Linkage Hierarchical Clustering Method,

makes the same assumptions as the work in Chapters 4 and 5. Chapter 6 makes the following contributions:

- The graphical relationships that underlie the above-described features of a distance graph are mathematically captured in a single equation.
- The means for finding meaningful levels is integrated into the new clustering method, at almost no extra cost to the new method. Thus, the new clustering method can be fully automated or used with minimal operator supervision.

3.4 Usefulness

The following are examples of applications where the new clustering method may be useful:

- Fault detection and diagnostics, machine condition monitoring, detecting anomalies in how devices and machines operate, and new defect prediction;
- The Internet of Things;
- The emerging area of ambient intelligence;
- Identifying patterns and creating templates for activities such as load balancing;
- Characterizing clinical outcomes for diagnosis, prognosis, and treatment; and
- Many kinds of bio-informatics and chem-informatics.

Chapter 4

INCLude Hierarchical Clustering: A Hierarchical Clustering Method Based Solely on Interpoint Distances

The INCLude (InterNodal Complete Linkage) hierarchical clustering method unwinds the assumptions that the standard complete linkage method makes. Instead of intercluster distances, interpoint distances are used to construct clusters; clusters can overlap; and data points can migrate between clusters. Consequently, when inherent structure in a data set is not taxonomic, the new method is substantially more accurate than the standard complete linkage method.

As the project progressed, it became clearer and clearer that these and other design decisions also had other important benefits. When evaluating pairs of data points for linkage is decoupled from constructing cluster sets and cluster sets are constructed *de novo*, agglomerative hierarchical clustering and divisive hierarchical clustering are unified; divisive hierarchical clustering becomes as easy to use as agglomerative hierarchical clustering; and it is easy to change how clusters are constructed or which subsets of data points are recognized as clusters. Moreover, it is possible to construct only the cluster sets for select, possibly noncontiguous levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence.

Then, as Chapter 5 will show, it becomes possible to construct only the cluster sets for meaningful levels of such a hierarchical sequence.

This chapter begins with an overview of the INCLude hierarchical clustering method. Afterwards, how the method is implemented as an algorithm is described in detail. Examples that illustrate how the algorithm works are provided along the way.

4.1 Design

4.1.1 Design Decisions

The new clustering method is a hierarchical clustering method because "in complex systems, relevant features are typically both local and global[,] and different levels of organization emerge at different [threshold distances] in a way that is intrinsically not reducible" [32]. The new method is founded on seven design decisions, which were made in view of the goal of this project.

First, instead of intercluster distances, interpoint distances are used to construct clusters. Whether a subset of data points can comprise a cluster depends only on the distances between these and no other data points. As Example 1, Chapter 2, shows, this property is important for constructing compact clusters, i.e., grouping data points that are the most alike.

Second, clusters can overlap. Many systems data sets have inherent structures that are more accurately described when clusters are allowed to overlap, and allowing clusters to overlap does not affect cluster set construction where inherent structure is taxonomic. For example, see the sensor system experiment described in Chapter 7. Moreover, as Example 1, Chapter 2, also shows, allowing clusters to overlap is needed for constructing maximally complete clusters.

Third, only maximally complete subsets of data points are recognized as clusters. Maximal completeness is consonant with the model for a measured value that scientists and engineers commonly use. It also is consonant with complete linkage and is used to construct clusters that overlap, which often occur at higher levels of hierarchical sequences. Rather than using a scheme such as fuzzy clustering, *see, e.g.,* [33], to resolve uncertainty caused by noise, the dimensionality of the data points is increased to reveal inherent structure in noisy data.

Fourth, evaluating pairs of data points for linkage is decoupled from constructing cluster sets. Consequently, linkage between the data points can be updated without constructing cluster sets. Also, agglomerative hierarchical clustering and divisive hierarchical clustering are unified. The results from either are the same, and divisive hierarchical clustering is as easy to use as agglomerative hierarchical clustering. Further, it is easy to switch from one cluster set construction module to another, in order to change how clusters are constructed or which subsets of data points are recognized as clusters.

Fifth, cluster sets are constructed *de novo* instead of updating previously constructed cluster sets. Unlike the standard complete linkage method, which must construct a cluster set for every level of an n-level hierarchical sequence, or Peay's hierarchical clique detection method [10], which must construct a clique set for every level of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence, the new method constructs cluster sets for an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence independently of one another. Because linkage between data points can be updated without constructing cluster sets, and cluster sets are constructed *de novo*, it is possible to construct only the cluster sets for select, possibly noncontiguous levels of such a hierarchical sequence. Also, by constructing cluster sets *de novo*, data points can migrate between clusters.

Sixth, data points can migrate between clusters. The migration problem does not arise when every maximally complete subset of data points is recognized as a cluster. *Cf.* [10]. When fewer clusters are recognized, rules are needed to define which clusters are recognized and which are not. For example, the standard complete linkage method recognizes only sets of clusters that nest. Here, the new clustering method recognizes the minimum number of complete linkage clusters that are exhaustive. As the data structures in Fig. 4.1 show, by doing so, the new method resolves the classical migration problem.

Seventh, like the standard complete linkage method, the new clustering method has simplicity. However, the model for a measured value that scientists and engineers commonly use has substantially broader applicability than the taxonomic model that is the basis for the standard complete linkage method.

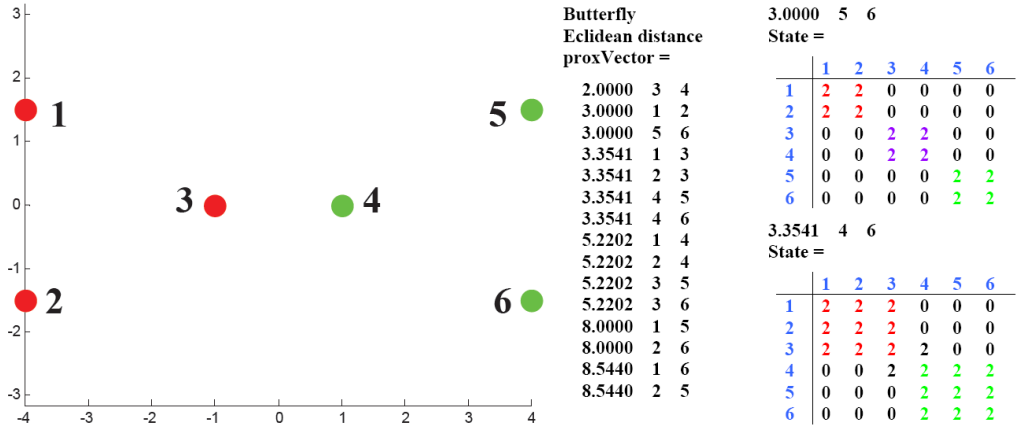


Figure 4.1: How the classical migration problem is resolved by INCLude hierarchical clustering. The different colored (gray scaled) numerals in the state matrices indicate which data points belong to the same cluster.

4.1.2 Overview of the INCLude Hierarchical Clustering Method

Fig. 4.2 is an activity diagram of agglomerative INCLude hierarchical clustering. The only input that the new method requires is a data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, where x_i, $i = 1, 2, ..., n$, are data points comprised of finite sequences of samples. The output includes a sequence of lists that comprise a hierarchical sequence of cluster sets and a sequence of *clusterTree*s, as described *infra*. The new method recognizes the minimum number of complete linkage clusters that are exhaustive. It assumes that clusters are globular or compact, and preferably comprised of maximally complete subsets of data points. When the new method constructs only the cluster sets for meaningful levels of a hierarchical sequence, it further assumes that the 2-norms and the 1-norms of the data points are calculable, that noise (random error) is the only random component in a measured value, that noise can be modeled as Gaussian random variables[1] , and that the noise that is embedded in each dimension (sample) of each data point is statistically independent. The new method can be divided into five tasks: loading a data set, calculating distances between the data points and constructing ordered triples, sorting

[1] Assuming that noise can be modeled as Gaussian random variables makes the proofs easier. When noise is assumed to be uniformly distributed, the results are analogous to those when noise is assumed to be normally distributed, indicating that the Gaussian random variables assumption is reasonable. *See* Chapter 5.

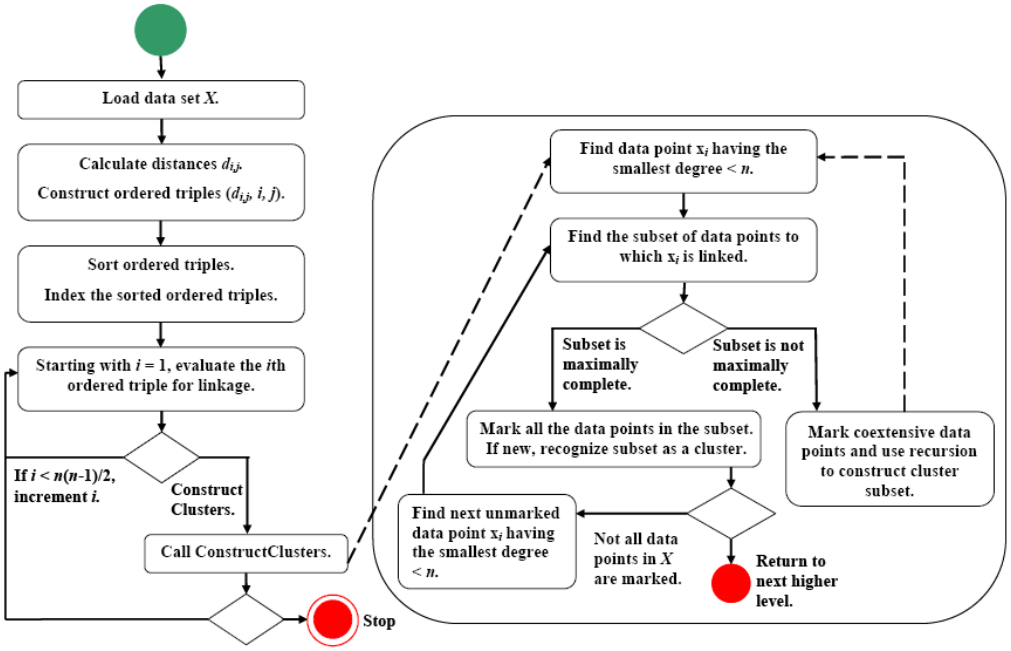


Figure 4.2: Activity diagram for agglomerative INCLude hierarchical clustering.

the ordered triples, evaluating pairs of data points for linkage, and constructing cluster sets. Optionally (not shown), the new method performs a test to determine which levels of a hierarchical sequence are deemed meaningful.

In brief, once the data points in data set $\mathcal{X}$ are loaded into memory, distances $d_{i,j}$ between data points x_i and x_j, $i, j = 1, 2, ..., n, i \neq j$, are calculated. These distances and the indices of the respective data points are used to construct ordered triples $(d_{i,j}, i, j)$, the ordered triples are sorted into rank order according to their distance elements, and the sorted ordered triples are indexed (the "rank order indices"). Next, the ordered triples (pairs of data points) are evaluated for linkage in ascending order. Linkage between the data points in a data set is controlled by a continuous variable, threshold distance (index) $d' \in R$. Data points x_i and x_j, $i, j = 1, 2, ..., n, i \neq j$, are linked if the distance between them is less than or equal to threshold distance d', i.e., $d_{i,j} \leq d'$. After the ith ordered triple is evaluated, $i = 1, 2, ..., \frac{n \cdot (n-1)}{2}$, a decision gets made whether to construct a cluster set for the ith level of the corresponding hierarchical sequence. If so, ConstructClusters is called. If not, the next ordered triple is evaluated.

Whether a level is meaningful can be used to decide whether to construct a cluster set. When the 2-norms or the 1-norms of the data points are calculable, Chapter 5 shows how to use distance graphs and high(er) dimensionalities to find meaningful levels of a hierarchical sequence. To mathematically capture the graphical relationships that underlie the features of a distance graph that correlate with meaningful levels of the corresponding hierarchical sequence, the rank order indices that coincide with meaningful levels of the corresponding hierarchical sequence, or the distance elements that coincide with the respective threshold distances d', must be identifiable *without* visually examining the curve of the distance graph. After each ordered triple is evaluated for linkage, the test described in Chapter 6 is performed to determine whether the corresponding level of the hierarchical sequence is deemed meaningful. If the test returns true after the ith ordered triple is evaluated, the ith level of the hierarchical sequence is constructed.

ConstructClusters constructs cluster sets. ConstructClusters begins by finding data point x_i having the smallest degree. *If at least one data point in each cluster is well-separated*, finding the data points having the smallest degree is a provably efficient way to find and construct only those clusters that are necessary for constructing a cluster set. (See the proof at the end of this chapter.) Further, the cluster sets are constructible without employing recursion to solve clustering subproblems (there are none). Once data point x_i is identified, ConstructClusters finds every data point to which x_i is linked.

If this subset of data points is maximally complete, it is recognized as a new cluster, and all the data points in the subset are marked. Marking data points prevents them from being (re)selected as one of the data points having the smallest degree. It is used to prevent redundant cluster construction, to narrow the search for necessary clusters without eliminating any of them, and to resolve the classical migration problem. If the subset is not maximally complete, it is treated as a clustering subproblem that can be solved by calling ConstructClustersSubpr. When a subset is not maximally complete, only those data points whose linkage is coextensive with that of data point x_i (including x_i) are marked. ConstructClusters continues by finding the next unmarked data point having the smallest degree and reiterates until all the data points in data set $\mathcal{X}$ are marked. Then, it returns, and the next ordered triple is evaluated.

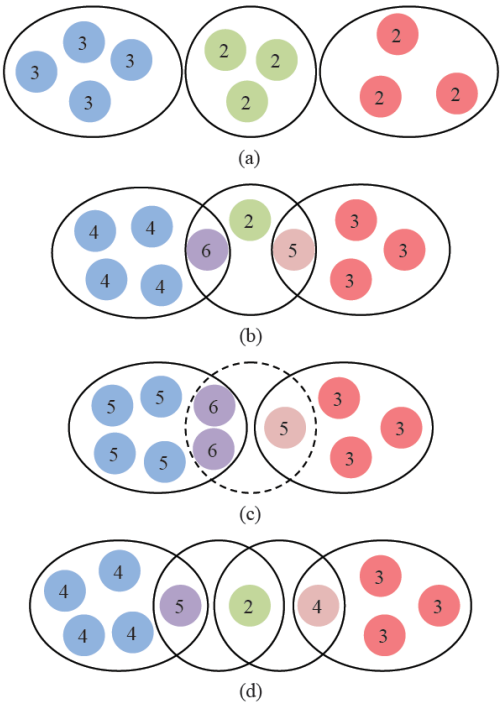


Figure 4.3: Cluster patterns encountered by the new clustering method. The dots represent data points, and the numbers represent the degrees of the respective data points.

Fig. 4.3 is an illustration of the four clustering patterns that are encountered by the new clustering method. In Fig. 4.3(a), at least one data point in each cluster is well-separated, and the clusters do not overlap. In Fig. 4.3(b), at least one data point in each cluster is well-separated, and the clusters overlap. The degree of a data point that belongs to overlapping clusters is greater than the degrees of those data points that belong to only one of the clusters. In Fig. 4.3(c), the data points encircled by the dashed line also belong to clusters wherein at least one data point is well-separated. These data points migrate to the other clusters, i.e., they are marked when the other clusters are constructed, so a cluster is not constructed for this subset. These three patterns comprise ideal circumstances, because at least one data point in each constructed cluster is well-separated. In these circumstances, cluster sets can be constructed without employing recursion to solve clustering subproblems.

As Fig. 4.3(d) illustrates, when a subset of data points is not maximally complete, nonetheless, it comprises a subset of overlapping clusters. The subset of data points is treated as a clustering subproblem, and recursion is employed to find the overlapping clusters. When inherent structure in a data set is weak[2] , it has been observed that recursion is often employed. Extensive use of recursion is undesirable for many if not most real-time systems. For at least this reason, it may be preferable to limit the depth to which recursion is employed and list the data points that comprise a (sub-)subproblem. If recursion is blocked completely, the maximum possible diameter of such a list will be $2d'$, and the maximum possible error will be d'. On the other hand, when inherent structure in a data set is strong, it is easy to identify meaningful levels of a hierarchical sequence and construct only the cluster sets for these levels. Then, recursion is employed less often, if at all. Where weak inherent structure is a consequence of noise that is embedded in the dimensions of the data points, under the above-mentioned, broadly applicable assumptions, increasing the dimensionality of (number of samples in) the data points can attenuate the effects of the noise on cluster set construction. *See* Chapter 5.

When a subset of data points is not maximally complete, each overlapping cluster is constructed because each cluster includes data point x_i. This may be viewed as an "equal dignity" rule. Because not all the other data points in the subset belong to each of the overlapping clusters, and some of the other data points belong to clusters to which data point x_i does not belong, only those data points whose linkage is coextensive with that of data point x_i (including x_i) are marked. Consequently, it is necessary to have one or more mechanisms that check for redundant cluster construction. Discovering a means for finding meaningful levels of a hierarchical sequence and designing these mechanisms were the two greatest challenges posed by the project.

4.1.3 Data Structures

The following nine data structures are used to implement the new clustering method. The last eight data structures, which are illustrated in Figs. 4.4 and 4.5, are used to construct clusters.

[2] Inherent structure is a more general notion than well-separateness. A data set can have inherent structure even though the clusters are not well-separated and may even overlap.

- *proxVector* is an $\frac{n\cdot(n-1)}{2}$ x 3 vector. An instance of *proxVector* is created and used in Pseudocode 1, described *infra*. Once the data points in data set $\mathcal{X}$ are loaded and stored in a data array and distances $d_{i,j}$ between data points x_i and x_j, $i, j = 1, 2, ..., n, i \neq j$, are calculated, these distances and the indices of the respective data points are used to construct ordered triples $(d_{i,j}, i, j)$. The ordered triples are stored in *proxVector*.

- *State* is an n x n symmetric matrix. An instance of *State* is created in Pseudocode 1, where it stores information about linkage. *State* is passed to ConstructClusters, where it is used to construct cluster sets. The row indices of *State* correspond to the indices of the data points.

- *Degrees* is an n x 2 list. An instance of *Degrees* is created in Pseudocode 1, where it stores the global (data array) index and the degree of each data point in data set $\mathcal{X}$. *Degrees* is passed with *State* to ConstructClusters, where it is used to construct cluster sets. The row indices of *Degrees* correspond to the row indices of *State* and the indices of the data points.

- *linkedNodesList* is a 2-column list. Instances of *linkedNodesList* are created in ConstructClusters and ConstructClustersSubpr, where they are used for cross-referencing between the global indices and the local indices of data points. Unless a data point is a singleton, a *linkedNodesList* is created for each subset described *supra.*

- *newState* is a symmetric submatrix that is constructed from entries in a *State* matrix. The row size of *newState* is the same size as the corresponding subset of data points. Instances of *newState* are created in ConstructClusters and ConstructClustersSubpr. When recursion is employed to solve a clustering subproblem, *newState* is passed to ConstructClustersSubpr, where it is used like *State* to construct cluster (sub)sets.

- *newDegrees* is a 2-column list. The row size of *newDegrees* is the same as that of the corresponding instance of *newState*. Instances of *newDegrees* are created in ConstructClusters and ConstructClustersSubpr. When recursion is employed to solve a clustering subproblem, *newDegrees* is passed to ConstructClustersSubpr, where it is used like *Degrees* to construct cluster (sub)sets. Unlike *newState*, however, *newDegrees* is not a sublist of *Degrees*. It is constructed *de novo*.

- *coverList* is a list that stores information about previously constructed clusters

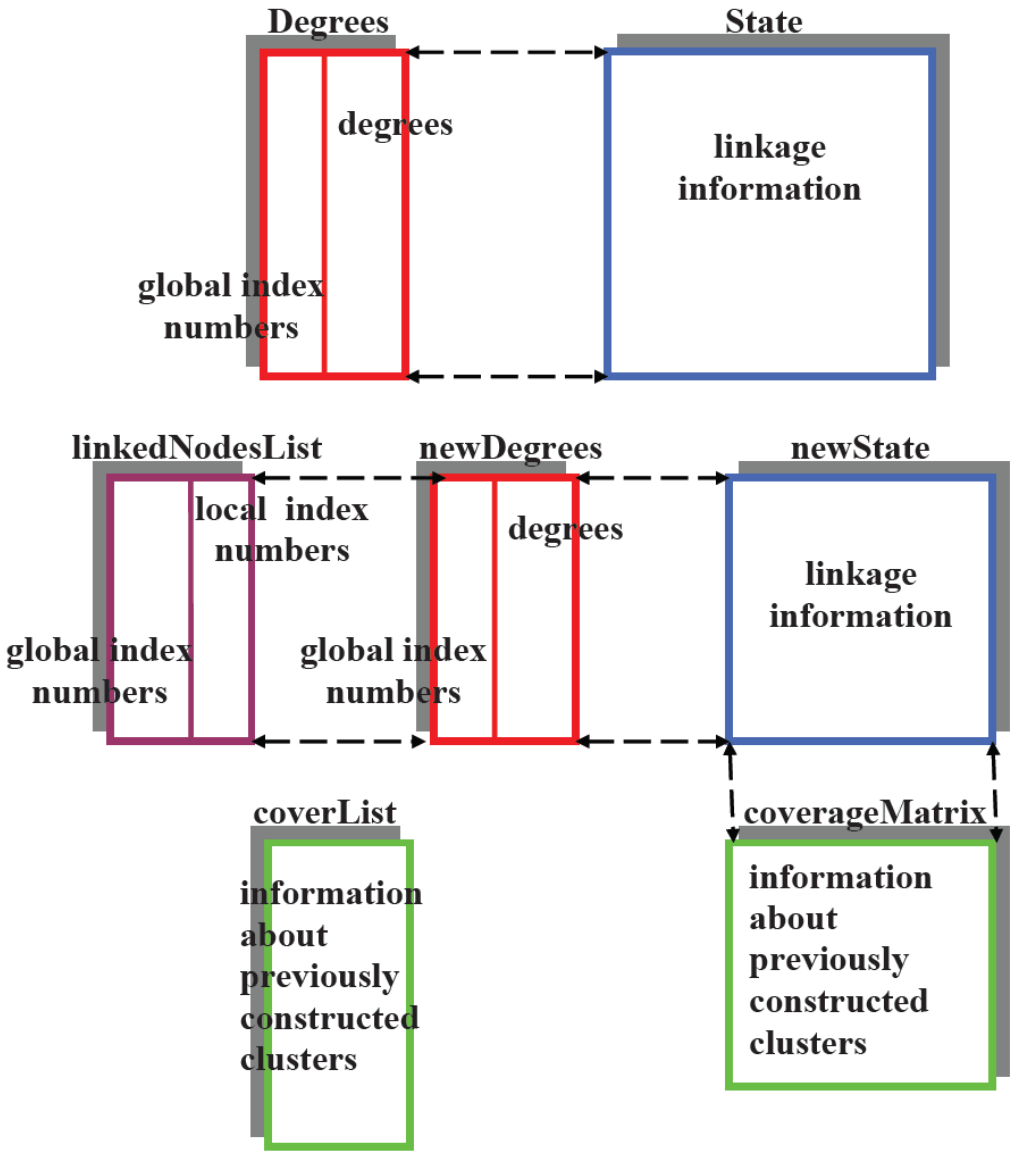


Figure 4.4: Seven of the nine data structures that are used in the new clustering method. The other two data structures are *proxVector* and *clusterTree*.

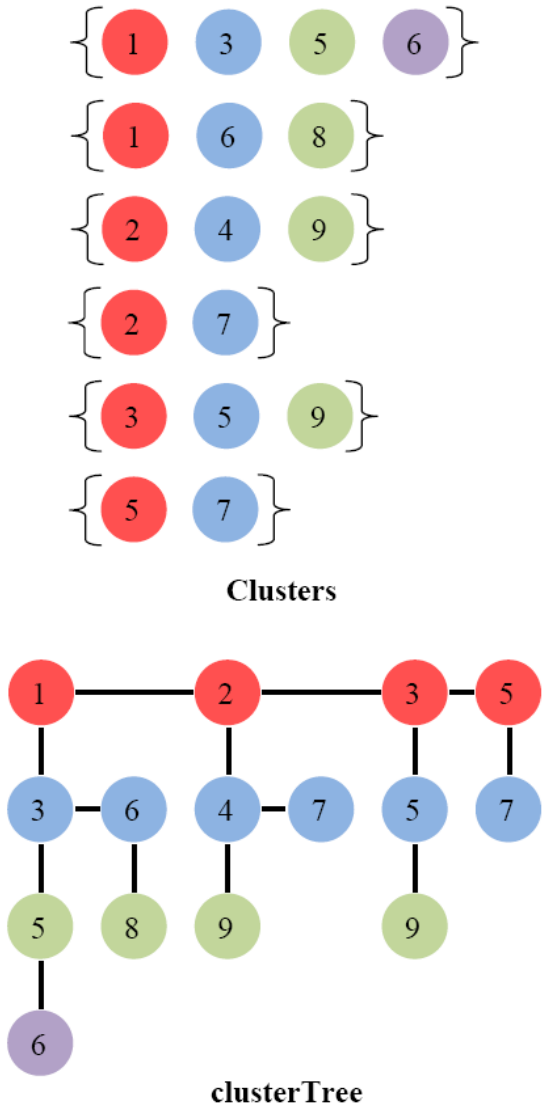


Figure 4.5: Illustration of a cluster set comprised of six clusters and their representation as a *clusterTree*. The dots represent data points, and the numbers represent the global indices of the respective data points.

(with respect to a particular cluster set). This information is used to avoid constructing redundant clusters. Instances of *coverList* are created in ConstructClusters.

- *coverageMatrix* is a symmetric matrix. Instances of *coverageMatrix* are created in ConstructClustersSubpr, where they store information about previously constructed clusters. *coverageMatrix* is the same size as the instance of *newState* that was passed to ConstructClustersSubpr.
- *clusterTree* is an adaptation of a binary search tree. It is used to store the clusters that comprise a cluster set. The right children are data points that have the same position in each of the clusters of a cluster set, where position is determined from the leftmost data point. A parent and a left child come from the same cluster. A left child is the data point that comes after the data point that is its parent.

4.2 Pseudocode

The following pseudocode describes how to implement the new clustering method. It is written for agglomerative hierarchical clustering. With minor modifications to Pseudocode 1, it can be used to implement divisive hierarchical clustering as well.

4.2.1 Pseudocode 1: Evaluating Ordered Triples for Linkage

Pseudocode 1 Line 1. The INCLude algorithm accepts data set $\mathcal{X}$, the size n of data set $\mathcal{X}$, a first optional control parameter *Guard*, and a second optional control parameter *stoppingCriteria* as input. Data points are stored as rows in a data array, the row indices of which are used to refer to the data points. n can be determined as or after data set $\mathcal{X}$ is input. *Guard* controls when or how often cluster sets are constructed, and *stoppingCriteria* controls when the algorithm terminates. Possible control parameters include rank order indices, distance elements $d_{i,j}$, and the numbers of clusters in cluster sets. So that the rank order of the ordered triples does not matter when ties arise between their distance elements, *Guard* and *stoppingCriteria* should be compared with $d_{i,j}$.

Pseudocode 1 Lines 2-5. After instances of *proxVector*, *State*, and *Degrees* are created, distances $d_{i,j}$ between data points x_i and $x_j, i, j = 1, 2, ..., n, i \neq j$, are calculated, ordered triples $(d_{i,j}, i, j)$ are constructed from these distances and the indices of

Pseudocode 1 Evaluating Ordered Triples for Linkage

1: **Input:** $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, $n \in I$, *Guard* $\in R$, and *stoppingCriteria* $\in R$
2: Define $\frac{n \cdot (n-1)}{2}$ x 3 array *proxVector* = 0, n x n array *State* = I_n, n x 2 array *Degrees* = [1:n;0:0]′, and linked = 2.
3: **for** $i \leftarrow 1$ **to** $n - 1$ **and** $j \leftarrow i + 1$ **to** n **do**
4: Calculate $d_{i,j}$ and store ordered triple $(d_{i,j}, i, j)$ in *proxVector*.
5: **end for**
6: Sort the ordered triples into rank order according to their distance elements $d_{i,j}$.
7: Define $k \leftarrow 1$. Get $i \leftarrow$ *proxVector*[k,2] and $j \leftarrow$ *proxVector*[k,3].
8: **while** $k \leq \frac{n \cdot (n-1)}{2}$ **and** $d_{i,j} <$ *stoppingCriteria* **do**
9: **if** *State*[i,i] == linked **then** *State*[i,j] ← linked.
10: **else** *State*[i,i] ← linked; *State*[i,j] ← linked.
11: **end if**
12: **if** *State*[j,j] == linked **then** *State*[j,i] ← linked.
13: **else** *State*[j,j] ← linked; *State*[j,i] ← linked.
14: **end if**
15: *Degrees*[i,2] ← *Degrees*[i,2] + 1; *Degrees*[j,2] ← *Degrees*[j,2] + 1.
16: **if** CallConstructClusters(*Guard*, other parameters) == true **then**
Call ConstructClusters(*State*, *Degrees*, n).
17: **end if**
18: $k \leftarrow k + 1$. Get $i \leftarrow$ *proxVector*[k,2] and $j \leftarrow$ *proxVector*[k,3].
19: **end while**

the respective data points, and the ordered triples are stored in *proxVector*. Each row of *proxVector* contains one ordered triple. *proxVector* contains the same information that a proximity matrix would initially contain.

State stores information about linkage that is embedded in the ordered triples, and *Degrees* stores the degrees of the data points. *State* is initialized as an identity matrix because, initially, each data point in data set $\mathcal{X}$ is assigned to its own cluster. See Example 2, *infra*, for an example of *proxVector* and *State*.

The algorithm can cluster metric or nonmetric data. The time complexity to calculate distances and construct ordered triples is $O(\frac{n \cdot (n-1) \cdot m}{2})$.

Pseudocode 1 Line 6. The ordered triples are sorted into rank order according to their distance elements, and the row indices of *proxVector* are used to index the ordered triples (the "rank order indices"). Other than shifting the threshold distances d' at which cluster sets are constructed, monotonic transformations of the distances $d_{i,j}$ have no effect on cluster set construction because the rank order of the ordered triples is preserved. (See the proof at the end of this chapter.) Ties between distance elements are resolved by comparing the indices of the respective data points. Using merge sort [34], the time complexity to sort the ordered triples is $O(\frac{n\cdot(n-1)}{2} \cdot log(\frac{n\cdot(n-1)}{2})) = O(n \cdot (n-1) \cdot log\sqrt{\frac{n\cdot(n-1)}{2}})$.

Pseudocode 1 Lines 7-15, 18, and 19. The ordered triples are evaluated in ascending order for linkage, which information is stored in *State*. The following numerals are used as symbols or indicators to represent information about linkage in *State*.

- A "2" indicates that 1) a data point is linked to another data point or a pair of data points are linked and 2) the data point or pair of data points have not been included in a previously constructed cluster (with respect to a particular cluster set).
- A "1" indicates that a data point is not linked to another data point, i.e., it is a singleton.
- A "0" indicates that a pair of data points are not linked.
- A "-2" indicates that 1) a data point is linked to another data point or a pair of data points are linked and 2) the data point or pair of data points have been included in a previously constructed cluster (with respect to a particular cluster set).

Each data point in each ordered triple is evaluated according to one of two rules. If data point x_i is a singleton (*Degree*[i,2] = 0 at the time of the evaluation), *State*[i,i] changes from "1" to "2", and *State*[i,j] changes from "0" to "2". If data point x_i is not a singleton (*Degree*[i,2] > 0 at the time of the evaluation), *State*[i,j] changes from "0" to "2". Analogous rules apply when data point x_j is evaluated. Even though *State* is symmetric, the upper and lower parts are completed to make cluster set construction easier. As the information about linkage is stored in *State*, the degrees of data points x_i and x_j are incremented in *Degrees*.

As the ordered triples are evaluated and the distance elements become larger, threshold distance d' implicitly increases from 0 to the maximum of all the distance elements. Although threshold distance d' can vary continuously from 0 (where each data point is

a singleton) to at least this maximum distance (where all the data points belong to the same cluster), the only values that matter are those $\frac{n\cdot(n-1)}{2}$ values that are equal to the distance elements $d_{i,j}$.

Each ordered triple is evaluated in turn until all the ordered triples are evaluated or *stoppingCriteria* is met. From the information about linkage that is stored in *State* and the degrees of the data points, a set of maximally complete clusters is constructible. The time complexity to evaluate the ordered triples is $O(\frac{n\cdot(n-1)}{2})$.

Pseudocode 1 Lines 16 and 17. After each ordered triple is evaluated, CallConstructClusters determines whether ConstructClusters is called. If CallConstructClusters returns true, *State*, *Degrees*, and n are passed to ConstructClusters. *State* is passed by value.

Example 2 *This example shows how ordered triples are evaluated. The data in Fig. 4.6 come from a sensor system experiment similar to that described in Chapter 7.*

4.2.2 Pseudocode 2-4: Cluster Set Construction

Pseudocode 2 Line 2. Cluster sets evolve as a function of threshold distance d'. Instances of *copyOfDegrees*, *recursionLevel*, and *maxRecursion* are created. *copyOfDegrees* is used to track which data points are marked and find unmarked data points having the smallest degree. Data point x_i is marked by increasing *copyOfDegrees*[i,2] to $n+1$, i.e., a number that is greater than the largest possible degree of any data point in data set $\mathcal{X}$. *Degrees* remains unchanged because it is used in a conditional, described *infra*. *recursionLevel* and *maxRecursion* are global variables. *recursionLevel* tracks the depth at which recursion is used to solve a subproblem, and *maxRecursion* limits the depth to which recursive calls are allowed.

Pseudocode 2 Line 4. ConstructClusters finds data point x_i having the smallest degree and uses this data point to initiate constructing at least one cluster. For example, if *Degrees*$[i,2] = 1$, x_i comprises a cluster with the data point to which it is linked. Ties are resolved by selecting the data point having the smallest index. *The new method is designed so that the order in which the data points are selected does not matter.* This is accomplished by how the data points are marked and with

9 Motes Sensing Light
Euclidean distance
proxVector =

96.64	1	8
128.22	6	8
141.72	4	9
171.46	1	6
187.13	2	9
285.69	2	4
922.47	2	7
964.10	3	5
987.37	7	9
…		
1874.55	4	6

d' = 00.00
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0

d' = 96.64
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	0	0	2	0	1
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	0	0	2	0	1
9	0	0	0	0	0	0	0	0	1	0

d' = 128.22
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	0	0	2	0	1
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	2	0	2	0	1
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	0	0	0	0	0	0	0	1	0

d' = 141.72
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	0	0	2	0	1
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	2	0	0	0	0	2	1
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	2	0	2	0	1
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	0	0	2	0	0	0	0	2	1

d' = 171.46
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	2	0	2	0	2
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	2	0	0	0	0	2	1
5	0	0	0	0	1	0	0	0	0	0
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	0	0	2	0	0	0	0	2	1

d' = 187.13
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	0	0	0	0	0	2	1
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	2	0	0	0	0	2	1
5	0	0	0	0	1	0	0	0	0	0
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

d' = 285.69
State =

	1	2	3	4	5	6	7	8	9	Degrees=
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	2	0	0	0	0	2	2
3	0	0	1	0	0	0	0	0	0	0
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	1	0	0	0	0	0
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

Figure 4.6: Ordered triples and data structures from a sensor system experiment similar to that described in Chapter 7. Euclidean distance was used to calculate the distances.

Pseudocode 2 ConstructClusters

1: **Input:** *State*, *Degrees*, and n.
2: Define *copyOfDegrees* = *Degrees*, global *recursionLevel* = 0, global *maxRecursion* $\in I$, global *clusterTree* = null, and mark = $n + 1$.
3: **repeat**
4: Find i so that *copyOfDegrees*[i,2] = min(*copyOfDegrees*[:,2]).
5: **if** *copyOfDegrees*[i,2] == 0 **then**
6: Recognize x_i as a singleton.
7: *State*[i,i] $\leftarrow$ -2.
8: *copyOfDegrees*[i,2] $\leftarrow$ mark.
9: **else if** *copyOfDegrees*[i,2] > 0 **then**
10: Define *linkCount* = 0 and n x 2 array *linkedNodesList* = [0:0,0:0]'.
11: **for** $j \leftarrow 1$ **to** n **do**
12: **if** $|State[i, j]| == 2$ **then**
13: *linkCount* $\leftarrow$ *linkCount* + 1.
14: Store *copyOfDegrees*[j,1] in column one of *linkedNodesList*.
15: Store j in column two of *linkedNodesList*.
16: **end if**
17: **end for**

clusterTree, an adaptation of a binary search tree that is used for checking redundant cluster construction.

Pseudocode 2 Lines 5-8. If the degree of data point x_i is zero, x_i is treated as a singleton and marked. *State*[i,i] changes from "1"to "-2", to indicate that x_i has been included in a cluster.

Pseudocode 2 Lines 9-17 and Pseudocode 4 Line 18. If the degree of data point x_i is greater than zero, instances of *linkCount* and *linkedNodesList* are created. *linkCount* stores the number of data points to which x_i is linked (including x_i), and *linkedNodesList* stores the global (data array) index and the local index of each such data point. The global indices are obtained from the first column of *Degrees* or *copyOfDegrees*. The local indices correspond to the indices of the rows in which these data points appear in *Degrees* or *copyOfDegrees*. To find the subset of data points to which x_i

Pseudocode 3 ConstructClusters

```
1:  Define linkCount x linkCount array newState = -2, linkCount x 2 array
    newDegrees = [linkedNodesList[1:linkCount,1], (linkCount-1):(linkCount-1)]',
    newClusterFlag = 0, linkCount^2 x 2 array coverList = [0:0,0:0]'.
2:  for j ← 1 to linkCount do
3:    for k ← 1 to linkCount do
4:      if State[linkedNodesList[j,2],linkedNodesList[k,2]] == 2 then
5:        newClusterFlag ← 1.
6:        newState[j,k] ← 2.
7:        Store j and k in coverList.
8:      else if State[linkedNodesList[j,2],linkedNodesList[k,2]] ≥ 0 then
9:        newState[j,k] ← 0.
10:       newDegrees[j,2] ← newDegrees[j,2] - 1.
11:     end if
12:   end for
13:   Mark those data points whose linkage is coextensive with that of x_i.
14: end for
```

is linked, ConstructClusters scans the ith row of *State* for linkage, i.e., whether $|State[i,j]| = 2, j = 1, 2, ..., n$. Where linkage is indicated, *linkCount* is incremented, and the global and local indices of x_j are stored in *linkedNodesList*.

Pseudocode 3 Lines 1-12 and 14. Two tasks are performed concurrently for each subset of data points. First, ConstructClusters determines whether a subset is maximally complete. Second, as linkage is being checked, a clustering subproblem is set up for the subset. If the subset is not maximally complete, recursion can be employed to solve the subproblem.

To set up a clustering subproblem, instances of *newState*, *newDegrees*, *coverList*, and *newClusterFlag* are created. The row size of *newState* and the row size of *newDegrees* equal *linkCount*. The entries in the first column of *newDegrees* are set to the entries in the first column of *linkedNodesList* (the global indices). As a design decision, the entries in the second column are initialized to *linkCount* - 1, the maximum possible degree of any data point in the subset, based only on linkage among these data points. Likewise,

Pseudocode 4 ConstructClusters

```
1:    if the subset of data points in linkedNodesList is maximally complete then
2:      Mark all the data points in the subset.
3:      if newClusterFlag == 1 or the subset does not appear in clusterTree then
4:        Recognize the subset as a new cluster, and
          include the new cluster in clusterTree.
5:      end if
6:    else if linkCount < n then
7:      recursionLevel ← recursionLevel + 1.
8:      if recursionLevel < maxRecursion then
9:        Call ConstructClustersSubpr(newState, newDegrees, linkCount).
10:     else
11:       List the data points in the subset.
12:     end if
13:     recursionLevel ← recursionLevel - 1.
14:   end if
15:   if indices have been stored in coverList then
16:     Set the corresponding entries in State to -2.
17:   end if
18:  end if
19: until min(copyOfDegrees[:,2]) = mark
```

all the entries in *newState* are initialized to "-2". For any two data points x_j and x_k that belong to the subset, $j, k = 1, 2, ..., linkCount$, if x_j is linked to x_k but x_j and x_k have not been included in the same previously constructed cluster, i.e., *State*[j,k] = 2, *newClusterFlag* is set to 1, the corresponding entry in *newState* is set to "2", and the indices of the data points are stored in *coverList*. If x_j is not linked to x_k, i.e., *State*[j,k] = 0, the corresponding entry in *newState* is set to "0", and the degree of x_j is decremented in *newDegrees*.

Pseudocode 3 Line 13. Those data points whose linkage is coextensive with that of data point x_i (including x_i) are marked. The linkage of a data point is coextensive with that of x_i when the data point is linked to every other data point in the subset,

and the degree of the data point, as provided in *Degrees*, equals the degree of x_i.

Pseudocode 4 Lines 1-5. If the subset is maximally complete, all the data points in the subset are marked. If *newClusterFlag* has been set, the subset is recognized as a new cluster and included in *clusterTree*. If *newClusterFlag* has not been set, *clusterTree* must be checked to determine whether the subset comprises a previously constructed cluster. Having all "-2" entries is a necessary but not a sufficient condition for identifying redundant cluster construction. If the subset is already included in *clusterTree*, it is ignored. If the subset is not already included in *clusterTree*, it is recognized as a new cluster and included.

Pseudocode 4 Lines 6-14. If the subset is not maximally complete, the subset nonetheless comprises two or more overlapping clusters. If *linkCount* is less than n, the subset is treated as a clustering subproblem. *linkCount* must be less than n to avoid redefining a clustering problem as a subproblem. If the depth of a recursive call exceeds the maximum allowable depth, further recursion is blocked. Otherwise ConstructClustersSubpr is called. *newState*, *newDegrees*, and *linkCount* are passed to ConstructClustersSubpr. *linkCount* becomes n inside ConstructClustersSubpr.

Pseudocode 4 Lines 15-17. *State* is updated to include information about any newly recognized clusters. The indices stored in *coverList* are used to change the corresponding entries in *State* from "2" to "-2", to indicate that these data points or pairs of data points have been included in a previously constructed cluster. Using *coverList* is a shortcut for a more complex mechanism that is used in ConstructClustersSubpr. When at least one data point in each cluster is well-separated, marking the data points is sufficient to avoid redundant cluster construction. When at least one data point in each cluster is not well-separated, additional mechanisms are needed.

Pseudocode 2 Line 3 and Pseudocode 4 Line 19. Once *State* is updated, ConstructClusters finds the next unmarked data point having the smallest degree and reiterates until all the data points are marked.

Example 3 *This example shows how clusters are constructed when recursion is unnecessary. The data in Figs. 4.7 and 4.8 come from a sensor system experiment similar to that described in Chapter 7.*

9 Motes Sensing Light
Euclidean distance
Rank order index = 6; d' = 285.69

State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	2	0	0	0	0	2	2
3	0	0	1	0	0	0	0	0	0	0
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	1	0	0	0	0	0
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

Data point having the smallest degree = 3
Singleton = 3
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	2	0	0	0	0	2	2
3	0	0	-2	0	0	0	0	0	0	10
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	1	0	0	0	0	0
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

Data point having the smallest degree = 5
Singleton = 5
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	2	0	0	0	0	2	2
3	0	0	-2	0	0	0	0	0	0	10
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	-2	0	0	0	0	10
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	1	0	0	0
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

Data point having the smallest degree = 7
Singleton = 7
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	2	0	0	0	0	2	0	2	0	2
2	0	2	0	2	0	0	0	0	2	2
3	0	0	-2	0	0	0	0	0	0	10
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	-2	0	0	0	0	10
6	2	0	0	0	0	2	0	2	0	2
7	0	0	0	0	0	0	-2	0	0	10
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	0	0	2	2

Data point having the smallest degree = 1
Cluster = {1, 6, 8}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	0	0	0	-2	0	-2	0	10
2	0	2	0	2	0	0	0	0	2	2
3	0	0	-2	0	0	0	0	0	0	10
4	0	2	0	2	0	0	0	0	2	2
5	0	0	0	0	-2	0	0	0	0	10
6	-2	0	0	0	0	-2	0	-2	0	10
7	0	0	0	0	0	0	-2	0	0	10
8	-2	0	0	0	0	-2	0	-2	0	10
9	0	2	0	2	0	0	0	0	2	2

Data point having the smallest degree = 2
Cluster = {2, 4, 9}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	0	0	0	-2	0	-2	0	10
2	0	-2	0	-2	0	0	0	0	-2	10
3	0	0	-2	0	0	0	0	0	0	10
4	0	-2	0	-2	0	0	0	0	-2	10
5	0	0	0	0	-2	0	0	0	0	10
6	-2	0	0	0	0	-2	0	-2	0	10
7	0	0	0	0	0	0	-2	0	0	10
8	-2	0	0	0	0	-2	0	-2	0	10
9	0	-2	0	-2	0	0	0	0	-2	10

Figure 4.7: Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 6. Euclidean distance was used to calculate the distances.

9 Motes Sensing Light
Euclidean distance
Rank order index = 27; d' = 1395.09

State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	2	0	2	0	2	2	2	2	0	5
2	0	2	2	2	2	0	2	0	2	5
3	2	2	2	2	2	2	2	2	2	8
4	0	2	2	2	2	0	2	0	2	5
5	2	2	2	2	2	2	2	2	2	8
6	2	0	2	0	2	2	2	2	0	5
7	2	2	2	2	2	2	2	2	2	8
8	2	0	2	0	2	2	2	2	0	5
9	0	2	2	2	2	0	2	0	2	5

Data point having the smallest degree = 1
Cluster = {1, 3, 5, 6, 7, 8}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	-2	0	-2	-2	-2	-2	0	10
2	0	2	2	2	2	0	2	0	2	5
3	-2	2	-2	2	-2	-2	-2	-2	2	10
4	0	2	2	2	2	0	2	0	2	5
5	-2	2	-2	2	-2	-2	-2	-2	2	10
6	-2	0	-2	0	-2	-2	-2	-2	0	10
7	-2	2	-2	2	-2	-2	-2	-2	2	10
8	-2	0	-2	0	-2	-2	-2	-2	0	10
9	0	2	2	2	2	0	2	0	2	5

Data point having the smallest degree = 2
Cluster = {2, 3, 4, 5, 7, 9}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	-2	0	-2	-2	-2	-2	0	10
2	0	-2	-2	-2	-2	0	-2	0	-2	10
3	-2	-2	-2	-2	-2	-2	-2	-2	-2	10
4	0	-2	-2	-2	-2	0	-2	0	-2	10
5	-2	-2	-2	-2	-2	-2	-2	-2	-2	10
6	-2	0	-2	0	-2	-2	-2	-2	0	10
7	-2	-2	-2	-2	-2	-2	-2	-2	-2	10
8	-2	0	-2	0	-2	-2	-2	-2	0	10
9	0	-2	-2	-2	-2	0	-2	0	-2	10

Figure 4.8: Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 27. Euclidean distance was used to calculate the distances.

4.2.3 Cluster Set Construction Subproblems

ConstructClustersSubpr is similar to ConstructClusters. Rather than describe ConstructClustersSubpr in detail, the differences between ConstructClustersSubpr and ConstructClusters are highlighted. Two notable differences distinguish ConstructClustersSubpr from ConstructClusters. First, ConstructClusters is responsible for identifying singleton data points, marking these data points, and updating *State* accordingly. Second, ConstructClustersSubpr and ConstructClusters use different albeit related mechanisms for tracking information about previously constructed clusters.

To track information about previously constructed clusters, ConstructClustersSubpr uses two data structures, *coverageMatrix* and *clusterTree*. ConstructClustersSubpr follows the same steps as ConstructClusters, except *coverageMatrix* is updated instead of storing indices in *coverList* and subsequently updating *State*. *coverageMatrix* is used to construct *newState* before it is passed to ConstructClustersSubpr. If a subset is maximally complete and *newClusterFlag* has been set, the subset is recognized as a new cluster and included in *clusterTree*. If *newClusterFlag* has not been set, *clusterTree* must be checked to determine whether the subset comprises a previously constructed cluster.

In summary, three mechanisms are used to avoid recognizing redundant clusters. First, data points are marked so that they cannot be (re-)selected as one of the data points having the smallest degree. Second, *coverList* and *coverageMatrix* are used to track information about previously constructed clusters. Third, *clusterTree* is used to store subsets of data points that have been recognized as new clusters. If a subset of data points is maximally complete but the entries in *State* for every data point and every pair of data points are "-2", *clusterTree* is used to determine whether the subset comprises a new cluster or a previously constructed cluster.

Example 4 *This example shows how clusters are constructed when recursion is employed. The data in Figs. 4.9 and 4.10 come from a sensor system experiment similar to that described in Chapter 7.*

Example 5 *The data in Fig. 4.11 come from the 17-points geometric pattern experiment described in Chapter 7. Euclidean distance was used to calculate the distances.*

The blue (gray scaled) subsets of data points were recognized as clusters for rank order index = 54. The right column includes a subset that was recognized as a cluster, even though the entries in State for every data point and every pair of data points in the subset were "-2"s, because the subset was not already included in clusterTree. The reader is encouraged to construct clusterTree for this example.

9 Motes Sensing Light
Euclidean distance
Rank order index = 12; d' = 1090.11

State =

	1	2	3	4	5	6	7	8	9	copyOfDegrees=
1	2	0	2	0	0	2	0	2	0	3
2	0	2	0	2	0	0	2	0	2	3
3	2	0	2	0	2	0	0	0	0	2
4	0	2	0	2	0	0	2	0	2	3
5	0	0	2	0	2	0	2	0	0	2
6	2	0	0	0	0	2	0	2	0	2
7	0	2	0	2	2	0	2	0	2	4
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	2	0	2	3

Data point having the smallest degree = 3
Recursive Call
State =

	1	3	5	copyOfDegrees=
1	2	2	0	1
3	2	2	2	2
5	0	2	2	1

Data point having the smallest degree = 1
Cluster = {1, 3}
State =

	1	3	5	copyOfDegrees=
1	2	2	0	4
3	2	2	2	4
5	0	2	2	1

Data point having the smallest degree = 5
Cluster = {3, 5}
State =

	1	3	5	copyOfDegrees=
1	2	2	0	4
3	2	2	2	4
5	0	2	2	4

Data point having the smallest degree = 3
Clusters = {1, 3}, {3, 5}
State =

	1	2	3	4	5	6	7	8	9	copyOfDegrees=
1	-2	0	-2	0	0	2	0	2	0	3
2	0	2	0	2	0	0	2	0	2	3
3	-2	0	-2	0	-2	0	0	0	0	10
4	0	2	0	2	0	0	2	0	2	3
5	0	0	-2	0	-2	0	2	0	0	2
6	2	0	0	0	0	2	0	2	0	2
7	0	2	0	2	2	0	2	0	2	4
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	2	0	2	3

Data point having the smallest degree = 5
Recursive Call
State =

	3	5	7	copyOfDegrees=
3	-2	-2	0	1
5	-2	-2	2	2
7	0	2	2	1

Figure 4.9: Data structures from a sensor system experiment similar to that described in Chapter 7. The data is for rank order index = 12. Euclidean distance was used to calculate the distances.

Data point having the smallest degree = 3
Redundant Cluster
State = **copyOfDegrees=**

	3	5	7	
3	-2	-2	0	4
5	-2	-2	2	2
7	0	2	2	1

Data point having the smallest degree = 7
Cluster = {5, 7}
State = **copyOfDegrees=**

	3	5	7	
3	-2	-2	0	4
5	-2	-2	2	4
7	0	2	2	4

Data point having the smallest degree = 5
Cluster = {5, 7}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	-2	0	0	2	0	2	0	3
2	0	2	0	2	0	0	2	0	2	3
3	-2	0	-2	0	-2	0	0	0	0	10
4	0	2	0	2	0	0	2	0	2	3
5	0	0	-2	0	-2	0	-2	0	0	10
6	2	0	0	0	0	2	0	2	0	2
7	0	2	0	2	-2	0	-2	0	2	4
8	2	0	0	0	0	2	0	2	0	2
9	0	2	0	2	0	0	2	0	2	3

Data point having the smallest degree = 6
Cluster = {1, 6, 8}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	-2	0	0	-2	0	-2	0	10
2	0	2	0	2	0	0	2	0	2	3
3	-2	0	-2	0	-2	0	0	0	0	10
4	0	2	0	2	0	0	2	0	2	3
5	0	0	-2	0	-2	0	-2	0	0	10
6	-2	0	0	0	0	-2	0	-2	0	10
7	0	2	0	2	-2	0	-2	0	2	4
8	-2	0	0	0	0	-2	0	-2	0	10
9	0	2	0	2	0	0	2	0	2	3

Data point having the smallest degree = 2
Cluster = {2, 4, 7, 9}
State = **copyOfDegrees=**

	1	2	3	4	5	6	7	8	9	
1	-2	0	-2	0	0	-2	0	-2	0	10
2	0	-2	0	-2	0	0	-2	0	-2	10
3	-2	0	-2	0	-2	0	0	0	0	10
4	0	-2	0	-2	0	0	-2	0	-2	10
5	0	0	-2	0	-2	0	-2	0	0	10
6	-2	0	0	0	0	-2	0	-2	0	10
7	0	-2	0	-2	-2	0	-2	0	-2	10
8	-2	0	0	0	0	-2	0	-2	0	10
9	0	-2	0	-2	0	0	-2	0	-2	10

Figure 4.10: Continuation of Fig. 4.9.

17-Points Geometric Pattern
Euclidean distance
Rank order index = 54

Cluster =
1 2 3 4
Cluster =
5 6 7 8
Cluster =
9 10 11 12
Cluster =
13 14 15 16
Cluster =
1 3 4 17
Cluster =
1 3 13 14 17
Possible Redundant Cluster =
1 3 4 17
Cluster =
1 4 5 6 17
Cluster =
5 6 7 17
Possible Redundant Cluster =
1 4 5 6 17
Possible Redundant Cluster =
5 6 7 17
Cluster =
5 7 9 10 17
Cluster =
9 10 11 17
Possible Redundant Cluster =
5 7 9 10 17
Possible Redundant Cluster =
9 10 11 17
Cluster =
9 11 13 15 17
Possible Redundant Cluster =
13 14 15 17
Possible Redundant Cluster =
1 3 13 14 17
Possible Redundant Cluster =
13 14 15 17
Possible Redundant Cluster =
9 11 13 15 17
Possible Redundant Cluster =
1 4 5 6 17
Cluster =
1 5 9 13 17
Possible Redundant Cluster =
1 3 13 14 17
Possible Redundant Cluster =
1 4 5 6 17
Possible Redundant Cluster =
5 7 9 10 17
Possible Redundant Cluster =
1 5 9 13 17
Possible Redundant Cluster =
1 5 9 13 17
Possible Redundant Cluster =
5 7 9 10 17
Possible Redundant Cluster =
9 11 13 15 17
Possible Redundant Cluster =
1 3 13 14 17
Possible Redundant Cluster =
1 5 9 13 17
Possible Redundant Cluster =
9 11 13 15 17

Figure 4.11: Trace for rank order index = 54 of the 17-points geometric pattern experiment described in Chapter 7. Euclidean distance was used to calculate the distances. The blue (gray scaled) subsets of data points were recognized as clusters.

4.3 Theorems and Proofs for the INCLude Hierarchical Clustering Method and Algorithm

4.3.1 Transformational Invariance

Theorem 1 *Let $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ be a finite set of data points. With respect to any distance, if a transformation of the distances between the data points in a data set is invariant, the results obtained from INCLude hierarchical clustering are invariant.*

Proof for Theorem 1 *Assume that the stated conditions are true. The INCLude algorithm sorts the ordered triples $(d_{i,j}, i, j)$, $i, j = 1, 2, ..., n, i \neq j$, into rank order and evaluates the ordered triples in ascending order. Thus, a first ordered triple is evaluated for linkage before a second ordered triple is evaluated for linkage if its distance element is smaller than that of the second ordered triple. In other words, for any first pair of data points x_h and x_i and any second pair of data points x_j and x_k, $h, i, j, k = 1, 2, ..., n, h \neq i, j \neq k$, if $d_{h,i} < d_{j,k}$, the information about linkage between x_h and x_i is stored in State before the information about linkage between x_j and x_k. As long as a transformation of the distance elements does not alter their relative sizes (less than, equal to, or greater than), the transformation will not affect the order in which the ordered triples are evaluated. Thus, the results obtained from INCLude hierarchical clustering are invariant.*
QED

For example, scaling and translational transformations of any p-norm and rotational transformations of the 2-norm are invariant, so these kinds of transformations do not affect cluster set construction.

4.3.2 Optimality

Theorem 2 *Let $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ be a finite set of data points. For any threshold distance d', if each cluster in a cluster set constructed by INCLude hierarchical clustering includes at least one data point that is well-separated, the cluster set is optimal.*

Proof for Theorem 2 *Assume that the stated conditions are true. In order to be optimal, a cluster set must satisfy each criterion for constructing cluster sets, and the clusters that comprise the cluster set must be both necessary and sufficient. By construction, the distances between data points from the same cluster are less than or equal to threshold distance d', the clusters are maximally complete, the clusters can overlap, and the data points can migrate between clusters. It remains to show that the constructed clusters are necessary and sufficient for constructing cluster sets comprised of the minimum number of complete linkage clusters that are exhaustive.*

Set threshold distance d' to any value that is greater than 0. The degree of every data point in data set $\mathcal{X}$ can be specified. First, construct clusters for all the singleton data points. Since none of these data points are linked to another data point, each must comprise a cluster, and each cluster is necessary and distinct.

Next, construct clusters for all the unmarked data points having the smallest degree whose degrees are equal to 1. At least one data point in each of these clusters is well-separated, and the degree of each such data point must be 1. Thus, the data points from these clusters can be linked only to data points whose degrees are equal to or greater than 1. Further, a data point whose degree is 1 will not be marked unless it belongs to one of these clusters, so every data point whose degree is 1 will be included in a cluster. Every data point from one of these clusters and whose degree is 1 is well-separated, because the clusters are maximally complete. Thus, data points whose degrees are 1 can belong to only one of these clusters, and each such cluster is necessary and distinct.

Next, construct clusters for all the unmarked data points having the smallest degree whose degrees are equal to 2. At least one data point in each of these clusters is well-separated, and the degree of each such data point must be 2. Thus, the data points from these clusters can be linked only to data points whose degrees are equal to or greater than 2. Further, a previously unmarked data point whose degree is 2 will not be marked unless it belongs to one of these clusters, so every data point whose degree is 2 will be

included in a cluster. Every data point from one of these clusters and whose degree is 2 is well-separated, because the clusters are maximally complete. Thus, previously unmarked data points whose degrees are 2 can belong to only one of these clusters, and each such cluster is necessary and distinct.

This construction continues only until all the data points are marked. Consequently, the cluster set is exhaustive and sufficient. Moreover, because each cluster is identifiable with a data point having the smallest degree, the clusters are constructed only once. Thus, the steps that INCLude hierarchical clustering takes to construct the cluster set also are efficient.
QED

NOTE–This proof covers only those cases where at least one data point in each cluster is well-separated (cluster patterns 1-3). It can be extended to cases where a data point having the smallest degree belongs to a subset of data points that comprise a subset of overlapping clusters (cluster pattern 4). Treating each of these clusters with "equal dignity", each cluster is constructed. Since only the data point having the smallest degree and all coextensive data points are marked, some of the overlapping clusters also may be included in other subsets of overlapping clusters. The problem becomes how to avoid constructing redundant clusters.

Chapter 5

Means for Finding Meaningful Levels of a Hierarchical Sequence Prior to Performing a Cluster Analysis

Because the computational power presently exists to cluster much larger data sets than before, the new clustering method unwinds the assumptions that underlie the standard complete linkage method. However, by unwinding these assumptions and letting the size of a hierarchical sequence revert back from n levels to $\frac{n\cdot(n-1)}{2}+1$ levels, the time complexity to construct cluster sets becomes $O(n^4)$. This is large even for small-n, large-m data sets. Moreover, the *post hoc* heuristics for cutting dendrograms are not suitable for finding meaningful levels or meaningful cluster sets of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence. Thus, with today's technology, the project went back more than 60 years to solve a problem that could not be solved then.

For the second part of the project, a means was developed for finding meaningful levels of an $\frac{n\cdot(n-1)}{2}+1$-level (complete linkage) hierarchical sequence *prior* to performing a cluster analysis. By finding meaningful levels of such a hierarchical sequence *prior* to performing a cluster analysis, it is possible to know which cluster sets to construct and construct only these cluster sets. This reduces the time complexity to construct

cluster sets from $O(n^4)$ to $O(ln^2)$, where l is the number of meaningful levels. *These are the cluster sets that can have real world meaning.* It is notable that the means does not depend on dendrograms or *post hoc* heuristics to find meaningful cluster sets. The second part also looked at how increasing the dimensionality of the data points helps reveal inherent structure in noisy data, which is necessary for finding meaningful levels.

5.1 Noise Attenuation

The means for finding meaningful levels is based on four assumptions. Let $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ be a data set that contains a finite number of data points n, where each data point has m dimensions. Further, suppose that each data point is a finite sequence of samples and that at any moment in time, with respect to each class or source, all the samples have the same true values and biases[1] . The means assumes that the 2-norms and the 1-norms of the data points are calculable. The means also assumes that noise (random error) is the only random component in a measured value, that noise can be modeled as Gaussian random variables, and that the noise that is embedded in each dimension (sample) of each data point is statistically independent. When a Gaussian distribution can over bound another distribution, the means should be applicable to the other distribution as well.

Within the context of the nearest neighbor problem, where high(er) dimensionality is considered to be a curse, Beyer et al. [35] show that, under broadly applicable conditions, if

$$\lim_{m\to\infty} Var[\frac{\|Y_m\|^p}{E[\|Y_m\|^p]}] = 0 \ , \tag{5.1}$$

then for every $\epsilon > 0$,

$$\lim_{m\to\infty} Prob[DMAX_m^p \leq (1+\epsilon)DMIN_m^p] = 1 \ . \tag{5.2}$$

Y_m is the difference between any independent data point $P_{i,m}, i = 1, 2, ..., n$, and Q_m, a query point that is chosen independently of all the data points; m is the dimensionality of $P_{i,m}$ and Q_m; $DMAX$ is the distance between Q_m and the farthest away data point; $DMIN$ is the distance between Q_m and the nearest data point; and p is the index of

[1] In real world terms, this is the same as calibrating the sensors.

the p-norm. Hinneburg et al. [36] extend this work by showing that

$$\lim_{m\to\infty} E[\frac{DMAX_m^p - DMIN_m^p}{m^{1/p-1/2}}] = C_p \ , \tag{5.3}$$

or

$$\lim_{m\to\infty} E[DMAX_m^p - DMIN_m^p] = C_p \cdot (m^{1/p-1/2}) \ . \tag{5.4}$$

C_p is a constant that depends on p. For the purposes of cluster analysis, these equations hint that classes of noisy data points may be spatially separable. However, they do not show how the distances between data points from different classes ("interclass" distances) relate to the distances between data points from the same class ("intraclass" distances). Also, C_p is unknown.

A set of theorems was proved to provide the missing pieces for the 2-norm and the 1-norm. Theorem 2, below, pertains to the 2-norm and includes Euclidean distance as a special case. Theorem 4 pertains to the 1-norm and includes city block distance as a special case. Since statistical independence is assumed only with respect to the Gaussian random variables (noise), the means (true values plus biases) of the dimensions (samples) may be highly correlated.

Theorem 1 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(0, \sigma_k^2)$, where σ_k is bounded from below by ϵ and above by a constant S. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = (\sum_{k=1}^m Y_k^2)^{\frac{1}{2}}$ is $\frac{\sum_{k=1}^m \sigma_k^4}{2\sum_{k=1}^m \sigma_k^2}$ plus an error term that converges to 0 as $m \to \infty$.*

Proof for Theorem 1 *The proof for Theorem 1 can be found at the end of this chapter.*

Theorem 2 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(\mu_k, \sigma_k^2)$, where σ_k is bounded from below by ϵ and above by a constant S, and $|\mu_k|$ is bounded from above by a constant M. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = (\sum_{k=1}^m Y_k^2)^{\frac{1}{2}}$ is $\frac{\sum_{k=1}^m \sigma_k^4}{2(\sum_{k=1}^m \sigma_k^2 + \sum_{k=1}^m \mu_k^2)} + \frac{\sum_{k=1}^m \sigma_k^2 \mu_k^2}{\sum_{k=1}^m \sigma_k^2 + \sum_{k=1}^m \mu_k^2}$ plus an error term that converges to 0 as $m \to \infty$.*

Proof for Theorem 2 *The proof for Theorem 2 also can be found at the end of this chapter.*

Let C_1 and C_2 be two classes (primitive clusters), each of which is comprised of a finite set of data points, i.e., $C_1 = \{x_{1,1}, x_{1,2}, ..., x_{1,n_1}\}$ and $C_2 = \{x_{2,1}, x_{2,2}, ..., x_{2,n_2}\}$. Also, let each data point have m dimensions, each of which is a statistically independent, Gaussian random variable, i.e., $X_{1,i,k} \sim N(\mu_{1,i,k}, \sigma^2_{1,i,k})$ and $X_{2,j,k} \sim N(\mu_{2,j,k}, \sigma^2_{2,j,k}), i = 1, 2, ..., n_1, j = 1, 2, ..., n_2$, and $k = 1, 2, ..., m$, where $\sigma_{1,i,k}$ or $\sigma_{2,j,k}$ (or both) are greater than $\epsilon' \geq \epsilon$. If $Y_{k,(i,j)} = X_{1,i,k} - X_{2,j,k}$, then $Y_{k,(i,j)}$ is statistically independent and $Y_{k,(i,j)} \sim N(\mu_{k,(i,j)}, \sigma^2_{k,(i,j)})$, where $\sigma_{k,(i,j)} \geq \epsilon$.

For $Y_{k,(i,j)} \sim N(0, 1)$, $k = 1, 2, ..., m$, the result in Theorem 2 is $\frac{1}{2}$. For $Y_{k,(i,j)} \sim N(0, \sigma^2_{k,(i,j)})$, where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$, the result in Theorem 2 is $\frac{1}{2}\sigma^2_{(i,j)}$. When $\sigma_{k,(i,j)}$ and $\mu_{k,(i,j)}$ are chosen from uniform distributions, using the Monte Carlo method shows that, as the bound M on $\mu_{k,(i,j)}$ increases, the sample mean for the result in Theorem 2 is $\frac{m}{3}S^2$ plus an error term that becomes relatively small, i.e., $\frac{error}{\frac{m}{3}S^2} \rightarrow 0$ as $m \rightarrow \infty$. More precisely, as m increases, the sample standard deviation becomes smaller relative to the magnitude of the sample mean. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$, using the Monte Carlo method shows that, as the bound M on $\mu_{k,(i,j)}$ increases, the sample mean for the result in Theorem 2 is mS^2 plus an error term that converges to 0 as $m \rightarrow \infty$. The sample standard deviation decreases to 0 absolutely. *See* Chapter 7.

Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$ and $\mu_{k,(i,j)} = \mu_{(i,j)}$, $k = 1, 2, ..., m$, the result in Theorem 2 becomes

$$\frac{\sigma^2_{(i,j)}}{2(1 + \frac{\mu^2_{(i,j)}}{\sigma^2_{(i,j)}})} + \frac{\mu^2_{(i,j)}}{(1 + \frac{\mu^2_{(i,j)}}{\sigma^2_{(i,j)}})} \, . \tag{5.5}$$

If $\sigma_{(i,j)}$ is held constant while $\mu_{(i,j)}$ is allowed to vary between 0 and $|\mu_{(i,j)}| \gg \sigma_{(i,j)}$, the result is a constant between $\frac{1}{2}\sigma^2_{(i,j)}$ and $\sigma^2_{(i,j)}$. As the graph in Fig. 5.1 shows, the first term in Equation 5.5 is monotonically decreasing while the second term is monotonically increasing. Moreover, the exemplary data in Fig. 5.2 show that limits calculated using Equation 5.5 are very consistent with empirical results from the sensitivity analysis described in Chapter 7.

Theorem 3 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(0, \sigma^2_k)$, where $\sigma_k > 0$. The variance $\sigma^2_{Z_m}$ of the random variable $Z_m = \sum_{k=1}^{m} |Y_k|$ is $\sum_{k=1}^{m} \sigma^2_k(1 - \frac{2}{\pi})$.*

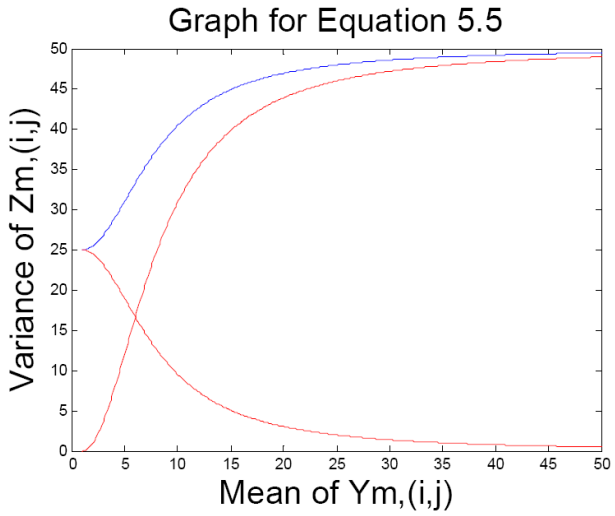


Figure 5.1: Graph for Equation 5.5. The blue curve (highest curve) describes Equation 5.5, the decreasing red curve describes the first term in Equation 5.5, and the increasing red curve describes the second term in Equation 5.5. The value 50 was used for all $\sigma^2_{k,(i,j)}$.

Proof for Theorem 3 *The proof for Theorem 3 can be found at the end of this chapter.*

Theorem 4 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(\mu_k, \sigma_k^2)$, where $\sigma_k > 0$. The variance $\sigma^2_{Z_m}$ of the random variable $Z_m = \sum_{k=1}^{m} |Y_k|$ is bounded by $\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|\mu_{k_1}||\mu_{k_2}| - \mu_{k_1}\mu_{k_2}(1 - 2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))(1 - 2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}})))$*

$$+ \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \frac{2\sigma_{k_2}}{\sqrt{2\pi}} (|\mu_{k_1}| - \mu_{k_1} \frac{(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{e^{\frac{\mu^2_{k_2}}{2\sigma^2_{k_2}}}})$$

$$+ \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \frac{2\sigma_{k_1}}{\sqrt{2\pi}} (|\mu_{k_2}| - \mu_{k_2} \frac{(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{e^{\frac{\mu^2_{k_1}}{2\sigma^2_{k_1}}}})$$

$$+ \sum_{k_1,k_2=1,k_1\neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} (1 - \frac{1}{e^{\frac{\mu^2_{k_1}}{2\sigma^2_{k_1}}} e^{\frac{\mu^2_{k_2}}{2\sigma^2_{k_2}}}})$$

$$+ \sum_{k=1}^{m} \sigma_k^2 (1 - \frac{2}{\pi e^{\frac{\mu^2_k}{\sigma^2_k}}}).$$

PAIRS	DIM	MEAN1	MEAN2	STD1	STD2	DMIN Normal	STDDIST Normal	LIMIT	DMIN Uniform	STDDIST Uniform
1000	10	2	2	2	200	200.6-256.1	131.7-143.5	141.4	490.0-553.1	158.5-163.8
1000	10	2	2	200	2	164.8-282.5	138.5-145.1	141.4	536.1-556.5	153.9-163.8
1000	10	2	2	20	20	20.9-36.4	19.2-20.2	20.0	47.5-65.2	28.8-30.0
1000	10	2	2B	20	20	6.324×10^{9}	27.9-29.0	28.2	6.324×10^{9}	47.7-49.2
1000	10	2	2B	200	200	6.324×10^{9}	278.1-291.8	282.8	6.324×10^{9}	464.3-497.0
1000	10	2	2B	2	200	6.324×10^{9}	195.8-207.0	200.0	6.324×10^{9}	337.6-352.3
1000	10	2	2B	200	2	6.324×10^{9}	198.2-204.9	200.0	6.324×10^{9}	335.4-351.1
1000	100	2	2	2	200	1507.8-1604.7	138.3-144.7	141.4	2891.3-3017.7	148.8-157.7
1000	100	2	2	200	2	1489.1-1603.1	137.5-144.8	141.4	2862.4-2919.1	153.4-155.7
1000	100	2	2	20	20	210.2-224.3	19.2-20.3	20.0	375.6-408.0	28.3-29.4
1000	100	2	2B	20	20	2.000×10^{10}	27.8-29.4	28.2	2.000×10^{10}	47.9-50.0
1000	100	2	2B	200	200	2.000×10^{10}	278.0-296.5	282.8	2.000×10^{10}	483.5-493.3
1000	100	2	2B	2	200	2.000×10^{10}	195.4-203.9	200.0	2.000×10^{10}	336.3-346.9
1000	100	2	2B	200	2	2.000×10^{10}	197.8-203.9	200.0	2.000×10^{10}	335.3-355.2
1000	1000	2	2	2	200	5756.0-5901.9	137.5-144.4	141.4	10,417.0-10,503.0	147.0-157.2
1000	1000	2	2	200	2	5772.8-5933.9	138.8-145.4	141.4	10,429.0-10,554.0	147.4-161.6
1000	1000	2	2	20	20	828.4-834.4	19.4-20.7	20.0	14,301.0-14,655.0	28.2-30.7
1000	1000	2	2B	20	20	6.324×10^{10}	27.4-28.7	28.2	6.324×10^{10}	47.5-50.8
1000	1000	2	2B	200	200	6.324×10^{10}	283.5-297.4	282.8	6.324×10^{10}	476.7-506.9
1000	1000	2	2B	2	200	6.324×10^{10}	194.8-202.7	200.0	6.324×10^{10}	335.3-353.2
1000	1000	2	2B	200	2	6.324×10^{10}	194.7-203.7	200.0	6.324×10^{10}	335.7-358.2

PAIRS = Number of data point pairs
DIM = Number of dimensions in each data point
MEAN1 = True value plus bias of first data point dimensions
MEAN2 = True value plus bias of second data point dimensions
STD1 = Std. dev. of noise embedded in first data point dimensions
STD2 = Std. dev. of noise embedded in second data point dimensions
DMIN Normal = DMIN when noise is normally distributed (5 trials)
STDDIST Normal = Std. devs. of the distances (5 trials) when noise is normally distributed
LIMIT = Limit calculations using Eq. 5.5
DMIN Uniform = DMIN when noise is uniformly distributed (5 trials)
STDDIST Uniform = Std. devs. of the distances (5 trials) when noise is uniformly distributed

Figure 5.2: Exemplary results from the sensitivity analysis described in Chapter 7. Using Euclidean distance, the minimum distances and the maximum distances (not shown) between data points from two different classes were calculated. Limits calculated using Equation 5.5 are very consistent with empirical results for STDDIST Normal. When noise is assumed to be uniformly distributed, the results are analogous to those when noise is assumed to be normally distributed, indicating that the Gaussian random variables assumption is reasonable.

Proof for Theorem 4 *The proof for Theorem 4 also can be found at the end of this chapter.*

For $Y_{k,(i,j)} \sim N(0,1)$, the bound on $\sigma^2_{Z_{m,(i,j)}}$ is $m(1-\frac{2}{\pi})$. For $Y_{k,(i,j)} \sim N(0,\sigma^2_{k,(i,j)})$, the bound on $\sigma^2_{Z_{m,(i,j)}}$ is $\sum_{k=1}^{m}\sigma^2_{k,(i,j)}(1-\frac{2}{\pi})$. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1,2,...,m$, the bound on $\sigma^2_{Z_{m,(i,j)}}$ is $m\sigma^2_{(i,j)}(1-\frac{2}{\pi})$. Where $|\mu_{k,(i,j)}| \gg \sigma_{k,(i,j)}$, the bound on $\sigma^2_{Z_{m,(i,j)}}$ reduces to $\sum_{k_1=1}^{m}\sum_{k_2=1}^{m}|\mu_{k_1,(i,j)}|\frac{2\sigma_{k_{2,(i,j)}}}{\sqrt{2\pi}} + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}|\mu_{k_2,(i,j)}|\frac{2\sigma_{k_{1,(i,j)}}}{\sqrt{2\pi}} + \sum_{k_1,k_2=1,k_1\neq k_2}^{m}\frac{2\sigma_{k_{1,(i,j)}}\sigma_{k_{2,(i,j)}}}{\pi} + \sum_{k=1}^{m}\sigma^2_{k,(i,j)}$. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$ and $\mu_{k,(i,j)} = \mu_{(i,j)}$, $k = 1,2,...,m$, the exemplary data in Fig. 5.3 show that, with respect to city block distance, the ratio DMIN (Normal)/STDDIST (Normal) converges to 0 as $m \to \infty$.

5.2 Finding Meaningful Levels and Cluster Sets

Often, as the dimensionality of the data points increases and the interclass distances become larger, the standard deviations of the interclass distances, i.e., $\sigma_{Z_{m,(i,j)}}$, become relatively small or are constant. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$ and $\mu_{k,(i,j)} = \mu_{(i,j)}$, $k = 1,2,...,m$, Equation 5.5 shows that this is certainly so. In particular, when the distributions of the noise that is embedded in each dimension of each data point are the same, $\sigma_{Z_{m,(i,j)}}$ is a constant between $\frac{1}{\sqrt{2}}\sigma_{(i,j)}$ and $\sigma_{(i,j)}$. As the Monte Carlo simulations show, this also is so for Euclidean distance when the interclass distances grow at an expected rate that is much faster than $\frac{d(\sqrt{mS^2})}{dm} = \frac{S}{2\sqrt{m}}$. The case for the 1-norm and city block distance is similar.

When this scenario holds, the standard deviations of the intraclass distances also become relatively small or are constant. So, *even at higher dimensionalities*, data points from the same class link at about the same time. As Fig. 5.4(a) depicts, classes of data points can be close together at lower dimensionalities. When they are, the magnitudes of many interclass distances and intraclass distances (which are caused by noise) are about the same, and the two kinds of distances commingle. However, as Fig. 5.4(b) depicts, the classes of data points grow farther apart at higher dimensionalities, so the interclass distances and the intraclass distances segregate into bands. Higher dimensionalities can attenuate the effects of noise[2] that preclude finding meaningful levels of a hierarchical

[2] Attenuating the effects of noise refers to reducing the effects of noise on cluster set construction.

PAIRS	DIM	MEAN1	MEAN2	STD1	STD2	DMIN Normal	STDDIST Normal	DMIN/ STDDIST	DMIN Uniform	STDDIST Uniform
1000	10	2	2	2	200	523.7-699.9	372.4-396.6	0.54-0.73	1327.9-1475.6	534.3-572.1
1000	10	2	2	200	2	516.2-662.8	361.3-378.3	0.56-0.73	1118.0-1469.4	537.7-554.1
1000	10	2	2	20	20	68.5-99.8	52.0-56.0	0.54-0.80	145.4-172.3	87.8-90.7
1000	10	2	2B	20	20	$2.000x10^{10}$	86.7-99.1	$4.3x10^{-9}$-$4.7x10^{-9}$	$2.000x10^{10}$	149.6-157.2
1000	10	2	2B	200	200	$2.000x10^{10}$	874.1-932.3	$4.4x10^{-8}$-$4.7x10^{-8}$	$2.000x10^{10}$	1496.2-1567.5
1000	10	2	2B	2	200	$2.000x10^{10}$	630.0-639.9	$3.1x10^{-8}$-$3.2x10^{-8}$	$2.000x10^{10}$	1090.9-1122.0
1000	10	2	2B	200	2	$2.000x10^{10}$	618.3-658.0	$3.1x10^{-8}$-$3.3x10^{-8}$	$2.000x10^{10}$	1042.3-1110.4
1000	100	2	2	2	200	11,906-12,745	1156.5-1239.1	0.09-0.10	23,167-24,777	1707.4-1807.5
1000	100	2	2	200	2	11,635-12,849	1185.6-1221.8	0.09-0.11	23,020-25,150	1681.4-1820.9
1000	100	2	2	20	20	1584.9-1805.0	171.7-177.6	0.10-0.11	3123.9-3174.3	273.6-289.6
1000	100	2	2B	20	20	$2.000x10^{11}$	272.4-291.2	$1.4x10^{-9}$-$1.5x10^{-9}$	$2.000x10^{11}$	469.3-496.4
1000	100	2	2B	200	200	$2.000x10^{11}$	2743.1-2980.3	$1.4x10^{-8}$-$1.5x10^{-8}$	$2.000x10^{11}$	4796.6-5037.2
1000	100	2	2B	2	200	$2.000x10^{11}$	1910.6-2062.4	$9.6x10^{-9}$-$1.0x10^{-8}$	$2.000x10^{11}$	3324.3-3555.6
1000	100	2	2B	200	2	$2.000x10^{11}$	1953.4-2077.7	$9.8x10^{-9}$-$1.0x10^{-8}$	$2.000x10^{11}$	3336.9-3505.6
1000	1000	2	2	2	200	146,467-149,249	3745.1-3875.0	.025-.027	281,220-284,594	5107.2-5536.5
1000	1000	2	2	200	2	145,310-148,294	3768.2-4040.3	.026-.028	282,313-284,342	5365.0-5679.0
1000	1000	2	2	20	20	20,593-21,080	524.8-553.2	.025-.027	36,947-37,543	880.6-901.0
1000	1000	2	2B	20	20	$2.000x10^{12}$	903.9-929.7	$4.5x10^{-10}$-$4.6x10^{-10}$	$2.000x10^{12}$	1529.9-1585.2
1000	1000	2	2B	200	200	$2.000x10^{12}$	8617.6-9234.8	$4.3x10^{-9}$-$4.6x10^{-9}$	$2.000x10^{12}$	14,771-16,075
1000	1000	2	2B	2	200	$2.000x10^{12}$	6236.6-6451.9	$3.1x10^{-9}$-$3.2x10^{-9}$	$2.000x10^{12}$	10,439-10,984
1000	1000	2	2B	200	2	$2.000x10^{12}$	6187.2-6535.2	$3.1x10^{-9}$-$3.3x10^{-9}$	$2.000x10^{12}$	10,549-11,156

PAIRS = Number of data point pairs
DIM = Number of dimensions in each data point
MEAN1 = True value plus bias of first data point dimensions
MEAN2 = True value plus bias of second data point dimensions
STD1 = Std. dev. of noise embedded in first data point dimensions
STD2 = Std. dev. of noise embedded in second data point dimensions

DMIN Normal = DMIN when noise is normally distributed (5 trials)
STDDIST Normal = Std. devs. of the distances (5 trials) when noise is normally distributed
DMIN/STDDIST = DMIN Normal / STDDIST Normal
DMIN Uniform = DMIN when noise is uniformly distributed (5 trials)
STDDIST Uniform = Std. devs. of the distances (5 trials) when noise is uniformly distributed

Figure 5.3: Exemplary results from the same sensitivity analysis described in Fig. 5.2. Using city block distance, the minimum distances and the maximum distances (not shown) between data points from two different classes were calculated. As the dimensionality of the data points increases, the ratio DMIN/STDDIST decreases. When noise is assumed to be uniformly distributed, the results are analogous to those when noise is assumed to be normally distributed, indicating that the Gaussian random variables assumption is reasonable.

sequence at lower dimensionalities, so that the classes can be distinguished. Moreover, as Figs. 5.4(b) and 5.5 show, this pattern repeats itself as clusters become larger from including more data points.

Consequently, as the dimensionality of the data points increases, the distance graphs for a data set can exhibit features that correlate with meaningful levels of the corresponding hierarchical sequences. *These levels are the levels at which multiple classes of data points have finished linking to form new configurations of clusters.* In particular, assuming that the data set has good inherent structure, the curve of a distance graph takes on a shape whereby sections of the curve run nearly parallel to one of the graph axes. Where there is very little or no linking activity, the sections run nearly vertically. Where there is significant activity, i.e., where new configurations of clusters are forming, the sections run nearly horizontally. Thus, sections of the curve that come after the lower-right corners and before the upper-left corners indicate where new configurations of clusters have finished forming. As the schematic in Fig. 5.5 shows, a distance graph can be visually examined *prior* to performing a cluster analysis. It is not a summary of results obtained from the analysis. Since a distance graph can be used to find meaningful levels of a hierarchical sequence *prior* to performing a cluster analysis, it enables a user to construct only the cluster sets for these levels, i.e., cluster sets where new configurations of clusters have finished forming.

Finding meaningful levels is remarkably easy:

Step 1 Calculate the dissimilarities between data points x_i and x_j in data set $\mathcal{X}$, $i, j = 1, 2, ..., n, x_i \neq x_j$. Then, calculate the lengths or magnitudes of the vectors that contain the dissimilarities between the data points. Here, the dissimilarity measures are simple value differences, the 2-norm is used to obtain Euclidean distance, and the 1-norm is used to obtain city block distance.

Step 2 From these distances $d_{i,j}$ and the indices of the respective data points, construct ordered triples $(d_{i,j}, i, j)$. Sort the ordered triples into rank order according to their distance elements and assign indices to the sorted ordered triples (the "rank order indices"). The time complexity to calculate the distances is $O(\frac{n\cdot(n-1)\cdot m}{2})$. If ordinary merge sort [34] is used, the time complexity to sort the ordered triples is $O(\frac{n\cdot(n-1)}{2} \cdot log(\frac{n\cdot(n-1)}{2})) = O(n \cdot (n-1) \cdot log\sqrt{\frac{n\cdot(n-1)}{2}})$.

Step 3 Use the rank order indices and the ordered triples to construct a distance

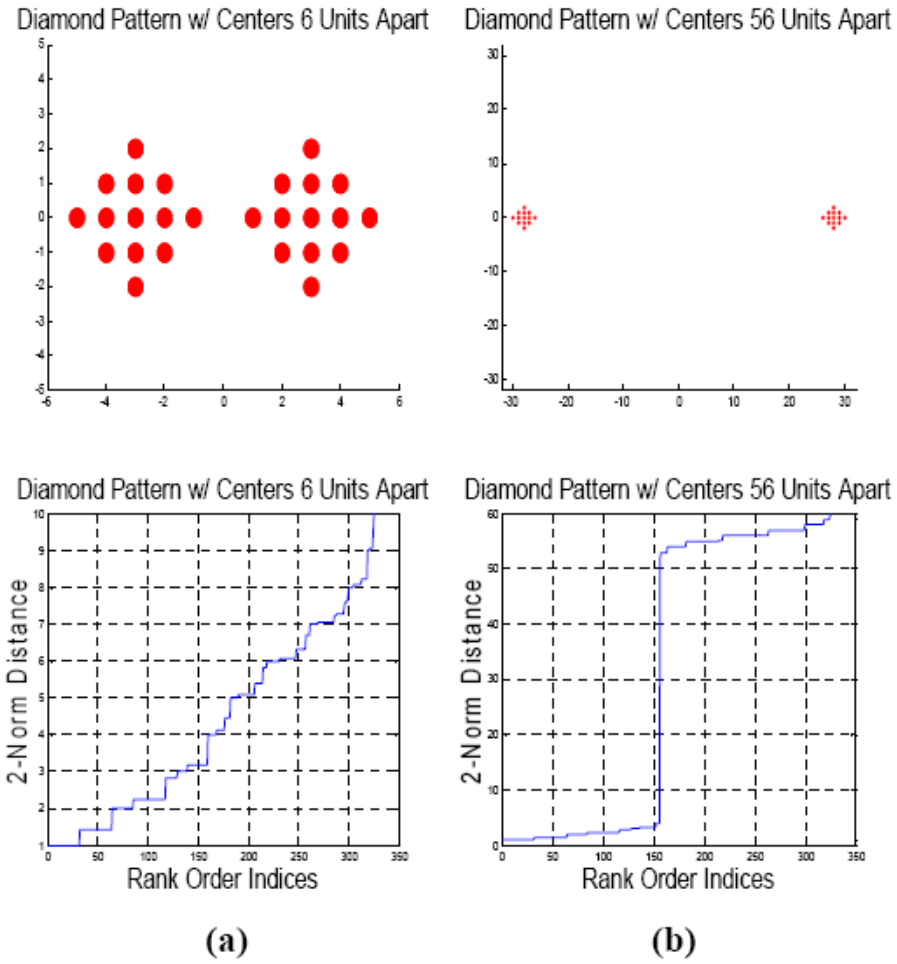


Figure 5.4: Illustrations that show how two classes of data points link as the classes grow farther apart.

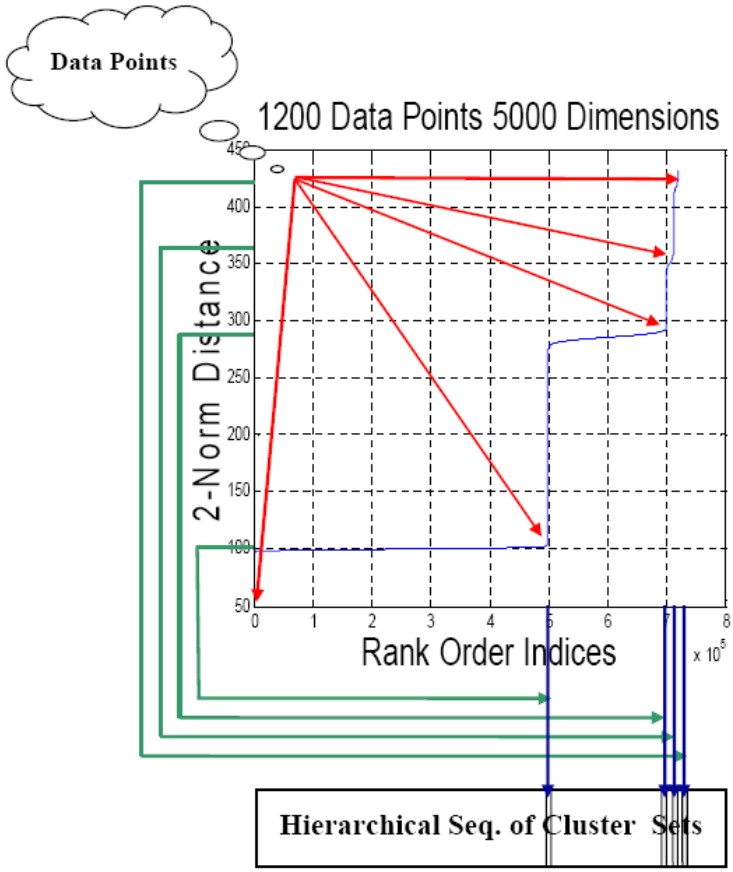


Figure 5.5: Schematic for finding meaningful levels of a hierarchical sequence. Inherent structure is revealed through features along the curve of the distance graph. These features correlate with those levels of the corresponding hierarchical sequence at which multiple classes of data points have finished linking to form new configurations of clusters.

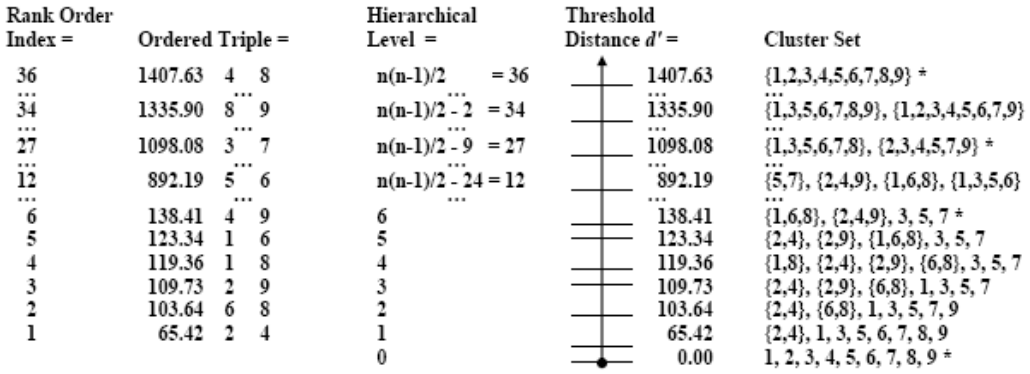


Figure 5.6: Illustration that shows how rank order indices align with levels of the corresponding hierarchical sequence and distance elements align with the respective threshold distances d'. The data come from the sensor system experiment described in Chapter 7. Euclidean distance was used to calculate the distances. The arrow in the column for the threshold distances signifies that threshold distance d' is a continuous variable. In the last column, the meaningful cluster sets are indicated by asterisks.

graph. As the structureless data sets experiment described in Chapter 7 shows, the curve of the distance graph will remain smooth, regardless of the dimensionality of the data points, where inherent structure is absent. Assuming that data set $\mathcal{X}$ has inherent structure, increase the dimensionality of the data points and repeat Steps 1 to 3 until the lower-right corners have good definition (or as good as is practically possible).

Step 4 Along the axes of the distance graph, locate the rank order indices and/or the distance elements that correspond to where the lower-right corners appear along the curve. When a data set has good inherent structure, the corners are nearly orthogonal. These rank order indices coincide with meaningful levels of the corresponding hierarchical sequence, and the distance elements coincide with the respective threshold distances d'.

For an example that shows how these four variables align, see Fig. 5.6. As part of the new clustering method described in Chapter 4, ordered triples are evaluated for linkage in ascending order. As the distance elements become larger, threshold distance d' implicitly increases from 0 to the maximum of all the distance elements. Although threshold distance d' can vary continuously from 0 (where each data point is a singleton)

to at least this maximum distance (where all the data points belong to the same cluster), the only values that matter are those $\frac{n \cdot (n-1)}{2}$ values that are equal to the distance elements $d_{i,j}$. Since the number of data points in a data set is finite, the number of ordered triples (distance elements) is finite. Thus, the number of threshold distances d' that matter and the number of levels of a hierarchical sequence are finite and equal the number of ordered triples (distance elements) plus one. The rank order indices, by virtue of the distance elements $d_{i,j}$, coincide with the last $\frac{n \cdot (n-1)}{2}$ levels of the hierarchical sequence.

Step 5 Use a complete linkage hierarchical clustering method, such as that described in Chapter 4, to construct only the cluster sets for meaningful levels. The number of clusters in a meaningful cluster set is ascertainable as an artifact of cluster set construction. Constructing only the cluster sets for meaningful levels of a hierarchical sequence reduces the time complexity to construct cluster sets from $O(n^4)$ to $O(ln^2)$.

5.3 Theorems and Proofs for Finding Meaningful Levels of a Hierarchical Sequence

5.3.1 Calculating the Variance of $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ When $Y_k \sim N(0, \sigma_k^2)$

Theorem 1 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(0, \sigma_k^2)$, where σ_k is bounded from below by ϵ and above by a constant S. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ is $\frac{\sum_{k=1}^{m} \sigma_k^4}{2\sum_{k=1}^{m} \sigma_k^2}$ plus an error term that converges to 0 as $m \to \infty$.*

Proof for Theorem 1 [3] *Assume that the stated conditions are true. The theorem will be proved using Taylor's series to find the variance of Z_m. Variance $\sigma_{Z_m}^2 = E[Z_m^2] - (E[Z_m])^2 = E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}\cdot 2}] - (E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}])^2$.*

The second moment of Z_m is $E[Z_m^2] = E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}\cdot 2}] = E[\sum_{k=1}^{m} Y_k^2] = \sum_{k=1}^{m} E[Y_k^2] = \sum_{k=1}^{m} \sigma_k^2$, where σ_k^2 is the central moment of $E[Y_k^2]$.

Taylor's series is used to find the expected value of Z_m. Let $x_0 = \sum_{k=1}^{m} (\sigma_k W_k)^2$, where W is a standard Gaussian random variable. Let $h = \sum_{k=1}^{m} ((\sigma_k W_k)^2 - E[(\sigma_k W_k)^2])$. Then,

$$
\begin{aligned}
E[Z_m] &= E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} (\sigma_k W_k)^2)^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} (E[(\sigma_k W_k)^2] + (\sigma_k W_k)^2 - E[(\sigma_k W_k)^2]))^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} (E[(\sigma_k W_k)^2] + h))^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} E[(\sigma_k W_k)^2])^{\frac{1}{2}} + \frac{h}{2(\sum_{k=1}^{m} E[(\sigma_k W_k)^2])^{\frac{1}{2}}} - \frac{h^2}{8(\sum_{k=1}^{m} E[(\sigma_k W_k)^2])^{\frac{3}{2}}} \\
&\quad + \frac{3h^3}{48(\sum_{k=1}^{m} E[(\sigma_k W_k)^2])^{\frac{5}{2}}} + ...]
\end{aligned}
$$

[3] *The author thanks Dr. Larry Gray, Department of Mathematics, University of Minnesota, for showing him how to prove Theorems 1 and 2.*

$$= (\sum_{k=1}^{m} \sigma_k^2)^{\frac{1}{2}} + \frac{E[h]}{2(\sum_{k=1}^{m} \sigma_k^2)^{\frac{1}{2}}} - \frac{E[h^2]}{8(\sum_{k=1}^{m} \sigma_k^2)^{\frac{3}{2}}} + \frac{3E[h^3]}{48(\sum_{k=1}^{m} \sigma_k^2)^{\frac{5}{2}}} + ... \quad (5.6)$$

$$\approx (\sum_{k=1}^{m} \sigma_k^2)^{\frac{1}{2}} - \frac{E[h^2]}{8(\sum_{k=1}^{m} \sigma_k^2)^{\frac{3}{2}}}. \quad (5.7)$$

Since $E[h] = E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - E[(\sigma_k W_k)^2])] = E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2)] = 0$, *the first-order term in Equation 5.6 drops out. The third-order term and all higher order terms converge to 0 as* $m \to \infty$ *(NOTE 1).*

When h^2 *is expanded and evaluated,*

$$\begin{aligned}
E[h^2] &= E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 - E[(\sigma_k W_k)^2]))^2] \\
&= E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2))^2] \\
&= E[\sum_{k_1=1}^{m}((\sigma_{k_1} W_{k_1})^2 - \sigma_{k_1}^2) \sum_{k_2=1}^{m}((\sigma_{k_2} W_{k_2})^2 - \sigma_{k_2}^2)] \\
&= E[\sum_{k_1=1}^{m}\sum_{k_2=1}^{m}((\sigma_{k_1} W_{k_1})^2 - \sigma_{k_1}^2)((\sigma_{k_2} W_{k_2})^2 - \sigma_{k_2}^2)] \quad (5.8) \\
&= \sum_{k_1,k_2=1,k_1\neq k_2}^{m} E[(\sigma_{k_1} W_{k_1})^2 - \sigma_{k_1}^2]E[(\sigma_{k_2} W_{k_2})^2 - \sigma_{k_2}^2] \\
&\quad + E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2)^2] \\
&= E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2)^2]. \quad (5.9)
\end{aligned}$$

Since Y_{k_1} *and* Y_{k_2} *are statistically independent, the cross-terms in Equation 5.8 can be evaluated separately. Both* $E[(\sigma_{k_1} W_{k_1})^2 - \sigma_{k_1}^2]$ *and* $E[(\sigma_{k_2} W_{k_2})^2 - \sigma_{k_2}^2]$ *evaluate to 0, and the cross-terms drop out. So,*

$$E[Z_m] \approx (\sum_{k=1}^{m} \sigma_k^2)^{\frac{1}{2}} - \frac{E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2)^2]}{8(\sum_{k=1}^{m} \sigma_k^2)^{\frac{3}{2}}}, \quad (5.10)$$

and

$$
\begin{aligned}
(E[Z_m])^2 &\approx ((\sum_{k=1}^{m}\sigma_k^2)^{\frac{1}{2}} - \frac{E[\sum_{k=1}^{m}((\sigma_k W_k)^2-\sigma_k^2)^2]}{8(\sum_{k=1}^{m}\sigma_k^2)^{\frac{3}{2}}})^2 \\
&= ((\sum_{k=1}^{m}\sigma_k^2)^{\frac{1}{2}})^2 \\
&\quad - 2(\sum_{k=1}^{m}\sigma_k^2)^{\frac{1}{2}}\frac{E[\sum_{k=1}^{m}((\sigma_k W_k)^2-\sigma_k^2)^2]}{8(\sum_{k=1}^{m}\sigma_k^2)^{\frac{3}{2}}} \\
&\quad + (\frac{E[\sum_{k=1}^{m}((\sigma_k W_k)^2-\sigma_k^2)^2]}{8(\sum_{k=1}^{m}\sigma_k^2)^{\frac{3}{2}}})^2 \\
&= \sum_{k=1}^{m}\sigma_k^2 - \frac{2(\sum_{k=1}^{m}\sigma_k^2)^{\frac{1}{2}}2\sum_{k=1}^{m}\sigma_k^4}{8(\sum_{k=1}^{m}\sigma_k^2)^{\frac{3}{2}}} + \frac{4(\sum_{k=1}^{m}\sigma_k^4)^2}{64(\sum_{k=1}^{m}\sigma_k^2)^3} \\
&= \sum_{k=1}^{m}\sigma_k^2 - \frac{\sum_{k=1}^{m}\sigma_k^4}{2(\sum_{k=1}^{m}\sigma_k^2)} + \frac{(\sum_{k=1}^{m}\sigma_k^4)^2}{16(\sum_{k=1}^{m}\sigma_k^2)^3} \qquad (5.11) \\
&= \sum_{k=1}^{m}\sigma_k^2 - \frac{\sum_{k=1}^{m}\sigma_k^4}{2(\sum_{k=1}^{m}\sigma_k^2)}. \qquad (5.12)
\end{aligned}
$$

The third term in Equation 5.11 converges to 0 as $m \to \infty$ (NOTE 2).

Thus, the variance $\sigma^2_{Z_m}$ of Z_m is

$$
\begin{aligned}
\sigma^2_{Z_m} &= E[Z_m^2] - (E[Z_m])^2 \\
&\approx \sum_{k=1}^{m}\sigma_k^2 - (\sum_{k=1}^{m}\sigma_k^2 - \frac{\sum_{k=1}^{m}\sigma_k^4}{2(\sum_{k=1}^{m}\sigma_k^2)}) \\
&= \frac{\sum_{k=1}^{m}\sigma_k^4}{2(\sum_{k=1}^{m}\sigma_k^2)}. \qquad (5.13)
\end{aligned}
$$

QED

NOTE 1–To show that the third-order term and all higher order terms in the Taylor series converge to 0, the terms are characterized by the general formula $\frac{c_1 \cdot E[h^{order}]}{(\sum_{k=1}^{m}\sigma_k^2)^{\frac{2\cdot order-1}{2}}}$, where order is the order of the term and c_1 is a constant. All nonnegative moments of a Gaussian random variable are finite, the central moments of $W^{2\cdot order}$ are $(2\cdot order-1)!!\cdot\sigma_k^{2\cdot order}$, and the central moments of $W^{2\cdot order+1}$ are 0. Thus, when $E[h^{order}]$ is

expanded and evaluated,

$$\frac{c_1 \cdot E[h^{order}]}{(\sum_{k=1}^{m} \sigma_k^2)^{\frac{2 \cdot order-1}{2}}} = \frac{c_1 \cdot c_2 \cdot \sum_{k=1}^{m} \sigma_k^{2 \cdot order}}{(\sum_{k=1}^{m} \sigma_k^2)^{\frac{2 \cdot order-1}{2}}}, \tag{5.14}$$

where c_2 is a constant. Let $\epsilon \leq c_3 \cdot \sigma_{min} = c_3 \cdot \min_k(\sigma_k) = \max_k(\sigma_k) = \sigma_{max} \leq S$, where c_3 is a constant. Then,

$$\begin{aligned} \frac{c_1 \cdot c_2 \cdot \sum_{k=1}^{m} \sigma_k^{2 \cdot order}}{(\sum_{k=1}^{m} \sigma_k^2)^{\frac{2 \cdot order-1}{2}}} &\leq \frac{c_1 \cdot c_2 \cdot m \cdot \sigma_{max}^{2 \cdot order}}{m^{\frac{2 \cdot order-1}{2}} \sigma_{min}^{2 \cdot order-1}} \\ &= \frac{c_1 \cdot c_2 \cdot c_3^{2 \cdot order-1} \cdot m \cdot \sigma_{max}^{2 \cdot order}}{m^{\frac{2 \cdot order-1}{2}} \sigma_{max}^{2 \cdot order-1}} \\ &= \frac{c_1 \cdot c_2 \cdot c_3^{2 \cdot order-1} \cdot m \cdot \sigma_{max}}{m^{\frac{2 \cdot order-1}{2}}}. \end{aligned} \tag{5.15}$$

$\frac{c_1 \cdot c_2 \cdot c_3^{2 \cdot order-1} \cdot m \cdot \sigma_{max}}{m^{\frac{2 \cdot order-1}{2}}} \rightarrow 0$ *as* $m \rightarrow \infty$.

NOTE 2–To show that $\frac{(\sum_{k=1}^{m} \sigma_k^4)^2}{16(\sum_{k=1}^{m} \sigma_k^2)^3}$ converges to 0, let $\epsilon \leq c \cdot \sigma_{min} = c \cdot \min_k(\sigma_k) = \max_k(\sigma_k) = \sigma_{max} \leq S$, where c is a constant. Then,

$$\begin{aligned} \frac{(\sum_{k=1}^{m} \sigma_k^4)^2}{16(\sum_{k=1}^{m} \sigma_k^2)^3} &\leq \frac{(m \cdot \sigma_{max}^4)^2}{16(m \cdot \sigma_{min}^2)^3} \\ &= \frac{m^2 c^6 \sigma_{max}^8}{16 m^3 \sigma_{max}^6} \\ &= \frac{m^2 c^6 \sigma_{max}^2}{16 m^3}. \end{aligned} \tag{5.16}$$

$\frac{m^2 c^6 \sigma_{max}^2}{16 m^3} \rightarrow 0$ *as* $m \rightarrow \infty$.

5.3.2 Calculating the Variance of $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ When $Y_k \sim N(\mu_k, \sigma_k^2)$

Theorem 2 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(\mu_k, \sigma_k^2)$, where σ_k is bounded from below by ϵ and above by a constant S, and $|\mu_k|$ is bounded from above by a constant M. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = (\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}$ is $\frac{\sum_{k=1}^{m} \sigma_k^4}{2(\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2)} + \frac{\sum_{k=1}^{m} \sigma_k^2 \mu_k^2}{\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2}$ plus an error term that converges to 0 as $m \to \infty$.*

Proof for Theorem 2 *Assume that the stated conditions are true. The theorem will be proved using Taylor's series to find the variance of Z_m. Variance $\sigma_{Z_m}^2 = E[Z_m^2] - (E[Z_m])^2 = E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2} \cdot 2}] - (E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}])^2$.*

The second moment of Z_m is $E[Z_m^2] = E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2} \cdot 2}] = E[\sum_{k=1}^{m} Y_k^2] = \sum_{k=1}^{m} E[Y_k^2] = \sum_{k=1}^{m} E[(\sigma_k W_k + \mu_k)^2] = \sum_{k=1}^{m} E[(\sigma_k W_k)^2 + 2\sigma_k W_k \mu_k + \mu_k^2] = \sum_{k=1}^{m} E[(\sigma_k W_k)^2] + \sum_{k=1}^{m} E[2\sigma_k W_k \mu_k] + \sum_{k=1}^{m} E[\mu_k^2]$, where W is a standard Gaussian random variable. The expected value in the middle term of the last expression equals 0 and drops out, so $E[Z_m^2] = \sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2$, where σ_k^2 is the central moment of $E[(\sigma_k W_k)^2]$.

Taylor's series is used to find the expected value of Z_m. Let $x_0 = \sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2)$

and $h_1 = \sum_{k=1}^{m} 2\sigma_k W_k \mu_k$. *Then,*

$$
\begin{aligned}
E[Z_m] &= E[(\sum_{k=1}^{m} Y_k^2)^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} (\sigma_k W_k + \mu_k)^2)^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + 2\sigma_k W_k \mu_k + \mu_k^2))^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2 + 2\sigma_k W_k \mu_k))^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2) + h_1)^{\frac{1}{2}}] \\
&= E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}} + \frac{h_1}{2(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}} \\
&\quad - \frac{h_1^2}{8(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}} + \frac{3h_1^3}{48(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{5}{2}}} + ...] \\
&= E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}] + E[\frac{h_1}{2(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}}] \\
&\quad - E[\frac{h_1^2}{8(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}] + E[\frac{3h_1^3}{48(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{5}{2}}}] + ... \qquad (5.17) \\
&\approx E[(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}] - E[\frac{h_1^2}{8(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}]. \qquad (5.18)
\end{aligned}
$$

Since $2\sigma_k W_k \mu_k$ *is symmetric,* $E[\frac{h_1}{2(\sum_{k=1}^{m} ((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}}] = 0$ *in Equation 5.17. As in Theorem 1, the third-order term and all higher order terms converge to 0 as* $m \to \infty$.

Thus, the variance $\sigma^2_{Z_m}$ *of* Z_m *is*

$$\begin{aligned}
\sigma^2_{Z_m} &= E[Z_m^2] - (E[Z_m])^2 \\
&\approx \sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2 \\
&\quad - (E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}] - E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}])^2 \\
&= \sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2 - (E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}])^2 \\
&\quad + 2E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}]E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}] \\
&\quad - (E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}])^2 \\
&= \sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2 - (\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2 \\
&\quad - \frac{\sum_{k=1}^{m} \sigma_k^4}{2(\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2)} + \frac{(\sum_{k=1}^{m} \sigma_k^4)^2}{16(\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2)^3}) \\
&\quad + 2E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}]E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}] \\
&\quad - (E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}])^2 \qquad (5.19) \\
&= \frac{\sum_{k=1}^{m} \sigma_k^4}{2(\sum_{k=1}^{m} \sigma_k^2 + \sum_{k=1}^{m} \mu_k^2)} \\
&\quad + E[(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}]E[\frac{\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}]. \qquad (5.20)
\end{aligned}$$

The third, fourth, fifth, and sixth terms in Equation 5.19 are derived using that part of Theorem 1 for finding the square of the expectation, except that $(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{1}{2}}$ *is substituted for* $(\sum_{k=1}^{m}(\sigma_k W_k)^2)^{\frac{1}{2}}$. *The first four terms in Equation 5.19 cancel. As in Theorem 1, the sixth and the eighth terms converge to 0 as* $m \to \infty$. *When* $E[\frac{h_1^2}{8(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}]$ *is expanded and evaluated, by symmetry, the cross-terms drop out, leaving* $E[\frac{\sum_{k=1}^{m}(2\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}((\sigma_k W_k)^2 + \mu_k^2))^{\frac{3}{2}}}]$.

Both σ_k and $|\mu_k|$ are bounded from above, so $E[\frac{\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2))^{\frac{3}{2}}}]$ is bounded from above by $\frac{M^2}{(\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2))^{\frac{1}{2}}}$, and $\frac{E[(\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2))^{\frac{1}{2}}]}{(\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2))^{\frac{1}{2}}}$ converges to 1 almost surely as $m \to \infty$ and also in the L_1 norm. Thus, by the dominated convergence theorem, it suffices to calculate

$$E[\frac{\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2)}] \tag{5.21}$$

in place of the second term in Equation 5.20.

Using Taylor's series once more, let $x_0 = \sum_{k=1}^{m}\sigma_k^2 + \sum_{k=1}^{m}\mu_k^2$, $f(x_0) = \frac{1}{x_0} = (\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^{-1}$ and $h_2 = \sum_{k=1}^{m}((\sigma_k W_k)^2 - E[(\sigma_k W_2)^2]) = \sum_{k=1}^{m}((\sigma_k W_k)^2-\sigma_k^2)$, as in Theorem 1. Then, $\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2) = \sum_{k=1}^{m}(\sigma_k^2+\mu_k^2+(\sigma_k W_k)^2-\sigma_k^2) = \sum_{k=1}^{m}\sigma_k^2 + \sum_{k=1}^{m}\mu_k^2 + \sum_{k=1}^{m}((\sigma_k W_k)^2-\sigma_k^2) = \sum_{k=1}^{m}\sigma_k^2 + \sum_{k=1}^{m}\mu_k^2 + h_2$, and

$$\begin{aligned}
&E[\frac{\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{\sum_{k=1}^{m}((\sigma_k W_k)^2+\mu_k^2)}] \\
&= E[\frac{\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2} - \frac{h_2\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^2} \\
&\quad + \frac{h_2^2\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^3} - \frac{h_2^3\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^4} + ...] \\
&= \frac{E[\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]}{\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2} - \frac{E[h_2\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^2} \\
&\quad + \frac{E[h_2^2\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^3} - \frac{E[h_2^3\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]}{(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)^4} + ... \qquad (5.22) \\
&\approx \frac{\sum_{k=1}^{m}\sigma_k^2\mu_k^2}{\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2}. \qquad (5.23)
\end{aligned}$$

When $E[h_2\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]$ in Equation 5.22 is expanded and evaluated, the cross-terms drop out, leaving $E[\sum_{k=1}^{m}((\sigma_k W_k)^2 - \sigma_k^2)(\sigma_k W_k \mu_k)^2]$. Likewise, when $E[h_2^{order}\sum_{k=1}^{m}(\sigma_k W_k \mu_k)^2]$ is expanded and evaluated, where order ≥ 2, the number of summations decreases by at least one. As in Theorem 1, the first-order term and all higher order terms converge to 0 as $m \to \infty$. Thus,

$$\sigma_{Z_k}^2 \approx \frac{\sum_{k=1}^{m}\sigma_k^4}{2(\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2)} + \frac{\sum_{k=1}^{m}\sigma_k^2\mu_k^2}{\sum_{k=1}^{m}\sigma_k^2+\sum_{k=1}^{m}\mu_k^2}. \tag{5.24}$$

QED

5.3.3 Calculating the Variance of $Z_m = \sum_{k=1}^{m} |Y_k|$ When $Y_k \sim N(0, \sigma_k^2)$

Theorem 3 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(0, \sigma_k^2)$, where $\sigma_k > 0$. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = \sum_{k=1}^{m} |Y_k|$ is $\sum_{k=1}^{m} \sigma_k^2 (1 - \frac{2}{\pi})$.*

Proof for Theorem 3 *Assume that the stated conditions are true. The theorem will be proved by calculating the variance of Z_m algebraically. Variance $\sigma_{Z_m}^2 = E[Z_m^2] - (E[Z_m])^2 = E[(\sum_{k=1}^{m} |Y_k|)^2] - (E[\sum_{k=1}^{m} |Y_k|])^2$.*

The second moment of Z_m is

$$\begin{aligned}
E[Z_m^2] &= E[(\sum_{k=1}^{m} |Y_k|)^2] \\
&= E[(\sum_{k_1=1}^{m} |Y_{k_1}|)(\sum_{k_2=1}^{m} |Y_{k_2}|)] \\
&= E[\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|Y_{k_1}||Y_{k_2}|)] \qquad (5.25) \\
&= \sum_{k_1,k_2=1,k_1 \neq k_2}^{m} (E[|Y_{k_1}|]E[|Y_{k_2}|]) + \sum_{k=1}^{m} E[|Y_k|^2] \\
&= \sum_{k_1,k_2=1,k_1 \neq k_2}^{m} (\frac{2\sigma_{k_1}}{\sqrt{2\pi}})(\frac{2\sigma_{k_2}}{\sqrt{2\pi}}) + \sum_{k=1}^{m} \sigma_k^2 \qquad (5.26) \\
&= \sum_{k_1,k_2=1,k_1 \neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m} \sigma_k^2. \qquad (5.27)
\end{aligned}$$

Since Y_{k_1} and Y_{k_2} are statistically independent, the cross-terms in Equation 5.25 can be evaluated separately. Equation 5.26 uses the first and second central absolute moments of Y_k.

The expected value of Z_m is

$$\begin{aligned}
E[Z_m] &= E[\sum_{k=1}^{m} |Y_k|] \\
&= \sum_{k=1}^{m} E[|Y_k|] \\
&= \sum_{k=1}^{m} \frac{2\sigma_k}{\sqrt{2\pi}}, \qquad (5.28)
\end{aligned}$$

so

$$
\begin{aligned}
(E[Z_m])^2 &= (\sum_{k=1}^{m} \frac{2\sigma_k}{\sqrt{2\pi}})^2 \\
&= (\sum_{k_1=1}^{m} \frac{2\sigma_{k_1}}{\sqrt{2\pi}})(\sum_{k_2=1}^{m} \frac{2\sigma_{k_2}}{\sqrt{2\pi}}) \\
&= \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} \\
&= \sum_{k_1,k_2=1,k_1\neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m} \frac{2\sigma_k^2}{\pi}. \qquad (5.29)
\end{aligned}
$$

Thus, the variance $\sigma_{Z_m}^2$ *of* Z_m *is*

$$
\begin{aligned}
\sigma_{Z_m}^2 &= E[Z_m^2] - (E[Z_m])^2 \\
&= \sum_{k_1,k_2=1,k_1\neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m} \sigma_k^2 \\
&\quad - (\sum_{k_1,k_2=1,k_1\neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m} \frac{2\sigma_k^2}{\pi}) \\
&= \sum_{k=1}^{m} (\sigma_k^2(1 - \frac{2}{\pi})). \qquad (5.30)
\end{aligned}
$$

QED

5.3.4 Calculating the Variance of $Z_m = \sum_{k=1}^{m} |Y_k|$ When $Y_k \sim N(\mu_k, \sigma_k^2)$

Theorem 4 *Let $Y_k, k = 1, 2, ..., m$, be statistically independent, Gaussian random variables such that $Y_k \sim N(\mu_k, \sigma_k^2)$, where $\sigma_k > 0$. The variance $\sigma_{Z_m}^2$ of the random variable $Z_m = \sum_{k=1}^{m} |Y_k|$ is bounded by $\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|\mu_{k_1}||\mu_{k_2}| - \mu_{k_1}\mu_{k_2}(1 - 2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))(1 - 2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))) + \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \frac{2\sigma_{k_2}}{\sqrt{2\pi}}(|\mu_{k_1}| - \mu_{k_1} \frac{(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}}) + \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \frac{2\sigma_{k_1}}{\sqrt{2\pi}}(|\mu_{k_2}| - \mu_{k_2} \frac{(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}}) + \sum_{k_1,k_2=1,k_1 \neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi}(1 - \frac{1}{e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}} e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}}) + \sum_{k=1}^{m} \sigma_k^2(1 - \frac{2}{\pi e^{\frac{\mu_k^2}{\sigma_k^2}}})$.*

Proof for Theorem 4 *Assume that the stated conditions are true. The theorem will be proved by calculating the variance of Z_m algebraically. Variance $\sigma_{Z_m}^2 = E[Z_m^2] - (E[Z_m])^2 = E[(\sum_{k=1}^{m} |Y_k|)^2] - (E[\sum_{k=1}^{m} |Y_k|])^2$.*

The second moment of Z_m is

$$
\begin{aligned}
E[Z_m^2] &= E[(\sum_{k=1}^{m} |Y_k|)^2] \\
&= E[(\sum_{k_1=1}^{m} |Y_{k_1}|)(\sum_{k_2=1}^{m} |Y_{k_2}|)] \\
&= E[\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|Y_{k_1}||Y_{k_2}|)] \\
&= E[\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|\mu_{k_1} + Y_{k_1} - \mu_{k_1}||\mu_{k_2} + Y_{k_2} - \mu_{k_2}|)] \\
&\leq E[\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} ((|\mu_{k_1}| + |Y_{k_1} - \mu_{k_1}|)(|\mu_{k_2}| + |Y_{k_2} - \mu_{k_2}|))] \qquad (5.31) \\
&= E[\sum_{k_1=1}^{m} \sum_{k_2=1}^{m} (|\mu_{k_1}||\mu_{k_2}| + |\mu_{k_1}||Y_{k_2} - \mu_{k_2}| \\
&\quad + |\mu_{k_2}||Y_{k_1} - \mu_{k_1}| + |Y_{k_1} - \mu_{k_1}||Y_{k_2} - \mu_{k_2}|)]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}||\mu_{k_2}| + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} E[|\mu_{k_1}||Y_{k_2}-\mu_{k_2}|] \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} E[|\mu_{k_2}||Y_{k_1}-\mu_{k_1}|] \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} E[|Y_{k_1}-\mu_{k_1}||Y_{k_2}-\mu_{k_2}| \qquad (5.32) \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}||\mu_{k_2}| + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} E[|\mu_{k_1}||Y_{k_2}-\mu_{k_2}|] \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} E[|\mu_{k_2}||Y_{k_1}-\mu_{k_1}|] \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m} E[|Y_{k_1}-\mu_{k_1}|]E[|Y_{k_2}-\mu_{k_2}|] + \sum_{k=1}^{m} E[|Y_k-\mu_k|^2] \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}||\mu_{k_2}| + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}|\frac{2\sigma_{k_2}}{\sqrt{2\pi}} + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_2}|\frac{2\sigma_{k_1}}{\sqrt{2\pi}} \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m} (\frac{2\sigma_{k_1}}{\sqrt{2\pi}})(\frac{2\sigma_{k_2}}{\sqrt{2\pi}}) + \sum_{k=1}^{m}\sigma_k^2 \qquad (5.33) \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}||\mu_{k_2}| + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_1}|\frac{2\sigma_{k_2}}{\sqrt{2\pi}} + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m} |\mu_{k_2}|\frac{2\sigma_{k_1}}{\sqrt{2\pi}} \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m} \frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m}\sigma_k^2. \qquad (5.34)
\end{aligned}
$$

Equation 5.31 uses the fact that $|a+b| \leq |a| + |b|$. Since Y_{k_1} and Y_{k_2} are statistically independent, the cross-terms in Equation 5.32 can be evaluated separately. Equation 5.33 uses the first and second central absolute moments of Y_k.

The expected value of Z_m is

$$\begin{aligned} E[Z_m] &= E[\sum_{k=1}^{m} |Y_k|] \\ &= \sum_{k=1}^{m} E[|Y_k|]. \end{aligned} \tag{5.35}$$

The expected value of $|Y_k|$, or the expected value of the folded normal distribution, is

$$\begin{aligned} E[|Y_k|] &= \int_{-\infty}^{\infty} |y_k| \frac{1}{\sigma_k\sqrt{2\pi}} e^{\frac{-(y_k-\mu_k)^2}{2\sigma_k^2}} dy_k \\ &= \frac{2\sigma_k}{\sqrt{2\pi}e^{\frac{\mu_k^2}{2\sigma_k^2}}} + \mu_k(1 - 2\Phi(-\frac{\mu_k}{\sigma_k})), \end{aligned} \tag{5.36}$$

so

$$E[Z_m] = \sum_{k=1}^{m} (\frac{2\sigma_k}{\sqrt{2\pi}e^{\frac{\mu_k^2}{2\sigma_k^2}}} + \mu_k(1 - 2\Phi(-\frac{\mu_k}{\sigma_k}))), \tag{5.37}$$

and

$$
\begin{aligned}
(E[Z_m])^2 &= (\sum_{k=1}^{m}(\frac{2\sigma_k}{\sqrt{2\pi}e^{\frac{\mu_k^2}{2\sigma_k^2}}} + \mu_k(1-2\Phi(-\frac{\mu_k}{\sigma_k}))))^2 \\
&= (\sum_{k=1}^{m}\frac{2\sigma_k}{\sqrt{2\pi}e^{\frac{\mu_k^2}{2\sigma_k^2}}} + \sum_{k=1}^{m}\mu_k(1-2\Phi(-\frac{\mu_k}{\sigma_k})))^2 \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}(\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))\cdot\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))) \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_2}\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_1}\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}} \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_1}\sigma_{k_2}}{\pi e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}(\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))\cdot\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))) \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_2}\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_1}\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}} \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m}\frac{2\sigma_{k_1}\sigma_{k_2}}{\pi e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} + \sum_{k=1}^{m}\frac{2\sigma_k^2}{\pi e^{\frac{\mu_k^2}{\sigma_k^2}}}. \qquad (5.38)
\end{aligned}
$$

Thus, the variance $\sigma^2_{Z_m}$ of Z_m is

$$\begin{aligned}
\sigma^2_{Z_m} &= E[Z_m^2] - (E[Z_m])^2 \\
&\leq \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}|\mu_{k_1}||\mu_{k_2}| + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}|\mu_{k_1}|\frac{2\sigma_{k_2}}{\sqrt{2\pi}} \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}|\mu_{k_2}|\frac{2\sigma_{k_1}}{\sqrt{2\pi}} + \sum_{k_1,k_2=1,k_1\neq k_2}^{m}\frac{2\sigma_{k_1}\sigma_{k_2}}{\pi} + \sum_{k=1}^{m}\sigma_k^2 \\
&\quad - (\sum_{k_1=1}^{m}\sum_{k_2=1}^{m}(\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))\cdot\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))) \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_2}\mu_{k_1}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_1}\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{\sqrt{2\pi}e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}} \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m}\frac{2\sigma_{k_1}\sigma_{k_2}}{\pi e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}} + \sum_{k=1}^{m}\frac{2\sigma_k^2}{\pi e^{\frac{\mu_k^2}{\sigma_k^2}}}) \\
&= \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}(|\mu_{k_1}||\mu_{k_2}| - \mu_{k_1}\mu_{k_2}(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))) \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_2}}{\sqrt{2\pi}}(|\mu_{k_1}| - \mu_{k_1}\frac{(1-2\Phi(-\frac{\mu_{k_1}}{\sigma_{k_1}}))}{e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}}) \\
&\quad + \sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\frac{2\sigma_{k_1}}{\sqrt{2\pi}}(|\mu_{k_2}| - \mu_{k_2}\frac{(1-2\Phi(-\frac{\mu_{k_2}}{\sigma_{k_2}}))}{e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}}) \\
&\quad + \sum_{k_1,k_2=1,k_1\neq k_2}^{m}\frac{2\sigma_{k_1}\sigma_{k_2}}{\pi}(1-\frac{1}{e^{\frac{\mu_{k_1}^2}{2\sigma_{k_1}^2}}e^{\frac{\mu_{k_2}^2}{2\sigma_{k_2}^2}}}) \\
&\quad + \sum_{k=1}^{m}\sigma_k^2(1-\frac{2}{\pi e^{\frac{\mu_k^2}{\sigma_k^2}}}). \qquad (5.39)
\end{aligned}$$

QED

Chapter 6

Closing the Loop on a Complete Linkage Hierarchical Clustering Method

To find meaningful levels of an $\frac{n \cdot (n-1)}{2}$ + 1-level hierarchical sequence, the means described in Chapter 5 includes constructing at least one distance graph, which is visually examined for features that correlate with meaningful levels of the corresponding hierarchical sequence. As a visual tool, however, a distance graph is not well suited for automation. Thus, the third part of the project resolved how to mathematically capture the graphical relationships that underlie these features and integrate the means into the new clustering method. By doing so, the new method becomes self-contained and can be fully automated or used with minimal operator supervision.

The same four assumptions that apply when distance graphs are visually examined also apply when the means is integrated into the new clustering method. The approach assumes that the 2-norms and the 1-norms of the data points are calculable. The approach also assumes that noise (random error) is the only random component in a measured value, that noise can be modeled as Gaussian random variables, and that the noise that is embedded in each dimension (sample) of each data point is statistically independent.

Evaluating ordered triples for linkage is decoupled from constructing cluster sets.

9 Motes Test Bed C
Euclidean distance
proxVector =

580.79	4	9
608.36	6	8
635.75	5	9
638.88	4	5
652.05	1	8
662.77	1	6
692.70	1	3
707.32	3	6
756.23	3	8
1224.18	2	7
1235.86	1	2
1251.43	2	3
1290.54	2	6
1299.89	2	8
1358.70	4	7
1391.82	2	4
1412.03	5	7
1417.16	2	5
1428.15	7	9
1436.95	3	7
1483.72	7	8
1485.47	2	9
...		
1938.60	1	9

d' = 1428.15
State =

	1	2	3	4	5	6	7	8	9
1	1	1	1	0	0	1	0	1	0
2	1	1	1	–	–	1	–	1	0
3	1	1	1	0	0	1	0	1	0
4	0	–	0	2	2	0	2	0	2
5	0	–	0	2	2	0	2	0	2
6	1	1	1	0	0	1	0	1	0
7	0	–	0	2	2	0	2	0	2
8	1	1	1	0	0	1	0	1	0
9	0	0	0	2	2	0	2	0	2

d' = 1485.47
State =

	1	2	3	4	5	6	7	8	9
1	1	1	1	0	0	1	0	1	0
2	1	1,2	1	2	2	1	2	1	2
3	1	1	1	0	0	1	–	1	0
4	0	2	0	2	2	0	2	0	2
5	0	2	0	2	2	0	2	0	2
6	1	1	1	0	0	1	0	1	0
7	0	2	–	2	2	0	2	–	2
8	1	1	1	0	0	1	–	1	0
9	0	2	0	2	2	0	2	0	2

Figure 6.1: Proximity vector and state matrices from a sensor system experiment similar to that described in Chapter 7. The different colored (gray scaled) numerals in the state matrices indicate which data points (sensor nodes) belong to the same cluster. For a more detailed explanation about how these data structures are used, see Chapter 4.

In contrast to storing intercluster distances in a proximity matrix, ordered triples comprised of interpoint distances and the indices of the respective data points are stored in proximity vectors. As the data structures in Fig. 6.1 show, only information about linkage is stored in a state matrix. Because cluster set construction is based solely on linkage between the data points, ordered triples can be evaluated for linkage and a state matrix can be updated without constructing cluster sets. Further, as described in Chapter 4, the new clustering method is *not* an updating method. Cluster sets are constructed *de novo*. Because a state matrix can be updated without constructing cluster sets, and the cluster set for every level of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence is constructed independently of the cluster sets for the other levels, it is possible to construct only the cluster sets for meaningful levels of a hierarchical sequence.

To mathematically capture the graphical relationships that underlie the above-described features of a distance graph, the rank order indices that coincide with meaningful levels of the corresponding hierarchical sequence, or the distance elements that

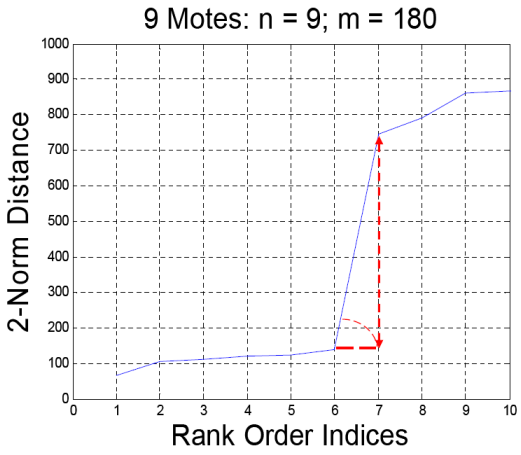


Figure 6.2: Lower left portion of a distance graph from the sensor system experiment described in Chapter 7. The enlargement shows one of the angles used to find meaningful levels of the corresponding hierarchical sequence. The dashed arrow represents $DISTROI_{i+1} - DISTROI_i$. Here, $DISTROI_{i+1}$ is the distance element of the 7th ordered triple and $DISTROI_i$ is the distance element of the 6th ordered triple.

coincide with the respective threshold distances d', must be identifiable *without* visually examining the curve of the distance graph. In other words, this objective must be attainable by looking only at the rank order indices and the information that is contained within the ordered triples. As described in Chapter 5, those sections of a curve that come after the lower-right corners and before the upper-left corners indicate where new configurations of clusters have finished forming. The approach focuses on the lower-right corners. These are the features that correspond to where the distance elements of every evaluated ordered triple and no others are less than the respective threshold distances d'. As the graph in Fig. 6.2 shows, these relationships can be mathematically captured by comparing 1) the tangent of the angle that is formed by the curve of the distance graph at each rank order index i, $i = 1, 2, ..., \frac{n\cdot(n-1)}{2} - 1$, and the positive x-axis of the graph with 2) the difference between the distance elements of the $i+1$th and ith ordered triples. The empirical experiments described in Chapter 7 show that this angle is typically between 60 degrees and 90^- degrees, or nearly orthogonal.

Proximity vectors are well suited for finding these angles. The ordered triples stored in a proximity vector are a permanent record of the interpoint distances between the

data points in a data set. After each ordered triple is evaluated for linkage, a test is performed to determine whether the corresponding level of the hierarchical sequence is meaningful. The ith level of a hierarchical sequence is *deemed* meaningful if the following test returns true:

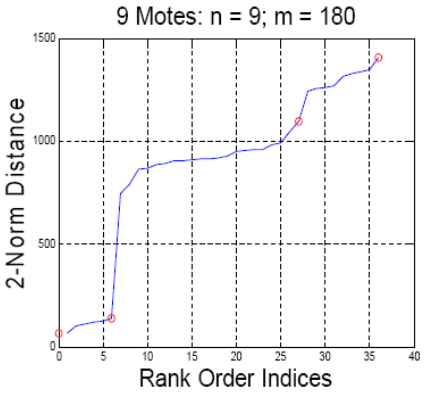
$$DISTROI_{i+1} - DISTROI_i \geq \tan(cutoffAngle) \cdot MAXDIST/MAXROI.$$

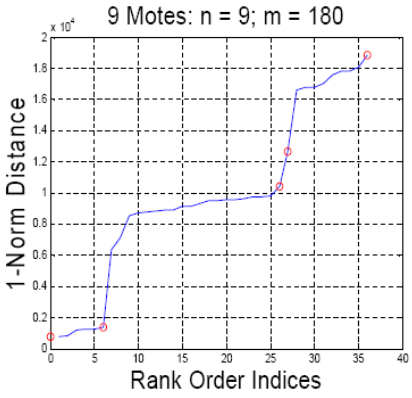
$DISTROI_{i+1}$ is the distance element of the $i+1$th ordered triple; $DISTROI_i$ is the distance element of the ith ordered triple; $cutoffAngle$ is the minimum angle that the curve of the distance graph must form with the positive x-axis of the graph at rank order index i, $i = 1, 2, ..., \frac{n\cdot(n-1)}{2} - 1$; $MAXDIST$ is the maximum distance element; and $MAXROI$ is the number of ordered triples or $\frac{n\cdot(n-1)}{2}$. The normalization factor is on the right side of the equation to reduce the number of multiplications. As a general rule, the first cluster set (all the data points are singletons) and the last cluster set (all the data points belong to the same cluster or stopping criteria are met) are always constructed.

Two parameters need tuning. One is the dimensionality beyond which inherent structure in a data set has good definition (or as good as is practically possible). The other is $cutoffAngle$. These can be tuned online, with minimal operator supervision, or hardwired, based on domain knowledge. Alternatively, it should be possible to learn them. The results for a data set can be characterized by the data set and the index $m(\angle cutoffAngle)$, where m is the dimensionality of the data points.

Example 6 *The data in Fig. 6.3 are from the sensor system experiment described in Chapter 7. The tables in Fig. 6.3 show how the test for finding deemed meaningful levels is applied when* $cutoffAngle = 70$ *degrees. For both Euclidean distance and city block distance, because the test returned true for rank order index = 6, the cluster sets for hierarchical level = 6 were constructed. The test returned false for rank order index = 0, but, as a general rule, the cluster sets for hierarchical level = 0 (all the data points are singletons) and hierarchical level* $= \frac{n\cdot(n-1)}{2} = 36$ *(all the data points belong to the same cluster) are always constructed. The small red circles on the curves of the distance graphs indicate which levels were deemed meaningful.*

Rank Order Index =	Euclidean Distance Between Sensor Nodes =			City Block Distance Between Sensor Nodes =		
1	65.42	2	4	782.00	2	9
2	103.64	6	8	798.00	2	4
3	109.73	2	9	1201.00	6	8
4	119.36	1	8	1234.60	4	9
5	123.34	1	6	1256.68	1	8
6	138.41	4	9	1388.54	1	6
7	744.69	3	5	6326.20	3	5
8	792.09	5	7	7115.20	5	7
9	861.82	1	3	8535.08	1	3
10	866.74	3	6	8724.80	3	6
11	887.06	1	5	8754.84	1	5
12	892.19	5	6	8863.80	5	6
13	903.30	5	9	8907.00	2	5
14	905.32	3	9	8929.40	5	9
15	911.62	2	7	9142.40	3	9
16	914.92	2	5	9168.40	2	3
17	914.94	7	9	9351.40	3	8
18	916.47	2	3	9507.40	4	5
19	927.69	3	8	9512.40	5	8
20	951.42	4	7	9567.50	1	7
21	953.45	5	8	9586.60	2	7
22	958.88	4	5	9627.40	4	7
23	960.06	3	4	9739.60	7	9
24	980.37	1	7	9747.60	3	4
25	990.93	6	7	9839.40	6	7
26	1044.64	7	8	10399.60	7	8
27	1098.08	3	7	12681.00	3	7
28	1243.18	1	9	16603.60	1	9
29	1256.73	1	2	16803.00	6	9
30	1258.61	6	9	16824.60	1	2
31	1270.49	2	6	17024.00	2	6
32	1316.28	1	4	17622.60	1	4
33	1330.22	4	6	17822.00	4	6
34	1335.90	8	9	17858.00	8	9
35	1348.05	2	8	18079.00	2	8
36	1407.63	4	8	18877.00	4	8





Rank Order Index	$DISTROI_{i+1}$ - $DISTROI_i$ (Euclidean Distance)	tan(*cutoffAngle*) • MAXDIST/MAXROI	Construct Cluster Set?
0	157.97 – 0.00 = 157.97	tan(70°)*(3136.20/36) = 239.35	YES
6	1681.71 - 287.96 = 1393.75	tan(70°)*(3136.20/36) = 239.35	YES
9	1920.82 - 1893.52 = 27.30	tan(70°)*(3136.20/36) = 239.35	NO
18	2088.72 - 2088.69 = 0.03	tan(70°)*(3136.20/36) = 239.35	NO
26	2488.62 - 2322.14 = 166.48	tan(70°)*(3136.20/36) = 239.35	NO
27	2731.73 - 2488.62 = 243.11	tan(70°)*(3136.20/36) = 239.35	YES
36	n/a	tan(70°)*(3136.20/36) = 239.35	YES

Rank Order Index	$DISTROI_{i+1}$ - $DISTROI_i$ (City Block Distance)	tan(*cutoffAngle*) • MAXDIST/MAXROI	Construct Cluster Set?
0	3731.50 – 0.00 = 3731.50	tan(70°)*(93661.00/36) = 7148.09	YES
6	33206.00 – 6723.20 = 26482.80	tan(70°)*(93661.00/36) = 7148.09	YES
9	42376.55 – 41981.00 = 395.55	tan(70°)*(93661.00/36) = 7148.09	NO
18	48412.80 – 46460.90 = 1951.90	tan(70°)*(93661.00/36) = 7148.09	NO
26	64391.60 – 50805.40 = 13586.20	tan(70°)*(93661.00/36) = 7148.09	YES
27	81358.71 – 64391.60 = 16967.11	tan(70°)*(93661.00/36) = 7148.09	YES
36	n/a	tan(70°)*(93661.00/36) = 7148.09	YES

Figure 6.3: Proximity vectors, distance graphs, and test results for Example 6.

Chapter 7

Empirical Experiments

This chapter describes empirical results from nine experiments. The first two experiments corroborate the theoretical work described in Chapter 5. The next three experiments show that the new clustering method performs well with respect to different kinds of data sets, including a data set that has no inherent structure, a data set comprised of nonmetric data, and a data set comprised of multiple attributes. The last four experiments show that the new clustering method compares favorably with the standard complete linkage method and k-means clustering with respect to accuracy and time. Further, they show that cluster sets constructed for meaningful levels can have real world meaning.

The data sets are representative of other data sets that have inherent structure. Euclidean distance and city block distance were used to calculate the distances. *level* is a variable that refers to individual meaningful levels, and d' refers to the respective threshold distances d'. Whether sample or population statistics are being described should be clear from the context.

7.1 Metrics for Evaluating the INCLude Algorithm

The following are metrics that were used to evaluate the INCLude algorithm:

1. The Rand Index was used to measure accuracy, i.e., how close cluster sets came to what they ought to have been. Although the Rand Index requires a benchmark, it is adaptable to each data set and can be used with ordinary statistics. *Accord*, [37].

To account for clusters that overlap, the scoring used the number of clusters to which a data point belongs. Singletons were not considered in the initial scoring. Instead, the number of singletons was added to an initial score to obtain a final score.

2. For each meaningful level of a hierarchical sequence, the number of recursive calls that were made and the depths at which they were made were recorded.

3. CPU time splits were recorded for how long it took to load a data set, calculate the distances between the data points and construct ordered triples, sort the ordered triples, evaluate ordered triples for linkage, perform the test for finding deemed meaningful levels, and construct a cluster set for every meaningful level. Each reported time is an average of five trials run on a Pentium 4 processor.

4. A binary score was used to record whether an operator could find all the meaningful levels of a hierarchical sequence by visually examining a distance graph.

5. Sample standard deviations for $\sigma_{Z_m,(i,j)}$ for Euclidean distance were calculated. The reader is encouraged to compare these values with the distances between the data points in a data set.

6. When the test for finding deemed meaningful levels was used, the number of false positives and the number of false negatives were recorded for *cutoffAngle*s between 60 degrees (inclusive) and 90 degrees (exclusive).

7. When the test for finding deemed meaningful levels was used, the indices $m(\angle cutoffAngle)$ were recorded when there were no false positives or false negatives.

7.2 Experiments

7.2.1 Sensitivity Analysis

A sensitivity analysis was conducted to explore how different parameters affect the distances between data points from two different classes (recall that a class refers to data points that have the same means or true values plus (measurement) biases). Suppose that each data point is a finite sequence of samples. The analysis assumed that the 2-norms and the 1-norms of the data points are calculable. It also assumed that noise (random error) is the only random component in a measured value, that noise can be modeled as Gaussian random variables, and that the noise that is embedded in each dimension (sample) of each data point is statistically independent.

Synthetic data sets were used to conduct the sensitivity analysis on four parameters: the number of data points in each of two classes (primitive clusters); the dimensionality of the data points; the biases between the means of the dimensions of two data points, one data point from one class and the other data point from the other; and the standard deviations of the normally distributed noise that was embedded in each dimension of each data point. The analysis had three parts. In the first part of the analysis, the number of data points was allowed to vary from 1 to 1000 in increments of a magnitude, the dimensionality of the data points was allowed to vary from 1 to 1000 in increments of a magnitude, constant biases between 1 and 100 were selected, and the standard deviations of the noise were 0.0, 1.0, 5.0, or 25.0. In the second part of the analysis, sinusoids that were +90, 180, or -90 degrees out-of-phase were used in place of constant biases. The standard deviations of the noise were either 1.0 or randomly selected by a pseudo-random number generator. In the third part of the analysis, the biases and the standard deviations of the noise were selected. The results from the third part of the analysis are described in Chapter 5. Qualitatively, the results for each part were the same.

Five trials were run for each combination of parameters, wherein the number of iterations equaled the number of data points from each class. For each iteration, a data point from each class was constructed, and Euclidean distance and city block distance were used to calculate the distances between the data points. For each trial, the smallest distance, the largest distance, the mean distance, and the standard deviation of all the distances were recorded. The following Matlab code was used for the first part of the

analysis:

```
%  Sensitivity Analyis
Distances = [];
Table = [];
numPoints = 1;
while (numPoints < 1001)
    for trials = 1:5
        for j = 1:numPoints
            DataPt1 = [];
            DataPt2 = [];
            for numDimensions = 1:100
                DataPt1 = [DataPt1; (3.0 + randn)];
            end
            for numDimensions = 1:100
                DataPt2 = [DataPt2; (0.0 + randn)];
            end
            temp = sqrt((DataPt1-DataPt2)'*(DataPt1-DataPt2));
            %temp = sum(abs(DataPt1-DataPt2));
            Distances = [Distances; temp];
        end
        minimum = min(Distances);
        maximum = max(Distances);
        ave = mean(Distances);
        stdDev = std(Distances);
        Table = [Table; minimum, maximum, ave, stdDev];
        Distances = [];
    end
    Table = [Table; 0,0,0,0 ]
    numPoints = 10*numPoints;
end
Table
```

The following is typical output from the first part of the analysis (the biases and the standard deviations of the noise were 10.0 and 5.0, respectively):

```
Tables for Euclidean distance:
For n = 1; m = 1:
```

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	16.1651	16.1651	16.1651	0
2	9.1883	9.1883	9.1883	0
3	1.6869	1.6869	1.6869	0
4	16.1340	16.1340	16.1340	0
5	10.7633	10.7633	10.7633	0

For n = 10; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.0606	24.2387	9.6921	7.7948
2	0.5574	21.9665	8.3474	7.2049
3	1.8745	24.8450	11.0895	6.7883
4	3.7943	19.6649	11.8897	5.1015
5	7.1364	17.5249	12.1706	3.7973

For n = 100; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	1.2533	25.3231	11.1883	6.1664
2	0.2383	27.2680	10.4899	5.9061
3	0.0907	26.2117	10.7417	6.0132
4	0.2317	31.7898	10.8938	6.5072
5	0.3766	28.3641	11.0207	6.7900

For n = 1000; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.0324	31.6841	10.5771	6.1735
2	0.0017	37.3624	10.7895	6.2767
3	0.0267	38.1430	10.9453	6.2786
4	0.0026	29.2737	10.4502	6.2401
5	0.0838	30.3125	10.5513	6.3312

For n = 1; m = 10:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	45.4196	45.4196	45.4196	0
2	45.1068	45.1068	45.1068	0
3	32.8506	32.8506	32.8506	0
4	38.9145	38.9145	38.9145	0
5	29.1670	29.1670	29.1670	0

For n = 10; m = 10:

```
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1    31.2182    48.1576    41.3633     5.5340
  2    27.4856    47.6731    39.2772     5.6044
  3    25.9285    49.5086    39.1750     7.4143
  4    30.0528    46.6554    39.2768     5.2174
  5    28.6109    43.5719    36.6400     5.5859
For n = 100; m = 10:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1    22.2757    58.4241    38.1750     6.7589
  2    24.3377    59.9263    37.8590     6.5527
  3    21.4018    50.8403    37.0385     5.9522
  4    25.2687    54.2406    38.9792     5.8918
  5    21.6336    55.3075    37.3137     6.6279
For n = 1000; m = 10:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1    17.3322    61.9282    38.5746     6.5473
  2    18.5492    59.5971    37.8223     6.4325
  3    15.9774    61.4785    37.9247     6.5178
  4    15.2459    59.6280    38.3079     6.5476
  5    17.2705    59.1476    38.0482     6.1844

For n = 1; m = 100:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   119.4742   119.4742   119.4742     0
  2   128.6743   128.6743   128.6743     0
  3   127.9521   127.9521   127.9521     0
  4   111.6458   111.6458   111.6458     0
  5   121.6402   121.6402   121.6402     0
For n = 10; m = 100:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   113.1612   129.2771   123.1337     4.5768
  2   105.5584   132.6450   120.6894     8.6416
  3   115.5458   131.8609   122.8502     5.6153
  4   111.4364   126.3339   122.1023     4.4261
  5   120.7219   143.2402   126.1494     7.2045
For n = 100; m = 100:
```

```
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   105.0716   142.8127   123.0314     6.6904
  2   110.7377   140.1337   122.6052     6.1432
  3   101.5216   135.4184   121.9489     6.6673
  4   103.5988   133.7046   122.4045     6.4847
  5   102.1891   146.2738   121.2834     7.1694
For n = 1000; m = 100:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   104.4055   142.9757   122.7153     6.5100
  2    97.3414   142.3790   122.4735     6.3722
  3    98.6294   142.2861   122.6025     6.5961
  4   102.8023   141.2818   122.2836     6.3307
  5   102.9883   142.0297   122.2572     6.3724

For n = 1; m = 1000:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   391.7118   391.7118   391.7118     0
  2   380.7883   380.7883   380.7883     0
  3   388.7398   388.7398   388.7398     0
  4   380.0285   380.0285   380.0285     0
  5   386.9480   386.9480   386.9480     0
For n = 10; m = 1000:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   376.8645   401.7325   385.3668     6.8676
  2   374.5326   401.0180   389.8071     7.7381
  3   373.8501   397.2981   386.8016     7.3196
  4   376.5763   390.7467   383.9229     4.0551
  5   373.4776   392.1919   382.6088     5.9764
For n = 100; m = 1000:
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1   370.8619   403.1648   386.4764     6.6938
  2   371.7582   407.5927   388.2864     6.5923
  3   374.4887   402.6155   386.7159     6.0678
  4   370.7308   400.9281   386.2859     5.7664
  5   370.6950   399.7883   386.0449     6.1712
For n = 1000; m = 1000:
```

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	365.1193	406.5452	387.1548	6.3642
2	364.0528	405.2790	387.0091	6.5229
3	367.3898	408.3097	387.5670	6.7097
4	364.0190	410.8071	387.4987	6.4123
5	367.1223	406.4071	387.4762	6.3882

Tables for City Block Distance:

For n = 1; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	4.9796	4.9796	4.9796	0
2	2.2437	2.2437	2.2437	0
3	15.8317	15.8317	15.8317	0
4	11.1481	11.1481	11.1481	0
5	2.6199	2.6199	2.6199	0

For n = 10; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.5575	22.0986	10.2290	7.5830
2	0.4358	19.3496	9.2499	6.0554
3	3.1088	18.7499	8.8628	4.6371
4	1.6976	19.5716	9.9743	6.6749
5	0.4523	20.1170	9.2068	6.4439

For n = 100; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.0155	31.0424	11.1560	6.7967
2	0.0508	29.6762	11.0468	5.9998
3	0.1959	27.7427	9.8014	6.8048
4	0.3100	27.9159	10.6043	6.0627
5	0.5396	25.2073	11.9620	6.1997

For n = 1000; m = 1:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.0189	34.9671	10.5881	6.3265
2	0.0147	31.1449	10.4381	6.4770
3	0.0105	38.7314	10.7381	6.4823
4	0.0138	32.1403	10.6363	6.2659
5	0.0301	31.1154	10.4908	6.4765

For n = 1; m = 10:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	110.6455	110.6455	110.6455	0
2	78.3837	78.3837	78.3837	0
3	93.8307	93.8307	93.8307	0
4	92.5270	92.5270	92.5270	0
5	119.1971	119.1971	119.1971	0

For n = 10; m = 10:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	45.3414	121.0636	96.1800	20.5533
2	84.4374	140.0120	112.1939	20.7626
3	73.7437	140.1974	106.1177	20.7385
4	75.8930	138.7107	110.7964	17.3472
5	96.5520	131.5100	112.0467	9.9211

For n = 100; m = 10:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	61.1111	156.6582	104.8747	18.4651
2	62.2369	145.7452	105.8873	19.0308
3	62.1325	185.8941	105.1179	20.6575
4	55.0409	170.3202	107.2951	18.6647
5	68.8813	152.6990	104.6532	18.0238

For n = 1000; m = 10:

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	49.2489	167.4608	105.8137	20.6469
2	36.7438	172.8119	105.7570	20.0763
3	48.5811	164.4157	104.9397	20.2313
4	46.7143	170.3185	104.1931	20.0308
5	42.5079	160.9262	104.7849	20.1318

For n = 1; m = 100:

1.0e+003*values

TRIAL	MIN DIST	MAX DIST	MEAN DIST	STD DIST
1	0.9991	0.9991	0.9991	0
2	1.0480	1.0480	1.0480	0
3	1.0091	1.0091	1.0091	0

```
  4     1.1067     1.1067     1.1067     0
  5     0.9272     0.9272     0.9272     0
For n = 10; m = 100:
1.0e+003*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     0.9848     1.1018     1.0627     0.0417
  2     0.9786     1.1535     1.0449     0.0511
  3     0.9301     1.1669     1.0562     0.0800
  4     0.9670     1.1750     1.0746     0.0602
  5     0.9747     1.1202     1.0582     0.0520
For n = 100; m = 100:
1.0e+003*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     0.9273     1.1916     1.0463     0.0542
  2     0.9139     1.1846     1.0496     0.0619
  3     0.8486     1.1923     1.0512     0.0575
  4     0.8630     1.1993     1.0587     0.0642
  5     0.8752     1.2533     1.0560     0.0706
For n = 1000; m = 100:
1.0e+003*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     0.8648     1.2379     1.0507     0.0623
  2     0.8209     1.2461     1.0478     0.0629
  3     0.8289     1.2439     1.0512     0.0610
  4     0.8796     1.2291     1.0499     0.0623
  5     0.8433     1.2285     1.0484     0.0622

For n = 1; m = 1000:
1.0e+004*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     1.0720     1.0720     1.0720     0
  2     1.0706     1.0706     1.0706     0
  3     1.0443     1.0443     1.0443     0
  4     1.0622     1.0622     1.0622     0
  5     1.0498     1.0498     1.0498     0
For n = 10; m = 1000:
```

```
1.0e+004*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     1.0280     1.0638     1.0424     0.0120
  2     1.0349     1.0782     1.0576     0.0156
  3     1.0126     1.0699     1.0441     0.0192
  4     1.0185     1.0694     1.0451     0.0148
  5     1.0116     1.0774     1.0444     0.0234
For n = 100; m = 1000:
1.0e+004*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     1.0115     1.1048     1.0542     0.0204
  2     0.9990     1.1006     1.0483     0.0192
  3     1.0124     1.0927     1.0519     0.0178
  4     0.9916     1.1075     1.0485     0.0210
  5     1.0018     1.1012     1.0510     0.0198
For n = 1000; m = 1000:
1.0e+004*values
TRIAL MIN DIST   MAX DIST  MEAN DIST   STD DIST
  1     0.9938     1.1055     1.0499     0.0198
  2     0.9721     1.1075     1.0495     0.0198
  3     0.9910     1.1118     1.0509     0.0202
  4     0.9741     1.1186     1.0508     0.0203
  5     1.0026     1.1066     1.0514     0.0200
```

As the exemplary data in Fig. 7.1 show, with respect to Euclidean distance, the minimum distances decreased slightly and the maximum distances increased slightly as the number of data points increased. As the number of data points increased, the likelihood increased that two data points were very close together or far apart. Increasing the dimensionality of the data points or increasing the biases between two data points increased the distances between them. However, the number of data points, the dimensionality of the data points, and the biases between two data points had no appreciable effect on the standard deviations of the distances. Even as the distances between the data points increased, the standard deviations of the distances remained roughly unchanged. On the other hand, increasing the standard deviations of the noise increased the distances between the data points and increased the standard deviations

of the distances. These results are consistent with Equation 5.5 in Chapter 5.

With respect to city block distance, increasing the number of data points did not have an appreciable effect on the distances between the data points or the standard deviations of the distances. Increasing the dimensionality of the data points increased the distances between the data points and the standard deviations of the distances. Increasing the biases between two data points increased the distances between the data points but had no appreciable effect on the standard deviations of the distances. Increasing the standard deviations of the noise also increased the distances between the data points and the standard deviations of the distances. The increases in the standard deviations of the distances were much smaller than the increases in the distances between the data points.

These surprising results suggest that increasing the dimensionality of the data points or reducing the standard deviations of the noise can attenuate the effects of noise on cluster set construction. With respect to dimensionality, this can be achieved by either increasing the sampling rates or extending the sampling time intervals.

Euclidean Distance

PAIRS	DIM	BIAS	STD1	STD2	DMIN	DMAX	DMEAN	STDDIST
100	100	10.0	5.0	5.0	101.5	146.3	122.4	7.2
1000	100	10.0	5.0	5.0	97.3	143.0	122.5	6.6
100	1000	10.0	5.0	5.0	370.6	407.6	386.5	6.2
100	100	100.0	5.0	5.0	978.9	1021.6	1002.3	7.4
100	100	10.0	25.0	25.0	291.6	448.5	365.3	29.0
100	100	10.0	5.0	25.0	213.5	329.5	270.9	19.0
100	100	10.0	25.0	5.0	219.6	325.3	273.9	17.9

City Block Distance

PAIRS	DIM	BIAS	STD1	STD2	DMIN	DMAX	DMEAN	STDDIST
100	100	10.0	5.0	5.0	848.6	1253.3	1051.2	70.6
1000	100	10.0	5.0	5.0	820.9	1246.1	1049.9	62.9
100	1000	10.0	5.0	5.0	9916.0	11,075.0	10,510.0	210.0
100	100	100.0	5.0	5.0	9767.0	10,240.0	9995.0	76.0
100	100	10.0	25.0	25.0	2291.8	3529.6	2938.1	217.9
100	100	10.0	5.0	25.0	1716.0	2845.7	2193.1	187.7
100	100	10.0	25.0	5.0	1705.0	2690.5	2203.5	170.2

PAIRS = Number of data point pairs
DIM = Number of dimensions in each data point
BIAS = Bias between pairs of data points
STD1 = Std. dev. of noise embedded in first data point dimensions
STD2 = Std. dev. of noise embedded in second data point dimensions
DMIN = Minimum calculated distance (5 trials)
DMAX = Maximum calculated distance (5 trials)
DMEAN = Mean of the distances (5 trials)
STDDIST = Std. devs. of the distances (5 trials)

Figure 7.1: Exemplary results from the first part of the sensitivity analysis.

7.2.2 Monte Carlo Simulation

As described in Theorem 2, Chapter 5, the variance $\sigma^2_{Z_{m,(i,j)}}$ for the random variable $Z_{m,(i,j)} = (\sum_{k=1}^{m} Y^2_{k,(i,j)})^{\frac{1}{2}}$ is $\frac{\sum_{k=1}^{m} \sigma^4_{k,(i,j)}}{2(\sum_{k=1}^{m} \sigma^2_{k,(i,j)} + \sum_{k=1}^{m} \mu^2_{k,(i,j)})} + \frac{\sum_{k=1}^{m} \sigma^2_{k,(i,j)} \mu^2_{k,(i,j)}}{\sum_{k=1}^{m} \sigma^2_{k,(i,j)} + \sum_{k=1}^{m} \mu^2_{k,(i,j)}}$ plus an error term that converges to 0 as $m \to \infty$. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$ and $\mu_{k,(i,j)} = \mu_{(i,j)}$, $k = 1, 2, ..., m$, the result in Theorem 2 becomes $\frac{\sigma^2_{(i,j)}}{2(1+\frac{\mu^2_{(i,j)}}{\sigma^2_{(i,j)}})} + \frac{\mu^2_{(i,j)}}{(1+\frac{\mu^2_{(i,j)}}{\sigma^2_{(i,j)}})}$. If $\sigma_{(i,j)}$ is held constant while $\mu_{(i,j)}$ is allowed to vary between 0 and $|\mu_{(i,j)}| \gg \sigma_{(i,j)}$, the result is a constant between $\frac{1}{2}\sigma^2_{(i,j)}$ and $\sigma^2_{(i,j)}$. Likewise, if $\mu_{(i,j)}$ is held constant while $\sigma_{(i,j)}$ is allowed to vary between 0 and $\sigma_{(i,j)} \gg |\mu_{(i,j)}|$, the result is a constant between 0 and $\frac{1}{2}\sigma^2_{(i,j)} + \mu^2_{(i,j)}$.

To examine the behavior of the result when neither $\sigma_{k,(i,j)}$ nor $\mu_{k,(i,j)}$ is held constant, the Monte Carlo method was used. A uniform distribution pseudo-random number generator was used to select $\sigma_{k,(i,j)}$ and $|\mu_{k,(i,j)}|$, which were selected from the intervals $(0, S]$ and $(0, M]$, respectively. The dimensionality m of the data points was allowed to vary from 1 to 1000 in increments of a magnitude. For each combination of parameters, 1000 iterations were performed. The following Matlab code was used to implement the Monte Carlo method:

```
%Monte Carlo Simulation
limitSigma = 10000;     limitSigma is S.
limitMu = 1000000;      limitMu is M.
dimensions = 1000;
iterations = 1000;
sigma = zeros(1,dimensions);
mu = zeros(1,dimensions);
variances = zeros(1,iterations);
for i = 1:iterations
    for j = 1:dimensions
        sigma(j) = rand * limitSigma;
        mu(j) = rand * limitMu;
    end
    temp1 = sigma.^2;
    temp2 = sum(temp1);
    temp3 = temp1.^2;
    temp4 = sum(temp3);
```

```
    temp5 = mu.^2;
    temp6 = sum(temp5);
    variances(i) = (temp4/(2*(temp2+temp6))) + ((temp2*temp6)/(temp2+temp6));
end
variances
dimensions
meanVariance = mean(variances)
stdVariance = std(variances)
```

The results from the Monte Carlo method are provided in Figs. 7.2 to 7.7. As the bound M on $\mu_{k,(i,j)}$ increased, the mean for the result in Theorem 2 was $\frac{m}{3}S^2$ plus an error term that became relatively small, i.e., $\frac{error}{\frac{m}{3}S^2} \to 0$ as $m \to \infty$. More precisely, as m increased, the standard deviation became smaller relative to the magnitude of the mean. Where $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$, and as the bound M on $\mu_{k,(i,j)}$ increased, the mean for the result in Theorem 2 was mS^2 plus an error term that converged to 0 as $m \to \infty$. The standard deviation decreased to 0 absolutely.

m = 1		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	0.2164	0.3071	0.3417	0.3456	0.3262		
	Std. Dev.	0.1847	0.2788	0.2902	0.3003	0.2956		
S = 10	Mean	16.6453	22.8589	28.5900	32.7656	31.8203		
	Std. Dev.	14.4751	19.1804	26.4066	29.4626	29.3760		
S = 100	Mean	1.6012×10^3	1.7282×10^3	2.1367×10^3	3.2329×10^3	3.3126×10^3		
	Std. Dev.	1.4911×10^3	1.5033×10^3	1.8947×10^3	2.8162×10^3	2.9767×10^3		
S = 1000	Mean	1.6702×10^5	1.6587×10^5	1.6921×10^5	2.2064×10^5	3.1288×10^5		
	Std. Dev.	1.4714×10^5	1.4976×10^5	1.5357×10^5	1.9373×10^5	2.7916×10^5		
S = 10,000	Mean	1.6556×10^7	1.6972×10^7	1.6583×10^7	1.6544×10^7	2.0475×10^7	3.4521×10^7	3.2446×10^7
	Std. Dev.	1.4602×10^7	1.4803×10^7	1.4939×10^7	1.4879×10^7	1.8193×10^7	2.9631×10^7	2.9691×10^7

m = 10		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	1.7586	3.2844	3.3662	3.4546	3.3818		
	Std. Dev.	0.3665	0.8988	0.9538	0.9289	0.9568		
S = 10	Mean	31.7857	175.1464	330.7691	329.6480	333.9324		
	Std. Dev.	5.7100	35.9233	91.9334	91.1942	95.5827		
S = 100	Mean	2.8648×10^3	3.1915×10^3	1.7315×10^4	3.2725×10^4	3.3853×10^4		
	Std. Dev.	570.4094	562.9641	3.7006×10^3	9.4252×10^3	9.4242×10^3		
S = 1000	Mean	2.8521×10^5	2.8744×10^5	3.1581×10^5	1.7470×10^6	3.2791×10^6	3.3241×10^6	
	Std. Dev.	5.7724×10^4	5.5010×10^4	5.9153×10^4	3.6269×10^5	9.3245×10^5	9.2684×10^5	
S = 10,000	Mean	2.9110×10^7	2.8945×10^7	2.8921×10^7	3.1780×10^7	1.7476×10^8	3.2584×10^8	3.3471×10^8
	Std. Dev.	5.5798×10^6	5.5865×10^6	5.8189×10^6	5.7477×10^6	3.6945×10^7	9.2841×10^7	9.5245×10^7

Figure 7.2: Results from the Monte Carlo method where $m = 1$ and $m = 10$.

m = 100		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	16.7031	33.0165	33.1896	33.4825	33.3658		
	Std. Dev.	1.0231	2.8170	2.9347	2.9739	2.8974		
S = 10	Mean	62.8224	1.6790×10^{3}	3.2989×10^{3}	3.3316×10^{3}	3.3353×10^{3}		
	Std. Dev.	3.2911	106.9303	292.4774	288.4194	280.6561		
S = 100	Mean	3.0251×10^{3}	6.2374×10^{3}	1.6748×10^{5}	3.3046×10^{5}	3.3046×10^{5}	3.3374×10^{5}	
	Std. Dev.	164.3501	338.5670	1.0488×10^{4}	2.9327×10^{4}	2.9327×10^{4}	2.9792×10^{4}	
S = 1000	Mean	2.9921×10^{5}	3.0174×10^{5}	6.2583×10^{5}	1.6788×10^{7}	1.6788×10^{7}	3.3060×10^{7}	3.3363×10^{7}
	Std. Dev.	1.6419×10^{4}	1.5787×10^{4}	3.2450×10^{4}	1.0263×10^{6}	1.0263×10^{6}	2.8868×10^{6}	2.8925×10^{6}
S = 10,000	Mean	2.9941×10^{7}	2.9897×10^{7}	3.0242×10^{7}	6.2674×10^{7}	1.6768×10^{9}	3.2940×10^{9}	3.3417×10^{9}
	Std. Dev.	1.6087×10^{6}	1.6405×10^{6}	1.6213×10^{6}	3.3051×10^{6}	1.0760×10^{8}	3.0699×10^{8}	2.8946×10^{8}

m = 1000		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	166.8613	329.5349	333.4158	333.8253	333.0504		
	Std. Dev.	3.1991	9.0996	9.4794	9.1363	9.3547		
S = 10	Mean	359.9207	1.6676×10^{4}	3.2995×10^{4}	3.3331×10^{4}	3.3337×10^{4}		
	Std. Dev.	9.5009	343.2315	946.1907	979.5355	938.1644		
S = 100	Mean	3.3327×10^{3}	3.5946×10^{4}	1.6688×10^{6}	3.2933×10^{6}	3.3096×10^{6}	3.3323×10^{6}	
	Std. Dev.	51.7747	925.8842	3.3774×10^{4}	9.5124×10^{4}	9.4465×10^{4}	9.3662×10^{4}	
S = 1000	Mean	3.0054×10^{5}	3.3310×10^{5}	3.5986×10^{6}	1.6674×10^{8}	3.2991×10^{8}	3.3311×10^{8}	
	Std. Dev.	4.0973×10^{3}	5.3615×10^{3}	9.0531×10^{4}	3.2410×10^{6}	9.3370×10^{6}	9.6756×10^{6}	
S = 10,000	Mean	2.9985×10^{7}	3.0031×10^{7}	3.3321×10^{7}	3.5946×10^{8}	1.6669×10^{10}	3.3022×10^{10}	3.3329×10^{10}
	Std. Dev.	5.0396×10^{5}	5.1243×10^{5}	5.3067×10^{5}	9.5298×10^{6}	3.3793×10^{8}	8.9987×10^{8}	9.8015×10^{8}

Figure 7.3: Results from the Monte Carlo method where $m = 100$ and $m = 1000$.

m = 1		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	0.6110	0.9253	0.9919	0.9993	1.0000		
	Std. Dev.	0.0810	0.1193	0.0410	0.0079	7.0462×10^{-4}		
S = 10	Mean	50.1218	60.5451	93.0859	99.0939	99.9311		
	Std. Dev.	0.1512	7.9983	11.0970	4.8038	1.3021		
S = 100	Mean	5.0002×10^{3}	5.0161×10^{3}	6.0797×10^{3}	9.2554×10^{3}	9.9148×10^{3}	9.9808×10^{3}	9.9998×10^{3}
	Std. Dev.	0.1494	14.6616	807.5712	1.1829×10^{3}	469.1169	266.9066	5.6679
S = 1000	Mean	5.0000×10^{5}	5.0002×10^{5}	5.0160×10^{5}	6.0875×10^{5}	9.3492×10^{5}	9.9379×10^{5}	9.9861×10^{5}
	Std. Dev.	0.1503	14.7348	1.4380×10^{3}	7.9713×10^{4}	1.0825×10^{5}	3.6300×10^{4}	2.2511×10^{4}
S = 10,000	Mean	5.0000×10^{7}	5.0000×10^{7}	5.002×10^{7}	5.0171×10^{7}	6.0795×10^{7}	9.2650×10^{7}	9.2650×10^{7}
	Std. Dev.	0.1442	14.8831	1.5032×10^{3}	1.4917×10^{5}	8.1011×10^{6}	1.1670×10^{7}	1.1670×10^{7}

m = 1		M = 10 M	M = 100 M	M = 1B	M = 10 B	M = 100 B
S = 1	Mean					
	Std. Dev.					
S = 10	Mean					
	Std. Dev.					
S = 100	Mean					
	Std. Dev.					
S = 1000	Mean	10.0000×10^{5}				
	Std. Dev.	31.7178				
S = 10,000	Mean	9.9271×10^{7}	9.9821×10^{7}	9.9972×10^{7}	9.9999×10^{7}	10.0000×10^{7}
	Std. Dev.	4.0866×10^{6}	2.4941×10^{6}	8.8218×10^{5}	3.1393×10^{4}	289.2051

Figure 7.4: Results from the Monte Carlo method where $m = 1$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$.

m = 10		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	2.8228	9.6951	9.9969	10.0000	10.0000		
	Std. Dev.	0.5243	0.1070	0.0012	1.0701×10^{-5}	1.0412×10^{-7}		
S = 10	Mean	53.1896	285.6368	969.7313	999.6886	999.9969		
	Std. Dev.	0.9016	50.6558	10.3323	0.1113	0.0011		
S = 100	Mean	5.0031×10^{3}	5.3179×10^{3}	2.8429×10^{4}	9.6966×10^{4}	9.9969×10^{4}		
	Std. Dev.	0.8720	88.7740	5.1733×10^{3}	1.1237×10^{3}	11.0142		
S = 1000	Mean	5.0000×10^{5}	5.0032×10^{5}	5.3155×10^{5}	2.8814×10^{6}	9.6992×10^{6}	9.9969×10^{6}	10.000×10^{6}
	Std. Dev.	0.9009	86.1107	8.9166×10^{3}	4.9377×10^{5}	9.5614×10^{4}	1.0797×10^{3}	10.9602
S = 10,000	Mean	5.0000×10^{7}	5.0000×10^{7}	5.0031×10^{7}	5.3136×10^{7}	2.8369×10^{8}	9.7005×10^{8}	9.9969×10^{8}
	Std. Dev.	0.8693	87.7391	9.1727×10^{3}	8.8504×10^{5}	4.8420×10^{7}	9.5829×10^{6}	1.1186×10^{5}

m = 10		M = 10 M	M = 100 M	M = 1B	M = 10 B	M = 100 B
S = 1	Mean					
	Std. Dev.					
S = 10	Mean					
	Std. Dev.					
S = 100	Mean					
	Std. Dev.					
S = 1000	Mean					
	Std. Dev.					
S = 10,000	Mean	10.0000×10^{8}	10.0000×10^{8}			
	Std. Dev.	1.1102×10^{3}	10.9803			

Figure 7.5: Results from the Monte Carlo method where $m = 10$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$.

m = 100		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	25.2527	97.0851	99.9699	99.9997	100.0000		
	Std. Dev.	1.6603	0.2648	0.0028	2.6403×10^{-7}	2.6712×10^{-7}		
S = 10	Mean	83.0980	2.5290×10^{3}	9.7086×10^{3}	9.9970×10^{3}	10.0000×10^{3}		
	Std. Dev.	2.9572	163.8519	26.3153	0.2604	0.0027		
S = 100	Mean	5.0333×10^{3}	8.2932×10^{3}	2.5294×10^{5}	9.7090×10^{5}	9.9970×10^{5}		
	Std. Dev.	2.9400	300.7448	1.6125×10^{4}	2.5724×10^{3}	27.2397		
S = 1000	Mean	5.0003×10^{5}	5.0331×10^{5}	8.3067×10^{5}	2.5318×10^{7}	9.7078×10^{7}	9.9970×10^{7}	10.0000×10^{7}
	Std. Dev.	2.8669	309.0447	2.9639×10^{4}	1.6997×10^{6}	2.6043×10^{5}	2.6273×10^{3}	25.8470
S = 10,000	Mean	5.0000×10^{7}	5.0003×10^{7}	5.0331×10^{7}	8.3112×10^{7}	2.5408×10^{9}	9.7092×10^{9}	9.9970×10^{9}
	Std. Dev.	2.8434	300.8390	3.0111×10^{4}	2.9709×10^{6}	1.6630×10^{8}	2.5371×10^{7}	2.6321×10^{5}

m = 100		M = 10 M	M = 100 M	M = 1B	M = 10 B	M = 100 B
S = 1	Mean					
	Std. Dev.					
S = 10	Mean					
	Std. Dev.					
S = 100	Mean					
	Std. Dev.					
S = 1000	Mean					
	Std. Dev.					
S = 10,000	Mean	10.000×10^{9}	10.0000×10^{9}			
	Std. Dev.	2.7784×10^{3}	27.6581			

Figure 7.6: Results from the Monte Carlo method where $m = 100$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$.

m = 1000		M = 1	M = 10	M = 100	M = 1000	M = 10,000	M = 100,000	M = 1 M
S = 1	Mean	250.2091	970.8809	999.6996	999.9970	1000.0000		
	Std. Dev.	5.3045	0.7705	0.0085	8.3417×10^{-5}	8.6715×10^{-7}		
S = 10	Mean	381.5986	2.5044×10^{4}	9.7089×10^{4}	9.9970×10^{4}	10.0000×10^{4}		
	Std. Dev.	9.3851	544.8986	78.2120	0.8223	0.0084		
S = 100	Mean	5.3328	3.8210×10^{4}	2.5017×10^{6}	9.7087×10^{6}	9.9970×10^{6}		
	Std. Dev.	9.5372	925.6256	5.3507×10^{4}	8.1008×10^{3}	85.9297		
S = 1000	Mean	5.0033×10^{5}	5.3337×10^{5}	3.8209×10^{6}	2.5042×10^{8}	9.7086×10^{8}	9.9970×10^{8}	10.0000×10^{8}
	Std. Dev.	9.1666	951.3390	9.3948×10^{4}	5.3214×10^{6}	3.9915×10^{5}	8.2412×10^{3}	86.9226
S = 10,000	Mean	5.0000×10^{7}	5.0033×10^{7}	5.3333×10^{7}	3.8209×10^{8}	2.5022×10^{10}	9.7086×10^{10}	9.9970×10^{10}
	Std. Dev.	9.3420	920.3269	9.3829×10^{4}	9.3593×10^{6}	5.1293×10^{8}	8.243×10^{7}	86015×10^{5}

m = 1000		M = 10 M	M = 100 M	M = 1B	M = 10 B	M = 100 B
S = 1	Mean					
	Std. Dev.					
S = 10	Mean					
	Std. Dev.					
S = 100	Mean					
	Std. Dev.					
S = 1000	Mean					
	Std. Dev.					
S = 10,000	Mean	10.0000×10^{10}	10.0000×10^{10}			
	Std. Dev.	8.6889×10^{3}	81.9715			

Figure 7.7: Results from the Monte Carlo method where $m = 1000$ and $\sigma_{k,(i,j)} = \sigma_{(i,j)}$, $k = 1, 2, ..., m$.

7.2.3 Structureless Data Sets

This experiment examined what distance graphs look like for structureless data sets. A uniform distribution pseudo-random number generator was used to construct 100 data points, whose dimensionality was increased from 1 to 1 million by increments of two magnitudes. Euclidean distance and city block distance were used to calculate the distances between the data points, and the sets of distances were graphed.

For the data set where $m = 1000$ dimensions, the maximum standard deviation of the noise that was embedded within a data point was 3.34, and the mean standard deviation of the noise that was embedded within the data points was 2.87. As the distance graphs in Fig. 7.8 show, the means for finding meaningful levels did not impose structure on the data sets. All the graphs appear smooth, indicating the absence of multiple classes of data points that disassociate from one another.

The data set where $m = 10{,}000$ dimensions was used to evaluate the test for finding deemed meaningful levels. As the data in Fig. 7.9 show for Euclidean distance, 9 false positives occurred at 10K($\angle$60), 9 false positives at 10K($\angle$65), 3 false positives at 10K($\angle$70), 3 false positives at 10K($\angle$75), 1 false positive at 10K($\angle$80), and no false positives at 10K($\angle$85). For city block distance, 6 false positives occurred at 10K($\angle$60), 5 false positives at 10K($\angle$65), 5 false positives at 10K($\angle$70), 3 false positives at 10K($\angle$75), 1 false positive at 10K($\angle$80), and no false positives at 10K($\angle$85). The false positives came at either end of the hierarchical sequences for both distances. The numbers of false positives are 0.40 to 0.50 percent of the total number of levels.

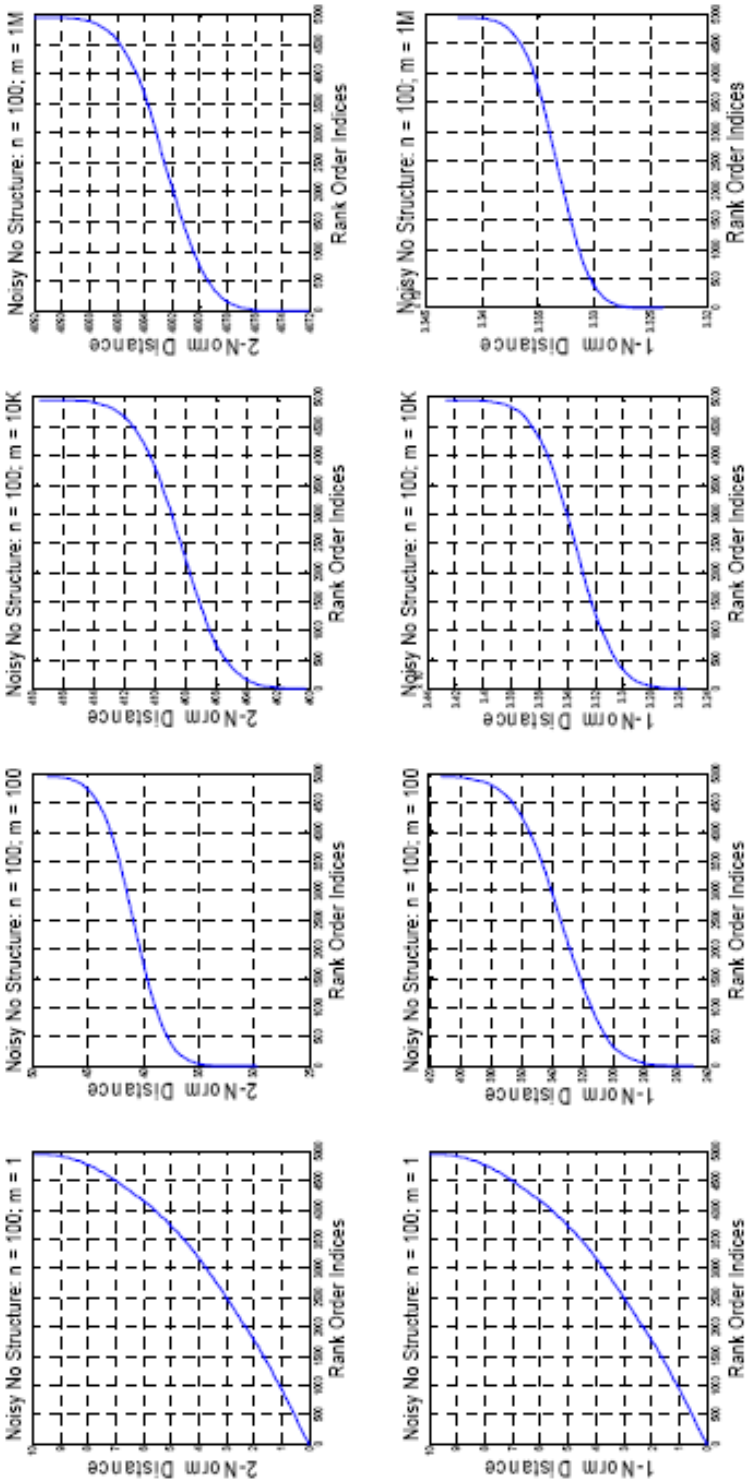


Figure 7.8: Distance graphs for the structureless data set. Inherent structure does not emerge as the dimensionality of the data points increases.

Human Inspect.	60 Degrees	65 Degrees	70 Degrees	75 Degrees	80 Degrees	85 Degrees	60 Degrees	65 Degrees	70 Degrees	75 Degrees	80 Degrees	85 Degrees
	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance
Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start
	1	1										
	2	2	2	2								
	3	3	3	3			3	3	3	3	3	
	4	4	4	4	4							
	4930	4930										
	4935	4935										
							4941	4941	4941			
	4944	4944					4944	4944	4944			
							4945					
	4946	4946										
							4947	4947	4947	4947		
	4948	4948										
							4949	4949	4949	4949		
4950	4950	4950	4950	4950	4950	4950	4950	4950	4950	4950	4950	4950

Figure 7.9: Deemed meaningful levels for the structureless data set where $m = 10{,}000$.

Ethnic Marriages
disparity matrix =

0	6	21	30	18	27	33	32
6	0	12	21	14	15	23	23
21	12	0	20	22	24	24	20
30	21	20	0	19	37	41	37
18	14	22	19	0	30	30	28
27	15	24	37	30	0	24	21
33	23	24	41	30	24	0	20
32	23	20	37	28	21	20	0

Figure 7.10: Disparity matrix for the ethnic marriages data set.

7.2.4 Ethnic Marriages Data Set

The 8-class, ethnic marriages data set [10] is a nonmetric data set that includes ties between intercluster distances and ties between interpoint distances. The data set consists of the number of marriages that occurred between different ethnic groups in Hawaii from 1948 to 1953. The data has been normalized for overall marriage rates, to correct for differences in size among the groups, converted into a disparity measure, and summarized in the disparity matrix shown in Fig. 7.10.

The new clustering method, the standard complete linkage method, and Peay's clique detection method were used to cluster the data points. The cluster sets are compared in Figs. 7.11 and 7.12. Two three-way ties between the intercluster distances caused the standard complete linkage method to branch twice, resulting in five competing hierarchical sequences of cluster sets. The results from the new clustering method and Peay's clique detection method are nearly the same, as expected. For 5 of the levels of the hierarchical sequence described in [10][1], Peay's clique detection method constructed more clusters (cliques) than the new clustering method, because it constructs clusters from which the data points migrate.

[1] There are $\frac{8\cdot(8-1)}{2} + 1 = 29$ levels in total. Where ties exist between interpoint distances, [10] provides results only for the last among these levels.

Hierarchical Level	Ordered Triple	Standard Complete Linkage Method	INCLude Hierarchical Clustering	Clique Detection
0		1, 2, 3, 4, 5, 6, 7, 8	1, 2, 3, 4, 5, 6, 7, 8	1, 2, 3, 4, 5, 6, 7, 8
1	6 1 2	{1,2}, 3, 4, 5, 6, 7, 8	{1,2}, 3, 4, 5, 6, 7, 8	{1,2}, 3, 4, 5, 6, 7, 8
2	12 2 3		{1,2}, {2,3}, 4, 5, 6, 7, 8	{1,2}, {2,3}, 4, 5, 6, 7, 8
3	14 2 5		{1,2}, {2,3}, {2,5}, 4, 6, 7, 8	{1,2}, {2,3}, {2,5}, 4, 6, 7, 8
4	16 2 6		{1,2}, {2,3}, {2,5}, {2,6}, 4, 7, 8	{1,2}, {2,3}, {2,5}, {2,6}, 4, 7, 8
5	18 1 5	{1,2,5}, 3, 4, 6, 7, 8	{1,2,5}, {2,3}, {2,6}, 4, 7, 8	{1,2,5}, {2,3}, {2,6}, 4, 7, 8
6	19 4 5		{1,2,5}, {2,3}, {4,5}, {2,6}, 7, 8	{1,2,5}, {2,3}, {4,5}, {2,6}, 7, 8
7	20 3 4	Path I: {1,2,5}, {3,4}, 6, 7, 8 Path III: {1,2,5}, {3,4}, {7,8}, 6	{1,2,5}, {2,6}, {2,3}, {3,4}, {4,5}, 7, 8	Not available
8	20 3 8	Path II: {1,2,5}, {3,8}, 4, 6, 7	{1,2,5}, {2,6}, {3,8}, {3,4}, {4,5}, 7	Not available
9	20 7 8	Path I: {1,2,5}, {3,4}, {7,8}, 6 Path III: {1,2,5}, {7,8}, 3, 4, 6	{1,2,5}, {2,6}, {7,8}, {3,4}, {4,5}, {2,3}, {3,8}	{1,2,5}, {2,6}, {7,8}, {3,4}, {4,5}, {2,3}, {3,8}
10	21 1 3		{1,2,3}, {1,2,5}, {2,6}, {7,8}, {3,4}, {4,5}, {3,8}	Not available
11	21 2 4		{1,2,3}, {1,2,5}, {2,3,4}, {2,4,5}, {2,6}, {7,8}, {3,8}	Not available
12	21 6 8		{1,2,3}, {1,2,5}, {2,3,4}, {2,4,5}, {7,8}, {2,6}, {6,8}, {3,8}	{1,2,3}, {1,2,5}, {2,3,4}, {2,4,5}, {7,8}, {2,6}, {6,8}, {3,8}
13	22 3 5		{1,2,3,5}, {2,3,4,5}, {7,8}, {2,6}, {6,8}	{1,2,3,5}, {2,3,4,5}, {7,8}, {2,6}, {6,8}, {3,8}
14	23 2 7		{1,2,3,5}, {2,3,4,5}, {2,6}, {6,8}, {2,7}, {7,8}, {3,8}	Not available

Figure 7.11: Cluster sets for levels 1 to 14 of the ethnic marriages data set.

Hierarchical Level	Ordered Triple	Standard Complete Linkage Method	INCLude Hierarchical Clustering	Clique Detection
15	23 2 8		{1,2,3,5}, {2,3,4,5}, {2,6,8}, {2,7,8}	{1,2,3,5}, {2,3,4,5}, {2,6,8}, {2,7,8}, {2,3,8}
16	24 3 6	Path IIA: {1,2,5}, {3,6,8}, 4, 7	{1,2,3,5}, {2,3,4,5}, {2,3,6,8}, {2,7,8}	Not available
17	24 3 7	Path IIB: {1,2,5}, {3,7,8}, 4, 6 Path IIC: {1,2,5}, {3,6,7,8}, 4	{1,2,3,5}, {2,3,4,5}, {2,3,6,8}, {2,3,7,8}	Not available
18	24 6 7	Path I: {1,2,5}, {6,7,8}, {3,4} Path IIA: {1,2,5}, {3,6,7,8}, 4 Path IIB: {1,2,5}, {3,6,7,8}, 4 Path IIC: {1,2,5}, {3,8}, {6,7}, 4 Path III: {1,2,5}, {3,4}, {6,7,8}	{2,3,6,7,8}, {1,2,3,5}, {2,3,4,5}	{2,3,6,7,8}, {1,2,3,5}, {2,3,4,5}
19	27 1 6		{2,3,6,7,8}, {2,3,4,5}, {1,2,3,5}, {1,2,3,6}	{2,3,6,7,8}, {2,3,4,5}, {1,2,3,5}, {1,2,3,6}
20	28 5 8		{2,3,6,7,8}, {2,3,4,5}, {1,2,3,5}, {1,2,3,6}	{2,3,6,7,8}, {2,3,4,5}, {1,2,3,5}, {1,2,3,6}, {2,3,5,8}
21	30 1 4	Path I: {1,2,3,4,5}, {6,7,8} Path II: {1,2,4,5}, {3,6,7,8} Path III: {1,2,3,4,5}, {6,7,8}	{1,2,3,4,5}, {2,3,6,7,8}	Not available
22	30 5 6		{1,2,3,4,5}, {2,3,6,7,8}	Not available
23	30 5 7		{2,3,5,6,7,8}, {1,2,3,4,5}	{2,3,5,6,7,8}, {1,2,3,4,5}, {1,2,3,5,6}
24	32 1 8		{2,3,5,6,7,8}, {1,2,3,4,5}	{1,2,3,5,6,8}, {2,3,5,6,7,8}, {1,2,3,4,5}
25	33 1 7		{1,2,3,5,6,7,8}, {1,2,3,4,5}	{1,2,3,5,6,7,8}, {1,2,3,4,5}
26	37 4 6		{1,2,3,5,6,7,8}, {1,2,3,4,5,6}	Not available
27	37 4 8		{1,2,3,4,5,6,8}, {1,2,3,5,6,7,8}	{1,2,3,4,5,6,8}, {1,2,3,5,6,7,8}
28	41 4 7	{1,2,3,4,5,6,7,8}	{1,2,3,4,5,6,7,8}	{1,2,3,4,5,6,7,8}

Figure 7.12: Cluster sets for levels 15 to 28 of the ethnic marriages data set.

7.2.5 Residential Heat Pump Data Set

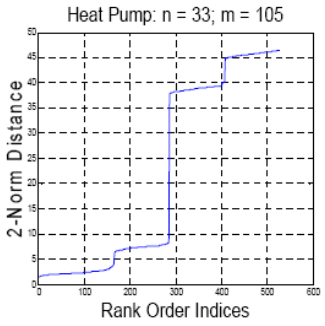
This experiment examined data sets that have data points comprised of multiple attributes. Three data sets were provided by the U.S. National Institute of Standards and Technology (NIST). The data sets were collected for a study described in [38]. There, they were used to analyze the performance of a residential heat pump that was operating in the cooling mode when a single external fault was imposed. The data sets are comprised of numerous kinds of measurements that were collected at approximately 12 second intervals for at least 17 minutes. While the second and third data sets were collected, the indoor air side flow rate was changed from 1000 scfm (std. cubic ft. per min.) to 500 scfm and from 1000 scfm to 1200 scfm, respectively.

Using no-fault, third-order polynomial correlations to calculate residuals, readings for the most informative seven kinds of measurements related to air flow were excerpted from each data set. Then, time synchronized sequences of readings representing each kind of measurement, which sequences included 15 consecutive time points, were concatenated to construct data points. In all, 11 data points having 105 dimensions (7 measurements x 15 time points) were constructed from each data set, or 33 data points in total. Euclidean distance and city block distance were used to calculate the distances between the data points, the sets of distances were graphed, and the new clustering method was used to cluster the data points.

As the top chart in Fig. 7.13 shows, the standard deviations of the measurements are relatively small. Assuming that the means and the standard deviations of the measurements are valid across all time points, the standard deviations of the interclass and intraclass distances for Euclidean distance are $\sigma_{Z,(1000,500)} = 0.14$, $\sigma_{Z,(1000,1200)} = 0.14$, $\sigma_{Z,(500,1200)} = 0.14$, $\sigma_{Z,(1000,1000)} = 0.18$, $\sigma_{Z,(500,500)} = 0.25$, and $\sigma_{Z,(1200,1200)} = 0.17$. Consequently, inherent structure emerges as early as $m = 105$ dimensions. The graphs in Fig. 7.13 suggest that the corresponding hierarchical sequences have five meaningful levels. Even though the data points can migrate, each of the data points continues to combine with other data points having the same type of fault. Recursion was not needed to construct the cluster sets. The fault pattern for the 500 scfm data appears at $level = 407$ ($d' = 40.10$ for Euclidean distance and $d' = 291.62$ for city block distance) while that for the 1200 scfm data appears at $level = 286$ ($d' = 8.28$ for Euclidean distance and $d' = 69.34$ for city block distance).

Measurement	1000 scfm (Normal) Mean	Std. Dev.	500 scfm Mean	Std. Dev.	1200 scfm Mean	Std. Dev.
Thermopile Indoor Air Temperature Change (F)	18.19	0.079	25.28	0.082	16.80	0.066
Compressor Discharge Line Wall Temperature (F)	157.63	0.153	157.67	0.213	157.78	0.153
Indoor Vapor Temperature at Saturation (F)	50.15	0.125	43.57	0.113	51.18	0.129
Indoor Coil Liquid Subcooling (F)	5.62	0.159	5.25	0.120	5.62	0.173
Outdoor Liquid Line Subcooling (F)	8.39	0.155	7.33	0.138	8.47	0.164
Outdoor Inlet Refrigerator Vapor Temperature at Saturation (F)	102.92	0.103	100.75	0.086	103.28	0.101
Compressor Suction Superheat (F)	20.91	0.233	20.32	0.311	21.16	0.202

n = 33; m = 105		
Hierarchical Level	Threshold Distance (Euclidean)	Cluster Set
0	0.00	1, 2, ..., 33
165	4.06	{1-11}, {12-22}, {23-33}
286	8.28	{1-11,23-33}, {12-22}
407	40.10	{1-11,23-33}, {1-11,12-22}
528	46.42	{1-33}



n = 33; m = 105		
Hierarchical Level	Threshold Distance (City Block)	Cluster Set
0	0.00	1, 2, ..., 33
165	29.08	{1-11}, {12-22}, {23-33}
286	69.34	{1-11,23-33}, {12-22}
407	291.62	{1-11,23-33}, {1-11,12-22}
528	332.85	{1-33}

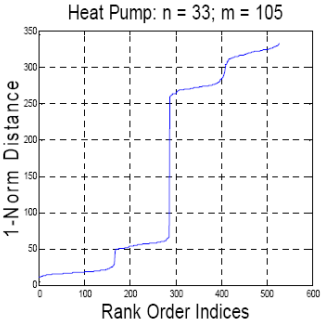


Figure 7.13: Means and standard deviations for the seven kinds of measurements that were excerpted from the NIST data sets, distance graphs for the 33-point data set, and meaningful cluster sets.

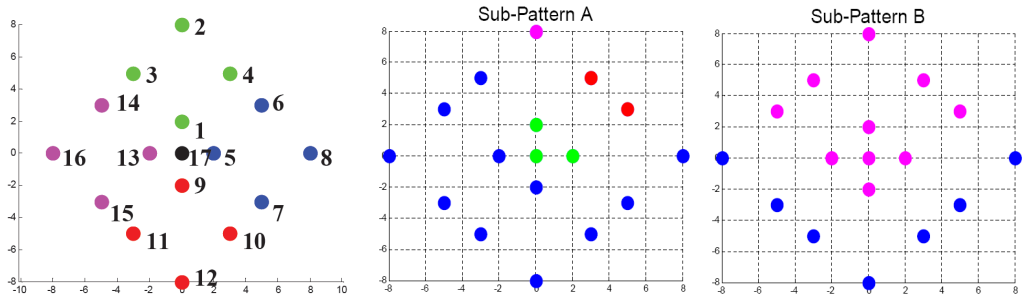


Figure 7.14: 17-points geometric pattern (left), sub-pattern A (center), and sub-pattern B (right).

7.2.6 17-Points Geometric Pattern Data Sets

A geometric pattern comprised of 17 points was constructed, as shown in Fig. 7.14. With a modest sized pattern, it is possible to observe which sub-patterns emerge as threshold distance d' increases and use these observations as a benchmark. A noiseless data set was constructed from five copies of each point (85 data points in total). Euclidean distance and city block distance were used to calculate the distances between the data points, the sets of distances were graphed, and the new clustering method and the standard complete linkage method were used to cluster the data points. Next, by resampling the points, the dimensionality of the data points was increased to 20,000 dimensions by increments of a magnitude, and noise ($N(0, 2^2)$) was added to each dimension of each data point. Euclidean distance and city block distance were used to calculate the distances between the data points in these data sets, the sets of distances were graphed, and the new clustering method was used to cluster the data set where m = 20,000 dimensions. The distance graphs for the noiseless and the noisy data sets are provided in Fig. 7.15. The meaningful levels of the hierarchical sequences are provided in Fig. 7.16, and the cluster sets (sub-patterns) for the noiseless data set are provided in Figs. 7.17 to 7.19.

As the data in Figs. 7.15 and 7.16 show, the distance that was used affected the numbers of meaningful levels and which levels were meaningful. Comparing the hierarchical sequences, however, shows that many of the meaningful levels are the same. Where they are, with one exception, the cluster sets also are the same. *See, e.g.*, Figs.

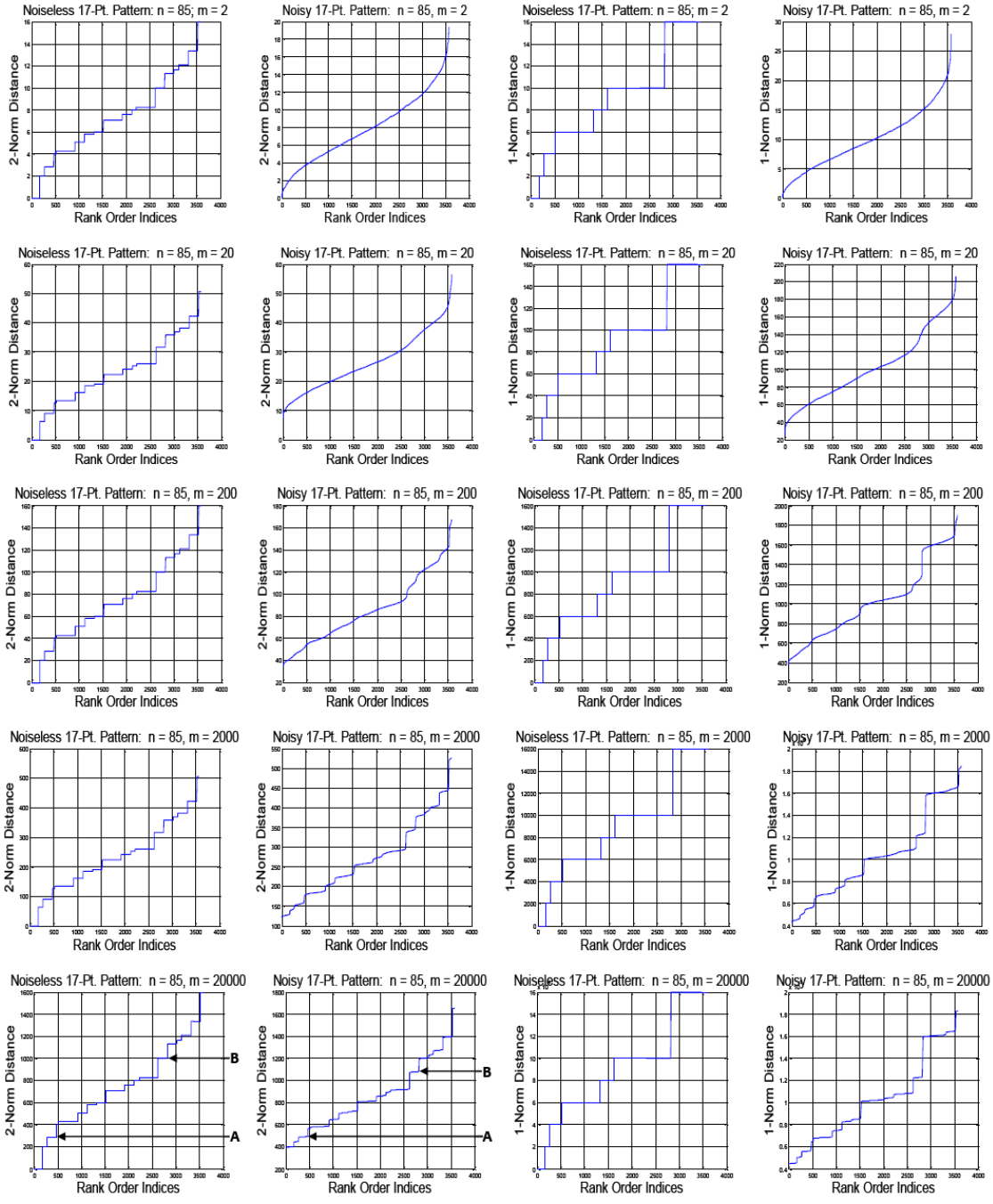


Figure 7.15: Distance graphs for the noiseless and the noisy 17-points geometric pattern data sets. The locations of sub-patterns A ($level = 470$ for Euclidean distance) and B ($level = 2820$ for Euclidean distance) are indicated by the arrows.

7.17 to 7.19 for the noiseless data set. For $level = 1320$, three different cluster sets were constructed (the cluster sets for city block distance are the same). For the noisy data set, the angle for this level is 58.23 degrees for Euclidean distance. For the other meaningful levels for both the noiseless and the noisy data sets, all the angles but one are greater than 80 degrees (for the noisy data set, the angle for $level = 2020$ is 63.04 degrees for city block distance). So, there is a significant degree of commonality. *Cf.* [39]. It appears that noise can add structure to a data set. While the meaningful levels for the noiseless and the noisy data sets are the same for Euclidean distance, they are not for city block distance. Still, they are remarkably similar. See also the synthetic gene expression data sets, *infra*.

The standard complete linkage method did not construct cluster sets for several meaningful levels of the hierarchical sequences, and several of the cluster sets that were constructed are inaccurate. For example, with respect to sub-pattern A in Fig. 7.14, the Rand Index score for the new clustering method is 1.0 while that for the standard complete linkage method is 0.84. For sub-pattern B in Fig. 7.14, the Rand Index score for the new clustering method is 1.0 while that for the standard complete linkage method is 0.48. The differences between these scores are due primarily to clusters that overlap. For $level = 1920$ for Euclidean distance, the differences also are due to the migration problem. Flat clustering methods such as k-means clustering likewise have difficulty constructing these cluster sets.

If the number of data points in the noiseless data set is reduced from 85 to 17 (one data point for each point in the pattern), the hierarchical sequence for Euclidean distance has 18 meaningful levels instead of 19. Since this number is greater than the number of data points, every meaningful level could not be included in an n-level hierarchical sequence. Moreover, the *post hoc* heuristics for cutting dendrograms are designed to find one "optimal" or maybe a few cluster sets. *See, e.g.,* [28]. Finding many more levels than this would be an enormous task.

Next, data sets where $m = 20{,}000$ to 80,000 dimensions were constructed to examine how many false positives and how many false negatives are incurred when the test for finding deemed meaningful levels is used. Most false positives were levels just to either side of the meaningful levels. As the data in Fig. 7.20 show, more false positives occurred at lower dimensionalities, due to poor definition, and at lower $cutoffAngle$s,

Noiseless Data n = 85;	m = 2; σ = 0	Noisy Data n = 85;	m = 20,000; σ = 2	Noiseless Data n = 85;	m = 2; σ = 0	Noisy Data n = 85;	m = 20,000; σ = 2
Hierarchical Level	Threshold Distance (Euclidean)	Hierarchical Level	Threshold Distance (Euclidean)	Hierarchical Level	Threshold Distance (City Block)	Hierarchical Level	Threshold Distance (City Block)
0	0.00	0	0.00	0	0	0	0
170	0.00	170	404.20	170	0	170	45,570
270	2.00	270	453.47	270	2	270	51,160
470	2.83	470	496.93			470	56,791
520	4.00	520	571.45	520	4	520	65,340
920	4.24	920	589.55			920	69,527
1120	5.10	1120	655.35			1120	75,530
1320	5.83	1320	712.25	1320	6	1320	83,833
1520	6.00	1520	726.98			1520	85,967
				1620	8		
1920	7.07	1920	819.82				
						2020	103,264
2120	7.62	2120	868.47				
2220	8.00	2220	898.97			2220	105,220
2620	8.25	2620	923.21			2620	109,040
2820	10.00	2820	1082.56	2820	10	2820	123,386
3020	11.31	3020	1207.94				
3120	11.66	3120	1239.77				
3320	12.08	3320	1279.14			3320	161,848
3520	13.34	3520	1398.38			3520	165,090
3570	16.00	3570	1653.00	3570	16	3570	183,318

Figure 7.16: Meaningful levels for the noiseless and the noisy 17-points geometric pattern data sets.

Noiseless Data n = 85; m = 2					
Hierarchical Level	Threshold Distance (Euclidean)	Recursive Calls Number/Level	INCLude Hierarchical Clustering	Standard Complete Linkage Method	Rand Index Score Std. Complete Linkage Method
0	0.00	0	1, 2, ..., 17	1, 2, ..., 17	1.00
170	0.00	0	1, 2, ..., 17	1, 2, ..., 17	1.00
270	2.00	0	{1,17}, {5,17}, {9,17}, {13,17}, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16	{13,17}, 1, 2, ..., 12, 14, 15, 16, 17	0.63
A 470	2.83	5/1 4/2	{1,5,17}, {1,13,17}, {5,9,17}, {9,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, 2, 8, 12, 16	{1,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, {5,9}, 2, 8, 12, 16	0.84
520	4.00	0	{1,5,9,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, 2, 8, 12, 16	{1,5,9,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, 2, 8, 12, 16	1.00
920	4.24	12/1	{1,5,9,13,17}, {2,3}, {2,4}, {6,8}, {7,8}, {10,12}, {11,12}, {14,16}, {15,16}, {1,3}, {3,14}, {1,4}, {4,6}, {5,6}, {5,7}, {7,10}, {9,10}, {9,11}, {11,15}, {13,14}, {13,15}		
1120	5.10	12/1	{1,5,9,13,17}, {1,3,13,14}, {1,4,5,6}, {5,7,9,10}, {9,11,13,15}, {2,3}, {2,4}, {6,8}, {7,8}, {10,12}, {11,12}, {14,16}, {15,16}		
1320	5.83	17/1	{1,5,9,13,17}, {1,3,13,14,17}, {1,4,5,6,17}, {5,7,9,10,17}, {9,11,13,15,17}, {2,3}, {2,4}, {6,8}, {7,8}, {10,12}, {11,12}, {14,16}, {15,16}		
1520	6.00	1/1 12/2	{1,5,9,13,17}, {1,3,13,14,17}, {1,4,5,6,17}, {5,7,9,10,17}, {9,11,13,15,17}, {1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}, {1,3,4,17}, {5,6,7,17}, {9,10,11,17}, {13,14,15,17}		
1920	7.07	17/1 32/2 24/3	{1,3,4,5,13,17}, {1,5,6,7,9,17}, {5,9,10,11,13,17}, {1,9,13,14,15,17}, {1,3,13,14,17}, {1,4,5,6,17},{5,7,9,10,17}, {9,11,13,15,17}, {1,2,3,4}, {1,2,4,6}, {1,2,3,14}, {4,5,6,8}, {5,7,8,10}, {5,6,7,8}, {7,9,10,12}, {9,11,12,15}, {3,13,14,16}, {9,10,11,12}, {13,14,15,16}, {11,13,15,16}	{1,5,9,13,17}, {11,15,16}, {7,10,12}, {2,3,14}, {4,6,8}	0.34
2120	7.62	17/1 80/2 96/3	{1,3,4,5,9,13,17}, {1,5,6,7,9,13,17}, {1,5,9,10,11,13,17}, {1,5,9,13,14,15,17}, {1,3,5,9,13,14,17}, {1,4,5,6,9,13,17}, {1,5,7,9,10,13,17}, {1,5,9,11,13,15,17}, {1,2,3,4}, {1,2,4,6}, {1,2,3,14}, {4,5,6,8}, {5,7,8,10}, {5,6,7,8}, {7,9,10,12}, {9,11,12,15}, {3,13,14,16}, {9,10,11,12}, {13,14,15,16}, {11,13,15,16}		

Figure 7.17: Cluster sets for the noiseless 17-points geometric pattern data set. Euclidean distance was used to calculate the distances. Only results for the first 17 data points are provided, since they are representative of the entire data set. The letters A and B denote sub-patterns A and B, respectively.

Noiseless Data n = 85; m = 2					
Hierarchical Level	Threshold Distance (Euclidean)	Recursive Calls Number/Level	INCLude Hierarchical Clustering	Standard Complete Linkage Method	Rand Index Score Std. Complete Linkage Method
2220	8.00	16/1 72/2 96/3	{1,3,4,5,9,13,17}, {1,5,6,7,9,13,17}, {1,5,9,10,11,13,17}, {1,5,9,13,14,15,17}, {1,3,5,9,13,14,17}, {1,4,5,6,9,13,17}, {1,5,7,9,10,13,17}, {1,5,9,11,13,15,17}, {1,2,3,4,17}, {1,2,4,6,17}, {1,2,3,14,17}, {4,5,6,8,17}, {5,6,7,8,17}, {5,7,8,10,17}, {7,9,10,12,17}, {9,10,11,12,17}, {9,11,12,15,17}, {13,14,15,16,17}, {3,13,14,16,17}, {11,13,15,16,17}		
2620	8.25	16/1 88/2 288/3 576/4 640/5 256/6	{1,3-5,9,13,14,17}, {1,3-6,9,13,17}, {1,3,5,9,13-15,17}, {1,4-7,9,13,17}, {1,4-9,17}, {1,5-7,9,10,13,17}, {1,5,7,9-11,13,17}, {1,5,9-11,13,15,17}, {1-6,13,17}, {1-5,13,14,17}, {1,5-10,17}, {5,7,9-13,17}, {1,5,,9,11,13-15,17}, {5,9-13,15,17}, {1,3,9,13-17}, {1,9,11,13-17}		
B 2820	10.00	0	{1-6,9,13,14,17}, {1,4-10,13,17}, {1,5,7,9-13,15,17}, {1,3,5,9,11,13-17}	{1,5,9,11,13,15,16,17}, {7,10,12}, {2,3,14}, {4,6,8}	0.48
3020	11.31	12/1 64/2 144/3	{1,2,4-6,8,9,13,17}, {1-3,5,9,13,14,16,17}, {1-6,9,13,14,17}, {1,3-7,9,13,17}, {1,3-5,7,9,13,15,17}, {1,3,5-7,9,11,13,17}, {1,3,5,7,9,11,13,15,17}, {1,3,5,9,11,13-17}, {1,3-5,9,13-15,17}, {1,3,5,6,9,11,13,14,17}, {1,4-10,13,17}, {1,4,5,7,9,10,13,15,17}, {1,4,5,9,10,13-15,17}, {1,4-6,9,10,13,14,17}, {1,5-7,9-11,13,17}, {1,5,6,9-11,13,14,17}, {1,5,7,9-13,15,17}, {1,5,7-10,12,13,17}, {1,5,9-11,13-15,17}, {1,5,9,11-13,15-17}		
3120	11.66	8/1 8/2	{1,2,4-6,8,9,13,17}, {1-3,5,9,13,14,16,17}, {1,5,7-10,12,13,17}, {1,5,9,11-13,15-17}, {1-6,9,13,14,17}, {1,3-7,9-11,13-15,17}, {1,3,5,9,11,13-17}, {1,4-10,13,17}, {1,5,7,9-13,15,17}		
3320	12.08	8/1 24/2 32/33	{1-9,13,17}, {1-5,9,13-17}, {1,3-11,13,17}, {1,5-13,17}, {1,5,9-17}, {1,3-5,9-11,13-17}, {1-7,9,13-15,17}, {1,3-7,9-11,13-15,17}, {1,5-7,9-15,17}		
3520	13.34	4/1	{1-11,13-15,17}, {1-7,9-11,13-17}, {1,3-15,17}, {1,3-7,9-17}	{1,5,7,9-13,15,16,17}, {2,3,4,6,8,14}	0.25
3570	16.00	0	{1-17}	{1-17}	1.00

Figure 7.18: Continuation of Fig. 7.17

Noiseless Data n = 85; m = 2					
Hierarchical Level	Threshold Distance (City Block)	Recursive Calls Number/Level	INCLude Hierarchical Clustering	Standard Complete Linkage Method	Rand Index Score Std. Complete Linkage Method
0	0	0	1, 2, ..., 17	1, 2, ..., 17	1.00
170	0	0	1, 2, ..., 17	1, 2, ..., 17	1.00
270	2	0	{1,17}, {5,17}, {9,17}, {13,17}, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16	{13,17}, 1, 2, ..., 12, 14, 15, 16, 17	0.63
520	4	0	{1,5,9,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, 2, 8, 12, 16	{1,5,9,13,17}, {3,14}, {4,6}, {7,10}, {11,15}, 2, 8, 12, 16	1.00
1320	6	0	{1,5,9,13,17}, {1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}		
1620	8	0	{1,2,3,4,17}, {5,6,7,8,17}, {9,10,11,12,17}, {13,14,15,16,17}		
2820	10	0	{1-6,9,13,14,17}, {1,4-10,13,17}, {1,5,7,9-13,15,17}, {1,3,5,9,11,13-17}	{1,3,5,9,11,13-17}, {4,6,7,8,10}, 2, 12	0.58
3570	16	0	{1-17}	{1-17}	1.00

Figure 7.19: Cluster sets for the noiseless 17-points geometric pattern data set. City block distance was used to calculate the distances. Only results for the first 17 data points are provided, since they are representative of the entire data set.

because the criterion for constructing cluster sets was less stringent. If they occurred, false negatives tended to occur at lower dimensionalities, due to poor definition, and at very high $cutoffAngle$s, because the criterion for constructing cluster sets was too stringent. From 70K($\angle$75) to 80K($\angle$85), there were no false positives or false negatives. These dimensionalities are higher than what is needed to visually examine a distance graph.

Dimensions m	60 degrees	65 degrees	70 degrees	75 degrees	80 degrees	85 degrees	60 degrees	65 degrees	70 degrees	75 degrees	80 degrees	85 degrees
	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance
20,000	32+/0-	21+/0-	10+/0-	4+/1-	0+/1-	0+/2-	28+/1-	18+/1-	11+/1-	6+/1-	3+/2-	0+/2-
30,000	13+/1-	7+/1-	5+/1-	2+/1-	0+/1-	0+/1-	14+/0-	6+/0-	2+/0-	1+/0-	0+/0-	0+/1-
40.,000	16+/0-	10+/0-	4+/0-	0+/0-	0+/0-	0+/1-	16+/0-	13+/0-	10+/0-	4+/0-	1+/0-	0+/1-
50,000	17+/0-	11+/0-	4+/0-	1+/0-	1+/0-	0+/1-	16+/0-	11+/0-	7+/0-	3+/0-	1+/0-	0+/1-
60,000	12+/0-	8+/0-	3+/0-	1+/0-	0+/0-	0+/1-	12+/0-	10+/0-	6+/0-	1+/0-	0+/0-	0+/1-
70,000	6+/0-	2+/0-	0+/0-	0+/0-	0+/0-	0+/0-	9+/0-	4+/0-	1+/0-	0+/0-	0+/0-	0+/0-
80.000	7+/0-	2+/0-	1+/0-	0+/0-	0+/0-	0+/0-	12+/0-	6+/0-	3+/0-	0+/0-	0+/0-	0+/0-

Figure 7.20: Number of deemed meaningful levels that were false positives or false negatives for the 17-points geometric pattern data sets where m = 20,000 to 80,000 dimensions.

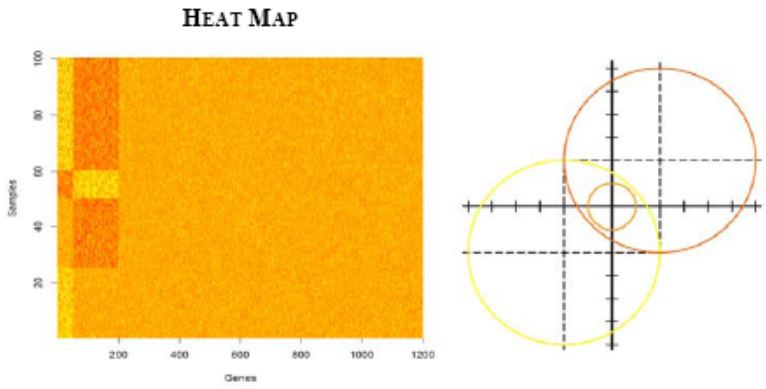


Figure 7.21: Heat map for the synthetic gene expression data sets.

7.2.7 Synthetic Gene Expression Data Sets

The heat map in Fig. 7.21 was provided by the Hollings Cancer Center at the Medical University of South Carolina. The data sets constructed from this heat map include three gene classes and four sample classes. The ratio for the gene classes is 50:150:1000 while the ratio for the sample classes is 40:10:25:25. The signal-to-noise ratio for the gene classes is 1.29/1.87, where noise is defined as the pooled estimate of the standard deviations for over ($N(2, 4^2)$, mostly in red-orange (dark gray)), under ($N(-2, 4^2)$, mostly in yellow (light gray)), and normally ($N(0, 1^2)$, mostly in orange (medium gray)) expressed genes.

The means of the three gene classes were used to construct a noiseless data set. Euclidean distance and city block distance were used to calculate the distances between the data points, the sets of distances were graphed, and the new clustering method, the standard complete linkage method, and k-means clustering were used to cluster the data points. As the distance graphs in the top row of Fig. 7.22 show, inherent structure emerges immediately for noiseless data. For the noisy data sets, inherent structure emerges as early as m = 5000 dimensions. The standard deviations of the interclass and intraclass distances for Euclidean distance are $\sigma_{Z,(over,under)} = 4.52$, $\sigma_{Z,(over,normal)} = 3.31$, $\sigma_{Z,(under,normal)} = 3.31$, $\sigma_{Z,(over,over)} = 3.96$, $\sigma_{Z,(under,under)} = 3.96$, and $\sigma_{Z,(normal,normal)} = 1.00$. The graphs in the bottom row of Fig. 7.22 suggest that the corresponding hierarchical sequences have five meaningful levels: $level = 0$ (d'

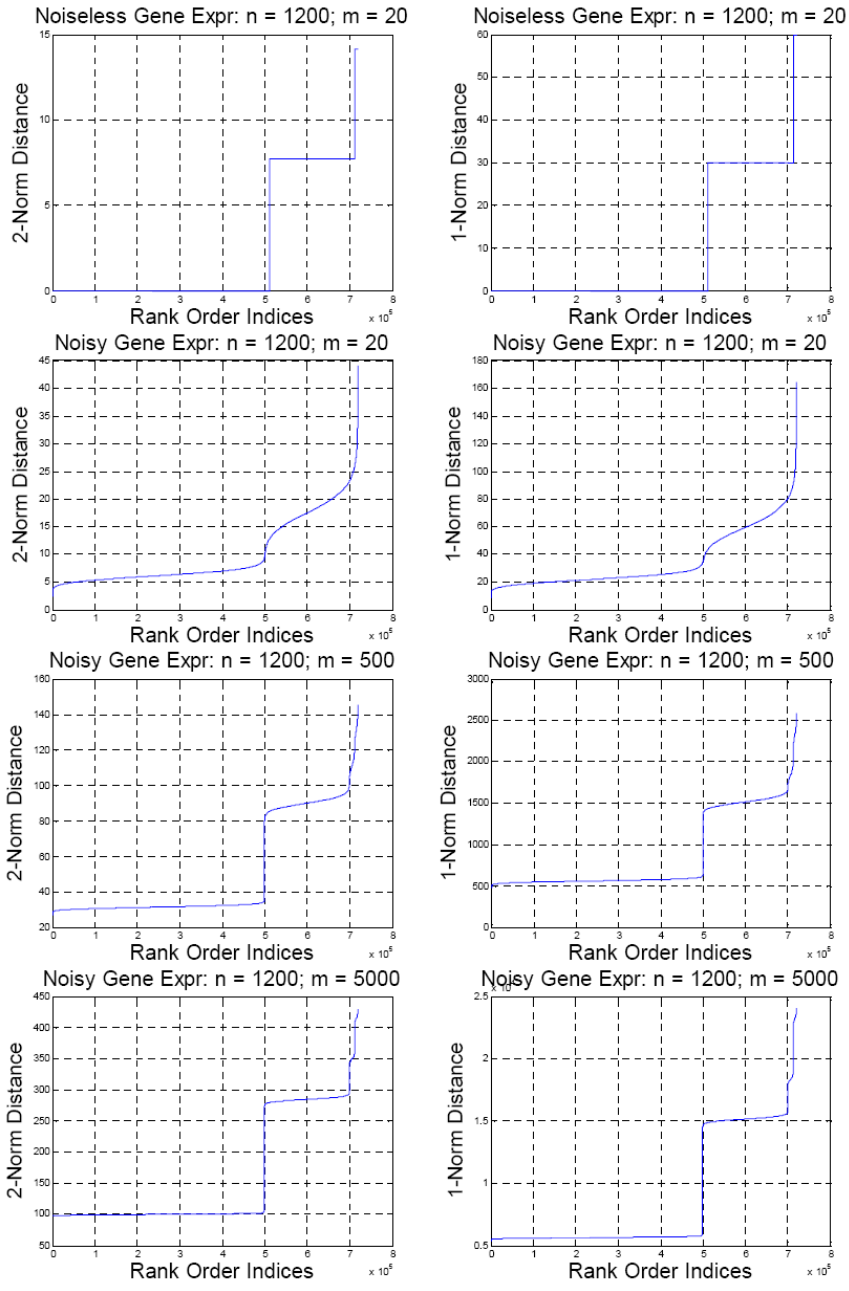


Figure 7.22: Distance graphs for the noiseless and three noisy synthetic gene expression data sets.

Noiseless Data n = 1200; m = 20		
Hierarchical Level	**Threshold Distance (Euclidean)**	**Cluster Set**
0	0.00	1, 2, …, 1200
511,900	0.00	{1-50}, {51-200}, {201-1200}
711,900	7.75	{1-50,201-1200}, {51-200,201-1200}
719,400	14.14	{1-1200}

Noiseless Data n = 1200; m = 20		
Hierarchical Level	**Threshold Distance (City Block)**	**Cluster Set**
0	0.00	1, 2, …, 1200
511,900	0.00	{1-50}, {51-200}, {201-1200}
711,900	30.00	{1-50,201-1200}, {51-200,201-1200}
719,400	60.00	{1-1200}

Noisy Data n = 1200; m = 5000		
Hierarchical Level	**Threshold Distance (Euclidean)**	**Cluster Set**
0	0.00	1, 2, …, 1200
499,500	105.28	1, 2, …, 200, {201-1200}
699,500	297.65	{1,201-1200}, {2,201-1200}, …, {200,201-1200}
711,900	365.58	{1-50,201-1200}, {51-200,201-1200}
719,400	429.81	{1-1200}

Noisy Data n = 1200; m = 5000		
Hierarchical Level	**Threshold Distance (City Block)**	**Cluster Set**
0	0.00	1, 2, …, 1200
499,500	5928.88	1, 2, …, 200, {201-1200}
699,500	15,855.38	{1,201-1200}, {2,201-1200}, …, {200,201-1200}
711,900	19,148.27	{1-50,201-1200}, {51-200,201-1200}
719,400	24,027.44	{1-1200}

Figure 7.23: Meaningful cluster sets for the noiseless and the noisy synthetic gene expression data sets.

= 0.00 for Euclidean distance and $d' = 0.00$ for city block distance), $level$ = 499,500 ($d' = 105.28$ for Euclidean distance and $d' = 5928.88$ for city block distance), $level$ = 699,500 ($d' = 297.65$ for Euclidean distance and $d' = 15{,}855.38$ for city block distance), $level$ = 711,900 ($d' = 365.58$ for Euclidean distance and $d' = 19{,}148.27$ for city block distance), and $level$ = 719,400 ($d' = 429.81$ for Euclidean distance and $d' = 24{,}027.44$ for city block distance). The cluster sets for these levels are constructible without constructing any of the other 719,396 cluster sets (which also is 1195 fewer cluster sets than an n-level hierarchical sequence), and recursion was not needed to construct the cluster sets. The tables in Fig. 7.23 show that noise attenuation is not the same as noise elimination. Nonetheless, the gene classes are discernible by examining the meaningful cluster sets.

This structure is not apparent when k-means clustering or the standard complete linkage method is used. For Euclidean distance and $k = 3$ clusters, k-means clustering correctly grouped the noisy data set ($m = 5000$) into clusters in three of five trials. The average number of errors was 22.4, and the average Rand Index score is 0.98. For city block distance and $k = 3$ clusters, k-means clustering correctly grouped the noisy data set into clusters in two of five trials. The average number of errors was 125.2, and the average Rand Index score is 0.86. For the standard complete linkage method, the Rand Index scores are 1.00, 0.0010, and 0.77 for $level$ = 499,500, $level$ = 699,500, and $level$ = 711,900, respectively.

To compare how much CPU time is used by the new clustering method and k-means clustering, a data set comprised of $n = 1200$ genes and $m = 10{,}000$ dimensions was constructed. Setting k equal to 2, 3, 4, and then 5 clusters, k-means clustering used, on average for 5 trials running on a Pentium 4 processor, 0.14 s., 312.60 s., 321.72 s., and 1092.82 s., respectively, to construct a cluster set. In comparison, the new clustering method used a maximum of 8.17 s. to load the data set, 1391.66 s. to calculate the distances between the data points and construct ordered triples, 267.83 s. to sort the ordered triples, 0.44 s. to evaluate the ordered triples for linkage, and 103.16 s. to construct cluster sets for all five meaningful levels. So, the new clustering method used less CPU time in total (1771.26 s.) than k-means clustering used for 2 trials when k equals 5. Since a range of settings has to be evaluated for k-means clustering, because k is unknown *a priori*, and several trials have to be run, because k-means clustering

is only locally optimal, comparatively speaking, using k-means clustering appears to be impractical when the meaningful levels of a hierarchical sequence are conveniently identifiable. Even if the number of clusters k were known *a priori*, k-means clustering is grossly inefficient when the number of clusters is large. For example, the cluster set for $level = 699{,}500$ has 200 clusters. k-means clustering used more than 2:20.00 hours of CPU time to construct this cluster set, i.e., to run one trial.

Next, data sets where $m = 5000$ to 25,000 dimensions were constructed to examine how many false positives and how many false negatives are incurred when the test for finding deemed meaningful levels is used. Most false positives were levels just to either side of the meaningful levels. As the data in Fig. 7.24 show, more false positives occurred at lower dimensionalities, due to poor definition, and at lower $cutoffAngle$s, because the criterion for constructing cluster sets was less stringent. There were no false negatives for this data set. Nor were there any dimensionalities at which there were no false positives or false negatives. The numbers of false positives are 0.013 to 0.54 percent of the total number of levels.

Dimensions m	80.00 degrees Euclid. Distance	82.50 degrees Euclid. Distance	85.00 degrees Euclid. Distance	87.50 degrees Euclid. Distance	88.75 degrees Euclid. Distance	80.00 degrees City B. Distance	82.50 degrees City B. Distance	85.00 degrees City B. Distance	87.50 degrees City B. Distance	88.75 degrees City B. Distance
5,000	3739+/0-	2582+/0-	1493+/0-	594+/0-	259+/0-	3897+/0-	2725+/0-	1562+/0-	605+/0-	259+/0-
10,000	2590+/0-	1771+/0-	1046+/0-	443+/0-	188+/0-	2494+/0-	1685+/0-	1014+/0-	431+/0-	203+/0-
15.,000	1979+/0-	1316+/0-	746+/0-	326+/0-	143+/0-	2040+/0-	1342+/0-	811+/0-	340+/0-	165+/0-
20,000	1489+/0-	1014+/0-	626+/0-	245+/0-	97+/0-	1488+/0-	1023+/0-	622+/0-	238+/0-	110+/0-
25,000	1363+/0-	923+/0-	573+/0-	252+/0-	114+/0-	1350+/0-	910+/0-	579+/0-	267+/0-	127+/0-

Figure 7.24: Number of deemed meaningful levels that were false positives or false negatives for the synthetic gene expression data sets having from 5000 to 25,000 dimensions.

7.2.8 Motes Sensing Luminescence Data Set

This experiment shows that meaningful cluster sets can have real world meaning while other cluster sets generally do not. Nine Crossbow® MicaZ motes with MTS300CA sensor boards attached thereto were configured into a 1x1 meter grid. The motes were programmed to concurrently take light readings (lux) of an overhead light source every 1 second. After calibrating the motes, canopies were placed over some of the motes to vary the time intervals during which these motes were exposed to direct light. Canopies were placed over motes 1, 6, and 8 during the entire experiment, so they were never exposed to direct light (the "full shade" motes); canopies were never placed over motes 2, 4, and 9, so they were always exposed to direct light (the "full sun" motes); and canopies were placed over motes 3, 5, and 7 for 1.5 minutes out of every 3-minute cycle (collectively, the "partial shade" motes). Further, the canopy for mote 3 was deployed at 30 seconds into each 3-minute cycle and removed at 120 seconds, the canopy for mote 5 was deployed at 60 seconds and removed at 150 seconds, and the canopy for mote 7 was deployed at 90 seconds and removed at 180 seconds. Data were collected for 15 minutes or 900 samples per mote (8100 samples in total), out of which 893 samples per mote (8037 samples in total) were usable[2] . Euclidean distance and city block distance were used to calculate the distances between the data points, the sets of distances were graphed, and the new clustering method and the standard complete linkage method were used to cluster the data points.

Typical direct light readings were about 909 lux while typical indirect light readings were about 813 lux. The standard deviations of the readings collected by each mote are all less than 11 lux, so inherent structure emerges as early as $m = 180$ dimensions. Assuming that the means and the standard deviations of the measurements, as provided in Fig. 7.25, are valid across all time points, the worst case scenario standard deviations of the interclass and intraclass distances for Euclidean distance are $\sigma_{Z,(fullsun,fullshade)} = 10.02$, $\sigma_{Z,(fullsun,ptshade)} = 13.37$, $\sigma_{Z,(ptshade,fullshade)} = 12.66$, $\sigma_{Z,(fullsun,fullsun)} = 7.73$, $\sigma_{Z,(fullshade,fullshade)} = 6.40$, and $\sigma_{Z,(ptshade,ptshade)} = 10.97$. Recursion was not needed to construct the cluster sets, and the graphs in Fig. 7.26 suggest that the corresponding hierarchical sequences have four meaningful levels. For $level = 6$ ($d' =$

[2] Seven packets from mote 9 were dropped during transmission.

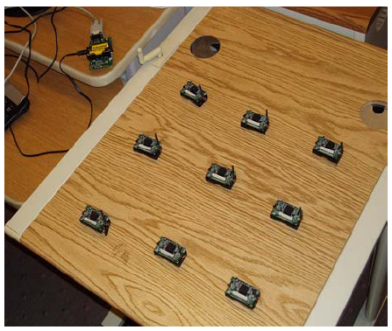
138.41 for Euclidean distance and $d' = 1388.54$ for city block distance), the cluster set includes five non-overlapping clusters, one for the full sun motes, another for the full shade motes, and one for each of the partial shade motes. For $level = 27$ ($d' = 1098.08$ for Euclidean distance and $d' = 12{,}681.00$ for city block distance), the cluster set includes two overlapping clusters, one for those motes that were exposed to direct light during all or part of the experiment (the full sun motes and the partial shade motes) and the other for those motes that were not exposed to direct light during all or part of the experiment (the full shade motes and the partial shade motes).

The results obtained from the standard complete linkage method do not reveal meaningful $level = 27$ for either hierarchical sequence. As the dendrograms in Fig. 7.25 show, while mote 7 combines with the full sun motes, motes 3 and 5 combine with the full shade motes. This disparity among the partial shade motes is difficult to understand without taking into consideration how the standard complete linkage method imposes taxonomic structure on data sets. The Rand Index score is 0.55

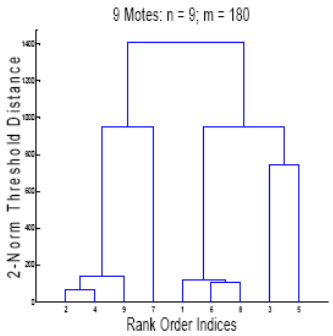
As the tables in Fig. 7.26 illustrate, the cluster sets for the meaningful levels have real world meaning. The cluster sets for the other levels generally do not, and the more so for levels that are not proximate to the meaningful levels. When multiple classes of data points have not finished linking to form a new configuration of clusters, the cluster sets are comprised of overlapping clusters whose differences are not related to inherent structure. These cluster sets are less transparent to domain experts.

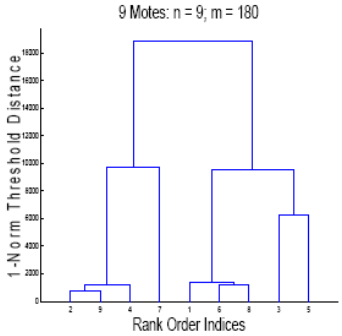
The performance of the new clustering method does not appear to be inhibited by the number of meaningful levels. In the 17-points geometric pattern experiment, described *supra*, as many as 19 meaningful levels were identified. In contrast, the *post hoc* heuristics are designed to find one "optimal" or maybe a few cluster sets. The gap statistic found the cluster set for $level = 6$ but not that for $level = 27$, because the latter cluster set includes overlapping clusters.

As the data in Fig. 7.27 show, when the test for finding deemed meaningful levels was used, the meaningful levels for Euclidean distance were identifiable from 180($\angle$60) to 180($\angle$70). At 180($\angle$65) and 180($\angle$70), the meaningful levels were identifiable without incurring any false positives or false negatives. The meaningful levels for city block distance were identifiable from 180($\angle$60) to 180($\angle$80). At 180($\angle$80), the meaningful levels were identifiable without incurring any false positives or false negatives.



1 Full Shade	2 Full Sun	3 Early Shade
4 Full Sun	5 Medial Shade	6 Full Shade
7 Late Shade	8 Full Shade	9 Full Sun

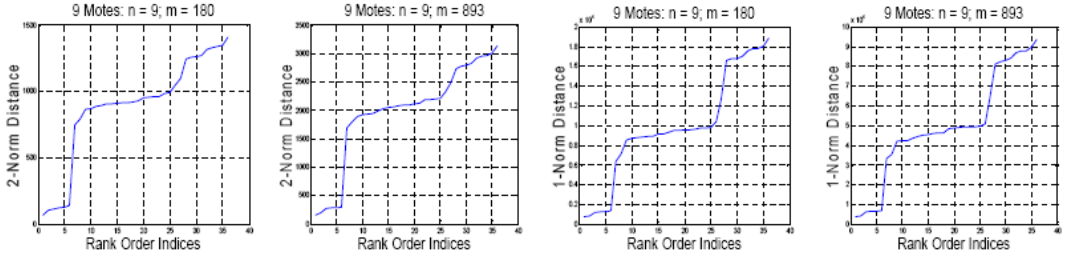




Sensor Node	Sun		Shade	
	Mean	Std. Dev.	Mean	Std. Dev.
1			814.30	4.19
2	907.77	2.34		
3	905.13	6.16	813.09	10.97
4	912.21	3.01		
5	907.81	2.72	812.14	6.89
6			813.19	6.40
7	918.36	2.48	813.45	5.28
8			807.33	3.13
9	906.54	7.73		

Figure 7.25: Configuration, dendrograms, and means and standard deviations for the motes sensing luminescence data set. The motes are classified according to the data sequences that were collected. The different colors (gray scales) represent the different clusters for $level = 6$.

Rank Order Index =	Euclidean Distance Between Sensor Nodes =			City Block Distance Between Sensor Nodes =		
1	65.42	2	4	782.00	2	9
2	103.64	6	8	798.00	2	4
3	109.73	2	9	1201.00	6	8
4	119.36	1	8	1234.60	4	9
5	123.34	1	6	1256.68	1	8
6	138.41	4	9	1388.54	1	6
7	744.69	3	5	6326.20	3	5
8	792.09	5	7	7115.20	5	7
9	861.82	1	3	8535.08	1	3
10	866.74	3	6	8724.80	3	6
11	887.06	1	5	8754.84	1	5
12	892.19	5	6	8863.80	5	6
13	903.30	5	9	8907.00	2	5
14	905.32	3	9	8929.40	5	9
15	911.62	2	7	9142.40	3	9
16	914.92	2	5	9168.40	2	3
17	914.94	7	9	9351.40	3	8
18	916.47	2	3	9507.40	4	5
19	927.69	3	8	9512.40	5	8
20	951.42	4	7	9567.50	1	7
21	953.45	5	8	9586.60	2	7
22	958.88	4	5	9627.40	4	7
23	960.06	3	4	9739.60	7	9
24	980.37	1	7	9747.60	3	4
25	990.93	6	7	9839.40	6	7
26	1044.64	7	8	10399.60	7	8
27	1098.08	3	7	12681.00	3	7
28	1243.18	1	9	16603.60	1	9
29	1256.73	1	2	16803.00	6	9
30	1258.61	6	9	16824.60	1	2
31	1270.49	2	6	17024.00	2	6
32	1316.28	1	4	17622.60	1	4
33	1330.22	4	6	17822.00	4	6
34	1335.90	8	9	17858.00	8	9
35	1348.05	2	8	18079.00	2	8
36	1407.63	4	8	18877.00	4	8



n = 9; m = 180		
Hierarchical Level	**Threshold Distance (Euclidean)**	**Cluster Set**
0	0.00	1, 2, 3, 4, 5, 6, 7, 8, 9*
1	65.42	{2,4}, 1, 3, 5, 6, 7, 8, 9
5	123.34	{2,4}, {2,9}, {1,6,8}, 3, 5, 7
6	138.41	{1,6,8}, {2,4,9}, 3, 5, 7*
7	744.69	{3,5}, {1,6,8}, {2,4,9}, 7
12	892.19	{5,7}, {1,6,8}, {2,4,9}, {1,3,5,6}
18	916.47	{1,6,8}, {2,4,9}, {1,3,5,6}, {2,3,5,9}, {2,5,7,9}
24	980.37	{1,5,7}, {1,3,5,6,8}, {2,3,4,5,9}, {2,4,5,7,9}
26	1044.64	{1,3,5,6,8}, {1,5,6,7,8}, {2,3,4,5,9}, {2,4,5,7,9}
27	1098.08	{1,3,5,6,7,8}, {2,3,4,5,7,9}*
28	1243.18	{1,3,5,6,7,8}, {2,3,4,5,7,9}
30	1258.61	{1,3,5,6,7,8}, {2,3,4,5,7,9}
35	1348.05	{1,2,3,5,6,7,8,9}, {1,2,3,4,5,6,7,9}
36	1407.63	{1,2,3,4,5,6,7,8,9}*

n = 9; m = 180		
Hierarchical Level	**Threshold Distance (City Block)**	**Cluster Set**
0	0.00	1, 2, 3, 4, 5, 6, 7, 8, 9*
1	782.00	{2,9}, 1, 3, 4, 5, 6, 7, 8
5	1256.68	{1,8}, {6,8}, {2,4,9}, 3, 5, 7
6	1388.54	{1,6,8}, {2,4,9}, 3, 5, 7*
7	6326.20	{3,5}, {1,6,8}, {2,4,9}, 7
12	8863.80	{5,7}, {1,6,8}, {2,4,9}, {1,3,5,6}
18	9507.40	{5,7}, {1,3,6,8}, {2,4,5,9}
24	9747.60	{1,5,7}, {2,4,5,7,9}, {1,3,5,6,8}, {2,3,4,5,9}
26	10,399.60	{1,3,5,6,8}, {1,5,6,7,8}, {2,3,4,5,9}, {2,4,5,7,9}
27	12,681.00	{1,3,5,6,7,8}, {2,3,4,5,7,9}*
28	16,603.60	{1,3,5,6,7,8}, {2,3,4,5,7,9}
30	16,824.60	{1,3,5,6,7,8}, {2,3,4,5,7,9}
35	18,079.00	{1,2,3,5,6,7,8,9}, {1,2,3,4,5,6,7,9}
36	18,877.00	{1,2,3,4,5,6,7,8,9}*

Figure 7.26: Proximity vectors, distance graphs, and exemplary cluster sets for the motes sensing luminescence data set. The meaningful cluster sets are indicated by asterisks.

Human Inspect.	60 degrees	65 degrees	70 degrees	75 degrees	80 degrees	85 degrees	60 degrees	65 degrees	70 degrees	75 degrees	80 degrees	85 degrees
	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	Euclid. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance	City B. Distance
Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start	Start
6	6	6	6	6	6	6	6	6	6	6	6	
	8						8	8				
							26	26	26	26		
27	27	27	27				27	27	27	27	27	
36	36	36	36	36	36	36	36	36	36	36	36	36

Figure 7.27: Deemed meaningful levels for the motes sensing luminescence data set.

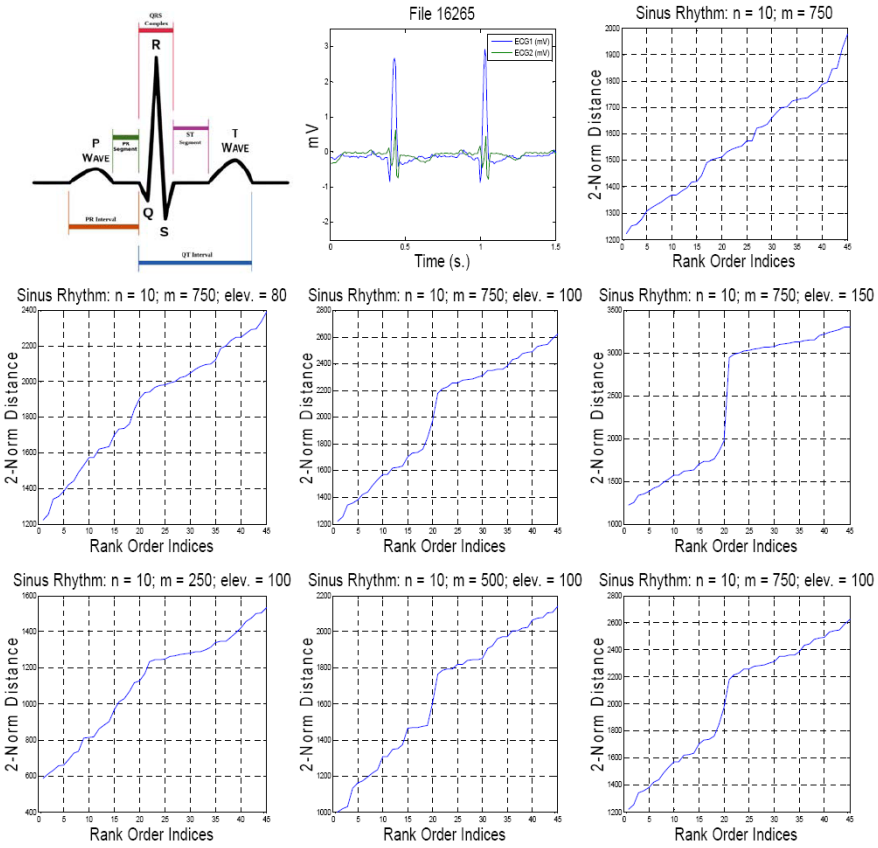


Figure 7.28: ECG and distance graphs for the sinus rhythm data sets.

7.2.9 Sinus Rhythm Data Sets

The data used in this experiment come from file 16265 of the MIT-BIH PhysioNet Normal Sinus Rhythm database [40]. This file contains ECG readings collected at 128 hertz. The P,Q,R,S,T interval of each heart beat, illustrated by the first two graphs in the top row of Fig. 7.28, describes how a heart pumps blood to other parts of a body. Here, 25 samples per beat, which include the Q,R,S complex and at least the left side of the ST element, were excerpted from the first 300 consecutive beats in the file. Then, the data set was divided into ten segments (approx. 25 seconds each). The last graph in the first row of Fig. 7.28 shows that this data set has almost no inherent structure.

An elevating ST element was simulated by adding a constant c_{elevST} between 10 and 150 mV to samples 11-22 of the excerpts that comprised the last five segments.

		1 Segment	2 Segments	3 Segments	4 Segments	5 Segments
m = 250	Euclid. Dist.	None	None	None	None	None
elev. = 80 mV	City B. Dist.	70-80	65-80	70-80	70-80	70-80
m = 250	Euclid. Dist.	None	None	None	None	None
elev. = 100 mV	City B. Dist.	65-85	65-85	65-85	65-85	65-85
m = 250	Euclid. Dist.	60-85	65-85	60-85	60-85	60-85
elev. = 150 mV	City B. Dist.	60-85	60-85	60-85	60-85	60-85
m = 500	Euclid. Dist.	None	65	65	None	None
elev. = 80 mV	City B. Dist.	70-85	60-85	60-85	65-85	50-85
m = 500	Euclid. Dist.	None	70-75	70-80	65-80	70
elev. = 100 mV	City B. Dist.	70-85	55-85	60-85	60-85	50-85
m = 500	Euclid. Dist.	65-85	60-85	55-85	60-85	70-85
elev. = 150 mV	City B. Dist.	65-85	50-85	50-85	55-85	40-85
m = 750	Euclid. Dist.	None	70	60-65	50-65	None
elev. = 80 mV	City B. Dist.	70-85	60-85	65-85	65-85	65-85
m = 750	Euclid. Dist.	70	70-80	60-80	45-80	70
elev. = 100 mV	City B. Dist.	70-85	55-85	60-85	60-85	60-85
m = 750	Euclid. Dist.	65-85	55-85	50-85	40-85	65-85
elev. = 150 mV	City B. Dist.	60-85	45-85	50-85	50-85	50-85

Figure 7.29: Results from clustering the sinus rhythm data sets, showing the ranges over which $cutoffAngle$ could vary without incurring any false positives or false negatives.

Increasing c_{elevST} added structure to the data set. The graphs in the second row show that the elevating ST is detectable as early as 80 mV, when the first five segments and the last five segments group into different clusters. The graphs in the bottom row show how inherent structure emerges as the dimensionality of the segments increases. Increasing the dimensionality of the segments did not add structure to the data set, however, and the law of diminishing returns eventually set in. At these elevations, the damage from ischemia and the risk of sudden death are still low.

To examine how early an elevated ST element is detectable without incurring any false positives or false negatives, an elevated ST element was simulated by adding a constant c_{elevST} equal to 80, 100, or 150 mV to samples 11-22 of the excerpts that comprised the last 1, 2, 3, 4, and then 5 segments. As the data in Fig. 7.29 show, increasing c_{elevST} or the dimensionality of the segments increased the ranges over which $cutoffAngle$ could vary without incurring any false positives or false negatives. Increasing c_{elevST} added structure to the data sets and had the biggest impact on the ranges. Increasing the dimensionality of the segments did not add structure to the data sets, and the law of diminishing returns eventually set in. The widest ranges were where both c_{elevST} and m were large. The number of segments to which c_{elevST} was added did

not show a clear trend. This is consistent with the view that the number of segments should not have an effect on the ranges.

Chapter 8

Conclusion

If today's visions for new cyber-physical systems are to become reality, computer science must develop the software to support these systems. This includes developing automated, intelligent control systems and data analysis that will enable a system to function autonomously under a set of conditions and to adapt to new conditions. The work presented in this thesis is a step towards fulfilling this need.

This thesis presents each part of a three-part research project. The goal of this project was to develop a general, complete linkage hierarchical clustering method that 1) substantially improves upon the accuracy of the standard complete linkage method and 2) can be fully automated or used with minimal operator supervision. The new clustering method is broadly applicable because it only assumes that clusters are globular or compact; that the 2-norms and the 1-norms of the data points in a data set are calculable; that noise is the only random component in a measured value; and that the noise that is embedded in each dimension of each data point is statistically independent and can be modeled as Gaussian random variables. The new method is consonant with the model for a measured value that scientists and engineers commonly use and thus exploits the structure of the clustering problems for which it was designed.

By using a clustering method that can construct only the cluster sets for select, possibly noncontiguous levels of an $\frac{n\cdot(n-1)}{2}+1$-level hierarchical sequence and by finding which levels of such a hierarchical sequence are meaningful *prior* to performing a cluster analysis, it is possible to construct only the cluster sets for meaningful levels of a hierarchical sequence and reduce the time complexity to construct cluster sets from

$O(n^4)$ to $O(ln^2)$. *These are the cluster sets that can have real world meaning.* The means for finding meaningful levels can be implemented as a distance graph that is visually examined for features that correlate with meaningful levels of the corresponding hierarchical sequence. Alternatively, it can be implemented as an equation that captures the graphical relationships that underlie these features. Additional work regarding enhancements to the algorithm, how noise influences inherent structure, designs for a reinforcement learning module, feature selection, and developing entirely new projects that bring machine learning methods over from the "computational side of things ... to the system ... kind of thinking" are possible future projects.

References

[1] R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, 2006.

[2] F. Murtagh. The remarkable simplicity of very high dimensional data: Application of model-based clustering. *J. of Classification*, 26:249–277, 2009.

[3] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer Science+Business Media, 2nd edition, 2011.

[4] National Science Foundation. www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286, Dec. 21, 2010.

[5] H. Gill. High confidence software and systems: Cyber-physical systems. In *NITRD National Workshop on High-Confidence Automotive Cyber-Physical Systems*, Troy, MI, April 3-4, 2008.

[6] G. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Science+Business Media, 2nd edition, 2005.

[7] M. Chui, M. Loffler, and R. Roberts. The internet of things. *Insights & Publications, www.mckinsey.com/insights/high_tech_telecoms_internet/the_internet_of_things*, March, 2010.

[8] en.wikipedia.org/wiki/ambient_intelligence.

[9] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition, 2003.

[10] E. Peay. Nonmetric grouping: Clusters and cliques. *Psychometrika*, 40(3):297–313, 1975.

[11] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[12] G. Lance and W. Williams. Computer programs for hierarchical polythetic classification ("similarity analyses"). *The Computer Journal*, 9(1):60–64, 1966.

[13] G. Lance and W. Williams. A general theory of classificatory sorting strategies i hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

[14] C. Biemann. *Structure Discovery in Natural Language*. Springer-Verlag, 2012.

[15] B. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. John Wiley & Sons, 5th edition, 2011.

[16] M. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[17] D. Kirk and W. Hwu. *Programming Massively Parallel Processors*. Elsevier Inc., 2nd edition, 2013.

[18] P. Berkhin. A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, chapter 2, pages 25–71. Springer-Verlag, 2006.

[19] E. Peay. Hierarchical clique structures. *Sociometry*, 37(1):54–65, 1974.

[20] E. Jewell and F. Abate editors. *The New Oxford American Dictionary*. Oxford University Press, 2001.

[21] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 5th edition, 2002.

[22] G. Karypis, E. Han, and V. Kumar. Multilevel refinement for hierarchical clustering. Technical report, Department of Computer Science & Engineering, University of Minnesota-Twin Cities, Minneapolis, MN, 1999.

[23] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson Education, 2006.

[24] A. Guénoche, P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification*, 8(1):5–30, 1991.

[25] H. Gill. CPS overview. In *Symposium on Control and Modeling Cyber-Physical Systems (www.csl.illinois.edu/video/csl-emerging-topics-2011-cyber-physical-systems-helen-gill-presentation)*, Champaign, IL, 2011.

[26] I. Matula. Graph theoretic techniques for cluster analysis algorithms. In J. Van Ryzin, editor, *Classification and Clustering*, pages 95–129. Academic Press, 1977.

[27] G. Gutin. Independent sets and cliques. In J. Gross and J. Yellen, editors, *Handbook of Graph Theory*, pages 389–402. CRC Press, 2004.

[28] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *J. of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[29] K. Daniels and C. Giraud-Carrier. Learning the threshold in hierarchical agglomerative clustering. In *Proceedings of the Fifth International Conference on Machine Learning and Applications (ICMLA '06)*, pages 270–278, Orlando, FL, 2006.

[30] H. Kim and S. Lee. A semi-supervised document clustering technique for information organization. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM '00)*, pages 30–37, McLean, VA, 2000.

[31] W. Navidi. *Statistics for Engineers and Scientists*. McGraw-Hill, 2006.

[32] W. Song, T. Matteo, and T. Aste. Hierarchical information clustering by means of topologically embedded graphs. *PLoS ONE*, 7(3):e31929, 2012.

[33] R. Nock and F. Nielsen. On weighting clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1223–1235, 2006.

[34] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2004.

[35] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor"meaningful? Technical report, Computer Sciences Department, University of Wisconsin-Madison, Madison, WI, 1998.

[36] A. Hinneburg, C. Aggarwal, and D. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB 2000)*, pages 506–516, Cairo, Egypt, 2000.

[37] L. Ferreira and D. Hitchcock. A comparison of hierarchical methods for clustering functional data. *Communications in Statistics–Simulation and Computation*, 38(9):1925–1949, 2009.

[38] M. Kim, W. V. Payne, and P. Domanski. Performance of a residential heat pump operating in the cooling mode with single faults imposed. Technical report, U.S. National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[39] P. Olver and C. Shakiban. *Applied Linear Algebra*. Pearson Education, 2006.

[40] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley. *PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; http://cir.ahajournals.org/cgi/content/full/101/ 23/e215].* June 13, 2000.