

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

A Tree-Based Measure for Hierarchical Data in Mixed Databases

By

DIMAN SIDDIQ HASSAN

A thesis submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

School of Computer Science

July 2016

”To my husband, Dler for his support and encouragement, my son Dadiyar and my daughter Dadveen who accompanied me to the UK, and also to my parents, my extended family and friends in Kurdistan for their continuing support, understanding and encouragement during this time”

Abstract

The structure of the data in a mixed database can be a barrier when clustering that database into meaningful groups. A hierarchically structured database necessitates efficient distance measures and clustering algorithms to locate similarities between data objects. Therefore, existing literature proposes hierarchical distance measures to measure the similarities between the records in hierarchical databases.

The main contribution of this research is to create and test a new distance measure for large hierarchical databases consisting of mixed data types and attributes, based on an existing tree-based (hierarchical) distance metric, the *pq*-gram distance metric. Several aims and objectives were pursued to fill a number of gaps in the current body of knowledge. One of these goals was to verify the validity of the *pq*-gram distance metric when applied to different data sets, and to compare and combine it with a number of different distance measures to demonstrate its usefulness across large mixed databases. To achieve this, further work focused on exploring how to exploit the existing method as a measure of hierarchical data attributes in mixed data sets, and to ascertain whether the new method would produce better results with large mixed databases. For evaluation purposes, the *pq*-gram metric was applied to The Health Improvement Network (THIN) database to determine if it could identify similarities between the records in the database. After this, it was applied to mixed data to examine different distance measures, which include non-hierarchical and other hierarchical measures, and to combine them to create a Combined Distance Function (CDF).

The CDF improved the results when applied to different data sets, such as the hierarchical National Bureau of Economic Research of United States (NBER US) Patent data set and the mixed (THIN) data set. The CDF was then modified to create a New-CDF, which used only the hierarchical *pq*-gram metric to measure the hierarchical attributes in the mixed data set. The New-CDF worked well, finding the most similar data records when applied to the THIN data set, and grouping them in one cluster using the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) clustering algorithm. The quality of the clusters was explored using two internal validation indices, Silhouette and C-Index, where the values showed good compactness and quality of the clusters obtained using the new method.

Acknowledgements

I would like to express my deep sense of gratitude to my supervisors, Professor Uwe Aickelin and Dr Christian Wagner, for their scientific supervision, continuous encouragement, unwavering support, advice and guidance during the period of the study.

I also wish to express my sincerest thanks to the staff of Intelligent Modelling and Analysis Research (IMA) group, particularly Dr. Daniele Soria especially during the first year of the study.

I am deeply grateful to the Ministry of Higher Education and Scientific Research Kurdistan Region of Iraq for providing me with an Overseas PhD Scholarship and also for their generous funding.

Contents

Abstract ii

Acknowledgements iii

Contents iv

List of Figures viii

List of Tables xii

List of Publications xiii

1 Introduction 1
- 1.1 Background and Motivation . . . 2
- 1.2 Aims and Objectives of This Thesis . . . 7
- 1.3 Thesis Organisation . . . 9
- 1.4 Contributions to Knowledge . . . 12

2 Literature Review 14
- 2.1 Non-Hierarchical Data . . . 14
- 2.2 Hierarchical Data . . . 16
- 2.3 Data Mining and Knowledge Discovery . . . 17
  - 2.3.1 Data Pre-processing . . . 18
  - 2.3.2 Clustering Data Mining Task . . . 18
    - 2.3.2.1 Clustering for the NBER U.S. Patent Database . . . 19
    - 2.3.2.2 Clustering for the THIN Database . . . 21
  - 2.3.3 Partitional Clustering . . . 22
    - 2.3.3.1 K-means Algorithm . . . 23
  - 2.3.4 Hierarchical Clustering . . . 24
  - 2.3.5 Distance Measures for Hierarchical Clustering . . . 25
  - 2.3.6 Distinction Between the Partitional and Hierarchical Clustering Approaches . . . 26
    - 2.3.6.1 The Hierarchical BIRCH Clustering Algorithm . . . 27

2.4 Proximity Measures . . . 28
2.4.1 Distance Measures Used for Hierarchical Data . . . 30
2.4.1.1 Background on Edit Distance Metric . . . 30
2.4.1.2 Background on pq-gram Metric . . . 32
2.4.1.3 Rules for Hierarchical Similarity Measures . . . 33
2.4.2 Distance Measures Used for Non-Hierarchical Data . . . 34
2.4.3 The Technical Details of Combining Distance Measures . . . 36
2.4.3.1 Adding Weights to the CDF . . . 40
2.5 Summary and Conclusions . . . 41

3 Methodology **43**
3.1 Background on Lenses Data Set . . . 43
3.2 Background to the Iris Data Set . . . 44
3.3 Background to the NBER US Patent Data Set . . . 44
3.3.1 Patent Data Pre-processing . . . 45
3.4 Background to the Medical THIN Database. . . . 46
3.5 The Hierarchical BIRCH Clustering Algorithm . . . 48
3.6 Geometric and Hamming Distance Metrics for Non-hierarchical Data Sets . . . 52
3.6.1 Euclidean Distance Metric . . . 52
3.6.2 Minkowski Distance Metric . . . 53
3.6.3 Manhattan Distance Metric . . . 53
3.6.4 Hamming Distance Metric . . . 53
3.7 Hierarchical Distance Metrics for Hierarchical Data Sets . . . 54
3.7.1 Edit Distance Metric . . . 54
3.7.2 Pq-gram Distance Metric . . . 55
3.7.3 Other Hierarchical Distance Measures . . . 59
3.8 Cluster Validation . . . 61
3.8.1 Silhouette Index . . . 63
3.8.2 C Validity Index . . . 64
3.8.3 Rand Index . . . 64
3.8.4 Jaccard Index . . . 65
3.9 Cluster Visualisation Based on PCA . . . 65
3.10 Summary . . . 66

4 Apply the pq-gram Distance Metric to Different Data Sets **69**
4.1 Introduction . . . 69
4.2 Apply the pq-gram Metric to Non-hierarchical Data Sets . . . 70
4.2.1 Lenses Data Set Pre-processing . . . 70
4.2.2 Apply the pq-gram Distance Metric to the Iris Data set . . . 72
4.3 Apply the pq-gram metric to a hierarchical data set . . . 75
4.4 Material and Methods for the Comparison . . . 76
4.4.1 THIN Data Set Preparation . . . 76
4.4.2 Distance Metrics Used for the Comparison . . . 77

4.5 Data Pre-processing . . . 79
4.5.1 Pq-gram and Edit Distance Preparation . . . 79
4.5.2 Geometric and Hamming Metrics Preparation . . . 80
4.5.3 Distances Calculation . . . 81
4.5.4 Clustering the Distances . . . 84
4.6 Discussion . . . 91
4.7 Summary and Conclusion . . . 94

5 Improve the Pq-gram Distance Metric for Mixed Data Sets 96
5.1 Introduction . . . 96
5.2 Comparison Between the Hierarchical Measures Based on the Hierarchical Similarity Rules . . . 97
5.3 Data Pre-processing . . . 99
5.4 Clustering and Visualising the Iris Data Set using BIRCH . . . 100
5.5 Using the New Measure for Patent Data Set . . . 105
5.5.1 Clustering the 200 Data Records Using the CDFs and BIRCH Algorithm- Example 1 . . . 106
5.5.2 Clustering the 500 Data Records Using the CDFs and BIRCH Algorithm- Example 2 . . . 111
5.5.3 Clustering the 1000 Data Records Using the CDFs and BIRCH Algorithm- Example 3 . . . 113
5.6 Replacing the Hierarchical Measures by the Pq-gram Distance Function 115
5.6.1 Example 1 . . . 115
5.6.2 Example 2 . . . 116
5.6.3 Example 3 . . . 117
5.7 Cluster Validation . . . 117
5.8 Using a Different Method to Compare the Results . . . 119
5.8.1 The new figures for 200 data records using both CDFs . . . 119
5.8.2 The New Figures for 500 Data Records Using Both CDFs . . . 121
5.9 Discussion . . . 122
5.10 Conclusion . . . 126

6 Apply the New Measure to A Mixed Data set 127
6.1 Introduction . . . 127
6.2 Data Pre-processing . . . 128
6.3 Distance Calculation and Clustering Using the BIRCH Algorithm for DOXEPIN Drug . . . 129
6.3.1 The Distance Calculation Using CDF . . . 129
6.3.2 Using Pq-gram Metric With the CDF . . . 131
6.3.3 Adding Weights to the CDF . . . 133
6.3.4 Replacing the Hierarchical Measures by the Pq-gram Distance Function . . . 138
6.4 Distance Calculation and Clustering Using BIRCH Algorithm for DESLORATADINE . . . 141
6.4.1 Using a Different Method to Compare the Results . . . 143

6.5 Distance Calculation and Clustering Using BIRCH Algorithm for LEVOFLOXACIN . . . 145
6.6 The Clusters Validation . . . 147
6.7 Summary and Conclusion . . . 149

7 Conclusions **152**
7.1 Contributions . . . 154
7.2 Limitations of the Contributions . . . 157
7.3 Evaluation of Aims . . . 158
7.4 Future Works . . . 160

A Appendix A **163**
A.1 The figures of DOXEPIN drug Using CDF . . . 163
A.2 The figures of DOXEPIN drug Using CDF with pq-gram . . . 166
A.3 The software used and examples of the code . . . 167
A.4 Figures for Patent and THIN Data Sets . . . 169

Bibliography **176**

List of Figures

1.1 Example of an unstructured data . . . 2
1.2 Example of a semi-structured data . . . 2
1.3 Example of a hierarchical data structure . . . 3

2.1 The data diagram . . . 15
2.2 Example of a branch of the THIN READ code tree . . . 17
2.3 Data Mining and Knowledge Discovery Process . . . 18
2.4 The CF Tree Structure . . . 28
2.5 An example of the Edit Distance Operations . . . 31

3.1 A screen shot of the created Patent data table . . . 46
3.2 THIN Database Structure . . . 48
3.3 The BIRCH algorithm overview . . . 50
3.4 A tree and its extended tree . . . 57
3.5 A *pq*-gram pattern with different examples . . . 57
3.6 The 2,3-grams of the example of tree $T^{2,3}$ . . . 58
3.7 The 2,3-grams profile of tree $T^{2,3}$ . . . 58
3.8 Example of a hierarchy tree . . . 60

4.1 The three pq-gram trees of Lenses data set . . . 71
4.2 A Sample of Iris Data and a pq-gram tree example . . . 72
4.3 1, 3-gram of Iris data using pq-gram metric . . . 73
4.4 2, 2-gram of Iris data using pq-gram metric . . . 73
4.5 The confusion matrix and statistics for the clusters of Iris data set . . . 74
4.6 The clusters of 100 patent data set using pq-gram metric . . . 75
4.7 Example of a tree T and its 2, 3-Extended tree . . . 78
4.8 Example of a single *pq*-gram from a THIN data tree . . . 78
4.9 Part of the THIN data extracted based on specific attributes . . . 80
4.10 A tree representation from THIN data . . . 80
4.11 The 2, 3-grams of a tree T . . . 81
4.12 The clusters of patients using Euclidean metric, DESLORATADINE drug 85
4.13 The clusters of patients using Minkowski metric, DESLORATADINE drug . . . 85
4.14 The clusters of patients using Manhattan metric, DESLORATADINE drug 86
4.15 The clusters of patients using Hamming metric, DESLORATADINE drug 86
4.16 The clusters of patients using *pq*-gram metric, DESLORATADINE drug 87

4.17 The clusters of patients using Edit Distance metric, DESLORATADINE drug . . . 87
4.18 The clusters of patients using Euclidean metric, DOXEPIN drug . . . 88
4.19 The clusters of patients using Minkowski metric, DOXEPIN drug . . . 88
4.20 The clusters of patients using Manhattan metric, DOXEPIN drug . . . 89
4.21 The clusters of patients using Hamming metric, DOXEPIN drug . . . 89
4.22 The clusters of patients using *pq*-gram metric, DOXEPIN drug . . . 90
4.23 The clusters of patients using Edit Distance metric, DOXEPIN drug . . . 90

5.1 Example of a Hierarchy Patent Data . . . 98
5.2 Hierarchical data have the same length . . . 100
5.3 Multi-dimensional scaling plot for Iris data set . . . 102
5.4 Plotting Iris data using CDF without pq-gram distance function . . . 102
5.5 Plotting Iris data using CDF with pq-gram distance function . . . 103
5.6 The confusion matrix and statistics for Iris data set . . . 103
5.7 The confusion matrix and statistics for Iris data set . . . 104
5.8 Clusters of 200 patent data records using CDF and clusplot function . . . 109
5.9 Clusters of 200 patent data records with 2,2-gram . . . 110
5.10 Clusters of 200 patent data records by giving a weight to the 2,2-gram metric . . . 110
5.11 Clusters of 500 patent data records without pq-gram distance function . . . 111
5.12 Clusters of 500 patent data records using BIRCH algorithm and 2,2-gram metric . . . 112
5.13 Clusters of 500 patent data records by giving a weight to the 2,2-gram metric . . . 112
5.14 Plotting 1000 patent data records using clusplot function . . . 114
5.15 Clusters of 1000 patent data records with 2,2-gram . . . 114
5.16 Clusters of 1000 patent data records by giving a weight to the 2,2-gram metric . . . 115
5.17 Clusters of 200 patent data records using the New-CDF . . . 116
5.18 Clusters of 500 patent data records using the New-CDF . . . 117
5.19 Clusters of 1000 patent data records using the New-CDF . . . 118
5.20 Clusters of 200 patent data records using the CDF with pq-gram . . . 120
5.21 Clusters of 200 patent data records using the New-CDF . . . 120
5.22 Clusters of 500 patent data records using the CDF with pq-gram . . . 122
5.23 Clusters of 500 patent data records using the New-CDF . . . 122

6.1 Screenshot of a THIN data sample . . . 130
6.2 The 7 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram . . . 132
6.3 The 8 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram . . . 132
6.4 The 9 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram . . . 133
6.5 The 10 clusters of 1172 THIN data records-DOXEPIN drug . . . 135

6.6 The 10 clusters of 1172 THIN data records-DOXEPIN drug Using CDF with pq-gram . . . 135
6.7 The 10 clusters of 1172 DOXEPIN drug data using CDF and weighted-pq-gram . . . 136
6.8 The 42 clusters of 1172 THIN data records-DOXEPIN drug representing 42 patients . . . 136
6.9 The 42 clusters of 1172 THIN data records-DOXEPIN drug Using CDF and pq-gram . . . 137
6.10 The 42 clusters of 1172 THIN data records-DOXEPIN drug Using CDF and weighted-pq-gram . . . 137
6.11 The 10 clusters of 1172 THIN data records-DOXEPIN drug Using New-CDF . . . 138
6.12 The 10 clusters of 1172 DOXEPIN drug data using CDF-pq-gram . . . 140
6.13 The 10 clusters of 1172 DOXEPIN drug data using New-CDF . . . 140
6.14 The 5 clusters of 1000 DESLORATADINE data records-using CDF . . . 141
6.15 The 5 clusters of 1000 DESLORATADINE data records using CDF with 2,2-gram . . . 142
6.16 The 5 clusters of 1000 DESLORATADINE data records using New CDF 143
6.17 The 5 clusters of 1000 DESLORATADINE data records using CDF-pq-gram and PCA . . . 144
6.18 The 5 clusters of 1000 DESLORATADINE data records using New-CDF and PCA . . . 145
6.19 The 3 clusters of 576 LEVOFLOXACIN data records using CDF . . . 146
6.20 The 3 clusters of 576 LEVOFLOXACIN data records using CDF with pq-gram . . . 146
6.21 The 3 clusters of 576 LEVOFLOXACIN data records using New CDF . 147

A.1 The 5 clusters of 1172 THIN data records-Doxepin drug . . . 163
A.2 The 6 clusters of 1172 THIN data records-Doxepin drug . . . 164
A.3 The 7 clusters of 1172 THIN data records-Doxepin drug . . . 164
A.4 The 8 clusters of 1172 THIN data records-Doxepin drug . . . 165
A.5 The 9 clusters of 1172 THIN data records-Doxepin drug . . . 165
A.6 The 5 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram . . . 166
A.7 The 6 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram . . . 166
A.8 The code of how the pq-gram distance calculated . . . 168
A.9 The code of how to run BIRCH algorithm . . . 168
A.10 The Patent table From Patent Data Set . . . 169
A.11 The inventor table From Patent Data Set . . . 169
A.12 The Assignee table From Patent Data Set . . . 170
A.13 A screen shot of the medical table contained within the THIN database. 171
A.14 A screen shot of the patient table contained within the THIN database. . 171
A.15 A screen shot of the therapy table contained within the THIN database. . 172
A.16 The DOXEPIN sample data . . . 173

A.17 The DESLORATADINE sample data 174
A.18 The LEVOFLOXACIN sample data 175

List of Tables

4.1 The original Lenses data set . . . 70
4.2 A subset of information from the database for two kinds of drugs . . . 77
4.3 The frequency table from THIN data . . . 81
4.4 Smallest normalised distances for patients taking DESLORATADINE drug . . . 83
4.5 The Shared Medical Events for Patients in Table 4.4 . . . 83
4.6 Smallest normalised distances for patients taking DOXEPIN drug . . . 83
4.7 The number of patients in each cluster for DESLORATADINE drug . . . 91
4.8 The number of patients in each cluster for DOXEPIN drug . . . 91

5.1 Similarity Values between Various Nodes in the Tree in Figure 5.1 . . . 99
5.2 All the data points (records) of the cluster number 3 in Figure 5.8 . . . 107
5.3 Some of the patent data records of cluster 1 in Figure 5.8 that have the most similarity . . . 107
5.4 Silhouette and C-index values for the 3 clusters of 200 patent data records 118
5.5 Silhouette and C-index values for the 4 clusters of 500 patent data records 118
5.6 Silhouette and C-index values for the 5 clusters of 1000 patent data records 119

6.1 Silhouette and C-index values for all the clusters . . . 134
6.2 Silhouette and C-index values for the 10 clusters of 1172 DOXEPIN data records . . . 148
6.3 Silhouette and C-index values for the 5 clusters of 1000 DESLORATADINE data records . . . 148
6.4 Silhouette and C-index values for the 3 clusters of 576 LEVOFLOXACIN data records . . . 148

List of Publications

Conference Papers

D. Hassan, U. Aickelin, and C. Wagner,

"*Comparison of distance metrics for hierarchical data in medical databases*",

in Neural Networks (IJCNN), 2014 International Joint Conference on. IEEE, 2014,

pp. 3636–3643. (Chapter 4).

Published

Journal Papers

D. Hassan, U. Aickelin, and C. Wagner,

"*New Distance Measure Applied to a Hierarchical Data Set*", Journal of Pattern

Recognition, 2016.

D. Hassan, U. Aickelin, and C. Wagner,

"*A New Distance Measure for Medical Mixed Data Set*", Journal of Information

Systems, 2016.

Chapter 1

Introduction

Handling mixed large data, especially data that is in a tree-based structure, is one of the most important topics on the agenda of contemporary researchers in the computer science domain. To manage these data types, powerful knowledge discovery tools, such as data mining techniques are essential. One of the most interesting data mining techniques available to manage mixed large data is the clustering technique, which groups data into clusters containing objects that are more similar than those in other clusters. To measure the similarity between data objects, proximity measures are used, and these are the basic building blocks of clustering techniques.

Recently, many hierarchical measures have been proposed to perform clustering by measuring the similarities present in hierarchical data. This thesis investigates the role of a new distance measure created based on an existing hierarchical distance metric, the *pq*-gram distance metric, to measure similarities between hierarchical data in a mixed database. This chapter first provides the background and motivation for the current study and then introduces its aims and objectives. It also gives an outline of the thesis organisation and concludes by describing the work's anticipated contribution to the field.

1.1 Background and Motivation

Big data can be defined as a data set where the larger it becomes the more difficult it is to process it using traditional database tools or database pre-processing applications [26]. It consists of three main data types: (i) unstructured data that does not follow any specified format or rules and the contents of which do not fit in a database, e.g. text or word processing documents such as those displayed in Figure 1.1; (ii) semi-structured data, e.g. csv files or XML data as shown in Figure 1.2, and (iii) structured data, in which information is arranged in rows and columns within tables or in tree-like structures, e.g. all databases in database management systems and hierarchical databases (as illustrated in Figure 1.3).

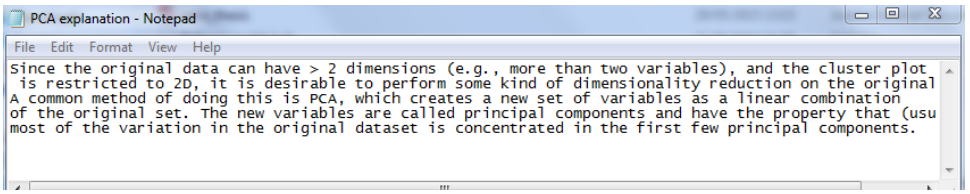


FIGURE 1.1: Example of an unstructured data

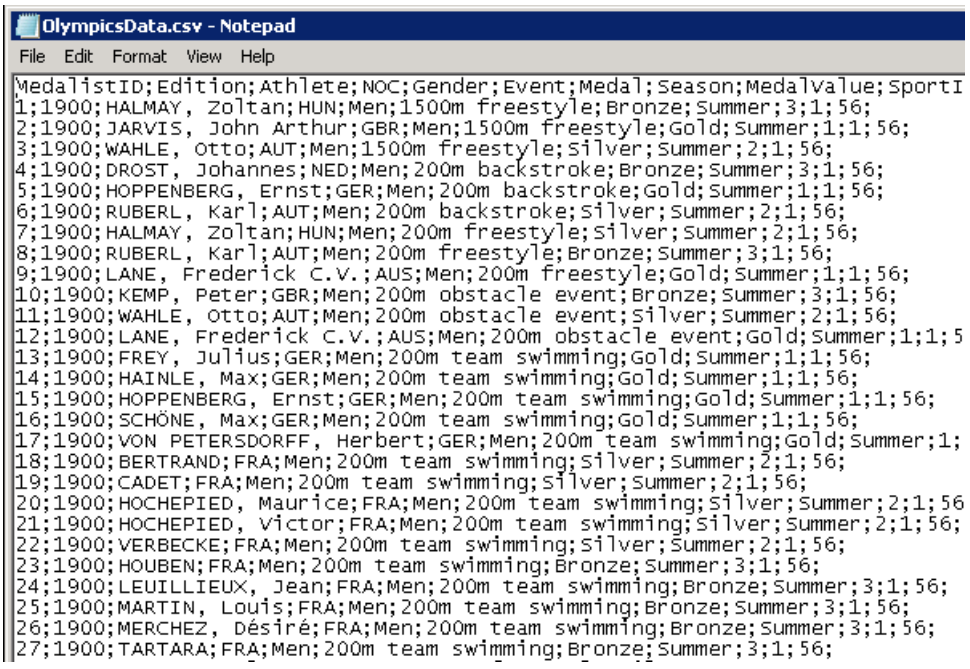


FIGURE 1.2: Example of a semi-structured data

Extracting knowledge from large mixed databases, including hierarchical ones, is a significant challenge in the world of computer science and knowledge discovery. The

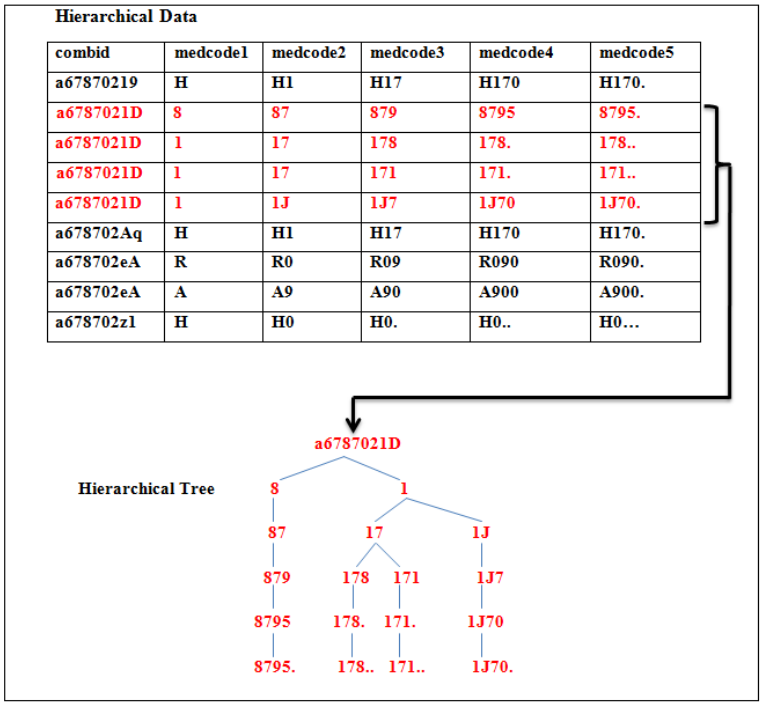


FIGURE 1.3: Example of a hierarchical data structure

extraction of information from such databases requires efficient techniques and tools, such as data mining and machine learning. The fundamental characteristic of data mining is that it is applied to large and complex data sets, comprising a very large number of objects describing various types of attributes and variables [49]. Thus, data mining algorithms need to be scalable and capable of dealing with different data types with various attributes. However, the scalability feature of most data mining algorithms is currently weak when applied to very large data sets, since they were initially developed for applications other than data mining, which involved small data sets [49].

One of the most important data mining tools is clustering, which aims to detect groups of similar objects within data to find structure present in the data set [8]. Despite the difficulties encountered when applying clustering to large mixed data [112], it is considered as a useful tool because it partitions the data into groups and can find hidden information in raw data. When clustering, an object is described either by a set of measurements or by a list of features [115]. Cluster analysis does not use category labels to tag objects with prior identifiers; this distinguishes cluster analysis from other forms of analysis, such as pattern recognition and decision analysis. The objective of cluster analysis is simply to find a convenient and valid approach to the organisation of the

data. Thus, it does not seek to establish rules for separating future data into categories [54]. To perform a cluster analysis, the data needs to have been deliberately prepared to avoid the high computational complexity and the occurrence of redundant and irrelevant features [15]. Thus, feature selection and feature extraction are two important steps when validating a clustering application to obtain suitable features for clustering [53] [129].

Different clustering algorithms exist for clustering mixed databases, including partitional or point assignment algorithms and hierarchical clustering algorithms. The most well-known partitional clustering algorithm is the k-means clustering algorithm [92], where k-means refers to the means of the groups represented which have been adjusted k times. This clustering algorithm was extended in [49], such that it can now efficiently cluster mixed data sets that contain different data types and attributes.

The second type of clustering algorithms is hierarchical clustering. This is used to produce a nested series of partitions sequentially, based on the application of similarity measures for merging or splitting clusters [53]. The result of hierarchical clustering is a tree-like nested structure termed a *dendrogram*, which represents the nested grouping of patterns and the similarity levels between all objects [129]. Recently, the data mining community has developed a number of fast hierarchical clustering algorithms that can cluster very large data sets, such as Clustering Algorithm based on Randomised Search (CLARANS) [81], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [33], and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [131]. The latter, BIRCH, is a well-known hierarchical algorithm for big data sets and is classified as a sample-based clustering technique [112]. These sampling-based algorithms perform clustering on a sample from the selected data set, which speeds up the computation process, and then generalise the results to the entire data set [112]. The BIRCH clustering algorithm has its own data structure, called a Cluster Feature (CF) tree, which it uses to solve the problem of large-sized data contrasted with memory size; i.e. the problem that the I/O cost dominates computational time. CF delivers a concise summary of each cluster, representing the data points or records in each cluster.

Many data sets, including hierarchical data sets, have been clustered using the aforementioned clustering algorithms such as the spatial data set [33] and network traffic data set [77]. A hierarchical data set is a collection of data records connected to each other through links, where each data record consists of a collection of attributes or variables each containing only one data value [113]. Examples of the hierarchical mixed data sets are longitudinal databases, patent databases, spatial data sets and XML data. These data sets either have hierarchical structures, such as the National Bureau of Economic Research of United States (NBER US) Patent database [45], or some attributes that are hierarchically structured, like the READ codes in The Health Improvement Network (THIN) data set (https://www.ucl.ac.uk/pcph/research-groups-themes/thin-pub/database). The fundamental characteristics of hierarchical structures are the compactness and dependence on the nature of the data. The hierarchical data structure can also save space, time and simplify operations such as search [106] since it can be based on the principle of recursive decomposition [107]. Currently, data mining clustering techniques for hierarchical data sets are an important consideration, since they comprise mixed data types including hierarchical data attributes. Also, since these data sets require the implementation of efficient structural similarity measures, which are not very commonly proposed, the data mining techniques employed for hierarchical data sets might be more sophisticated than those employed for non-hierarchical data sets.

As mentioned above, when clustering hierarchical data sets, efficient proximity measures need to be implemented to measure the similarity between the objects in the data sets, particularly the hierarchical distance measures specifically designed to manage such data sets. The hierarchical *pq*-gram distance metric was introduced to match hierarchical data from large data sources approximately, and this metric has been proven to be more efficient and provide less complexity in terms of time than other hierarchical distance measures, such as the edit distance metric [5]. This motivates users to use the *pq*-gram metric to measure the similar distances between the data records in large hierarchical databases. This work evaluates the *pq*-gram metric for the development of a new distance measure for a mixed data set like THIN.

The hierarchical data sets used in this work are the NBER US Patent and THIN data sets. The hierarchical NBER US Patent data set has been clustered in the literature and has been used to resolve different problems with different proximity measures [68] [59] [80]. However, using tree-based measures to measure the similar distances between patent data records is uncommon, which motivates the application of the hierarchical *pq*-gram distance metric in this study. Whilst different data mining techniques have been applied to the THIN database [96] [97] [72] [98], no clustering techniques have been applied to it to date. This is another motivation for this study, which aims to apply the clustering technique to such hierarchical databases, as well as to use the hierarchical *pq*-gram distance metric to measure the similarity between database records.

Several hierarchical measures and metrics (see the difference in Chapter 2, Section 2.4) have been introduced to measure the similarity in hierarchical databases such as *pq*-gram distance metric [5], Edit distance metric [118], Hierarchical Similarity measure 1 ($HSM1$) [77] and others. The first two metrics are compared with four other distance metrics that are suitable for use with non-hierarchical data sets, Minkowski [23], Euclidean, Manhattan [41] and Hamming [48] [86] distance metrics. The comparison, as explained in Chapter 4, investigates which of the distance metrics, specifically focusing on the *pq*-gram metric, performs better and is most suitable for handling THIN data in order to cluster patients with similar data records [47].

The *pq*-gram distance metric, as mentioned earlier, is used to approximately match hierarchical data from large sources between ordered labelled trees [5]. In this work, the metric is applied to different data sets, including simple non-hierarchical sets like the Iris [35] and lenses (`https://archive.ics.uci.edu/ml/datasets/Lenses`) data sets, a hierarchical data set named the NBER US Patent data, and finally a mixed data set named the THIN data set. Since the performance of the hierarchical metric is thought to be poor for non-hierarchical data sets, relatively good for hierarchical data sets, and good for the mixed data set, this metric may produce better results if used only to manage the hierarchical attributes within a mixed data set. However, using the *pq*-gram distance metric to measure the similarity between hierarchical attributes only within mixed data type records has not yet been investigated. Thus, to do so, the metric

has to be combined with other hierarchical and non-hierarchical distance measures, in a combined distance function (CDF), similar to the measure proposed in [77], as illustrated in Chapter 2, Section 2.4.3, so it can be used to group similar mixed data type records in a single cluster.

Further investigation is performed by adding weights to the different aspects of the CDF, specifically, the *pq*-gram distance function. This process aims to show the role of the *pq*-gram distance function among other hierarchical distance functions when measuring the similar distances between database records, as will be illustrated in Chapters 5 and 6.

By adding the weights to the parts of CDF with the *pq*-gram distance function, a new combined distance function can be obtained by choosing not to consider the other hierarchical distance measures like the Hierarchical Similarity Measure1 (D_{HSM1}) [77] by giving them the weight 0. The new combined distance function, termed the 'New-CDF' combines the *pq*-gram distance function for hierarchical attributes, the Euclidean distance function for numerical attributes, and a histogram of the frequencies with which the categorical attribute values appear. In this New-CDF, the aim is to investigate how effective the *pq*-gram distance function is at finding similar data records when used to replace the other hierarchical distance functions in the CDF, and when applied to the mixed data set THIN.

1.2 Aims and Objectives of This Thesis

The ultimate goal of this research is to design a new distance measure for mixed data sets, including hierarchical data, which combines the advantages of the *pq*-gram distance metric and other distance measures to deliver better overall performance. As mentioned earlier, the *pq*-gram distance metric first needs to be applied to different data sets to test its validity (simple non-hierarchical, hierarchical, and a mixed data set). It then needs to be improved prior to investigating whether the new measure proposed could produce better results when applied to mixed data sets. Several research questions and hypotheses were identified for this project and underlie this work. In addition

to being hierarchical, databases like the NBER US Patent and THIN are mixed and very large. Clustering of the hierarchical patent data set has been described in the literature, but this did not involve using hierarchical distance measures. In terms of the mixed medical THIN database, clustering of the data set using hierarchical distance measures to determine the similarity between data records has not yet been explored. Furthermore, a comparison of distance metrics for hierarchical data has not been conducted previously. This gap in the knowledge led to the formulation of the following research questions:

1. Can the *pq*-gram distance metric be applied to any data set to identify similarity between data objects reliably?
2. What is the performance of the *pq*-gram distance metric compared to other measures when measuring the similarity between the mixed THIN data and is the metric suitable for clustering the entire patient population from the data set?
3. How can the *pq*-gram metric improve the results when measuring the similar distances between mixed data type records in large databases if it produces poor results?
4. Can improvements be made by using or combining different distance measures for mixed data type attributes in large databases?
5. While combining between the different distance measures, can the new measure be useful for hierarchical data sets to find the similar distances between the data records and can it be suitable for clustering large mixed data sets while clustering using the hierarchical BIRCH algorithm?

In order to answer the research questions, the following objectives were identified:

1. To apply the *pq*-gram distance metric to simple non-hierarchical, hierarchical and mixed data sets for the purpose of a feasibility study.
2. To compare the *pq*-gram metric with other different distance metrics, hierarchical and non-hierarchical, to measure the similarities between the mixed data in THIN

database, and to show its usefulness and suitability for clustering the data set using the *k*-means clustering algorithm.

3. To create a new distance measure, using the *pq*-gram metric and different distance measures for the mixed data set, to investigate if the new measure produce better results.

4. To combine the distance functions of each data type within a mixed data record (hierarchical and non-hierarchical) for the creation of the new distance function that measures the similarity between data records and then cluster these using the BIRCH clustering algorithm.

5. To add weight to the different aspects of the combined distance function (CDF) to illustrate the impact of the attributes, particularly, the *pq*-gram distance metric, while calculating the distances between the data records using the CDF.

6. To replace the other hierarchical distance measures in the CDF using the *pq*-gram distance metric, and to combine it with other distance functions for non-hierarchical data to create a new CDF, New-CDF.

7. To investigate the suitability of the New-CDF for clustering patient populations in THIN data using the hierarchical BIRCH clustering algorithm.

1.3 Thesis Organisation

This thesis is structured as follows.

Chapter 2 presents a literature review of the data mining clustering task over very large hierarchical databases within the process of knowledge discovery in databases, and describes the concepts of hierarchical and non-hierarchical data sets. It also reviews general clustering techniques, details the clustering of two hierarchical databases and two non-hierarchical data sets as described in the literature. Furthermore, it offers a general description of proximity measures and the rules needed to create a hierarchical similarity measure are reported. The proximity measures for both the hierarchical and

non-hierarchical data sets used in this work are then reviewed to distinguish between them, and to explain how they can be combined to handle mixed data type entries, while clustering is described based on literature.

Chapter 3 describes all the research methodologies applied in this work, from the databases used to the techniques for validating the results. It begins by providing some general background about the four data sets and describes the preparation process for each data set prior to commencing the clustering task. Then to explain the similarities between the data sets, the hierarchical distance metrics for the hierarchical data sets and the distance metrics for non-hierarchical data sets are described. The chapter ends by describing the cluster validation techniques used, which include the internal and external validation indices.

Chapter 4 starts by performing a feasibility study to assess the validity of the *pq*-gram distance metric when applied to different data sets (non-hierarchical, hierarchical, and a mixed data set). The study of the non-hierarchical classified data sets is designed to check the level of agreement between the resulting clusters from these data sets, using the *k*-means clustering algorithm with the *pq*-gram distance calculation, and the original data set classes.

In addition, Chapter 4 compares two different groups of metrics for two types of data structure applied on subsets of the THIN database, namely Minkowski, Euclidean, Manhattan and Hamming metrics for the first group, and *pq*-gram and Edit Distance metrics for the second group. The comparison shows the *pq*-gram distance metric delivers good performance when measuring the similarity between the patients' data from the subsets.

In chapter 5, a new distance measure was created for mixed hierarchical data sets. This was done by employing different distance measures to different data types of data. In this chapter, the hierarchical NBER US Patent database is explored as a new data set to determine if the method could be applied to this data set to produce similar data records for mixed data types by using a CDF.

The *pq*-gram metric is applied to measure the similarity between NBER US Patent database records. The metric is first combined to work alongside another hierarchical distance function, named the D_{HSM1} for hierarchical attributes and non-hierarchical measures for numerical and categorical attributes. The distance functions for the final two attributes are the Euclidean distance function and the histogram of frequencies for the categorical attribute values. The CDF produced good results. The quality of the clusters was evaluated using cluster validation indices, internal indices such as the Silhouette and C-Index and external indices such as the Rand and Jaccard indices. Some of the clusters provided were more similar in terms of their data records than others according to their attributes. Adding the *pq*-gram distance function to the CDF produced slightly better results than when not using it. Thus, in order to concentrate more fully on the *pq*-gram distance function role along with the CDF, weights were added to different aspects of the CDF. The weighted *pq*-gram distance function influenced the results, and in some examples led to improved results.

The New-CDF is then created, consisting of only the *pq*-gram distance function for hierarchical attributes and other distance functions for non-hierarchical attributes. After applying the New-CDF, the resulted clusters were not suitably compact and separated when compared to the previous results that use the *pq*-gram distance function alongside the CDF. In addition, in this chapter, the Iris data set is used as a validation data set to ascertain whether the BIRCH algorithm could produce the same original classes of Iris (setosa, versiclor and virginica flowers) using CDF and New-CDF.

In Chapter 6, the new distance measure is investigated as a tool for use with the medical UK mixed THIN database. This database has the characteristic of multiple data records belonging to an individual patient, which provides an opportunity to develop well-separated and compact clusters with similar data records for patients. The new method is applied to three different data sets for three different medical drugs comprising three data sets; the New-CDF worked and produced the most similar data records in one cluster when clustering the data using the hierarchical BIRCH clustering algorithm.

The results when using the CDF without the *pq*-gram distance function and the

CDF with the *pq*-gram distance function were slightly different, but in general the clusters were judged to be compact and well separated when their quality was evaluated using the cluster validation indices.

The final chapter of this thesis, Chapter 7, concludes the work, emphasising its main contributions and highlighting several possible directions for future research.

1.4 Contributions to Knowledge

Three main contributions to knowledge result from this work. Firstly, it provides a comparison between the *pq*-gram distance metric and other different measures for hierarchical and non-hierarchical data management when determining the similarity of the objects in the mixed THIN database records. The comparison shows the *pq*-gram distance metric can measure similar distances and group them in one cluster by clustering the data using the *k*-means clustering algorithm. This research work led to a conference paper detailing this contribution, as set out in Chapter 4.

Secondly, this thesis creates a new distance measure using the *pq*-gram distance metric by looking at different distance measures that can be combined for mixed data types of attributes in large databases. This used to measure the similar distances between the data records. The CDF is produced, which consists of hierarchical measures for hierarchical attributes including the *pq*-gram distance metric to work along with the new measure, the Euclidean distance metric for numerical attributes and the histogram of frequencies of categorical values. The new approach delivered slightly better results when applied to mixed databases.

Finally, the objective of using the *pq*-gram distance function to measure only hierarchical attributes, in combination with other distance functions to measure non-hierarchical attributes, using a new combined distance function (New-CDF) has been accomplished. New-CDF, the new measure, produced reasonable results when applied to the mixed THIN database to group the most similar data records into one cluster.

The quality of the clusters was ascertained by the indices values used: Silhouette and C-index.

Chapter 2

Literature Review

This chapter provides a literature review of general clustering techniques, as well as those clustering approaches applied in studies of hierarchical mixed databases. It also reviews the distance measures used for hierarchical and non-hierarchical data sets, and examines the possibility of combining them into one distance function. One of the contributions of this work is the use of hierarchical distance measures to measure the similarity between hierarchical data in hierarchical databases. Therefore, this chapter also discusses previously published literature on hierarchical databases and hierarchical distance measures.

The aim of this chapter is to present relevant background information concerning all the research subjects previously used in the clustering of hierarchical data sets and to highlight gaps in the body of knowledge. This underscores the motivation of the thesis to apply hierarchical based-distance measures, specifically application of the *pq*-gram distance metric to mixed databases to measure the similar distances between databases entries.

2.1 Non-Hierarchical Data

Data, in general, comprises facts, such as numbers, words, measurements, observations, or simply descriptions of things. It is actual information used as a basis for reasoning,

discussion or calculation. Data can be stored in computers as processed information and can be of any type, e.g. text, numeric, video, audio clips, software programs, images, logical, hierarchical. Based on the size of the data sets, it can be divided into two categories, small data and big data. Small data includes specific attributes derived from big data to address specific problems, e.g. the small data sets used for the experiments in Chapters 4, 5 and 6. The second category is big data, which is defined and divided according to three main types, as described in the previous chapter (see section 1.1). Figure 2.1 provides a diagram of data types.

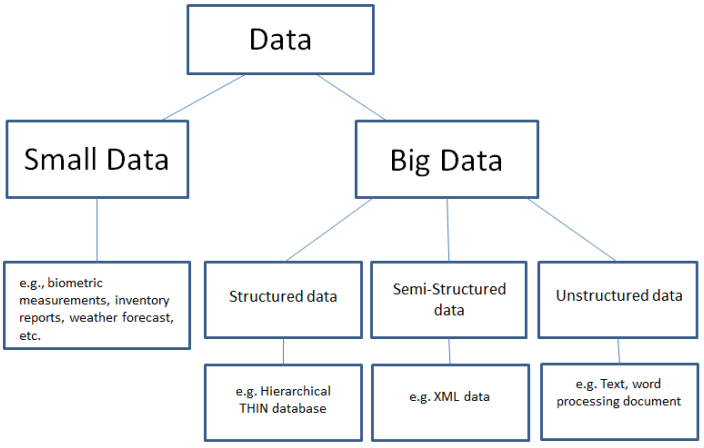


FIGURE 2.1: The data diagram

There are different ways in which data can be organised. It can be sorted, unsorted, structured, semi-structured, or simply unstructured. This study is interested in structured types of data, which include hierarchical and non-hierarchical data. The non-hierarchical data structure describes types of data that fit databases and which are not organised as tree-like structures. Data sets that have such a structure can be called non-hierarchical data sets. Examples of non-hierarchical data sets are: breast cancer data set, seeds data set, User Knowledge Modelling data set (http://archive.ics.uci.edu/ml/datasets.html) and others, which may contain either only numerical, only categorical, both numerical and categorical data or others.

Non-hierarchical data sets have been explored in different fields, such as data mining classifications for breast cancer data set [14], data mining regression for the breast

cancer Wisconsin data set [13] and data mining clustering on the numerical User Knowledge Modelling data set [60]. The latter data mining task, clustering, requires utilisation of proximity measures to measure the similarity between data set objects, which can be any of non-hierarchical measures as will be illustrated in Section 3.7 in the next chapter. On the other hand, the type of structured data that is most interesting to study in this work is the hierarchical data structure (described in the next section), which can also fit into a database, but can be organised as a tree-like structure.

2.2 Hierarchical Data

Hierarchical data, as mentioned in the previous section, is data that can be organised as a tree-like structure, for example, READ codes as in Figure 2.2. Thus, data sets comprising such data are known as hierarchical data sets. The latter is defined as "a collection of data records that are connected to each other through links where each data record, in turn, consists of a collection of attributes or variables that each contains only one data value" [113]. This collection of attributes or variables may contain hierarchical attributes, as well as other types of data attributes. The fundamental characteristics of hierarchical structures data are the compactness and dependence on the nature of the data. The hierarchically structured data can also save space, time and simplify operations such as search [106] since it can be based on the principle of recursive decomposition [107] which is similar to the divide and conquer method [128]. Longitudinal databases, patent database, spatial data, XML, and gene expression microarray data, are examples of mixed and hierarchical data sets or data sets with some hierarchical attributes. Many data mining techniques have been applied to hierarchical data sets, such as association and sequential patterns with the THIN database [96] [97], supervised classification with the gene expression microarray data [31] and unsupervised classification (clustering) of XML data [69] [22] [73] [68].

Clustering hierarchical mixed databases is an immense challenge in the world of

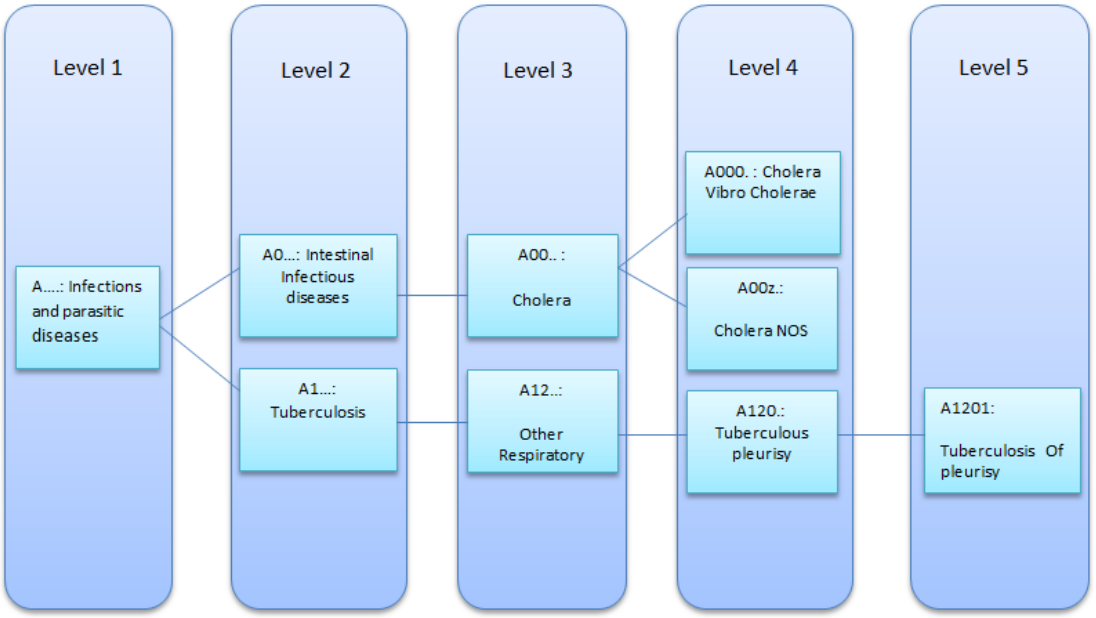


FIGURE 2.2: Example of a branch of the THIN READ code tree reproduced from [99, p.200]

data mining and knowledge discovery. Thus, the clustering data mining task is described, in the next section, in terms of the data mining process and knowledge discovery, including a review on the clustering of two hierarchical databases, NBER US Patent and THIN.

2.3 Data Mining and Knowledge Discovery

One of the Knowledge Discovery in Databases (KDD) process steps is data mining which includes the application of data analysis and discovery algorithms to obtain particular patterns or models over large databases [34]. Some researchers have synonymised data mining and knowledge discovery, and others have identified it as the main step in knowledge discovery. Data mining consists of some major tasks that are applied to large-scale databases, for instance, data description, association, classification, prediction and clustering [17]. Each of these tasks can be processed in the data-mining step of the knowledge discovery process, as shown in Figure 2.3. In the next subsections, the iterative steps of the knowledge discovery process are described for

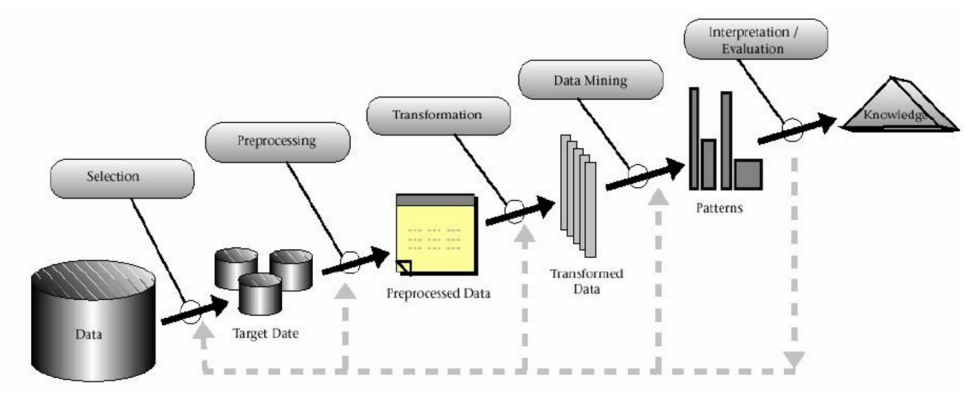


FIGURE 2.3: Data Mining and Knowledge Discovery Process [34]

when clustering data mining tasks are applied to big databases, such as the NBER US Patent and THIN databases.

2.3.1 Data Pre-processing

Data integration, data cleaning, data transformation and data selection are examples of data pre-processing [17]. The data pre-processing prior to the clustering data-mining task consists of *Feature selection or feature extraction*. The process of feature selection is to select an effective subset of original and important features for clustering, while the process of feature extraction is to create useful and notable features from one or more transformations of original features. Both techniques are important for the validation of clustering applications and either or both are used to obtain the most crucial subset of features for clustering [53] [129]. This step is important to avoid high computational complexity and the occurrence of redundant and irrelevant features [15]. For the NBER Patent and THIN databases, important features are selected as data pre-processing steps for clustering in Chapter 5, Section 5.3 and Chapter 6, Section 6.2, respectively.

2.3.2 Clustering Data Mining Task

Clustering is the process of assigning a set of data points or objects into groups called clusters, such that the data points or objects that belong to one cluster are more similar than those belonging to other clusters. Thus, the goal of clustering is to determine

essential groupings in an unlabelled data set [120] [53]. In data mining, clustering and classification are two types of learning methods and both partition a data set into subsets, but the difference between them is that the classification approach is based on supervised learning and is provided with a collection of pre-classified patterns. In contrast, the clustering approach is unsupervised and has no predefined classes, thus, it learns about the data from the data itself [38].

Proximity measures are used in cluster analysis to measure the similarity or distances between patterns in clustering data [15]. Although the similarity and distances measures are conceptually alternatives, there is a little difference in practice. For example, the distance measure is a complement of the similarity measure. The choice of similarity/distance measures depends on the representation of objects and the underlying problem [16]. The most used are the Euclidean distance and correlation coefficients. Proximity measures for both hierarchical and non-hierarchical data, as well as considerations of how to combine them, are reviewed in detail in Section 2.4, and the measures used in this study will be described in Chapter 3, in Sections 3.6 and 3.7.

Clustering is a powerful data mining technique and a useful tool for analysing mixed large data. Thus, this thesis reviews use of the clustering data mining technique for two known hierarchical databases, namely the US NBER patent database [45] and the UK THIN database (https://www.ucl.ac.uk/pcph/research-groups-themes/thin-pub/database).

2.3.2.1 Clustering for the NBER U.S. Patent Database

The NBER U.S. patent data set is considered a very good source of data, containing inventions and studies of technical changes. Recently, a number of research studies have been undertaken using the patent data set surrounding the problem of matching data. For example, identification of a co-authorship network of inventors from the data set has been achieved using a disambiguation algorithm and applying a Bayesian supervised learning approach [68]. The data were prepared in [68] by creating primary data sets named, ASSIGNEE, INVENTOR, PATENT, and classification data from the NBER

database and the US Patent and Trademark Office (USPTO), as well as used data from secondary data sources. These small data sets are merged, parsed, cleaned and then fed into the algorithm for the disambiguation process. This work uses almost the same process of patent data preparation for clustering but applies it only to the US NBER patent data set.

In [59], when clustering the patent data set it was considered as a text document and a clustering algorithm used to solve the problem of high dimensionality since the data set has the characteristics of a highly dimensional structure. Jun et al. [59] highlighted that because of the sparseness and highly dimensionality of the patent data set, it is difficult to cluster the document using traditional clustering approaches, like hierarchical and k-means clustering algorithms. Although traditional clustering approaches can prove challenging, this study used an efficient hierarchical clustering algorithm named BIRCH to cluster the mixed data types and represent the high dimensionality of the patent data set.

The duplicate detection problem has been addressed in the U.S. patent data set to overcome the company name matching problem [80] using string based-measures mentioned in [7]; i.e. the Levenshtein edit distance measure [71] and an agglomerative clustering algorithm like the Greedy Agglomerative Algorithm. The work concentrated on the ASSIGNEE field, which contains the name of the company owning the patent. The main goal of Medvedev and Ulanov's work [80] was to group names relating to the same company where more than two instances of the same company may have misspelling or simply different spellings. To do that, the authors used an approach of dividing the large data set into overlapping subsets similar to the approach used and described in [79].

The clustering techniques have been used in the patent data set using different similarity or distance measures as mentioned above. However, it is necessary to show that hierarchical distance measures are infrequently used while clustering patent data. Although the string based measures used by Medvedev and Ulanov [80] can be considered hierarchical, such as the Levenshtein edit distance measure, these measures are also known to have the drawback of worse case time complexity, at $O(n^4)$ which is high.

Thus, in order to cluster data sets with hierarchical attributes, hierarchical distance measures with low time complexity are better. For example, the *pq*-gram distance metric, which has time complexity equal to $O(n \log n)$ and the Hamming distance metric is a lower bound of Levenshtein edit distance measure. In this study, both metrics are applied and used with the mixed THIN data set to measure the similar distance between the data set records.

2.3.2.2 Clustering for the THIN Database

The THIN database is a known UK medical database with hierarchical attributes, named READ codes (see Figure 2.2), which refer to the medical events codes. Recently, a number of researchers [96] [97] have studied the THIN database using data mining techniques, such as association and sequential patterns. The aim is to detect associations between patient attributes (e.g. age, gender, medical history) and adverse events associated with drugs. Another study using case-control studies [72] of THIN data, aimed to detect the association between different diseases and between diseases and drugs, as well as the validity of the database to support pharmacovigilance. In 2014, Reps et al. [98] also presented a framework (computational meta-analysis) using semi-supervised clustering, which was able to signal many known rare and serious side effects for different drugs. No other data mining technique has since been applied to the THIN database, such as the unsupervised clustering technique. However, in order to cluster THIN data, clustering techniques require distance measures to represent similarities between data objects, such as patients with similar medical data or reported side effects from taking a specific drug. Furthermore, because the THIN database consists of mixed data types including hierarchical attributes, suitable distance measures are required to represent the similarities between the mixed data. Thus, hierarchical distance measures are the measures investigated in reference to such mixed data.

In order to select or design an appropriate clustering strategy, the characteristics of the problem outlined must be investigated first. Once this step is achieved, the outcome of clustering can be hard or fuzzy. Hard clustering partitions the data into groups (each data point belongs to only one cluster); while in fuzzy clustering, each pattern has a

degree of belonging to each of the output clusters. This chapter describes hard clustering. The main categories into which types of clustering are placed are hierarchical and partitional clustering. In each category, many subtypes and variants are applied to diverse types of clustering problems [53] [119].

In the following subsections, vital literature pertaining to both hierarchical and partitional clustering is identified and analysed.

2.3.3 Partitional Clustering

The aim of partitional clustering is to partition a data set into non-overlapping subsets or clusters such that each data object in the data set is in exactly one cluster or subset [119]. This clustering strategy obtains a singleton cluster (e.g. k-means clustering algorithm) from the data in contrast with hierarchical clustering strategies [53]. Partitional methods are advantageous in applications involving large data sets, for which the construction of a dendrogram is not computationally possible. The first process of clustering using partitional algorithms involves selecting a cluster centre, in which all the data objects that lie within the distance threshold are included in the resultant cluster. However, the problem with using a partitional algorithm is the choice of a number of desired output clusters, which either may be decided in advance, or determined as part of a procedure. Partitional techniques usually produce clusters by optimising a criterion function that is either defined locally (on a subset of the patterns like the Euclidean square-error measure criterion) or globally (defined over all of the patterns). Combinatorial search of a set of possible labelling to establish an optimum value of a criterion is clearly computationally prohibitive. In practice, therefore, the algorithm is typically run multiple times from different starting states, and the best configuration is then obtained based on the most frequent clustering (by majority vote) [62].

The squared error criterion is a criterion function commonly used in partitional clustering techniques, which tends to work well with separated and compact clusters. Consider a data set $X = \{x_1, x_2, ..., x_n\}$, containing n patterns, which is to be clustered into k groups. Let us call $C = \{c_1, c_2, ..., c_k\}$ the corresponding set of centres and k_j the

number of patterns in cluster j; also assuming that each pattern can only belong to one cluster. The squared error e^2 can be expressed as:

$$e^2 = \sum_{j=1}^{n} \sum_{i=1}^{k_j} \left\| x_i - c_j \right\|^2 \tag{2.1}$$

where x_i is the i-th pattern in the j-th cluster, c_j is the j-th cluster centre, and $\left\| x_i - c_j \right\|$ is the Euclidean distance between x_i and c_j [53] [62].

2.3.3.1 K-means Algorithm

The algorithms that belong to the partitional clustering category are also known as point-assignment algorithms. The k-means method it the most commonly used and the earliest algorithm established of this type, and it employs a squared error criterion [92]. The aim of the algorithm is to minimise the squared error criterion e^2 in Equation (2.1). Stated informally, the k-means procedure starts simply with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centres until a convergence criterion is met (e.g., there is no reassignment of any pattern from one cluster to another, or the squared error ceases to decrease significantly after some number of iterations). The k-means term refers to the means of the groups represented which are adjusted k times. The k-means clustering algorithm procedure can be described as follows [53]:

1. Choose k points that represent k cluster centres rather than those in different clusters.

2. Assign the points to their closest cluster centroids according to the similarity measure used.

3. Recalculate the positions of the k cluster centres based on current cluster memberships using the formula:

$$c_j = \frac{1}{k_j} \sum_{j=1}^{k_j} x_i \tag{2.2}$$

where j = 1,..., c and as for Equation (2.1), k_j is the number of patterns in cluster j, x_i is the i-th pattern in the j-th cluster, and c is the total number of clusters.

4. Reassign each pattern to the closest cluster centre.

5. Go to (3) if the centroids no longer move (i.e. the convergence criterion not met). The typical convergence criteria means no (or minimal) reassignment of patterns to new cluster centres.

2.3.4 Hierarchical Clustering

To provide a general overview of hierarchical clustering, it is first defined as a process that produces a nested series of partitions sequentially, based on some similarity measures for merging or splitting clusters [53]. A tree-like nested structure termed a dendrogram results from hierarchical clustering and represents the nested grouping of patterns and the similarity levels between them all [129].

For general data to be analysed using a standard method, a proximity measure as described in Section 2.4 is constructed for each pair of objects and consists of similarity and distance measures. Based on its algorithmic structure and operation, hierarchical clustering can be further divided into two approaches: **agglomerative** and **divisive-algorithms** [53] [93].

- The first algorithm uses bottom-up clustering and works by starting with each point in its own cluster, then larger clusters are agglomerated in each successive iteration by combining two smaller clusters when they satisfy established similarity criteria. The representations of the clusters, the decision as to which two clusters can be merged, and the decision to stop combining clusters are decided by the chosen agglomerative algorithm [92].

- The second algorithm is the divisive clustering approach, or top-down approach, which works in the opposite direction. Initially, all the objects are merged into a single cluster, after which they are gradually divided. In the cluster, the object

with the greatest dissimilarity from other objects is then separated to form a splinter group, and the remaining objects in the original cluster examined to ascertain whether additional elements should be added to the splinter group [40]. The result of the first step will then be two clusters. Next, the largest distance between observations in the same cluster referred to as the diameter of each cluster is computed to see which one is greater. The steps above are repeated with the cluster that has the greatest diameter until there are n singleton clusters.

2.3.5 Distance Measures for Hierarchical Clustering

Different methods can be used to define the distance for separate clusters for both the hierarchical clustering approaches (agglomerative and divisive approaches); such as the single linkage, complete linkage, average linkage and minimum-variant (Ward) algorithms [53] [55] [127]. In single linkage clustering, the distance between two clusters is taken as the minimum of all possible pairwise distances (nearest neighbour). This method finds the two objects separated by the shortest distance and places them in the first cluster. Then the next shortest distance is found, and either a third object then joins the previously formed cluster, or a new two-object cluster is formed. This process continues until all objects are in one cluster. In contrast, the distance between two clusters in complete linkage clustering is taken to be the maximum for all possible pairwise distances (furthest neighbour). In average linkage clustering, the distance between two clusters describes the average of the pairwise distances between two elements, one from the first cluster and the other from the second. The minimum-variant algorithm is distinct from all other methods, because it uses an analysis of variance approach to evaluate the distances between clusters. In short, this method attempts to minimise the sum of the squares of any two (hypothetical) clusters that can be formed at each step. Other types are centroid clustering, median clustering and Ward's hierarchical clustering [127].

2.3.6 Distinction Between the Partitional and Hierarchical Clustering Approaches

In both agglomerative and divisive clustering, the membership of an object or cluster is fixed once the cluster that it belongs to has formed during the clustering process. This means that once a cluster is formed, it cannot be split and it can only be combined with other clusters [40]. However, partitioning methods such as the *k*-means clustering algorithm can reconsider cluster assignments at every stage. This is an important distinction between hierarchical and partitioning (non-hierarchical) clustering strategies. Another important distinction is that, in hierarchical clustering procedures, the data is clustered regardless of type, whilst partitioning clustering requires that data be summarised by a set of representative entities, e.g. centroids in *k*-means clustering. For instance, there is no natural centroid underlying categorical and hierarchical data, and so the data needs to be modified while clustering. In contrast, hierarchical clustering requires a distance measure only and is applicable to many data types including categorical and hierarchical data. The hierarchical databases used in this thesis, such as NBER US Patent database and the THIN database, include multiple data types like, numerical, categorical and hierarchical. Thus, in order to cluster such data sets, hierarchical clustering algorithms could be a suitable method for grouping the data using appropriate distance measures that can measure the similarity between patterns of different data types.

Generally, partitional clustering is faster than hierarchical clustering [91] [53], but the disadvantage of it is that it can often be difficult to know how many clusters there should be, resulting in a considerable amount of repeated analysis. Another drawback of conventional clustering is that it can be very sensitive to the choice of initial cluster centres, so there is a need to try different approaches to assess their impact [95] [21]. However, the hierarchical clustering requires only a similarity measure and does not need input parameters.

2.3.6.1 The Hierarchical BIRCH Clustering Algorithm

BIRCH is a clustering algorithm that can be used to perform hierarchical clustering over very large data sets. It also has the ability to cluster multi-dimensional metric data points incrementally and dynamically [131] [53]. The advantage of this algorithm is that it produces the best quality clustering for the given memory and time constraints [131]. In most cases, BIRCH only requires a single scan of a database. In addition, Zhang et al. [131] claim that BIRCH is the "first clustering algorithm proposed in the database area to handle noise (data points that are not part of the underlying pattern) effectively comparing to other hierarchical clustering algorithms". The core of the BIRCH clustering algorithm is the CF (Cluster Feature) tree. Each CF in the tree represents a compact sub-cluster, which itself represents a cluster of records. The CF sustains a triple summary of the information about each cluster as a vector of three variables:

Definition 2.1. *Cluster Feature*: "Given N d-dimensional data points in a cluster: $\overrightarrow{X_i}$, where $i = 1, 2, \ldots, N$, the clustering feature (CF) vector of the cluster is defined as a triple: $CF = (N, \overrightarrow{LS}, SS)$, where N is the number of the data points in the cluster, $\overrightarrow{LS}$ is the linear sum of the N data points, i.e. $\sum_{i=1}^{N} \overrightarrow{X_i}$, and SS is the sum of squares of the N data points, i.e. $\sum_{i=1}^{N} \overrightarrow{X_i}^2$" [131, p.3].

The CF tree, as shown in Figure 2.4, is constructed dynamically as new data objects are inserted. In the figure, B is called the branching factor which represents the maximum number of entries that a non-leaf node has and L represents the maximum number of CF entries that a leaf node has (leaf node capacity), each of which satisfy a threshold $T \in [0 - 1]$, where the threshold determines the size of the clusters in the CF tree. Each CF in the figure consists of a number of data records instead of having a single data record. The CF tree is used to guide every new insertion into the correct sub-cluster for clustering purposes. The insertion of the data entry into the CF tree proceeds by identifying an appropriate leaf node starting from the root. Then the entry descends the CF tree choosing the closest child node according to the distance metric being used. The advantage of the BIRCH algorithm is that it scales linearly to larger

sizes of databases, which produce good clusters with a single scan and improves the quality of the clusters with additional scans. The second advantage is the computational complexity $O(n)$, where n is the number of objects.

The drawbacks of the algorithm are that it can handle and work with numerical data only, and it is sensitive to the order of the data records [90], with the result that the algorithm does not provide highly accurate clustering [6]. This work addresses the drawback of handling only numerical data by enabling the algorithm to handle different data type attributes (including the numerical attribute) to produce reasonable clusters when clustering large databases.

BIRCH has been used to solve two real-life problems: one is building an iterative and interactive pixel classification tool [66] [67], and the other is generating an initial codebook for image compression [132].

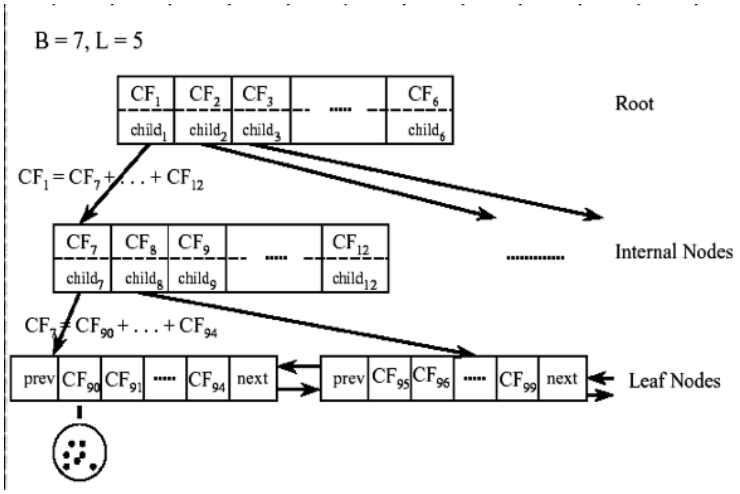


FIGURE 2.4: The Structure of CF Tree taken from [131]

.

2.4 Proximity Measures

The proximity between data objects can be measured according to distance or similarity, where the difference between them must be considered, especially during computations [15]. However, here it is important to outline the fundamental properties of the proximity measures. These measures can be divided into two groups: the metric and

non-metric, or simply called a measure [15]. For both types of measures, and for any given pair of points $x, y \in X$, where X is a set that has the concept of distance $d(x, y)$ between the pair of points, the following must be true [92] [85] [64] [15]

1. $d(\mathrm{x}, \mathrm{y}) \geq 0 \ \ \forall\ \mathrm{x}, \mathrm{y} \in X$. (Non-negativity).
2. $d(\mathrm{x}, \mathrm{y}) = 0 \iff \mathrm{x} = \mathrm{y}$ (Identity).
3. $d(\mathrm{x}, \mathrm{y}) = d(\mathrm{y}, \mathrm{x}) \ \ \forall\ \mathrm{x}, \mathrm{y} \in X$. (Symmetry).

Distances and similarities are metric in the mathematical sense, only if triangular inequality holds. This implies that a metric d on a set X is a function of d: $X \times X \rightarrow \mathbb{R}$ such that for all x, y and $z \in X$:

$$d(x, y) \leq d(x, z) + d(z, y) \tag{2.3}$$

The proximity measure is the main requirement for several data mining and knowledge discovery tasks, such as clustering. For data mining clustering, the basic problem is the choice of proximity measures when summarising the similarity or dissimilarity of patterns. Therefore, a measure of the similarity between two patterns drawn from the same feature space is essential to most clustering procedures. It is necessary for similarity measures to be chosen carefully, because of the variety of feature types and scales involved [15] [43].

In cluster analysis, the data used can be either numerical, categorical or hierarchical. However, it will be more sophisticated when it comprises a mixture of different types of variables. This is because in cluster analysis there needs to be some way of measuring the distance between observations, and the type of measure used depends on the type of data. In this thesis, the distance measures are divided into two new groups, distance measures for hierarchical data and distance measures for non-hierarchical data, which are reviewed in the following subsections.

2.4.1 Distance Measures Used for Hierarchical Data

The database system should consider the structure and the content features of the data objects when searching for similar objects. The key feature of applications utilised with mixed databases, for example, to classify new objects or to cluster database objects, is the need to use an efficient search to locate similar objects in the databases. As mentioned in Section 2.2, hierarchical data is the data type that can be organised as a tree-like structure, such as the READ codes in the longitudinal healthcare (THIN) database. Many different distance measures or metrics have been proposed in the literature to measure the similarity between hierarchical (tree structure) data types [61] [56] [110]. These measures have been applied, for example, to XML documents [82], web site analysis [125] and shape recognition [109]. The general issue affecting the measures proposed for hierarchical structure data is computational complexity, which renders the measures unsuitable for large databases [61].

2.4.1.1 Background on Edit Distance Metric

An editing algorithm named tree-to-tree editing for ordered labelled trees was proposed by Selkow et al. in 1977 [110], and was the first step towards the edit distance metric tree similarity which is used for string matching. The latter distance metric, as described in Section 3.7.1, is the most common and well-studied metric, so far proposed for the comparison of two trees. It is defined as the minimum cost sequence of edit operations (insertion, deletion and rename) that transforms one tree into another, as shown in Figure 2.5. The edit distance metric was first introduced by Tai in 1979 [118], as a generalisation of the well-known string edit distance problem [124], to determine the distance between two ordered labelled trees, T_1 and T_2. This assumes the sizes of the two trees are n and m, such that $n \geq m$ and n_{leaves} and m_{leaves} representing the number of leaves in each tree. Also, n_{height} and m_{height} denote the corresponding height of each tree, which can be as large as n and m, respectively. The time complexity for Tai's algorithm requires $O(n \times m \times (n_{leaves})^2 \times (m_{leaves})^2)$ time and space. This time complexity corresponds to the worst-case complexity $O(n^3 m^3) = O(n^6)$. This edit distance metric was

applied to do automatic error correction for the programming languages as well as to determine the largest common subtrees between two trees. A dynamic algorithm then was proposed in 1989 by Zhang and Shasha [130] to improve the time of the Tai's algorithm. The algorithm as stated by the authors is much simpler and easy to understand comparing to the previously proposed algorithms and can be computed also for the same two ordered labelled trees T_1 and T_2 in $O(n \times m \times \min(n_{height}, n_{leaves}) \times \min(m_{height}, m_{leaves}))$ time and the worst case complexity required $O(n^2m^2) = O(n^4)$. It was also used to find the distance between ordered labelled trees and it is efficiently parallelisable and can be generalised to approximate tree matching problems.

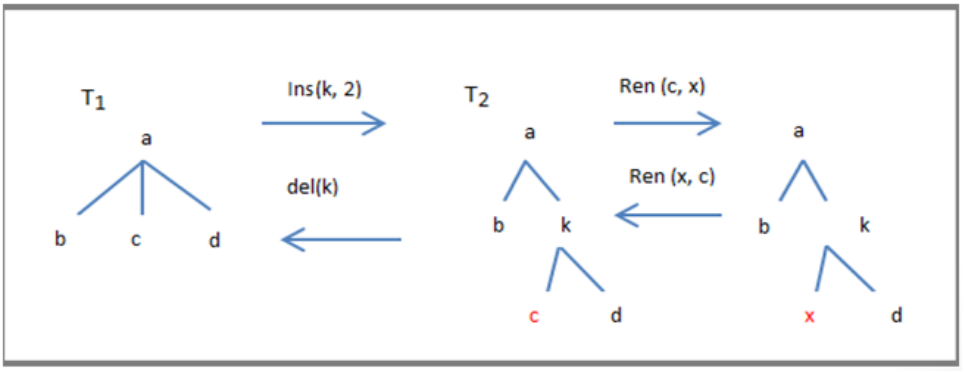


FIGURE 2.5: An example of the Edit Distance Operations

Later in [63] and [18], the edit distance metric was further improved and developed, by producing an $O(m^2 n \log n) = O(n^3 \log n)$ worst case complexity time algorithm and demonstrating that it is optimal under a reasonably wide class of algorithms. Better run-time complexities were found by Demaine in 2007 [25] with the worst case complexity time $O(nm^2(1 + \log \frac{n}{m}) = O(n^3)$ for the same ordered labelled tree T_1 and T_2 determined above. Another edit distance with low complexity named as the bottom-up distance is proposed by Valiente in 2001 [122], where the computation can be achieved in $O(n)$ time, i.e. it can be computed in a time linear to the size of the trees on rooted labelled ordered trees of unbounded degree. The bottom-up distance seeks to find the largest possible common sub-trees from each of two trees, taking the leaf nodes as starting nodes. In 2005, a distance measure named q-gram, which is a sub-tree with q nodes, was proposed for ordered unlabelled trees [84]. A tree consists of (q - 1) different q-gram shapes and represented by a q-gram vector with (q – 1) dimensions. Each dimension counts the q-grams of a specific shape. The q-gram distance between two trees can be

calculated using the L_1 norm distance (Taxicab norm or Manhattan distance) between the q-gram vectors and it has a linear time algorithm to compute the q-grams.

2.4.1.2 Background on pq-gram Metric

As mentioned in the previous section, the principal limitation of the edit distance metric relates to the computational complexity, with worse case $O(n^2)$ time, as well as the data objects with exact matches that represent the same real-world objects often failing due to a lack of common keys. As a result, a new metric was presented by Augsten et al. in 2010 [5], known as the pq-gram distance metric, to be used to approximately match hierarchical data for large sources. It also defines between ordered labelled trees as an effective and efficient approximation of the fanout weighted tree edit distance. The pq-gram metric time complexity is computed in $O(n \log n)$ and $O(n)$ space. The pq-grams for a tree are all its sub-trees that have a specific shape; such that if two trees have many pq-grams in common, they are considered similar. The shape of the pq-grams is specified by two parameters, p and q, and by adjusting them a new distance measure will allow control of the relevance of the tree structure.

Arguably, the empirical results of [5] showed that the pq-grams are a good approximation of the tree edit distance. Then, it has been extended with real world experiments on XML data and proven that the pq-gram distance provides a lower bound for the fanout weighted tree edit distance. Moreover, the normalisation used by Augsten et al. in 2005 [3] was not a metric from the outset, in particular, it does not satisfy the triangle inequality. Later, the authors provided the normalisation to satisfy the triangle inequality in 2010 [5]. In 2012, Augsten et al. [2] proposed the Windowed pq-grams to approximate the tree edit distance between unordered trees, where the sibling order is not relevant, while in 2010 they assumed ordered trees, where sibling order matters. The pq-grams have been extended for consideration in addition to the structural similarity between trees and the string similarity between leaf nodes [102].

2.4.1.3 Rules for Hierarchical Similarity Measures

The design of hierarchical measures is based on the requirements of the underlying problem domain. They may work well for one type of domain and may not work well in another. In [77], six hierarchical measures were used by Mahmood et al. in 2008 to exploit the hierarchical structure of the Network Traffic database called Echidna, and were tested based on some of the rules desired for hierarchical similarity measures. The first three measures are: Generalised Cosine Similarity Measure (GCSM), Optimistic Genealogy Measure (OGM) and Similarity-Based Cosine Metric (SS), which are mentioned in detail in [37] and [58], respectively. The author went through them and discovered they are not suitable based on the following rules established. The rules are:

- Rule 1: The similarity measure should be symmetric. For example, for any two nodes I and J, $S\ (I, J) = S\ (J, I)$.
- Rule 2: The similarity between a node X and its ancestor Y should be greater than the similarity between X and a sibling Z of Y.
- Rule 3: The similarity between a node and its parent should be less than the similarity of a node to itself.
- Rule 4: Siblings at lower levels should be more similar to each other than siblings at higher levels.
- Rule 5: Any two non-sibling nodes at the same level should be at least as similar as one of the nodes and the other node's parent.

Since these rules are important when inventing new hierarchical measures, Mahmmod et al. proposed three new measures: *Hierarchical Similarity Measure1* (HSM1), *Hierarchical Similarity Measure2* (HSM2) and *Hierarchical Similarity Measure3* (HSM3) based on them. He then tested these measures and found that HSM1 is the only measure that satisfies the properties of a hierarchical similarity measure, and is the best for exploiting the hierarchical structure of hierarchical data sets.

Based on these rules, and using an example from the Patent data set, as will be illustrated in Chapter 5, Section 5.2, these measures have all been tested. S_{HSM1}, and by chance, S_{HSM2}, are the two similarity measures among the other measures that follow all the rules. In this study, only S_{HSM1} will be used, in combination with other measures used for numerical and categorical attributes, to measure the similarity between the data records in the Patent data set.

2.4.2 Distance Measures Used for Non-Hierarchical Data

Examples of non-hierarchical data types are numerical, categorical, nominal and others. Many distance measures in the literature have been proposed for use with clustering algorithms for non-hierarchical data types. Most of these distance measures are metric functions, and the most common metrics used in cluster analysis include Euclidean, Manhattan [41], Minkowski [23] and hamming distance metrics [48] [86]. The first three metrics are also geometric measures and are generally used in partitional algorithms. The distance measure Minkowski is a general metric that measures the distance between two multivariate points in continuous data sets. The two most widely used distance measures for continuous data are Manhattan (Minkowski Distance of order 1) and Euclidean (Minkowski Distance of order 2). The key observation regarding the above measures is that they are independent of the underlying data set to which the two points belong [12]. They are easy to test and implement, and are easily generalised to higher dimensions. The drawbacks of the metrics are that they do not work well for image data and document classification, and the results from the Euclidean metric are influenced greatly by those variables with the largest value [86].

In [114], the geometric measures were used with the *k*-means clustering algorithm on dummy data to find similar objects and compare the performance of the algorithm with the basic *k*-means algorithm, which was implemented via the Euclidean distance metric for two-dimensional data. The comparison showed the distortion in *k*-means using the Manhattan distance metric was less than that of *k*-means using Euclidean distance metric. Another comparison was made between the geometric metrics and actual measures, to estimate the distance in spatial analytical models (to estimate distances

from patient residence to hospital)[111]. The results give more accurate measurements for actual distances (road distance and travel time measurements) than geometric metrics. A new distance measure [49] for categorical attributes, based on total mismatches of the categorical attributes of two data records, is proposed for the *k*-modes algorithm. In the literature, two distance measures have been used for mixed data types (numerical and categorical attributes), the weighted sum of Euclidean distance for numerical data with the k-prototypes algorithm, and the proposed distance measure for categorical attributes. In reference to the weighted sum of Euclidean distance, Huang et al. [49] claimed that choosing improper weights might result in biased treatment of different attribute types, thus the weights need to be determined in advance.

Vijay et al. in [123] used the concept of hamming distance for categorical data. The data was transformed into its binary representation during the processing step. They then used proposed algorithm to group data points into clusters. To analyse the performance of their work, the authors experimented with data sets from the UCI machine-learning repository and demonstrated promising results using their proposed algorithm.

Recently, a comparison has been made for [47] between metrics for hierarchical data like Edit distance and *pq*-gram distance and metrics for non-hierarchical data like geometric and hamming distances metrics. The comparison was to find, particularly for the *pq*-gram distance metric, the similarities in patients' data (in the THIN database), for patients who have the same close distances as well as suitable metrics for clustering the entire patient population using the *k*-means clustering algorithm. The results showed the two groups of metrics performed differently, as they represented different data structures. For the first group of metrics, a tree-like structure is used . For the second group of metrics, the data is converted into a frequency table or matrix. Nevertheless, all the metrics could represent some similarities in patient data, as well as discriminating sufficiently well when clustering the patient population using the *k*-means clustering algorithm.

2.4.3 The Technical Details of Combining Distance Measures

Since the efficiency of the distance/similarity measures depends on the size of the data [28], as well as each of them having its own drawback, it is worth searching for an alternative way to show the best performing similarity measures while applying clustering data mining techniques to mixed data sets. Furthermore, a specific technique may be good when used for some measures and poor for others. Thus, using a combination of distance measures to weight performance is provided in many fields, such as time series classification [29], content-based image retrieval [1], Protein Fold Classification [117], the clustering of fiber tracts in Diffusion Tensor Imaging (DTI) [104], and biological networks and gene expression data [46].

In [77], a Combined Distance Function (CDF) was proposed for mixed types of attributes, including numerical with real values, categorical with unranked nominal values and hierarchical attributes for a one-pass hierarchical clustering algorithm. The key issue raised by Mahmood et al.'s work is how to represent a distance function such that it can incorporate hierarchical attributes to help find meaningful clusters, while clustering multidimensional network traffic data using the BIRCH clustering algorithm [131] (the BIRCH clustering algorithm was described in the previous chapter, Section 2.3.6.1)

The CDF as in Equation 2.15 is represented for comparisons between records with three types of attributes, e.g. numerical, categorical, and hierarchical. In this section, the following equations (2.4) to (2.10) have been reproduced from [77] for the numerical and categorical attributes. The notation in the equations has been changed to follow the notation employed in this thesis, the cluster C becomes Cl, the N data points becomes K, a data point in a cluster $x[i]$ becomes $s[i]$ and the centroid $\bar{c}$ becomes cen. For the hierarchical attributes, this work uses a different calculation, since it is different from the hierarchical attribute used in [77]. Thus, to do the calculation for each attribute, the following notation for records and clusters are defined.

$\mathbb{R}$ represents the real number field , the set of Integers is represented by $\mathbb{Z}$ and $\mathbb{R}^d$ is the d-dimensional vector space on $\mathbb{R}$. Cl represents a cluster with K points or records, that is, $Cl = (X_j : j = 1, ..., K)$, where X is a d-dimensional vector representing a point

in the cluster, $X = (s[i] : i = 1, ..., d)$. For example, if the data points are described by continuous attributes, then $X \in \mathbb{R}^d$, $s[i] \in \mathbb{R}$.

- To represent the numerical attribute, a scalar $s[i] \in \mathbb{R}$ is used. For a numerical attribute i in a cluster of K points, the centroid $cen_n[i]$ of it is given by:

$$cen_n[i] = \frac{1}{K}\sum_{j=1}^{K} s_j[i] \tag{2.4}$$

  Also for a numerical attribute i, the distance dis_n between the centroids of two clusters Cl_1, Cl_2 is calculated using the Euclidean distance metric as follows

$$dis_n^i(Cl_1, Cl_2) = \left\| cen_{n_1}[i] - cen_{n_2}[i] \right\| = \left[\left(cen_{n_1}[i] - cen_{n_2}[i] \right)^2 \right]^{\frac{1}{2}} \tag{2.5}$$

- For the categorical attributes, a vector is given as $s[i] \in \mathbb{Z}^c$, where c is the number of possible values that a categorical attribute i can take. For a d-dimensional vector X of categorical attributes, the i-th attribute $s[i]$ is represented as $s[i] = (x_1, x_2 \cdots x_c)$, where $x_k \in (0, 1)$. In a record, each categorical attribute can only take one value at a time. If $x_k = 1$, then $x_j = 0 \; \forall j \neq k$ so that the sum of x_k of a single record is always one, that is,

$$\sum_{j=1}^{c} x_k = 1, \quad x_k \in s[i] \tag{2.6}$$

  To represent the centroid $cen_c[i]$ of a categorical attribute i in a cluster Cl having K points, a histogram of the frequencies of the attribute values is used:

$$cen_c[i] = \frac{1}{K}\sum_{j=1}^{K} s_j[i] = \left(\frac{I_1}{K}, \frac{I_2}{K}, \cdots, \frac{I_c}{K} \right) \tag{2.7}$$

  Where

$$I_k = \sum_{j=1}^{K} x_{k,j}, \; s_j[i] = \left(x_{1,j}, x_{2,j}, \cdots, x_{c,j} \right) \tag{2.8}$$

  and

$$\sum_{j=1}^{c} I_k = K \tag{2.9}$$

The distance between clusters Cl_1, Cl_2 for the single categorical attribute i is given by

$$dis_c^i(Cl_1, Cl_2) = \left\| cen_{c_1} - cen_{c_2} \right\| = \frac{1}{K}\left[\sum_{k=1}^{c}(I_{k,1} - I_{k,2})^2\right]^{\frac{1}{2}} \tag{2.10}$$

For example, by considering an attribute $x[i]$ that represents Fruits, which has values Banana, Apple, Orange. A value $x[i] \in (1,0,0)$ represents the fruit Banana. In addition, if considering a cluster C_1 contains five records described by the Fruit attribute and there are three instances of Banana, two instances of Apple, and zero instances of Orange, then $c_c[i] = (0.6, 0.4, 0)$

- In the case of hierarchical attributes of the network traffic data from [77], equations (2.13) to (2.15) also have been reproduced with some changes to notations to follow the notations employed in this thesis. The calculation of the hierarchical distance function dis_h depended on two main definitions, namely, Prefix of an IP address and CommonPrefix. The generalisation hierarchy has been formed by the set of all prefixes on the domain of 32-bit IP addresses, where a prefix IP=p corresponds to a node at level p in the generalization hierarchy.

  Definition 2.2. Prefix of an IP address. "Given a 32-bit IP address, the p-bit binary prefix of an address IP is defined as the p most significant bits of the address IP and written as IP=p, where $0 \leq p \leq 32$. Note that a prefix IP=p corresponds to a set of 2^{32-p} different IP addresses. For example, 203.190.32.0/24 has the prefix $p = 24$, and it represents a group of 2^8 different IP addresses, all having the first 24 bits in common" [77, p.758].

  Definition 2.3. CommonPrefix and **AggregateIP**. "The CommonPrefix($IP_1; IP_2$) is the longest prefix p such that $IP_1/p = IP_2/p$. The CommonPrefix is equivalent to

$$|path(root, n_1) \cap path(root, n_2)| \tag{2.11}$$

  AggregateIP. The definition of the AggregateIP(IP_1, IP_2) of two IP addresses IP_1 and IP_2 as $IP = p$, such that p = CommonPrefix(IP_1, IP_2), and the least significant 32-p bits of IP are set to zero. Note that it is trivial to

extend this definition to a set of IP addresses. For example, AggregateIP (203.190.32.127; 203.190.32.128) = 203.190.32.0/24" [77, p.758].

Based on these definitions, the hierarchical distance measure can be derived as a complement of the hierarchical similarity measure as the following example reproduced from [77, p.758]:

$$\begin{aligned} &"D_{HSM1}(n_1, n_2) = 1 - S_{HSM1}(n_1, n_2) \\ &= (L - |path(root, n_1) \cap path(root, n_2)|)/L" \end{aligned} \tag{2.12}$$

The centroid cen_h of the hierarchical attribute i in a cluster Cl that corresponds to an IP address is represented by an IP prefix IP/p, which is the aggregate of the IP addresses in that cluster. The hierarchical distance dis_h between two clusters Cl_1 and Cl_2 with centroids $IP_1 = p_1$ and IP_2=p_2 is

$$dis_h^i(Cl_1, Cl_2) = \begin{cases} (32 - p)/24 & p > 8 \\ 1 & p \leq 8 \end{cases} \tag{2.13}$$

where $p = CommonPrefix(IP_1 = p_1, IP_2 = p_2)$.

By using the distance functions dis_n, dis_c, and dis_h, the combination of these for comparisons between records having all three attribute types is simple. Thus, the distance functions $Dis_i(x, y)$ for the numerical, categorical, hierarchical attributes is as follows, based on type of attribute, i:

$$Dis_i(Cl_1, Cl_2) = \begin{cases} dis_n^i(Cl_1, Cl_2), & i \in numerical\ attribute, \\ dis_c^i(Cl_1, Cl_2), & i \in categorical\ attribute, \\ dis_h^i(Cl_1, Cl_2), & i \in hierarchical\ attribute. \end{cases} \tag{2.14}$$

The CDF will be as follows:

$$Dis(Cl_1, Cl_2) = \left[\sum_{i=1}^{d} Dis_i(Cl_1, Cl_2) \right]^{\frac{1}{2}} \tag{2.15}$$

Generally, to find the distance functions for both numerical and categorical attributes, it is possible to use the Equations 2.5 and 2.10, respectively. However, finding the distance function for hierarchical attributes will depend on how the centroid of the hierarchical attribute will be calculated.

In this thesis and in the case of hierarchical attributes, the distance measure can be derived as a complement of the hierarchical similarity measure, like in Equation 2.12, and the centroid for each cluster is represented by the Least Common Ancestor (LCA) of the generalisation hierarchy in each cluster. The Least Common Ancestor between any two nodes in a tree T is defined as the node of greatest depth that is an ancestor of those nodes in T. For example, if the root of a tree is A and its children are B, C and D, the children of B are E and F, then the LCA(B, C) is A and LCA(B, F) is B.

2.4.3.1 Adding Weights to the CDF

Using different similarity measures for a data set might not reveal which similarity measure is best suited for the chosen data set. For this reason, some authors [117] [29] [1] have proposed combined distance functions that consist of distance functions $\{Dis_1, Dis_2, ..., Dis_n\}$ similar to that in Equation 2.15 in the previous section, where n is the number of distance functions, by assigning them as weights $\{w_1, w_2, ..., w_n\}$ to make the combined distance measure applicable to their data sets. For example, using the combined distance function in Equation 2.15 with weights will appear as follows:

$$Dis\,(Cl_1, Cl_2) = \left[\sum_{i=1}^{d} Dis_i\,(Cl_1, Cl_2) \times w_i\right]^{\frac{1}{2}} \tag{2.16}$$

The reason for assigning weights to combined distance measures is to test the performance of the measures, and the impact of one measure among others when measuring the distances between data objects. It can also be done to replace measures that have a weak impact with new measures; for example, replacing an existing hierarchical measure with a new hierarchical measure.

2.5 Summary and Conclusions

This chapter has provided a general overview of both the clustering techniques used within the literature, and the distance measures used for clustering large mixed data sets. The purpose of clustering is to group objects so that those within the same cluster have the greatest similarity, while those belonging to different clusters have high dissimilarity. Furthermore, the clustering process used for large data sets that consist of mixed data types, and different distance measures used and defined in literature were also discussed. As one of the principal objectives of this study is to emphasise the importance of clustering large mixed data sets that include hierarchical attributes, general consideration was given to both kinds of clustering techniques (partitional and hierarchical), and particular consideration given to the different hierarchical distance measures previously developed for representing similarity between hierarchical data objects. To achieve a clearer overview of the measures used when clustering large mixed data sets, a general description of proximity measures was reviewed for both hierarchical and non-hierarchical data, in order to distinguish between them.

Different data sets have also been used herein; two data sets that have mixed data types, including the NBER US Patent data set and the UK THIN database with hierarchical attributes. This chapter presented a brief literature review of these data sets and the data mining fields they were used in (see Section 2.1 and 2.2). The technical details of these data sets and two non-hierarchical data sets will be presented in the next chapter. In order to determine if the distance metrics sufficiently discriminate the data sets while clustering, a partitional clustering algorithm and a hierarchical clustering algorithm have been used.

To conclude, a number of aspects remain open for further investigation. Firstly, it appears that a comprehensive and systematic comparison of distance metrics for hierarchical data has not been carried out yet. This gap in the knowledge can be filled by determining which of the metrics is useful for a hierarchical data set and by comparing different distance metrics for hierarchical and non-hierarchical data. Moreover, this comparison will resolve the first and second research questions in this study.

Secondly, it is also clear from the published literature that the hierarchical patent data set has been clustered using different clustering techniques, while finding the similar distances between data records using hierarchical distance measures has not yet been exploited widely. Thus, using hierarchical distance measures while clustering the Patent data set is one of the goals of this study.

Finally, different data mining techniques have been used on the medical THIN database, as presented in the literature; although, the clustering data mining technique has not yet been explored for this type of mixed data set. Thus, clustering this data is another objective of this study, because both the Patent data set and THIN data consist of mixed data types, and different distance measures will be used for measuring the similar distances between their data objects. For clustering inputs with mixed data types, it is challenging to measure distances using a single measure. Therefore, some authors [29], [1], [117], [104] and [46] have combined distance measures by assigning distance functions with some weights to create a combined distance measure to suit their data set. Using the *pq*-gram distance metric for hierarchical attributes, in combination with distance metrics available for measuring other different data types has not yet been explored which leaves scope for further investigation.

In the next chapter, an overview of the data sets, and background information relevant to this thesis as well as a description of how the data has been collected will be given. It will also describe, the methods used in this thesis when applying distance metrics for both hierarchical and non-hierarchical data, as well as the clustering validation techniques used.

Chapter 3

Methodology

This chapter introduces the research methodology used for this study and explains how it guided the data preparation, analysis and development of theory. Firstly, essential background and fundamental guidelines common to different data sets are provided. The first four sections describe the data preparation of four different data sets. This chapter also includes technical details about the distance measures used for both the hierarchical and non-hierarchical data sets, as well as the validation techniques for clustering, which include internal and external validation indices.

3.1 Background on Lenses Data Set

The original Lenses data set has three classes, numbered 1, 2 and 3 (http://archive.ics.uci.edu/ml/datasets/Lenses). The first class refers patients who should be fitted with hard contact lenses, the second class refers to patients who should be fitted with soft contact lenses, and the last class to patients that do not require contact lenses. The data has 24 instances and four attributes. The first attribute is the age of the patient, which is described by one of three data values: (1) young, (2) prepresbyopic, and (3) presbyopic. The second attribute is the spectacle prescription: (1) myope, (2) hypermetrope, the third attribute refers to if the patient has astigmatic: (1) no, (2) yes. The final attribute is tear production rate, which is classified as: (1) reduced, or (2) normal.

Four of the patients are classified as suitable for fitting with hard contact lenses, and five patients are classified as needing to be fitted with soft contact lenses, and the remaining 15 patients are classified as not to be fitted with contact lenses.

3.2 Background to the Iris Data Set

The Iris flower data set [35] is a non-hierarchical data set (see Chapter 2, Section 2.1). It comprises multivariate data, consisting of 150 data observations and four variables. The 150 data rows represent 150 samples of the Iris flower from three species (classes), named iris setosa, iris virginica and iris versicolor. Each species class contains 50 samples, and four features from each sample were measured; i.e. the length and width of the sepals and petals in centimetres. Based on Fisher's linear discriminant model [35], this data set has become a typical test case for many classification techniques in machine learning, such as the support vector machine [121]. The use of this data set in a cluster analysis, however, is uncommon, since the data set only contains two clusters with rather obvious separation. One of the clusters contains Iris setosa, while the other cluster contains both Iris virginica and Iris versicolor, and is inseparable from the species information used by Fisher [35] [78]. In this work, the data set is used as a validation data, i.e. to determine if the clustering process using the BIRCH algorithm produces the same original Iris classes of data.

3.3 Background to the NBER US Patent Data Set

The NBER U.S. Patent data set [45] consists of detailed information about approximately 3 million patents awarded between 1963 and 1999. The citations for these patents number over 16 million and were added between 1975 and 1999. In addition, other details are included, such as data about inventors, assignees, grants, dates, etc. Patent data sets are considered an excellent source of information for studies into innovation and technical change.

The criteria when patenting an invention, according to USPTO (United State Patent and Trademark Office), are novelty, and non-triviality, and the patented object must have commercial application, although not all inventors meet these criteria. Another limitation is that patent files were not originally entirely computerised because of the very large size of the data. However, fast progress in computer technology has removed virtually all these difficulties, such that now the entire set of data resides in personal computers, and can be analysed with the assistance of standard personal computer software.

The main data set extends from January 1963 through to December 1999 (37 years), and includes all utility patents granted during that period, totalling 2,923,922 patents; this refers to the patent file. The file includes two main sets of variables, the original variables from the Patent Office and the constructed variables created from them. The citations file includes all the citations made by patents granted in 1975-1999, totalling 16,522,438 citations. In addition, the data includes details on inventors, assignees, etc. The patent data itself was procured from the Patent Office, except for the citations from patents granted in 1999, which came from the MicroPatent. The patent file occupies less than 500 MB, and the citation file is about 260 MB. This inventor file contains the full names and addresses of each of the multiple inventors listed in each patent (most patents have multiple inventors, the average being over 2 per patent). Both the names of the inventors and their geographical locations offer a wealth of research data that has yet to be exploited fully. Further details can be found in [45].

3.3.1 Patent Data Pre-processing

For this thesis, the NBER US Patent data set was downloaded from the NBER website `http://www.nber.org/patents` in compressed (.zip) format: ASCII comma separated variable (.csv) file. The data files were then loaded from the source to the Microsoft SQL server 2008 R2, then each file was cleaned by removing all the records with missing values, to generate one complete table per data set. The tables for the data sets used are: Patent, Inventor and Assignee (see Figures A.10, A.11 and A.12 in Appendix A). These small data sets were consolidated and then fed into the algorithm

Results Messages

	PATENT	GYEAR	GDATE	ASSIGNEE	COMPNAME	ASSCODE	LASTNAM	FIRSTNAM	CITY	COUNTRY	NCLASS	CAT	SUBCAT
1	3882149	1975	5604	24425	AMERICAN CYANAMID COMPANY	2	Bernstein	Seymour	New City	US	549	1	14
2	3882151	1975	5604	225545	GLAXO LABORATORIES LIMITED	3	Phillipps	Gordon	Wembley	GB	552	1	14
3	3882151	1975	5604	225545	GLAXO LABORATORIES LIMITED	3	Newall	Christopher	London	GB	552	1	14
4	3882153	1975	5604	323385	KUREHA CHEMICAL INDUSTRY CO. LTD.	3	Seki	Shigeru	Iwaki	JP	554	1	14
5	3882153	1975	5604	323385	KUREHA CHEMICAL INDUSTRY CO. LTD.	3	Sato	Koji	Iwaki	JP	554	1	14
6	3882154	1975	5604	640980	ZOECON CORPORATION	2	Henrick	Clive	Palo Alto	US	554	1	14
7	3882154	1975	5604	640980	ZOECON CORPORATION	2	Siddall	John	Palo Alto	US	554	1	14
8	3882155	1975	5604	454215	PROCTER + GAMBLE COMPANY	2	Wharton	Harry	Colerain Township; H	US	554	1	14
9	3882156	1975	5604	640980	ZOECON CORPORATION	2	Henrick	Clive	Palo Alto	US	558	1	14
10	3882156	1975	5604	640980	ZOECON CORPORATION	2	Siddall	John	Palo Alto	US	558	1	14
11	3882157	1975	5604	640980	ZOECON CORPORATION	2	Henrick	Clive	Palo Alto	US	558	1	14
12	3882160	1975	5604	165265	E. I. DU PONT DE NEMOURS AND COMPANY	2	Fuchs	Julius	Wilmington	US	560	1	14
13	3882173	1975	5604	759416	MONSANTO COMPANY INC.	2	Gash	Virgil	Ballwin	US	562	1	14
14	3884642	1975	5618	33480	APPLIED MATERIALS INC.	2	Benedict	Theodore	Los Gatos	US	117	1	19
15	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Shiba	Keisuke	Minami-Ashigara	JP	430	1	19
16	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Hinata	Masanao	Minami-Ashigara	JP	430	1	19
17	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Ohi	Reiichi	Minami-Ashigara	JP	430	1	19
18	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Sato	Arkira	Minami-Ashigara	JP	430	1	19
19	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Yamasue	Koutarou	Minami-Ashigara	JP	430	1	19
20	3887380	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Kondo	Tokiharu	LATE OF Minami-A...	JP	430	1	19
21	3887381	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Shiba	Keisuke	Kanagawa	JP	430	1	19
22	3887381	1975	5632	211155	FUJI PHOTO FILM CO. LTD	3	Tsubota	Hotohiko	Kanagawa	JP	430	1	19
23	3887392	1975	5632	218370	GENERAL DIODE CORPORATION	2	Tang	Arthur	Framingham	US	134	6	69
24	3887403	1975	5632	361390	MCDONNELL DOUGLAS CORP.	2	Coggins	Dolphus	Berkeley	US	216	1	19
25	3887407	1975	5632	252865	HITACHI LTD	3	Ono	Minoru	Kodaira	JP	438	4	46
26	3887407	1975	5632	252865	HITACHI LTD	3	Momoi	Toshimitu	Tokyo	JP	438	4	46

FIGURE 3.1: A screen shot of the created Patent data table

used for hierarchical clustering (BIRCH clustering algorithm), by creating one table (374,230 data records) containing all the information. Figure 3.1 is an example of a patent data table created in this way.

3.4 Background to the Medical THIN Database.

The THIN database is an example of an electronic health-care longitudinal database. It contains electronic medical records for over 3.5 million patients and 11.5 million patients in total, extracted directly from over 500 General Practices (GPs) throughout the United Kingdom [42]. The number of GPs included within the database is constantly expanding, which results in concurrent expansion of the database over time. These types of databases contain temporal medical data about patients, which accrue over many years [74]. The database contains information about each patient registered within each general practice, including their personal details, such as gender, date of birth, date of registration, and family connections. In addition, the data on all drugs prescribed and the associated sets of symptoms based on the drug prescribed are also included. Blak et al. [9] found that the THIN database offers a suitable representation of the UK, and

that because of the validation procedures applied to the database, it is unusual to find duplicated or missing data.

The structure of the THIN database is detailed in Figure 3.2 [32], which includes all the related tables. The THIN database structure contains raw files, and is simple and flexible. The data is organised first by practice, then by patient ID, and then by dates, to ensure all the information and records pertaining to a single patient are kept together. All the practice data is partitioned into four standard ASCII fixed width text files and three linked files. The four partitions in the file are patient, medical, therapy and additional health data (AHD) (which includes information on preventive health-care, tests and immunisations). The three linked files are postcode variable indicators (PVI), consultants, and staff [32]. The first three partitions are the main tables, shown in the Figures A.13, A.14 and A.15 in Appendix A, and each patient within the THIN database is represented by a unique patient id, named the combid. The patient table contains the attributes of each patient, e.g. year of birth, smoking habits, registration date and date of death if they have died. The medical table stores the patients' temporal medical events data. The therapy table contains records regarding drug prescriptions, each of which contain the combid, referring to the patient being prescribed the drug, a drugcode corresponding to the drug being prescribed, and the date of the prescription. The drugcode is introduced due to database normalisation and has no clear structure. However, each drugcode is linked to three British National Formulary (BNF) codes, corresponding to the main chemical components of the drug [99].

In the medical table, the most important attribute is medical events, which are represented by codes, named READ codes. The READ codes arise from database normalisation [99], and the advantage of using READ codes rather than string descriptions to represent a medical event is that they have a hierarchical structure, which is useful when applying a data analysis. All Read Codes consist of up to seven characters, which may be either a digit 0-9, or a character A-Z, a-z excluding upper and lower case I, V, W and X. The code is hierarchical, from left to right. The medical events become more specific as the level increases, an example of the THIN READ codes is given in Figure 2.2 [99]. The drawbacks of the READ codes are they have redundancies and a

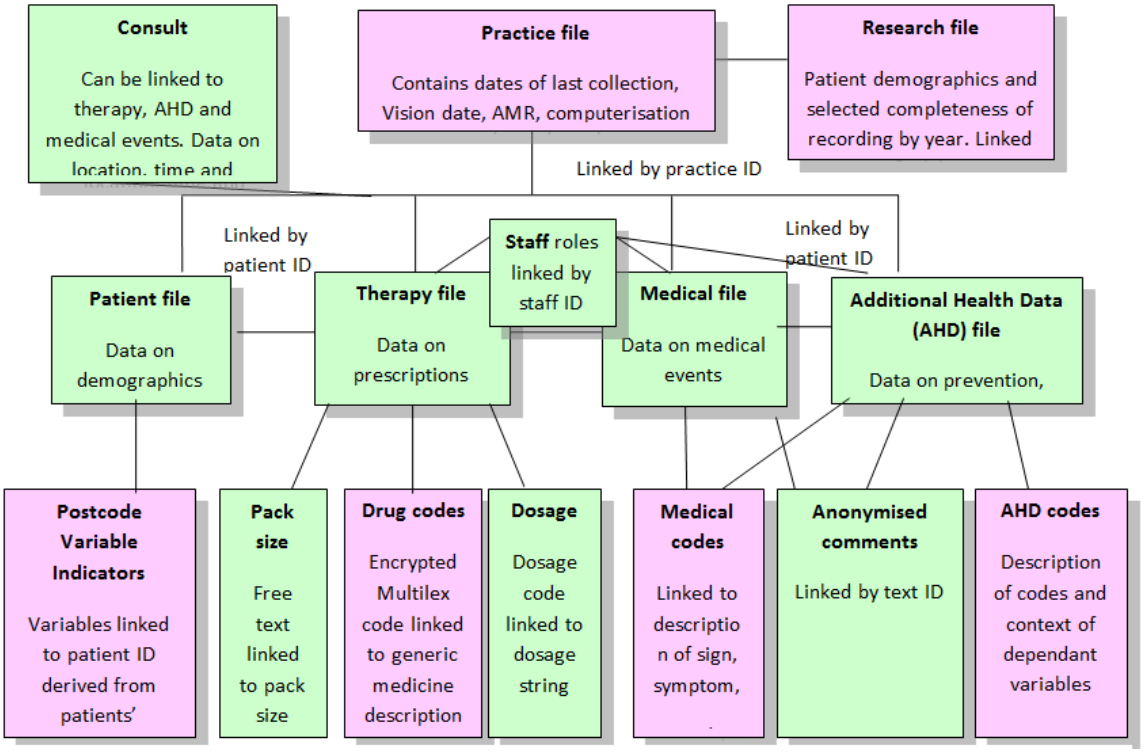


FIGURE 3.2: THIN Database Structure

single medical event might have multiple corresponding READ codes found in widely disparate branches of the READ code tree. There are also problems with READ code usage inconsistencies by medical staff. For example, some staff may frequently enter high-level specific READ codes, while others may have a tendency to enter low-level less specific READ codes.

3.5 The Hierarchical BIRCH Clustering Algorithm

Data clustering has been studied in many areas including machine learning, statistics and databases communities, and involves using different methods. The approaches used in areas like the probability-based and distance-based approaches result in difficulties when dealing with large data sets that must fit the main memory; i.e. the main memory is typically much smaller than the size of the data set required to perform clustering accurately, while keeping the cost of the input/output low.

The BIRCH clustering algorithm [131] was introduced to tackle the above problems by making the time and memory constrains explicit, so that they would be appropriate to large data sets. It also introduces some additional advantages, as mentioned in Chapter 2, Section 2.3.6.1. To understand how the algorithm works, the definition and use of the centroid, radius and diameter are described for a cluster based on the terminology of vector space as follows: Given N d-dimensional data points in a cluster: $\overrightarrow{X_i}$, where $i = 1, 2, \ldots, N$, the centroid $\overrightarrow{X}0$, radius R and diameter D of the cluster are defined as:

$$\overrightarrow{X}0 = \frac{\sum_{i=1}^{N} \overrightarrow{X_i}}{N} \tag{3.1}$$

$$R = \left(\frac{\sum_{i=1}^{N} \left(\overrightarrow{X_i} - \overrightarrow{X}0 \right)^2}{N} \right)^{\frac{1}{2}} \tag{3.2}$$

$$D = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(\overrightarrow{X_i} - \overrightarrow{X}0 \right)^2}{N\,(N-1)} \right)^{\frac{1}{2}} \tag{3.3}$$

R is the average distance from the member points to the centroid and D is the average pairwise distance within a cluster. These are two alternative measures of the tightness of the cluster around the centroid. Furthermore, Zhang et al. in [132] and [131] have defined five alternative distances for measuring closeness between two clusters named: $D0$, $D1$, $D2$, $D3$, $D4$. The first two distances are the centroid Euclidean distance and centroid Manhattan distance between two clusters, and the last three are the average inter-cluster distance, average intra-cluster distance and variance increase distance between the two clusters, respectively. $D3$ is actually D of the merged cluster and $\overrightarrow{X}0$, R and D are treated as properties of a single cluster, and $D0$, $D1$, $D2$, $D3$ and $D4$ describe the properties between two clusters.

The core of BIRCH clustering are the Cluster Feature (CF) and the CF tree (see Definition 2.1 in the previous chapter, specifically Section 2.3.6.1). Figure 3.3 illustrates

the main steps of the clustering algorithm, which consists of four phases: Loading, Optional Condensing, Global Clustering, and Optional Refining. The main task in Phase 1 is to scan all the data and build an initial in-memory CF-tree, using the given amount of memory and recycling space available on the disk. This CF-tree tries to reflect the clustering information of the data set in as much detail as possible, subject to the memory constraints. Once crowded data points are grouped into sub-clusters, and sparse data points are removed as outliers, this phase creates an in-memory summary of the data.

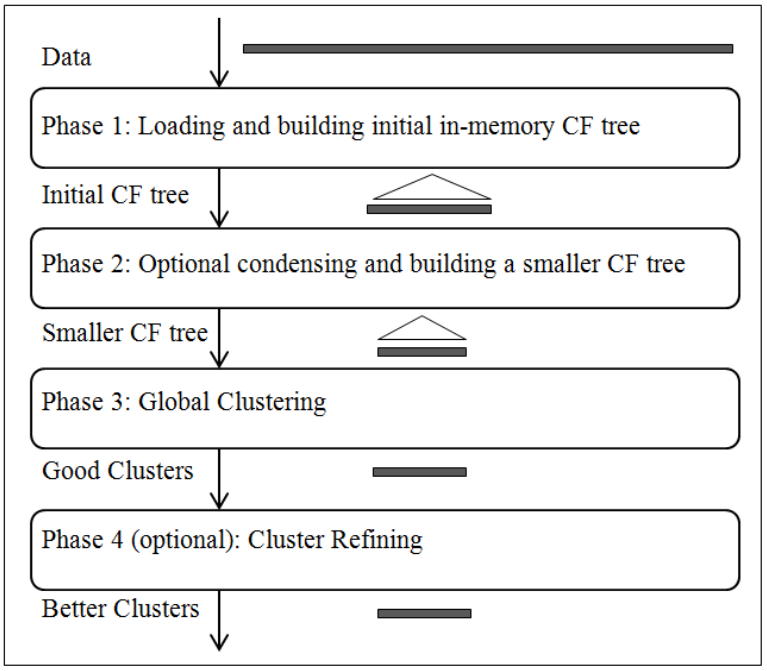


FIGURE 3.3: The BIRCH algorithm overview adapted from [131]

After Phase 1, subsequent computations in later phases will be: *fast* because no I/O operations are needed, and the problem of clustering the original data has been reduced to the smaller problem of clustering the sub-clusters in the leaf entries; *accurate* because the outliers can be eliminated, and the remaining data is described at the finest granularity that can be achieved given the available memory; and *less order sensitive* because the leaf entries of the initial tree form an input order, containing better data locality compared with the arbitrary original data input order.

Phase 2 is an optional phase. It acts as a cushion between Phase 1 and Phase 3 and bridges the gap between them: the leaf entries in the initial CF-tree are scanned to

rebuild a smaller CF-tree, while removing more outliers and grouping the more crowded sub-clusters into larger ones.

After Phase 3, a set of clusters that captures the major distribution patterns in the data are obtained. However, minor and localised inaccuracies might exist because of the rare misplacement problems and the fact that Phase 3 is applied to a coarse summary of the data. Phase 4 is optional and entails the cost of additional passes over the data to correct inaccuracies and refine the clusters further. Up to this point, the original data has only been scanned once, although the tree may have been rebuilt multiple times.

Phase 4 uses the centroids of the clusters produced by Phase 3 as seeds, and redistributes the data points to the closest seed to obtain a set of new clusters. Not only does this allow points belonging to a cluster to migrate, but it also ensures that all copies of a given data set point to the same cluster. Phase 4 can be extended with additional passes if needed which provide better clusters [39]. As a bonus, during this pass, each data point can be labelled with the cluster that it belongs to, if the identification of data points in each cluster is required. Phase 4 also provides the option to discard outliers. Thus, a point that is too distant from its closest seed can be treated as an outlier and excluded from the results.

To explain how the algorithm works, the BIRCH hierarchical clustering algorithm takes each record from a chosen data set and iteratively builds a hierarchical CF tree. The cluster formation performs by adding each data input record X to a cluster C in a leaf node of the CF tree, such that C is closest to X using a distance function or a CDF similar to that described in Section 2.4.3. If the addition of X to C would cause the radius R of the cluster to grow above a threshold $T \in [0 - 1]$, then a new cluster is created. After the insertion, the CF vector describing sufficient statistics of the cluster is updated with X. If a node (sub-cluster) is full, that is, there is no more space in the node for a new CF entry, then the node is split into two, which creates a new node.

Three main parameters should be considered when constructing the CF tree: the first is the threshold, T, which determines the size of the clusters. The tree size is a function of T. The larger T is, the smaller the tree and the faster the algorithm. If the CF tree outgrows the present memory limit, M (the second parameter), then the threshold

is increased, and all the CFs at the leaf level are re-entered into the tree based on the new threshold, T. The third parameter is the branching factor, B, which determines the maximum number of entries to the CF tree at each non-leaf node. This parameter affects the training and testing time and might influence detection accuracy [131] [77].

3.6 Geometric and Hamming Distance Metrics for Non-hierarchical Data Sets

As mentioned in Section 2.4.2 and according to [86], geometric and hamming metrics are easy to test and implement, as well as easily generalised to higher dimensions. They are also metrics, which satisfy all the distance properties (see Section 2.4). Of these, the Euclidean distance metric is still the most used geometrical tool by applied scientists, despite the development of different types of geometry [41]. The following are the non-hierarchical distance metrics used in this thesis for clustering the THIN database and are compared in the next chapter with the hierarchical distance metrics used for the same purpose.

3.6.1 Euclidean Distance Metric

The Euclidean metric measures distance d on the space $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as a distance between any two points in space $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ that one would measure with a ruler [41]. It is the straight-line distance between two points. In one dimension, the distance between two points on the real line is the absolute value of their numerical differences. Thus, if (x and y) are two points on the real line, then the distance between them is given by:

$$d(x, y) = \sqrt{(x - y)^2} \tag{3.4}$$

For two dimensions and in the Euclidean plane, if $x = (x_1, x_2)$ and $y = (y_1, y_2)$ then the distance is given by

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{3.5}$$

In general, for an n-dimensions space, the distance is:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^n} \tag{3.6}$$

where $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n)$ [41].

3.6.2 Minkowski Distance Metric

The Minkowski metric is a p-metric between n-dimensional points $x = (x_i)$ and $y = (y_i)$ defined as:

$$d(x, y) = \sqrt[p]{\sum_{i=1}^{n} |(x_i - y_i)^p|} \tag{3.7}$$

If $p = 2$ it is called Euclidean distance and if $p = 1$ it is called Manhattan or city block distance. If $p = \infty$, then it is called Chebyshev or maximum distance [23].

3.6.3 Manhattan Distance Metric

The Manhattan distance metric is a special case of the Minkowski metric when $p = 1$ [23]:

$$d(x, y) = \sum_{i=1}^{n} |(x_i - y_i)| \tag{3.8}$$

where $x = (x_1, x_2,..., x_n)$ and $y = (y_1, y_2,..., y_n)$.

3.6.4 Hamming Distance Metric

Hamming distance is an upper boundary of the Levenshtein edit distance measure [71], where two strings are the same size. It is used for the detection and correction of errors in digital communications, and is defined as the number of different symbols between two equal length sequences. It is equal to the minimum number of substitutions that transform one string into the other, or the minimum number of errors that could transform one string into the other. For example, the hamming distance between "**t**o**n**e**d**" and "**r**o**s**e**s**" is 3 and between 21**7**3**8**9 and 21**3**3**7**9 is 2.

3.7 Hierarchical Distance Metrics for Hierarchical Data Sets

In order to understand the Edit distance [118] and *pq*-gram [5] metrics computations, some notations and basic concepts must be introduced here. Firstly, the definition of a tree T is a directed, acyclic, connected graph with nodes $N(T)$ and edges $E(T)$, where each node has a maximum of one incoming edge. A forest F is a graph in which each connected component is a tree; each tree is also a forest. Each node has a label, which is not necessarily unique. The nodes of forest F, such as v and w, are strictly and totally ordered such that $v > w$ for any edge $(v, w) \in E(F)$. The tree traversal that visits all nodes in ascending order is the postorder traversal.

In an edge (v, w), node v is the parent and w is the child: $p(w) = v$. A node can have at most one parent, and nodes that share the same parent are siblings. A total order $<$ is defined for each group of sibling nodes. Two siblings s_1, s_2 are contiguous if and only if $s_1 < s_2$ and they have no sibling x such that $s_1 < x < s_2$. A node c is the i-th child of v if and only if $i = |x \in N(T)/(v, x) \in E(T), x \leq w|$. The number of children of v is its *fanout* f_p. A node with no parent is a root node, a node without children is a leaf. Node x is an ancestor of node v if an only if $x = p(v)$ or x is an ancestor of $p(v)$, x is a descendant of v if and only if v is an ancestor of x. A node v is to the left (right) of w if and only if $v < w$ ($v > w$) and v is not a descendant (or ancestor) of w [88] [5].

Example 1 The nodes of tree T (in Figure 2.5 in Section 2.4.1.1) are $N(T) = \{a, b, c, d\}$, the edges are $E(T) = \{(a, b), (a, c), (a, d)\}$. The root of T is $r(T) = a$, and $|T| = 4$. T_a is the subforest of T with $N(a) = \{b, c, d\}$ and $E(a) = \{(c)\}$, $r_L(a) = b$, $r_R(a) = d$

3.7.1 Edit Distance Metric

According to Kialing et al. [61], the definition of the Edit Distance measured between two trees T_1 and T_2 is the minimum cost of all edit sequences that transform T_1 to T_2: Edit Distance(T_1, T_2) = min{$c(S)\backslash S$ and a sequence of edit operations transform T_1

into T_2}. Kialing et al. claimed the advantage of using the edit distance as a similarity measure was that it provided a mapping between the nodes in two trees during the term of the edit sequence (Insertion, Deletion and Relabeling nodes in a tree T). The main drawback of the Edit distance metric is that the cost of all the change operations are performed in the same way and to the same value [36]. Regardless of cost/weights, a key advantage is its property, which comprises all of the edit distances: When x and y share a common prefix, this prefix has no effect on distance. In other words, when $x = uv$ and $y = uw$, then $d(x, y) = d(v, w)$ [108], this allows the speeding up of multiple computations involving edit distance and edit scripts, since common prefixes and suffixes can be skipped in linear time. The example in Figure 2.5 illustrates the edit distance operations (the insertion, the deletion and the renaming). In brief, the Edit Distance distances are equal to the minimum number of edit operations (insert, delete or rename nodes) when converting one tree to another. Each edit operation has a cost 1 and is based on the distance being equal to the minimum cost of converting T_1 to T_2. The Tree Edit Distance Normalisation (TED_NORM) is:

$$TED_NORM(T_1, T_2) = \frac{TED(T_1, T_2)}{(|T_1| + |T_2|)} \tag{3.9}$$

Where $(|T1| + |T2|)$ means the sum of the two trees' nodes.

3.7.2 Pq-gram Distance Metric

Pq-grams distance metric [5] is different from the Tree Edit Distance metric, and it is defined as a measure of similarity between two trees which is based on the number of *pq*-grams that are different between the indices or profiles of the two trees. The *pq*-grams of a single tree are all sub-trees of a specific shape. The specific shape can be adjusted by applying two integer parameters: p and q, where both are greater than 0. After adjusting the two parameters, the new distance measure allows the user to control the relevance of the tree structure and two trees are considered similar if they have many *pq*-grams in common. Augsten et al. [5] proposed the *pq*-gram distance metric

to match ordered labelled trees approximately. They show that the metric is an effective and efficient approximation of the fanout weighted tree edit distance and that it can be computed in $O(n \log n)$ time and $O(n)$ space for two trees of size n. The authors also show that the normalisation of the pq-gram distance satisfies the triangle inequality by comparing the study in [5] to their work in 2005 [3]. To include each node of a tree T in at least one pq-gram, the tree has to be extended with new nodes called dummy nodes or null nodes (*), as shown in Figure 3.4, such that the pq-grams are defined as sub-trees extracted from the extended tree.

In order to form the extended tree, $T^{p,q}$, from a tree T and the two parameters p and q, the dummy nodes should be inserted as follows: add p-1 nodes called ancestors to the root node, insert q-1 nodes (children) before the first and after the last child of each non-leaf node, and add q children to each leaf node of T. Then, from the extended tree, $T^{p,q}$, a pq-gram can be extracted and defined such that a pq-gram is a sub-tree of $T^{p,q}$ and has q leaf nodes and p non-leaf nodes. All leaf nodes of the pq-gram are children of a single node with fanout q, called the anchor node, and they are also consecutive siblings in $T^{p,q}$ [5]. Examples of the pq-gram patterns with different values for p and q and a bag of 2, 3-grams are shown in Figures 3.5 and 3.6, respectively. The bag of pq-grams represents the pq-gram profile or index (as in Figure 3.7) of size $O(n)$ and defined for a tree T with l leaves and i non-leaves is as follows:

$$|P^{p,q}(T)| = 2l + qi - 1. \tag{3.10}$$

Where the leaf nodes are 4, non-leaf nodes are 2 and q is 3, then the size of the pq-gram profile in the Figure 3.7 is 13.

From all above, the pq-gram distance, $dist^{p,q}(T_1, T_2)$, between two trees, T_1 and T_2, where each tree has a pq-gram profile or index, $I_1 = I^{p,q}(T_1)$, $I_2 = I^{p,q}(T_2)$, with $p > 0$, $q > 0$ can be defined as the symmetric difference between the respective indices [5]:

$$dist^{p,q}(T_1, T_2) = |I_1 \uplus I_2| - 2\,|I_1 \upharpoonright I_2| \tag{3.11}$$

The *pq*-gram indices I_1 and I_2 are also known as the bags of Label-tuples for all *pq*-grams of T_1 and T_2, respectively. The bag, also known as multiset, is an unordered collection of elements that are allowed to appear more than once. For instance, in Figure 3.6 and 3.7 the *pq*-grams (a, b,*,*,*) and (a, b,*,*,*) are similar and appear twice and both should be used in the calculation of the *pq*-gram distance. The symbol $\uplus$ refers to the bag or multiset union between I_1 and I_2 and the symbol $\upharpoonright$ refers to the bag intersection between the same indices. Since the same number of different *pq*-grams may be considered as a small change if the two trees are large and as a big change if the trees are small, the normalisation of the *pq*-gram distance considering the tree size has been defined [5]. The normalised *pq*-gram distance is 1 if two trees share no *pq*-grams and 0 for identical trees.

$$dist_norm^{(p,q)}(T_1, T_2) = \frac{dist^{(p,q)}(T_1, T_2)}{|I_1 \uplus I_2| - |I_1 \upharpoonright I_2|} \tag{3.12}$$

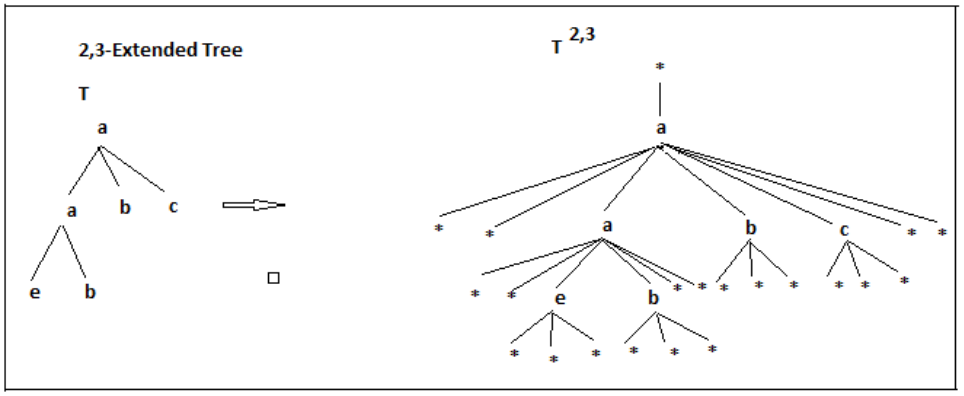


FIGURE 3.4: A graphical representation of tree $T^{2,3}$ and its extended tree reproduced from [4]

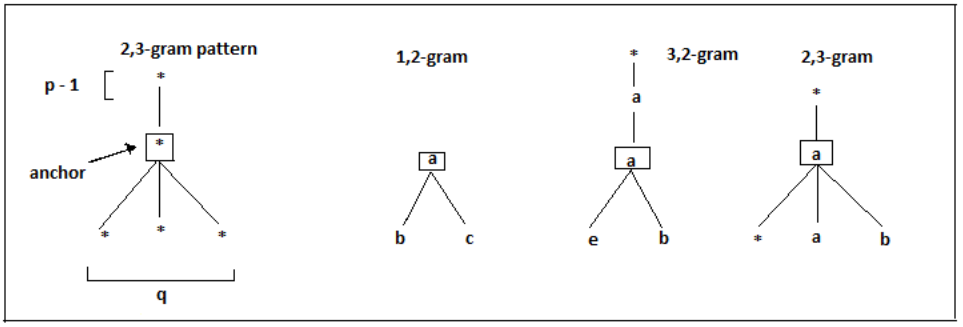


FIGURE 3.5: A *pq*-gram pattern with different examples reproduced from [4]

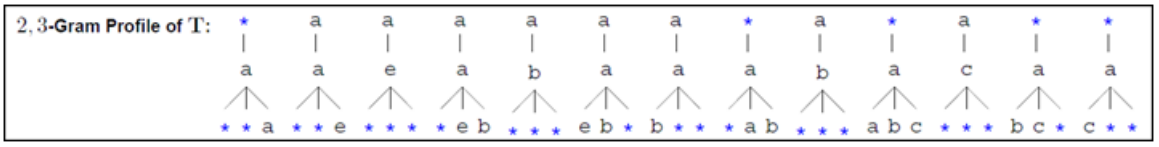


FIGURE 3.6: The 2,3-grams of the example of tree $T^{2,3}$ reproduced from [4]

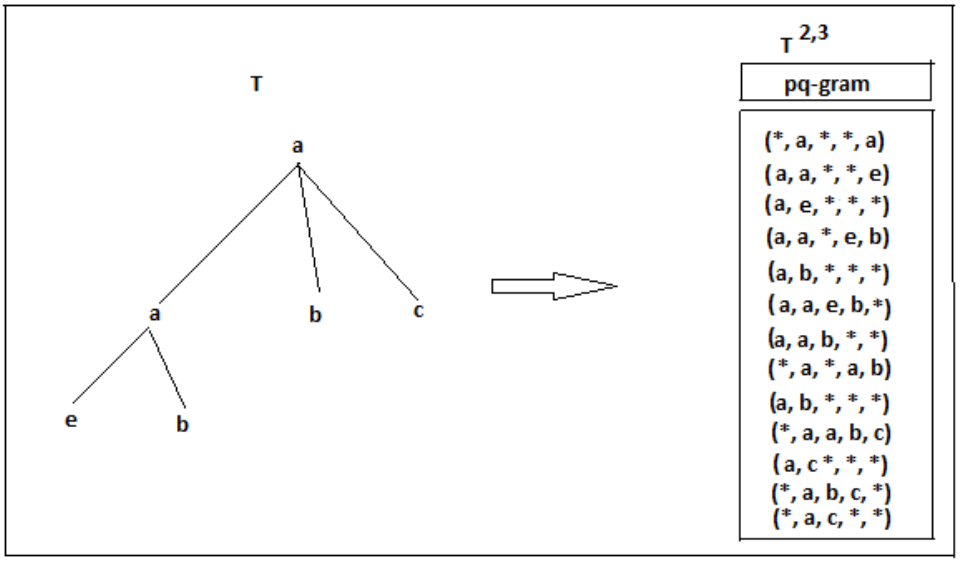


FIGURE 3.7: The 2,3-grams profile of tree $T^{2,3}$ reproduced from [4]

3.7.3 Other Hierarchical Distance Measures

According to [77], six hierarchical measures have been used to exploit the hierarchical structure of Network Traffic data in a clustering approach named Echidna. The first three measures are: Generalised Cosine Similarity Measure (GCSM), Optimistic Genealogy Measure (OGM) and the Similarity-based Cosine Metric (SS), mentioned in detail in [37] and [58], respectively.

The GCSM is a new generation of the traditional cosine similarity measure of the vector space model where the similarity relies on the size of the smallest sub-tree in a hierarchical tree. The OGM measure is also a new generation, but for the Dice's coefficients and computes the similarity between collections of objects rather than between pair of nodes and the SS measure is used to measure the similarity between user sessions of Web access logs. This measure uses the modified cosine similarity metric between two user sessions, i.e. two elements representing URLs at the leaf level.

The other three measures that have been proposed by Mahmood et al. [77] are named as Hierarchical Similarity Measure1 (HSM1), Hierarchical Similarity Measure2 (HSM2) and Hierarchical Similarity Measure3 (HSM3). Based on the hierarchical similarity rules (see Section 2.4.1) and using the example given in Chapter 5 taken from a data set used in this work, it was determined that HSM1 and by chance HSM2 also, are the only two measures that follow the rules that match the requirements of this work. However, only HSM1 will be used in this work, as it is the best hierarchical measure amongst those found in the literature. The description of the last three hierarchical similarity measure, as described in [77], is as follows:

1. HSM1:

   The length of the common segment between the paths from nodes n_1 and n_2 to the root of a hierarchical tree is determined by the hierarchical similarity measure 1 comparing this distance to the maximum depth L which is the depth of the tree. It is as follows:

$$S_{HSM1}(n_1, n_2) = \frac{|path(root, n_1) + path(root, n_2)|}{L} \tag{3.13}$$

For example, in Figure 3.8 the $S_{HSM1}(h, j) = 1/3 = 0.333$, where the numerator is the common paths from h and j to the common root between the two nodes which is at level 1. The depth of the hierarchical tree is 3.

2. HSM2:

   The hierarchical similarity measure 2 measures the similarity between two nodes in a hierarchical tree using the path length between them, rather than the length of the common segment in the paths to the root. It is defined as:

$$S_{HSM2}(n_1, n_2) = 1 - \frac{|path(n_1, n_2)|}{(depth(n_1) + depth(n_2))} \tag{3.14}$$

   Using the example from Figure 3.8, $S_{HSM2}(h, i) = 1 - 0.3333 = 0.6667$

3. HSM3:

   This measure is similar to the previous one, except that in this case, like HSM1, the normalisation is performed using the depth of the tree as follows:

$$S_{HSM3}(n_1, n_2) = 1 - \frac{|path(n_1, n_2)|}{2L}. \tag{3.15}$$

   Using the example from Figure 3.8, $S_{HSM3}(h, l) = 0$.

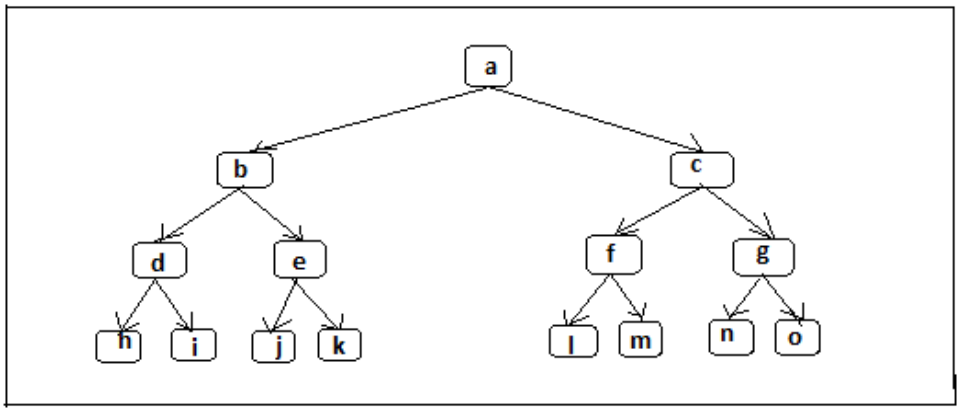


FIGURE 3.8: Example of a hierarchy tree

In this thesis, a CDF was created from both kinds of distance measures (hierarchical and non-hierarchical). The CDF is used for three types of attributes: hierarchical, numerical and categorical attributes. It has been used to compare the data records of

the Patent and THIN data sets, since they consist of mixed data types. The distance for numerical attributes between the centroids of two clusters, and using the Euclidean distance metric is calculated using the Equation 2.5, while the distance for categorical attributes between the centroids of two clusters and using a histogram of the frequencies of the attribute values is calculated using Equation 2.10. In the case of a hierarchical attribute, the distance measure is derived as a complement of the hierarchical similarity measure, for example $D_{HSM1}(n_1, n_2) = 1 - S_{HSM1}(n_1, n_2)$ and the Least Common Ancestor (LCA) of the generalisation hierarchy in each cluster represents the centroid of each cluster.

3.8 Cluster Validation

Cluster validation is considered an important step in cluster analysis, since it may answer some questions regarding the quality of the partition, and even whether there is a better way to partition the data [129] [119]. This step may assist assessment of the clustering procedure output, by using suitable validation indices. Several validity indices were mentioned in [27] [103] that can be used to validate the robustness and separation of clusters resulting from clustering techniques.

In clustering processes, predefined classes are not determined in advance, nor are examples given to show what kind of desirable relations would be valid among the data. This is why it is called an unsupervised process [44]. Therefore, the choice of an optimal number of clusters is one of the most important issues to consider in cluster analysis to assess the resulting clusters produced by a single algorithm [115]. Conceptually, they measure the compactness of each cluster and the separation of cluster centres [65]. In general, clustering result validations may answer questions like “how many clusters are there in the data set?”, “does the resulting clustering scheme fit our data set?”, and “is there a better partitioning for our data set?” [44] [101] [115].

Generally, cluster validation studies are divided into three types, named below [126] [115] [53] [44].

- *External criteria*: Measure performance by comparing a clustering structure to a priori structure. An external criterion can also measure the degree to which data confirms a priori ideas without performing a formal cluster analysis.

- *Internal criteria*: Assess the fitness between the structure and the data, using only the data itself. For example, an internal criterion would measure the degree to which a partition, obtained from a clustering algorithm, is justified by the given proximity matrix.

- *Relative criteria*: Compare two structures and determine which is better in some sense, such as being more stable or more appropriate for the data.

The fundamental idea of these validation criteria (for the first two criteria) is to determine whether results are meaningful or not. It is also to check the validity of the clustering structure, i.e. to ascertain if the clustering structure has occurred by chance (randomly structured or not) or as an artifact of a clustering algorithm [115]. In addition, the indices related to these approaches aspire to measure the degree to which the data set conforms to a pre-specified clustering scheme. The fundamental idea for the third criteria is to look for the best clustering structure from a set of schemes, defined based on pre-specified criteria, as well as not involving statistical tests [44]. The drawback of cluster validity indices is that they cannot work well with arbitrary shaped clusters. The reason for this is that they choose a representative point from each cluster and calculate the distance of these points and some additional parameters based on these points [65] [24].

To verify the quality and correctness of a clustering algorithm's results, and to evaluate its performance, appropriate criteria and methods need to be used. The evaluation procedure must be performed on the clustering algorithm results, since the clusters are not known a priori. Where priori knowledge is not available from a data set, internal validation indices are used. On the other hand, when priori information is required from a data set, external validation indices are used.

In this study, the first two types of validity indices are produced. Although there is no global definition of good clustering [10], these indices are used to measure how good

the clustering results are in comparison with others created using alternative clustering algorithms or others created using the same algorithms but different parameter values [65]. Nevertheless, several evaluation criteria have been applied in the study belonging to the internal categories. These include the Silhouette index [105] and C-index [50], to evaluate the results when clustering the NBER US Patent and THIN databases, the external category, Rand [94] and Jaccard [51] indexes, and to evaluate the results of clustering the Iris data set.

3.8.1 Silhouette Index

The Silhouette Index [105] is a method of interpretation and validation of clusters of data, which describes how well each data object lies within its cluster by providing a graphical representation. Its range is [-1, 1] and the better the clustering the closer the index to 1. Mathematically, it is defined as follows:
For a cluster C_j, the silhouette technique assigned to the i-th sample of a quality measure is $s(i) = (i = 1, ..., m)$, known as the silhouette width. This value is a confidence indicator on the membership for the i-th sample in the cluster C_j and is defined as:

$$s(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \tag{3.16}$$

where a(i) is the average dissimilarity of i with all the other data in the same cluster and b(i) is the lowest average dissimilarity of i to any other cluster. Both are calculated based on the following formula:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & if a(i) < b(i), \\ 0, & if a(i) = b(i), \\ \frac{b(i)}{a(i)} - 1, & if a(i) > b(i). \end{cases} \tag{3.17}$$

and $-1 \leq s(i) \leq 1$

3.8.2 C Validity Index

The C-index [50] [19] is a method of evaluating the clustering models by comparing the pairwise dissimilarities within clusters with those directly observed in the data set irrespective of the cluster membership. Mathematically it is defined as:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}} \tag{3.18}$$

Where S is the sum of distances over all pairs of objects n from the same cluster, and S_{min} is the sum of the n minimal distances when considering all pairs of different instances from the data set. Whereas, S_{max} is the sum of the n maximal distances of the same pairs. The C-index is limited to the interval [0, 1] where the smaller the value the better the clustering; thus this value should be minimised.

3.8.3 Rand Index

Rand index [94] measures the number of pairwise agreements between the set of discovered clusters K and a set of class labels C, as given by:

$$R = \frac{a + d}{a + b + c + d} \tag{3.19}$$

where a denotes the number of pairs of data points with the same label in C, assigned to the same cluster in K; b denotes the number of pairs with the same label, but in different clusters, c denotes the number of pairs in the same cluster, but with different class labels; and, d denotes the number of pairs with a different label in C that were assigned to a different cluster in K. The indexing results in $0 \leq R \leq 1$, where a value of 1 indicates that C and K are identical. A high value for this index generally indicates a high level of agreement between clustered and natural classes.

3.8.4 Jaccard Index

The Jaccard index [51] is used to assess the similarity between different partitions of the same data set, the level of agreement between a set of class labels C and a clustering result K is determined by the number of pairs of points assigned to the same cluster in both partitions:

$$J = \frac{a}{a + b + c} \tag{3.20}$$

where a denotes the count of pairs of points with the same label in C and is assigned to the same cluster in K, and b denotes the count of pairs with the same label, but in different clusters, and c denotes the number of pairs in the same cluster, but with different class labels. The index results in $0 \leq J \leq 1$, where a value of 1 indicates C and K are identical.

3.9 Cluster Visualisation Based on PCA

After clustering, the clusters need to be visualised to demonstrate how they are presented. Various methods are available to display the results when clustering a data set. One of the methods is to use the *Clusplot* function [89] from R software, in which all observations are represented by points on the plot using principal components or multidimensional scaling. The principal components (PCs) are the underlying data structures. They are the directions in which there is the most variance, or where the data is most spread out.

PCA [70] is a statistical procedure mainly used to reduce the dimensionality of a data set with a large number of correlated variables, to retain as much as possible of the variation present in the data set [57]. This can be achieved by transforming it into principal components (PCs), which are ordered and uncorrelated variables such that the first few retain most of the variation present in the original variables [57]. In other words, the transformation aims to rotate the original axes to produce uncorrelated orthogonal axes. The procedure of rotating involves a linear transformation of the original data set and all information is preserved if the variables are included in this procedure [52].

The first component can be described as an orthogonal projection into one dimension, which maximises the variance of the projected points; removing this dimension and its associated variance. The second component is the projection into one dimension, which maximises the variance of the projected new array of points, and so on [30].

The *Clusplot* function represents the data points in a bivariate plot, and the clusters as differently sized and shaped ellipses; Pison et al. first introduced this function in [89] to display multidimensional data. It is a bivariate plot of objects relative to the first two principal components (PC) (i.e., it projects the data onto the first two PCs). When the data is two-dimensional from the start, it displays the data relative to the PCs rather than to the original axes. On the other hand, for the k-dimensional data set $X = \{(x_{i_1}, x_{i_2}, \cdots, x_{i_k}), i = 1, \cdots, n\}$, the dimensions need to be reduced to two dimensions using the PCA, yielding a first component with maximal variance, and then a second component with a second maximal variance perpendicular to the first component. The ellipses resulting from the *clusplot* are based on the average and covariance matrices of each cluster, and their size is such that they contain all the points of their cluster, which is why an object exists on the boundary of each ellipse.

Another way of visualising the resultant clusters in this work is to use the function called Multi-Dimensional Scaling (MDS) [11], which is defined as a means of visualising the level of similarity for individual cases in a data set. It is a technique used in information visualisation, particularly for displaying the information contained in a distance matrix. The number of dimensions of an MDS plot can exceed 2, and when $N = 2$ it optimises an object's location for a two-dimensional scatterplot. In R, the function used is *cmdscale*.

3.10 Summary

All the methodologies used for this study were presented in this chapter. Four data sets were described for later use in subsequent chapters for clustering. The Lenses and Iris data sets are numerical data sets, which are used as validation data. The other data sets are the main research databases in this study; i.e. the NBER US Patent database and

the UK medical THIN database. Both these databases are considered to be mixed hierarchical databases, containing hierarchical attributes in addition to other attributes. The research on these databases aims to measure the similarity between the data records, using hierarchical similarity measures in combination with measures for other non-hierarchical data attributes, while clustering the data.

Since the big hierarchical databases comprise mixed data types (numerical, categorical, hierarchical, etc.) it might be difficult to determine which similarity measures best suit the chosen data sets. This is because different distance metrics are combined for different data types in one distance function. The CDF is intended to measure the similarity between the data records, particularly the hierarchical distance measures for hierarchical attributes.

The *pq*-gram distance metric is contributed in the CDF in two steps to create the New-CDF. The first step is to work along with the CDF and the second step is to replace the other hierarchical distance measures in the CDF. Applying the *pq*-gram to work along with the CDF means adding the *pq*-gram to the CDF, because it is considered an entire database entry, while other hierarchical measures (like D_{HSM1}) are considered as only one column for each data entry (record). In the second step, the *pq*-gram metric replaces the hierarchical distance measures to create a new CDF, New-CDF. Both the CDFs were then tested on the patent and THIN databases to determine the distances between the data records while clustering using the BIRCH algorithm. Experiments and details will be presented for the US NBER Patent database in Chapter 5 and for the THIN database in Chapter 6.

Different validation techniques have been described for the results attained when clustering using the BIRCH algorithm. Two internal validity indices (Silhouette and C index) are used for validating the results of the clustering patent and THIN data, as they do not provide a priori clusters from the data sets. Two external validity indices (Rand and Jaccard) for validating other data sets, that provide a priori information from the sets. The validation indices aim to validate the compactness of the clusters.

After clustering the data, clusters will be visualised using R - which is a free software environment for statistical computing and graphic functions - using *clusplot* and *cmdscale* functions.

Chapter 4

Apply the pq-gram Distance Metric to Different Data Sets

4.1 Introduction

In this chapter, the first contribution is to answer the first two research questions. The aim of this chapter is also to apply the *pq*-gram distance metric to a mixed data set. Before achieving this aim, the validity of the distance metric needs to be examined first by applying it to very simple non-hierarchical data sets, such as Iris and lenses, the hierarchical NBER US Patent data set and then mixed (THIN) data set. The application of the *pq*-gram distance metric to the THIN data set involves a comparison between this metric and other different hierarchical and non-hierarchical metrics.

The THIN data should be converted into a tree-like structure for the hierarchical group of metrics. For the non-hierarchical group of metrics, the data is converted into a frequency table or matrix, and for all metrics the distances are found and normalised. This is to attempt to ascertain the similarities in the patient data and the suitability of the metrics for clustering the entire patient population, using the *k*-means clustering algorithm.

4.2 Apply the pq-gram Metric to Non-hierarchical Data Sets

4.2.1 Lenses Data Set Pre-processing

The data set used for the purpose of the *pq*-gram feasibility study is the Lenses data set https://archive.ics.uci.edu/ml/datasets/Lenses, as shown in Table 4.1 and described in Section 3.1 in the previous chapter.

TABLE 4.1: The original Lenses data set

age	spectacle	astigmatic	tear	Class
1	1	2	2	1
1	2	2	2	1
2	1	2	2	1
3	1	2	2	1
1	1	1	2	2
1	2	1	2	2
2	1	1	2	2
2	2	1	2	2
3	2	1	2	2
1	1	1	1	3
1	1	2	1	3
1	2	1	1	3
1	2	2	1	3
2	1	1	1	3
2	1	2	1	3
2	2	1	1	3
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	2	1	1	3
3	2	2	1	3
3	2	2	2	3

The data set holds information about patients, and the number of patients is the number of data instances, which is equal to 24. However, the data set does not have

a unique ID or code referring to each patient, such that the *pq*-gram metric could distinguish patients' records from one another. The data set places the age attribute in the first column, and there are only three values, each of which will provide a unique root for a *pq*-gram tree creation. This means, there will only be three *pq*-gram trees for the whole data set, as shown in Figure 4.1.

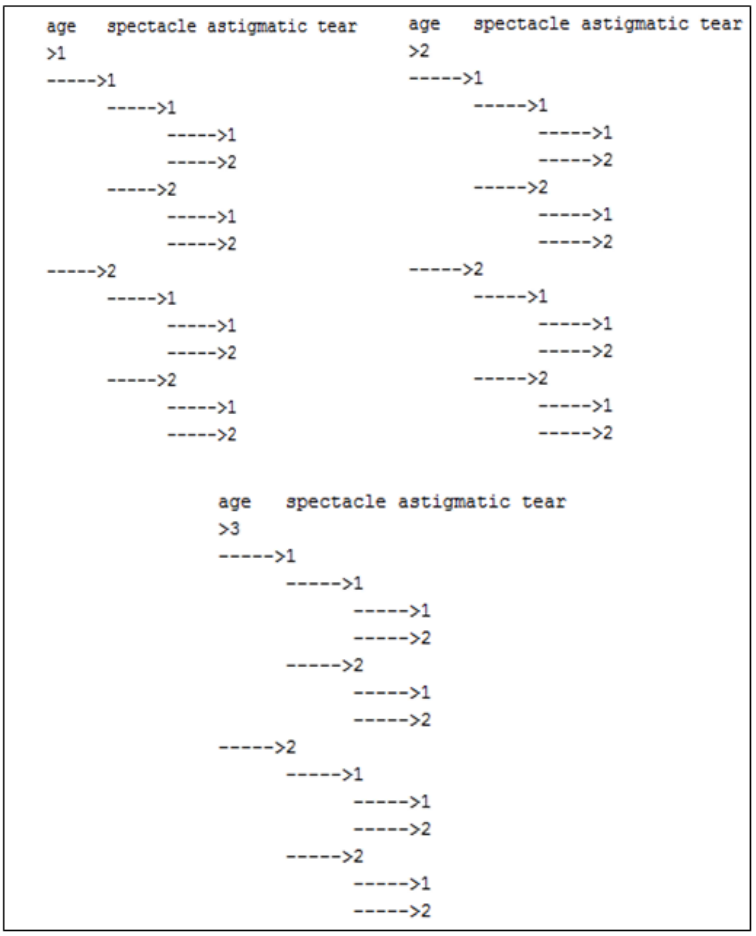


FIGURE 4.1: The three pq-gram trees of Lenses data set

Consequently, in order to cluster the data set using the *k*-means algorithm and *pq*-gram distance metric, there will only be three data points (*pq*-gram trees) to be clustered to calculate the distances between them. The trees are a *pq*-gram tree for all the patients in age category 1 (young), a *pq*-gram tree for all patients in age category 2 (pre-presbyopic) and a *pq*-gram tree for all patients who are in age category 3 (presbyopic). Based on this, the number of observations made is three; because there are three classes for the data set, the number of *k* (clusters) cannot equal the number of observations.

Thus, it can be said that applying the *pq*-gram distance metric to the Lenses data set is not possible if the number of clusters required is three.

4.2.2 Apply the pq-gram Distance Metric to the Iris Data set

The Iris data set, as described in Section 3.2, is a non-hierarchical data set that has only numerical data attributes. However, although the *pq*-gram distance metric is a hierarchical measure used for hierarchical data, it is applied to this data set to measure the distances between Iris data species, and to investigate if the metric can produce the same classes as the original numerical data set. The *pq*-gram distance metric generates small trees from the data set, where each one has a unique root node representing the attribute value from the first column of the Iris data set, and the branches are from data records belong to the same root or attribute values, as shown in Figure 4.2.

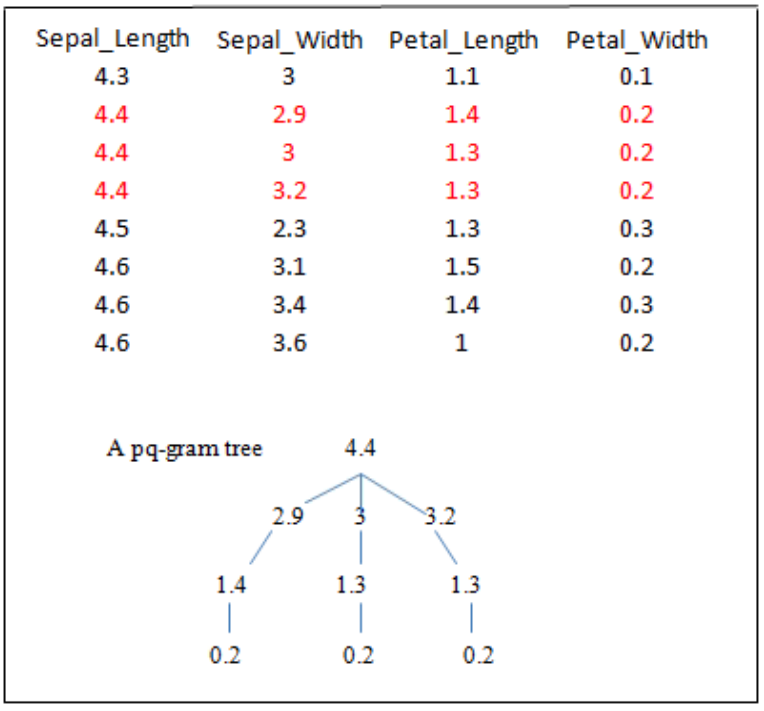


FIGURE 4.2: A Sample of Iris Data and a pq-gram tree example

Each *pq*-gram tree from the data is expanded, and the *pq*-grams are extracted to create the *pq*-gram profile. From each of these two profiles, the distances and their normalisation are calculated using Equations 3.11 and 3.12. By taking two examples of the p and q values, such that the values of $p = 1$ and $q = 3$ and $p = 2$ and $q = 2$, the

results when clustering the Iris data using the k-means algorithm are given in Figures 4.3 and 4.4.

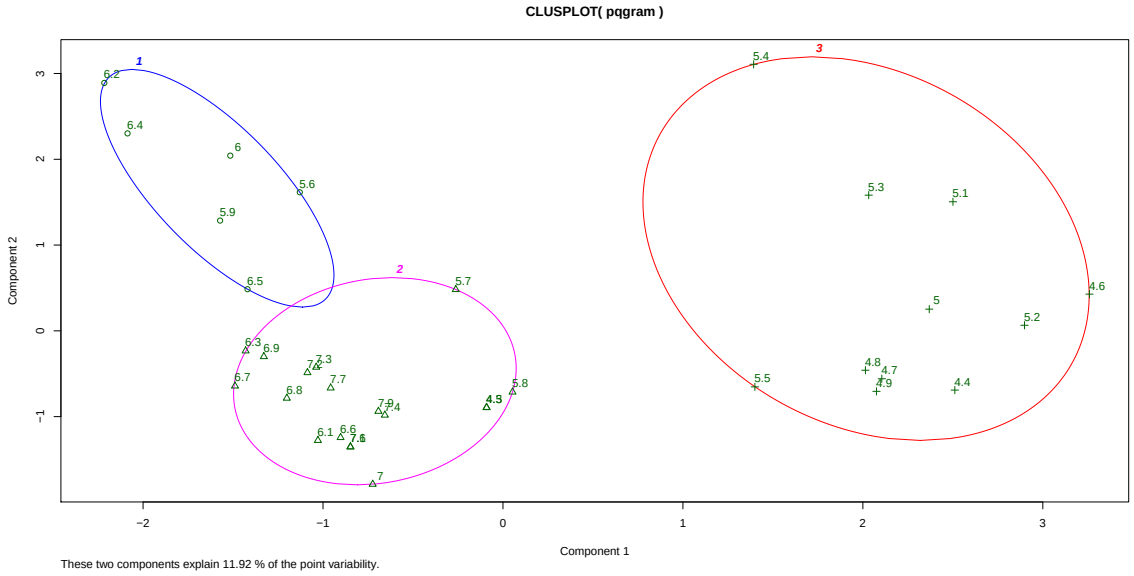


FIGURE 4.3: 1, 3-gram of Iris data using pq-gram metric

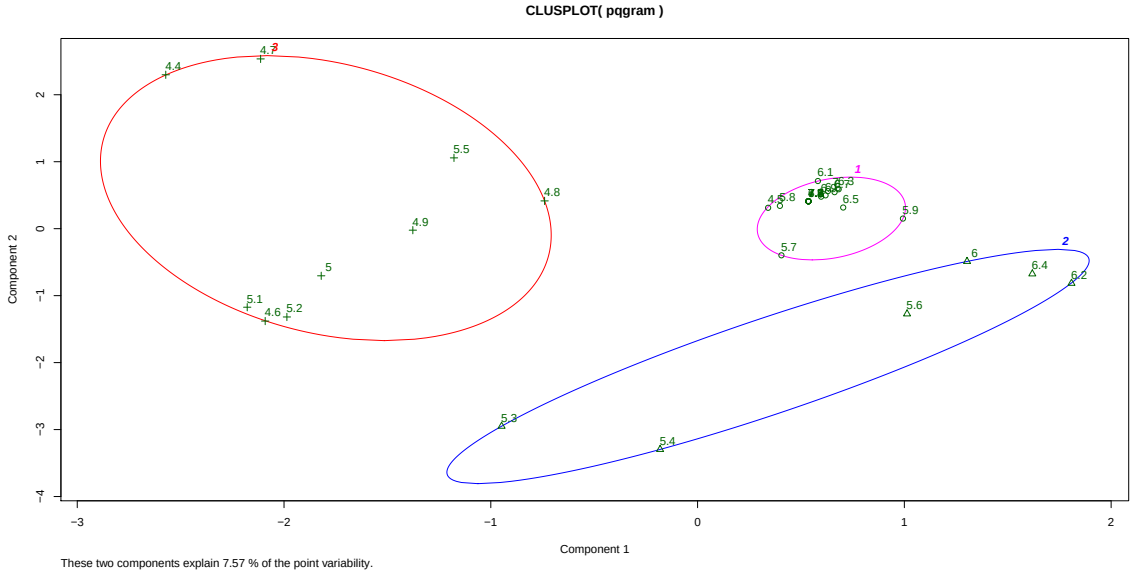


FIGURE 4.4: 2, 2-gram of Iris data using pq-gram metric

In Figures 4.3 and 4.4, the clusters are well separated, but the data points in each cluster are not similar to those in the original classes. This is because of the method used for calculating the distances using the *pq*-gram distance metric, as mentioned above. Each data point in a cluster represents a *pq*-gram tree generated from the original Iris data set (see the example in 4.2). For example in the original Iris data set, the data

records beginning with 4.3, 4.4 and 4.5 are in one cluster, representing the Iris setosa species, while in the new clusters both 4.3 and 4.5 are assigned to both cluster 2 and 4.4 to cluster 3. Thus, based on the comparison between the original classes and the new clusters, the *pq*-gram distance metric could not produce the necessary classes. However, the resultant clusters could be new classes for the Iris data, although it is hard to interpret them. The accuracy of the new clusters when compared to the original classes is shown in the following figure, and reveals low agreement between the existing classes and the new clusters.

```
> confusionMatrix(cl$cluster, iris$Species)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa          0         20        12
  versicolor     48          8         0
  virginica       2         22        38

Overall Statistics

               Accuracy : 0.3067
                 95% CI : (0.2341, 0.3871)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 0.781

                  Kappa : -0.04
 Mcnemar's Test P-Value : 7.674e-09

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 0.0000           0.16000           0.7600
Specificity                 0.6800           0.52000           0.7600
Pos Pred Value              0.0000           0.14286           0.6129
Neg Pred Value              0.5763           0.55319           0.8636
Prevalence                  0.3333           0.33333           0.3333
Detection Rate              0.0000           0.05333           0.2533
Detection Prevalence        0.2133           0.37333           0.4133
Balanced Accuracy           0.3400           0.34000           0.7600
```

FIGURE 4.5: The confusion matrix and statistics for the clusters of Iris data set

It can be concluded from the above experiments that the performance of the distance metric was not good using these data sets, since the *pq*-gram metric is a hierarchical metric designed for hierarchical data sets only. Therefore, in the next section, the distance metric will be applied to the hierarchical NBER US Patent data set to investigate its validity and feasibility, using a sample extracted from the hierarchical data set.

4.3 Apply the pq-gram metric to a hierarchical data set

In this section, the *pq*-gram distance metric is applied to a sample of the Patent data set, consisting of 100 data records extracted from Table 3.1. It aims to investigate if the metric can perform better when clustering this type of data set than when handling a non-hierarchical data set. The sample consists of five attributes: patent ID, last name of the inventor, first name of the inventor, the category that the patent belongs to and the city the patent was awarded in. The data set is clustered into three clusters using the *k*-means clustering algorithm.

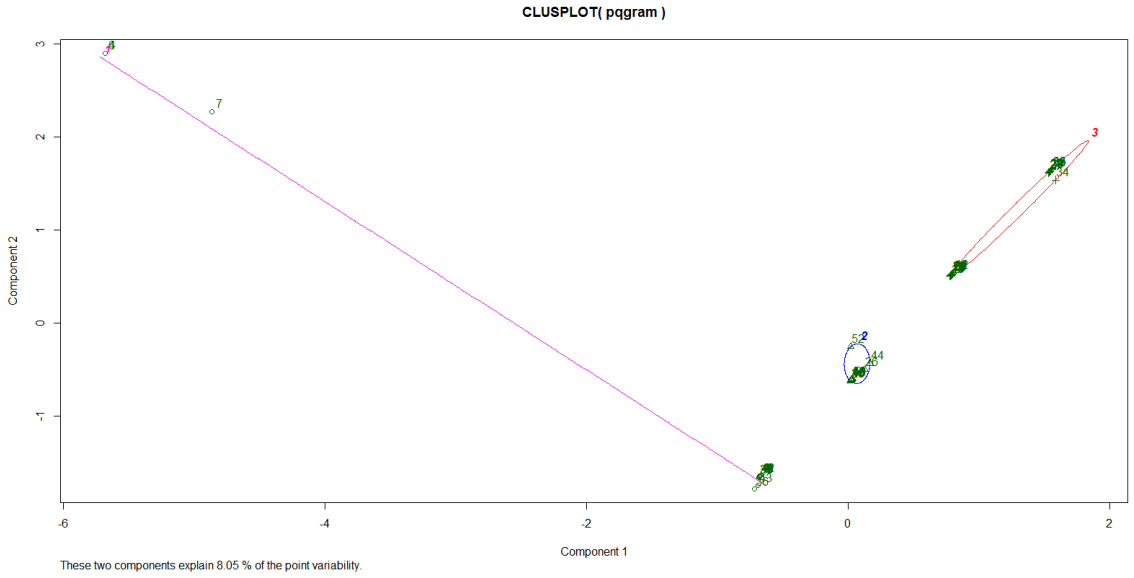


FIGURE 4.6: The clusters of 100 patent data set using pq-gram metric

The data set was converted into 52 *pq*-gram trees, representing those inventors that have the same patent. Since each patent has few records that relate to it, the *pq*-gram tree for the patent will not be big enough to express similarity to other patent trees. Thus, a small number of *pq*-gram trees share similarity, as most were allocated in cluster number 1 in Figure 4.6. For example, two *pq*-gram trees of patent codes, 3882154, 3882156 have a distance similarity of 0.583333. The two trees are associated with the same inventors' names (Henrick Clive and Siddall John), the same patent category number (1), and the same city (Palo Alto). The other clusters contain *pq*-gram trees with no similarity distances between each other, which are equal to 1, and distributed between cluster number 2 and cluster number 3.

From this experiment, it is apparent that the *pq*-gram distance metric produced relatively good results because the data records that have similarity in distances are grouped in one cluster. Also, the results could be better for each patent in the data set if, for example, there were lots of data records creating bigger *pq*-gram trees, which may increase the chances of finding similar *pq*-grams between each two patents' *pq*-gram trees. In the next section, the medical THIN data set, which is a hierarchical mixed data set, will be used to ascertain whether the *pq*-gram distance metric could produce better results (clusters) than when used the patent data set. This data set has the advantage of containing many data records belonging to someone or something, i.e. data records that relate to each patient in the data set.

4.4 Material and Methods for the Comparison

4.4.1 THIN Data Set Preparation

This experiment was trialled on a group of patients aged between 0 and 17 years old, who are considered children, since the clinical trails on this group are very difficult. Thus, an algorithm or new method that could work for this group will be more valuable. The information shown in Table 4.2 was extracted from the THIN data set, providing information about two kinds of drugs that were chosen based on the number of prescriptions and because they have sufficient amount of data. These drugs with some others also have been tested before by Reps et al. in 2013 [100], but for different groups of patients to enable a comparison between different existing electronic healthcare database signal detecting algorithms. Because the same drugs have been used in other works, this means that these drugs are of good quality and can be used in this work as well.

The first drug, DESLORATADINE is from the group 'Antihistamines', and was widely prescribed to treat allergies. The second drug was less heavily prescribed, and is from the family 'Tricyclics' that are used as antidepressant drugs [20]. For the experiments, for the first drug, a sample size of 9949 prescriptions for 30 days of the drug

(representing 988 patients), out of a total 53,995 prescriptions (representing 18,293 patients), were tested to find the similarities between them. For the second drug, all the prescriptions (1172), for 30 days of drug use (representing 42 patients), were tested.

TABLE 4.2: A subset of information from the database for two kinds of drugs

	DESLORATADINE	DOXEPIN
All drug's codes in THIN data set	6	15
All prescription	358,768	72448
All patients	81,000	6152
All prescription(0-17 years)	53,995	2014
All patients (0-17 years)	18,293	60
All presc.(0-17) after 30 days	9949	1172
All Patients(0-17) after 30 days	988	42

4.4.2 Distance Metrics Used for the Comparison

The distance metrics used in this chapter for comparison were six metrics, as described in Chapter 3, Section 3.6 and 3.7, respectively. For the *pq*-gram distance metric, another example is shown here, although using the THIN database.

The *pq*-gram distance [5] is mainly used for computing distances between ordered labelled trees; for a single tree, that is all sub-trees of a specific shape. The parameter values of p and q are the basis for determining the specific shape of the *pq*-grams. The tree T shown in Figure 4.7 is expanded by inserting dummy nodes (*) to make sure that each node appears at least in one *pq*-gram.

The expansion of each tree is done by inserting $p-1$ before the root node, inserting $q-1$ before the first and after the last child of each non-leaf node, and inserting q nodes into each leaf node (see Section 3.7.2); for example $p = 2$, $q = 3$ in Figure 4.8. After the expansion process, the 2, 3-grams are extracted to produce a list of *pq*-grams. In the example of a single 2, 3-gram, which is given in Figure 4.8, p = (*, a6706022p) is the stem and q = (*,*, 1) is the base. The trees that have a large number of common *pq*-grams are considered more similar than those with less; furthermore, the *pq*-gram distance is used to match hierarchical data from large sources using Equation 3.11, and to ensure the normalisation of the *pq*-gram distances using Equation 3.12.

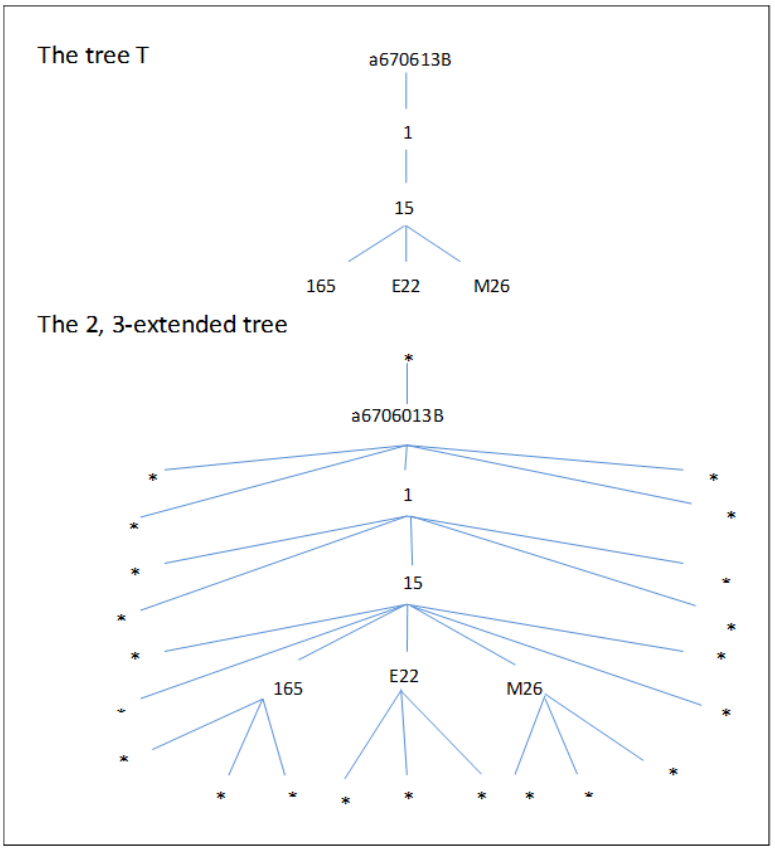


FIGURE 4.7: Example of a tree T and its 2, 3-Extended tree

.

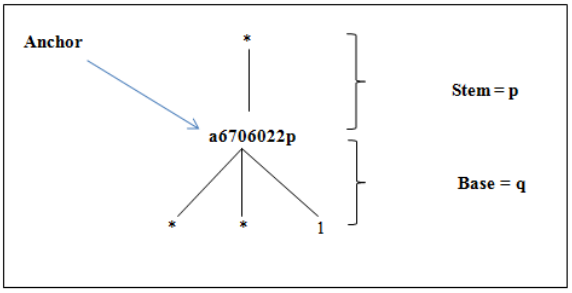


FIGURE 4.8: Example of a single pq-gram from a THIN data tree

.

The pq-gram metric was proposed originally to match similar hierarchical information from autonomous sources that might have a different representation in the sources approximately [5]. The pq-gram metric has the advantage of computational efficiency, and can be computed in $O(n \log n)$ time and $O(n)$ space. Another advantage of the pq-gram distance is that it can be tuned by adjusting the two parameters p and q [116]. The determination of p and q values depends on the underlying semantics of the data. In general, increasing the values of p and q makes the distance between the two

trees more sensitive to the structure of the trees rather than to the data, while decreasing them makes the distance more sensitive to the data. For instance, different values of p and q: for $p = 1$ and $q = 3$ and for $p = 2$, $q = 3$ are used in the experiments and the results are shown in Table 4.4 and 4.6. The results reveal that better distances are obtained when $p = 1$ and $q = 3$, which means that the obtained distances are more sensitive to the data than to the structure of the trees. In other words, the *pq*-grams were more similar to each other in the data that they consist of rather than that in the structure of the trees.

4.5 Data Pre-processing

The THIN data is converted into trees before applying it to the *pq*-gram and Edit Distance metrics, while the data for the geometric and Hamming distance metrics is converted into a frequency table. The data extracted from THIN is based on different patient's attributes, such as unique ID, gender, age when first taking the specified drug, and the medical codes related to the drug. The medical events are chosen at level 3 (the first three digits of the codes read like H33). Figure 4.9 shows part of this information as represented in THIN for three patients, which have unique identifiers in the database (a6706013B, a6706015R, a670601o8).

4.5.1 Pq-gram and Edit Distance Preparation

From the data in Figure 4.9, each patient's records were converted into a tree, as depicted in Figure 4.10, to enable the computation of both *pq*-gram and Edit Distance metrics. For the *pq*-gram metric, each tree is expanded in the same way as shown in Figure 4.7, and for the experiments, ($p = 1$, $q = 3$) and ($p = 2$, $q = 3$) were used. After the process of tree expansion, the *pq*-grams were extracted for each tree. Figure 4.11 shows the 2, 3-grams for the tree in Figure 4.10. The *pq*-gram distance between the two trees is created by all the common *pq*-grams between them, and computed using Equation 3.11; while the calculation of the distances for the Edit Distance is performed by inserting, deleting

combid	sex	age	medcode	description
a6706013B	1	15	165	Feels hot/feverish
a6706013B	1	15	E22	Erectile dysfunction
a6706013B	1	15	M26	Acne vulgaris
a6706015R	2	10	168	Tired all the time
a6706015R	2	10	195	Gastric reflux
a6706015R	2	10	197	Epigastric pain
a6706015R	2	10	1B1	C/O - a headache
a6706015R	2	10	1BK	Worried
a6706015R	2	10	1C2	Tinnitus symptoms
a6706015R	2	10	730	Syringe ear to remove wax
a6706015R	2	10	C38	Localised adiposity - fat pad
a6706015R	2	10	F50	Other otitis externa
a6706015R	2	10	F51	Eustachian tube dysfunction
a6706015R	2	10	F58	Ear pain
a6706015R	2	10	F58	Otalgia
a670601o8	1	12	171	Cough
a670601o8	1	12	19C	Constipation
a670601o8	1	12	1A5	Pain in penis
a670601o8	1	12	1B1	C/O - a headache

FIGURE 4.9: Part of the THIN data extracted based on specific attributes

.

or re-labelling nodes to convert one tree into another. The single edit operation cost 1 and the Edit Distance between two trees is equal to the minimum cost or minimum number of edit operations needed to convert one tree into another.

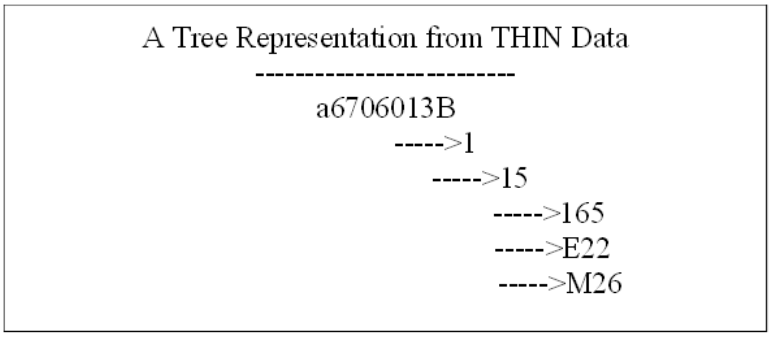


FIGURE 4.10: A tree representation from THIN data

.

4.5.2 Geometric and Hamming Metrics Preparation

The THIN data for the Euclidean, Minkowski, Manhattan and Hamming metrics was converted into a frequency table, as shown in Table 4.3. The table represents how many times each patient had a specific symptom after taking the specified drug, for example,

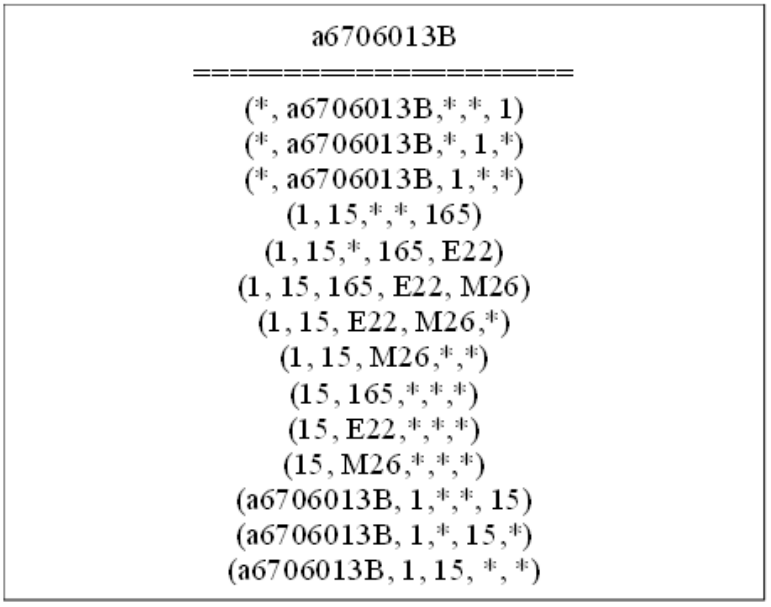


FIGURE 4.11: The 2, 3-grams of a tree T

.

patient a6706015R had symptom F58 twice and symptoms 195 and 168 once. The

TABLE 4.3: The frequency table from THIN data

Patient's ID	The medical events															Patient's ages				Sex
	168	171	195	19C	1A5	730	F58	H17	M0.	M26	N24	N32	SD.	SL.	ZL5	10	11	12	15	
a6706013B	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
a6706015R	1	0	1	0	0	1	2	0	0	0	0	0	0	0	0	1	0	0	0	2
a670601o8	0	1	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	1
a670601yJ	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	2

table also contains additional columns, one for the patient's gender (In THIN, 1 = male, 2 = female) and others for the age classifications of each patient taking the drug. In Table 4.3, the ages of the patients are 10, 11, 12 and 15.

4.5.3 Distances Calculation

The distances, using all six metrics and as applied to the THIN data, are calculated and normalised. The normalisation of the distances is to demonstrate that the small distances close to 0 indicate similar patients, while the large distances near to 1 indicate dissimilar patients. In the case of Euclidean, Minkowski and Manhattan metrics, the data in Table 4.3 was used to calculate the distances using Equations 3.6, 3.7 and

3.8, respectively. For the calculation of Hamming distances, the number of different values between two equal length sequences from Table 4.3 was taken into account. The normalisation of the distances was then calculated using the formula:

$$norm - dist\,(x) = \frac{x - \min\,(x)}{\max\,(x) - \min\,(x)} \tag{4.1}$$

where x refers to the distance between two patients, $\min\,(x)$ is the minimum distance value among all the distances and $\max\,(x)$ is the maximum distance value among all the distances.

Regarding the *pq*-gram metric, the distance between the two trees of patients is defined as the symmetric difference between the two sets of *pq*-grams using Equation 3.11; while the normalisation of the *pq*-gram distances is calculated using Equation 3.12. Conversely, the Edit Distance distances are equal to the minimum number of edit operations (insert, delete or rename nodes) when converting one tree into another. Each edit operation cost is 1, and was based on the distance being equal to the minimum cost of converting T_1 to T_2. Equation 3.9 is used to normalise the tree's Edit Distances. The results when calculating the distances using all six metrics are summarised in 4.4 and Table 4.6 for DESLORATADINE and DOXEPIN, respectively. The tables contain the smallest normalized distances for patients (the most similar data) among the other distances.

The results for the first drug show geometric and hamming metrics could define similar patients as the distances between two patients is equal to zero. In contrast, the *pq*-gram and Edit Distance metrics produced few similar patients, like (a670605Up, a670602uS) and (a67340327, a681001KN), alongside others who have some similarity or close to identical distances between patients. The reason for this relates to the structure of the data, which is a hierarchical tree structure.

On the other hand, the experiment for the second drug also produced a number of similar patients relative to medical events, based on the geometric and hamming metrics, as shown in Table 4.6. Although, for the *pq*-gram and Edit Distance metrics the table showed no similar distances. The reason for this could be the lack of data

TABLE 4.4: Smallest normalised distances for patients taking DESLORATADINE drug

The Normalised Distances							
patient's ID	Euclidean	Minkowski	Manhattan	Hamming	Edit Distance	1, 3-Grams	2, 3-Grams
a670605Up, a670602uS	0	0	0	0	0.25	0	0
a6732002X, a673200WF	0	0	0	0	0.25	0.888889	1
a6732002X, a673201@y	0	0	0	0	0.25	0.888889	1
a673200tm, a673201j7	0	0	0	0	0.25	0.888889	1
a673201@y, 673200WF	0	0	0	0	0.25	0.888889	1
a673201Wt,a6732025y	0	0	0	0	0.25	0.888889	1
a673201wI, a678701pI	0	0	0	0	0.6666	0.888889	1
a67340327, a681001KN	0	0	0	0	0.25	0	0
a677505bO, a677505pe	0	0	0	0	0.25	0.888889	1
a683104@Y, 677505bO	0	0	0	0	0.6666	0.888889	1
a683104@Y, a677505pe	0	0	0	0	0.6666	0.888889	1
a673201wI, a777805mH	0	0	0	0	0.25	0.888889	1
a673402zw, a683105Bk	0	0	0	0	0.25	0.888889	1
a678701pI, a777805mH	0	0	0	0	0.25	0.888889	1
a791600uB,a777806FG	0	0	0	0	0.25	0.888889	1
a7916065T, a777800Gj	0	0	0	0	0.25	0.888889	1

TABLE 4.5: The Shared Medical Events for Patients in Table 4.4

Patient's ID	The medical events For the first patient	The medical events for the second patient	The description of the event
a670605Up, a670602uS	1B8	1B8	Itchy eye symptom
a6732002X, a673200WF	17Z	17Z	Respiratory symptom NOS
a6732002X, a673201@y	17Z	17Z	Respiratory symptom NOS
a673200tm, a673201j7	ZL5	ZL5	Referral to orthopaedic surgeon
a673201@y, 673200WF	17Z	17Z	Respiratory symptom NOS
a673201Wt,a6732025y	740	740	Submucous diathermy to turbinate of nose
a673201wI, a678701pI	H05	H05	Upper respiratory tract infection NOS
a67340327, a681001KN	171	171	Cough
a677505bO, a677505pe	8B3	8B3	Medication requested
a683104@Y, 677505bO	8B3	8B3	Medication requested
a683104@Y, a677505pe	8B3	8B3	Medication requested
a673201wI, a777805mH	H05	H05	Upper respiratory tract infection NOS
a673402zw, a683105Bk	H17, 8B3	H17, 8B3	Hay fever or pollens, Medication requested
a678701pI, a777805mH	H05	H05	Upper respiratory tract infection NOS
a791600uB,a777806FG	H17	H17	Hay fever or pollens
a7916065T, a777800Gj	A78	A78	Verrucae warts or Molluscum contagiosum

TABLE 4.6: Smallest normalised distances for patients taking DOXEPIN drug

The Normalised Distances							
patient's ID	Euclidean	Minkowski	Manhattan	Hamming	Edit Distance	1, 3-Grams	2, 3-Grams
a793901c8,a9910027z	0	0	0	0	0.4761	0.971	0.967
b977401S1,a999104cU	0	0	0	0	0.4	0.923	1
g989501KB,a999104cU	0	0	0	0	0.375	1	1
g989501KB,b990804AL	0	0	0	0	0.5	1	1
b990804AL, a999104cU	0	0	0	0	0.375	1	1

records for each patient regarding the second drug. In other words, the more the data there is regarding a patient, the larger the *pq*-gram tree and then the more chances for the *pq*-gram profile to have the same *pq*-grams with another *pq*-gram profile of another patient's *pq*-gram tree.

To conclude from the tables of both drugs, the *pq*-gram distance metric could not be the metric that produces better results or similar distances compared to the other metrics. The reason could be that the THIN data set contains mixed data that is not all hierarchical (i.e. consists of only one attribute that is hierarchical) and the *pq*-gram is designed for only hierarchical data. In other words, the *pq*-gram metric could produce better similar distances if it uses to measure the similarity for only hierarchical attributes in a mixed data.

4.5.4 Clustering the Distances

The results in Table 4.4 and Table 4.6 show the similarities and closest distances between patients using the previously mentioned metrics. The following step is to use a clustering method to give the first insight into how the data looks and to determine if the *pq*-gram distance metric can represent similar distances better than other metrics. The clustering process will also show whether all the similar distances in Tables 4.4 and 4.6 fall into one cluster, or if they are distributed over all or some clusters. In this chapter, the *k*-means method is used and three clusters are chosen. For the first drug, Figures 4.12 to Figure 4.17 show the clusters of patients using all the distance metrics. Figures 4.18 to Figure 4.23 show the clusters of patients using the same metrics for the second drug. Generally, all the distance metrics produce similarity between the patient data for both DESLORATADINE and DOXEPIN. The geometric and hamming distance metrics provide greater similar data than the hierarchical metrics, as do the number of identical distance values between the patients. Also, the *pq*-gram and Edit distance metrics produced some identical distances, as illustrated in Tables 4.4 and 4.6; although when compared to the geometric and hamming metrics it cannot be considered better than the other distance metrics.

Since the *k*-means algorithm is known to be biased by the starting positions, it needs to be re-run more than once. Consequently, more than one outcome may be obtained. The figures of the clusters represented in this work result from the most frequent clustering (the majority vote, after running the experiment 10 times). In order to distinguish between the clusters, Tables 4.7 and 4.8 describe the number of patients

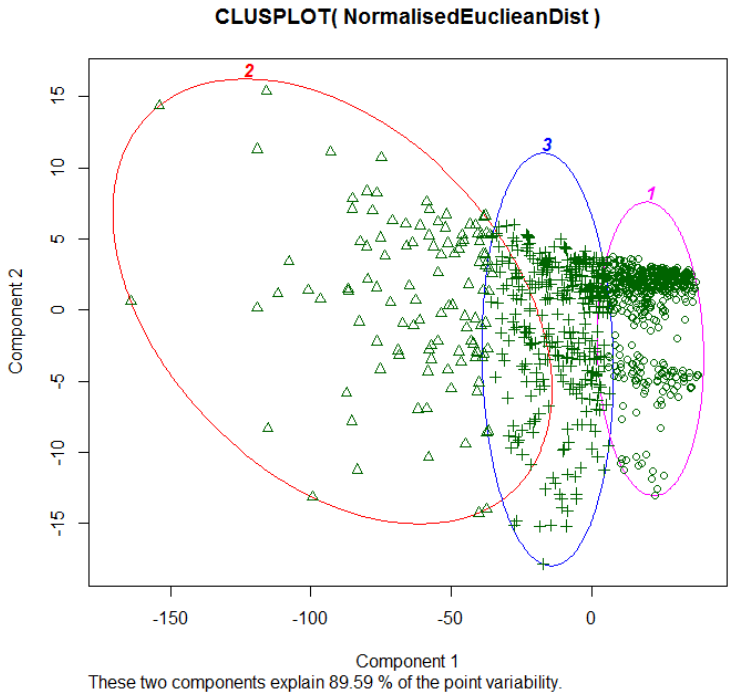


FIGURE 4.12: The clusters of patients using Euclidean metric, DESLORATADINE drug

.

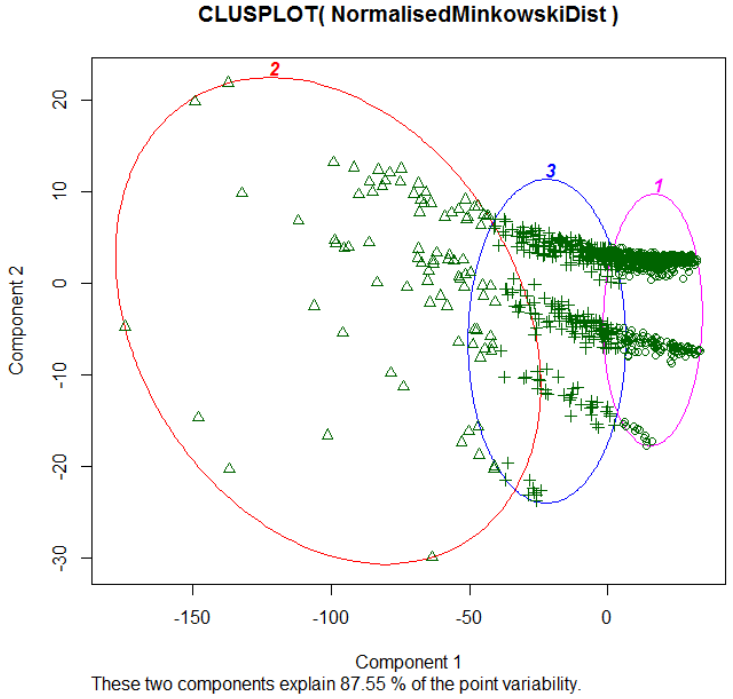


FIGURE 4.13: The clusters of patients using Minkowski metric, DESLORATADINE drug

.

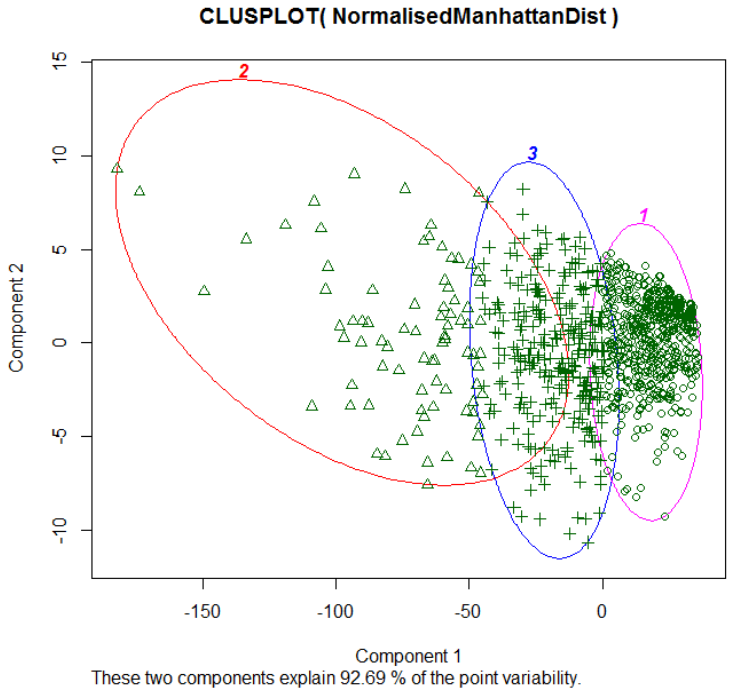


FIGURE 4.14: The clusters of patients using Manhattan metric, DESLORATADINE drug

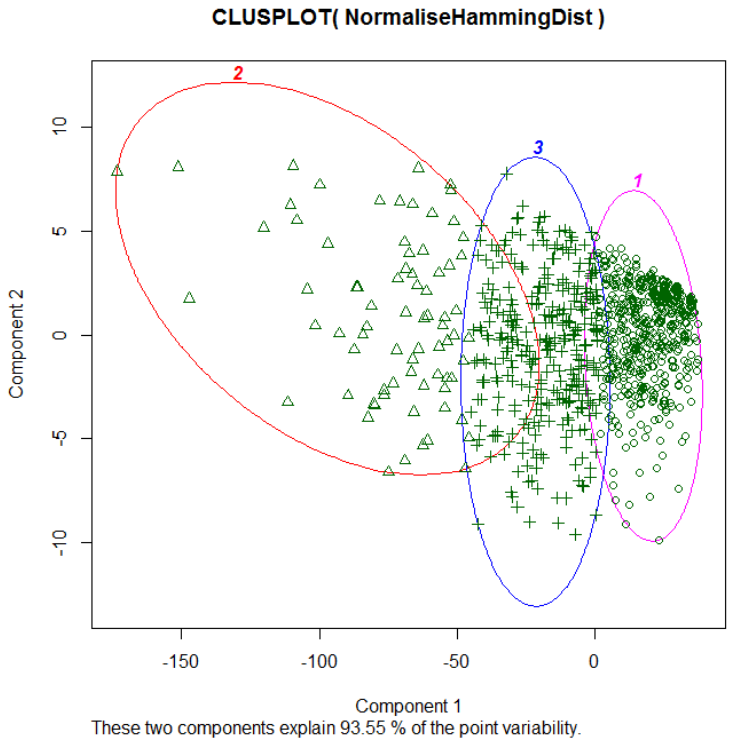


FIGURE 4.15: The clusters of patients using Hamming metric, DESLORATADINE drug

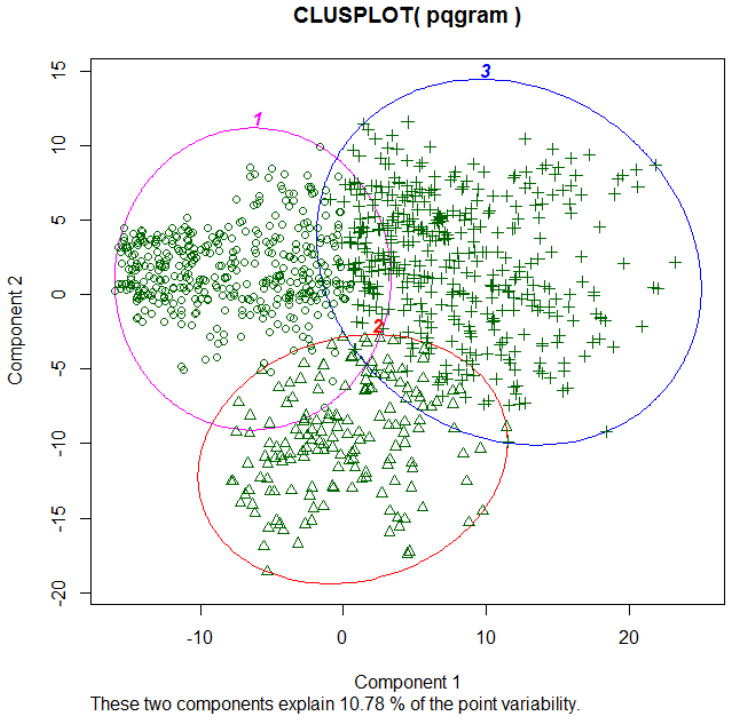


FIGURE 4.16: The clusters of patients using *pq*-gram metric, DESLORATADINE drug

.

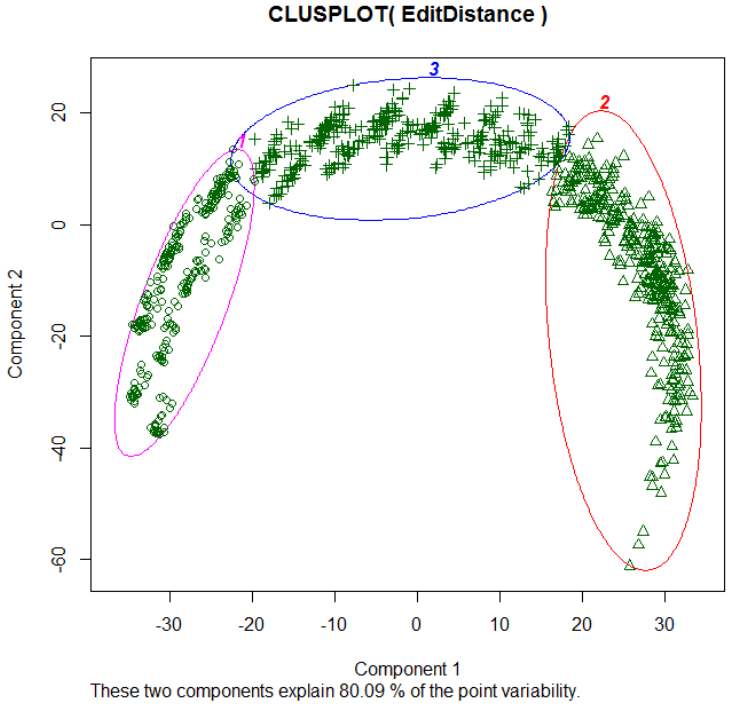


FIGURE 4.17: The clusters of patients using Edit Distance metric, DESLORATADINE drug

.

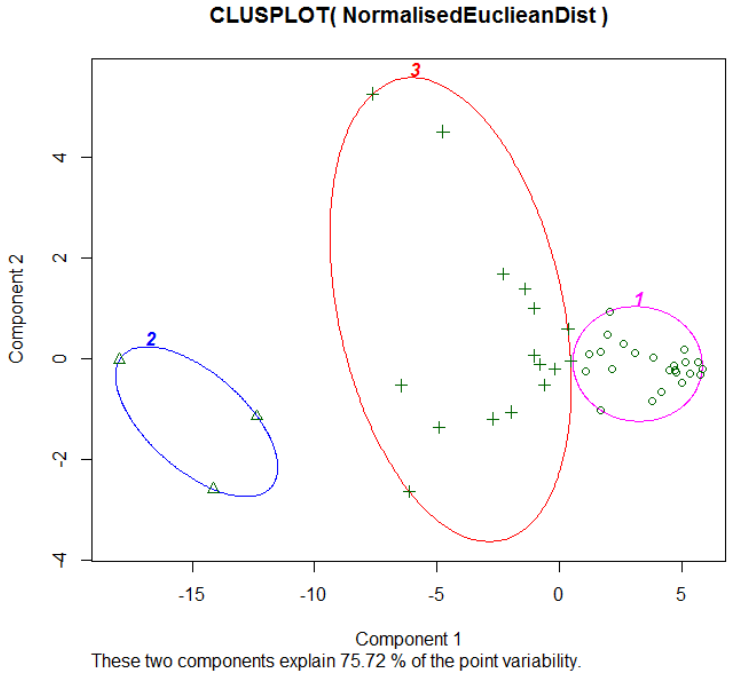


FIGURE 4.18: The clusters of patients using Euclidean metric, DOXEPIN drug

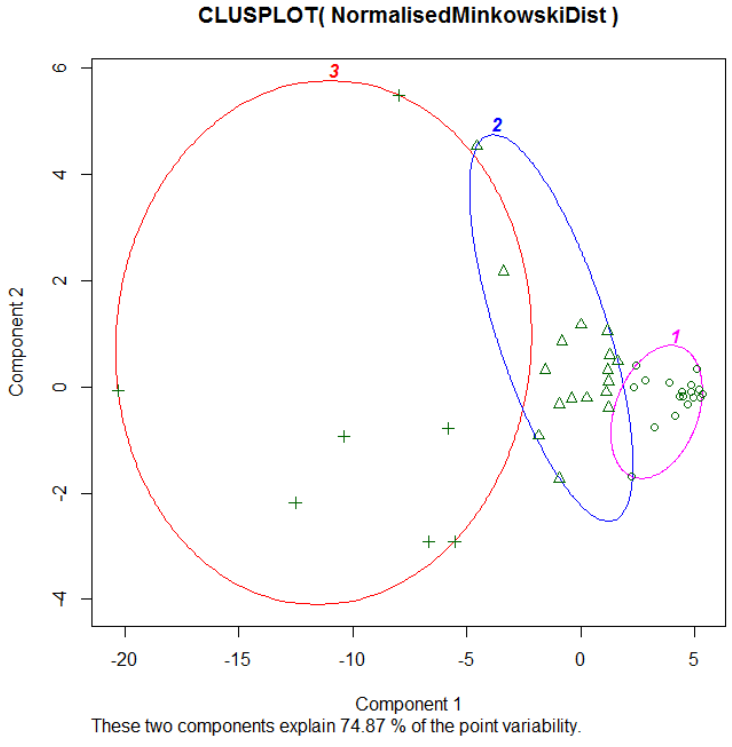


FIGURE 4.19: The clusters of patients using Minkowski metric, DOXEPIN drug

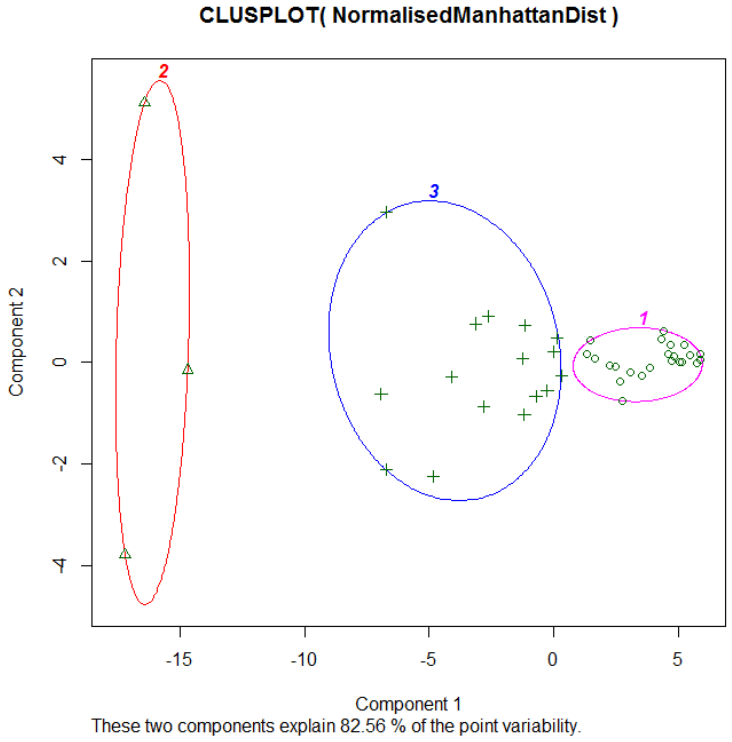
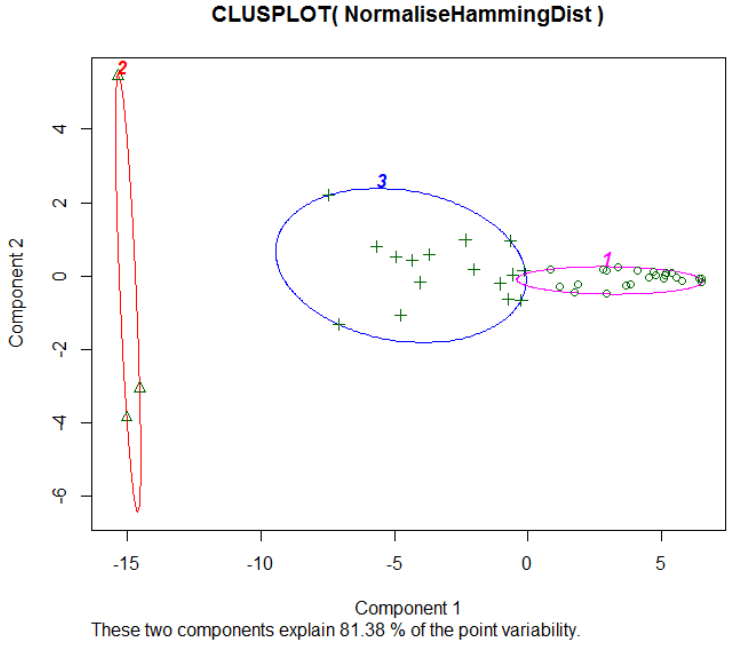


FIGURE 4.20: The clusters of patients using Manhattan metric, DOXEPIN drug



.

FIGURE 4.21: The clusters of patients using Hamming metric, DOXEPIN drug

.

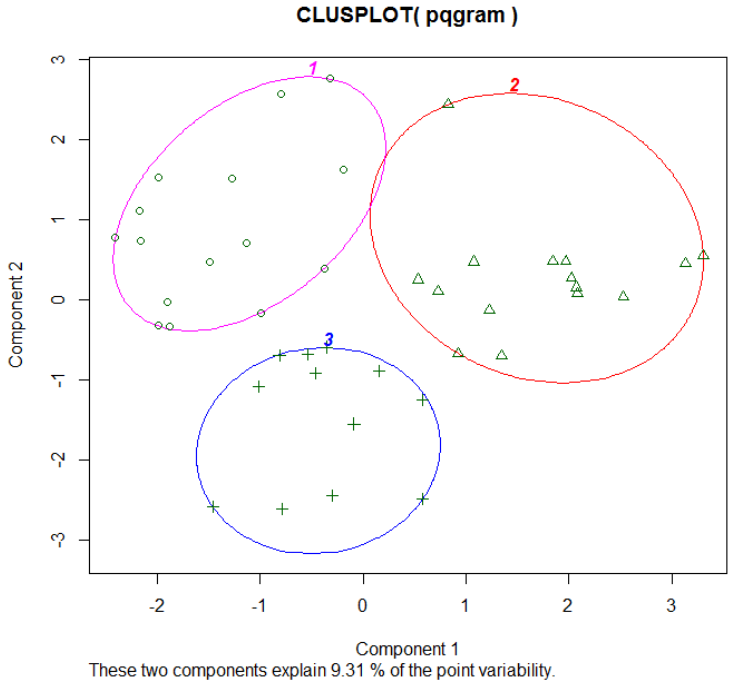


FIGURE 4.22: The clusters of patients using *pq*-gram metric, DOXEPIN drug

.

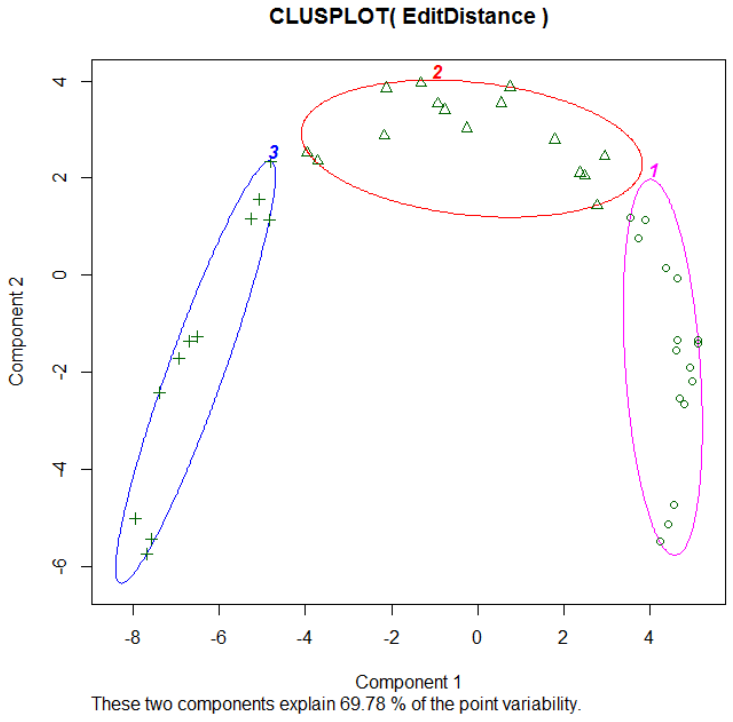


FIGURE 4.23: The clusters of patients using Edit Distance metric, DOXEPIN drug

.

in each cluster for the first and second drug, respectively. *Cluster*1 in the tables contains the number for all patients who are similar to each other, whereas *Cluster*2 contains all patients who have large distances between them. The remaining patients are grouped in *Cluster*3.

TABLE 4.7: The number of patients in each cluster for DESLORATADINE drug

	Cluster 1 (similar)	Cluster 2 (Non-similar)	Cluster 3 others
Euclidean Metric	513	114	361
Minkowski, p=3	578	89	321
Manhattan Metric	602	89	304
Hamming Metric	579	75	334
pq-gram Metric	409	164	415
Edit Distance Metric	284	332	372

TABLE 4.8: The number of patients in each cluster for DOXEPIN drug

	Cluster 1 (similar)	Cluster 2 (Non-similar)	Cluster 3 others
Euclidean Metric	23	3	16
Minkowski, p=3	81	7	17
Manhattan Metric	23	3	16
Hamming Metric	23	3	16
pq-gram Metric	15	15	12
Edit Distance Metric	16	15	11

4.6 Discussion

In this chapter, the *pq*-gram distance metric was applied to different data sets to test its validity and feasibility when used to cluster mixed data sets. The Lenses and Iris data sets were the simplest data sets used and they are non-hierarchical. The *pq*-gram metric has forced these data sets to be hierarchical when applied to them. In case of Lenses data set, the data set could not be clustered, since the number of observations (*pq*-gram trees) obtained after using the pq-gram metric was equal to the number of clusters required, which is equal to the number of the data set classes as well. In the case of Iris data set which is also non-hierarchical data, the data also has been forced to be hierarchical and the clusters obtained to some extent were good in separation, but they were not similar to the original classes of the data sets. In other words, the

results when applying the *pq*-gram distance metric to this classified data did not show good results (the agreement between the new clusters and the existing classes was very low 0.3067), as each new cluster has many dissimilar data points, despite the excellent separation between clusters.

After this, the NBER US Patent data set was used, which is hierarchical. The results, as expected were reasonable when compared to the non-hierarchical data sets, since the data did not need to be forced to be hierarchical by the *pq*-gram metric. The *pq*-gram metric produced some similarity between the objects in the data sets. The disadvantage of this data set was the lack of data records for the patents, which makes it difficult for the *pq*-gram metric to find similar *pq*-gram trees between, for example, each of two patents. Thus, a mixed data set, named the THIN data set was used. This data set comprises rich data records for almost every patient in the data set. The *pq*-gram distance metric was been applied to THIN when comparing between the *pq*-gram metric and a number of other distance metrics. In total, six different distance metrics were applied to the THIN database to isolate details pertaining to the drugs DESLORATADINE and DOXEPIN. The main objective was to find which metric is most useful for measuring distances in THIN data, with an emphasis on the *pq*-gram metric because it is designed for hierarchical data, such as the READ codes present in THIN. The distance metrics were implemented using two different types of data structures, and their results compared. The two data structures were the tree-like structure of the *pq*-gram and Edit Distance metrics groups, as shown in Figure 4.10, and the frequency table or matrix for the group of geometric and Hamming metrics, as shown in Table 4.3.

Regarding the *pq*-gram, the distances were found to depend mainly on the number of intersected *pq*-grams between two trees, as well as the values of the parameters p and q. Choosing the correct values for p and q requires trade-offs. In [116], Srivastava et al. analysed the sensitivity of *pq*-gram distances when comparing values p and q, and concluded that increasing p relative to q implies greater importance is being given to the ancestors than to the children of the trees; i.e. two nodes are considered to be the same only when they share p common ancestors. Thus, in this case the smaller the value of p relative to q, the greater the probability of finding intersected *pq*-grams between

two trees and the more importance is given to the data rather than the structure of the trees. Based on this, the results in the seventh column in Tables 4.4 and 4.6 are better compared to the results in the eighth column of the same tables.

In general, the *pq*-gram metric performed well, compared to that when using the aforementioned data sets, but it was not the best metric because it depends on many parameters (p, q and the tree structure). It was able to highlight some similar patients by measuring the similarity between their data, as shown in Table 4.4 (e.g. patients a670605Up, a670602uS and patients a67340327, a681001KN). On the other hand, Table 4.6 contains some non-similar distances produced by the *pq*-gram and Edit Distance metrics; for example, the two patients (g989501KB and a999104cU) have a normalised distance equal to 1, which means there is no similarity between these patients' data. The reason for this could be the lack of data about the drug DOXEPIN. That is to say, the more data (data records) are available the greater the probability that there will be similar data for patients in the THIN database.

After finding all the distances using the chosen metrics, the results were verified by considering the entire patient populations for each drug and by checking whether these distance metrics discriminate sufficiently using clustering of the patient population. Figures 4.12 to Figure 4.17 for the first drug and Figure 4.18 to Figure 4.23 for the second drug show the results of clustering using the k-means algorithm. The latter details the simplest clustering method and requires the number of clusters to be known in advance. Hence, here it was defined that the number of clusters should be equal to 3. However, more accurate data analysis is required for future work, so more than three clusters might be considered.

The clusters were plotted using the *clusplot* function from R software, which represents all the observations by points in plots using the Principal Component Analysis (PCA) [75], see Chapter 3, Section 3.9. Here, PCA is used in the data set for the purpose of visualisation and no feature selection has been carried out. The clusters are labelled using numbers (1, 2 and 3) as shown in Figure 4.12 and the geometric and Hamming metrics discriminate successfully between the populations for both drugs. Tables 4.7 and 4.8 show the number of patients in each cluster. The patients in Table 4.4 are

grouped in *cluster*1 for all the metrics used, while the patients in Table 4.6 are grouped in *cluster*1 for the geometric and Hamming distance metrics only. In contrast, the distances for the same patients using the *pq*-gram and Edit Distance metrics have very poor similarity. Thus, *cluster*1 for both metrics contains some similar distances other than those in Table 4.6. The probable reason for this is the lack of data related to the second drug. In general, *cluster*1 in Table 4.7 and Table 4.8 contains similar patients, all of whom have medical events related to the drug in common, while *cluster*2 contains non-similar ones. The remaining patients, i.e. those not in *cluster*1 or *cluster*2, are grouped in *cluster*3, as shown in Figures 4.12 to 4.23.

4.7 Summary and Conclusion

The *pq*-gram distance metric was applied to three different data sets to test its feasibility and reliability when measuring the distances for mixed data sets. It was not effective for managing simple non-hierarchical data sets, or for producing the same classes as the original data sets. Then, it was applied to large hierarchical databases, including the NBER US Patent data, so the metric could improve performance by providing some similarity between the *pq*-gram trees for the data set. This provided motivation to expand the investigation by using the *pq*-gram metric on mixed data sets, such as the THIN data set. The metric was applied to the data set during the process of comparing different distance metrics, including the *pq*-gram metric, to measure the similarity between the THIN data set objects.

The comparison shows that, generally, all the metrics produced similarity in distances between the patients' data. Regarding the *pq*-gram metric, it was not the best metric for the THIN data, but it could measure similar distances and group them into one cluster. That is to say, it highlighted some known medical events related to the drugs taken, for example cough and itchy eye symptoms related to the drug DESLORATADINE. Based on this experiment, it was apparent that the *pq*-gram distance metric can be used to measure similarity in the THIN data, but that the results are not outstanding when compared to other distance measures. Thus, to prompt the hierarchical distance

measure to produce better results and to investigate whether it is suitable for use with large mixed databases the *pq*-gram distance metric was used in different way, to work for hierarchical data attributes only. For example, by combining the hierarchical *pq*-gram metric for hierarchical attributes and other distance metrics for non-hierarchical attributes into one distance function to create a new distance measure.

Use of the *pq*-gram distance metric in a CDF has not been carried out yet, and this gap in knowledge can be filled by creating CDFs from the hierarchical distance measures for hierarchical attributes and non-hierarchical measures for other types of attributes, and then clustering the data sets using the BIRCH hierarchical clustering algorithm.

In the next chapter, a new distance measure will be introduced named as the Combined Distance Function (CDF) using the *pq*-gram distance metric for hierarchical attributes. It will be used to measure the similarity between data records in the NBER US Patent data set. The data will also be clustered using the hierarchical clustering algorithm BIRCH, but will first be applied to the Iris data set to determine if it produces the same original classes as Iris.

Chapter 5

Improve the Pq-gram Distance Metric for Mixed Data Sets

5.1 Introduction

There is a growing need for efficient distance measures and clustering algorithms to find similarities between data objects in hierarchical mixed data sets that include hierarchical attributes. Such hierarchical data sets can also contain different data types, such as numeric and categorical attributes; in which case they are called mixed data sets. In the previous chapter, the performance of the hierarchical *pq*-gram distance metric was found to be better when producing similarity data from the mixed THIN data set, than when applied to non-hierarchical and hierarchical only data sets. In this chapter, the *pq*-gram distance metric is used in a different way to deliver better performance and functionality for large mixed data sets. The aim is to use different distance measures for attributes with mixed data types, such that the *pq*-gram metric, in combination with other distance measures, produces a novel distance measure for mixed data sets.

The new distance measure, the CDF, will be first applied to non-hierarchical and hierarchical only data sets, to investigate its performance. This will provide a simple comparison between the performance of the *pq*-gram metric for the data set in the previous chapter and the new distance function detailed in this chapter. A hierarchical

algorithm, named BIRCH is applied in this chapter, which is a well-known hierarchical algorithm for big data sets. It can be used to perform clustering on samples of data sets, to speed up the computation, and then generalise it to the entire data set. The results of the clustering are visualised using R software functions and evaluated using validation indices.

5.2 Comparison Between the Hierarchical Measures Based on the Hierarchical Similarity Rules

According to [77], six hierarchical measures were used to exploit the hierarchical structure of the Network Traffic data, called Echidna. The first three were unsuitable, based on the desired rules and hierarchical similarity measures (see Chapter 2, Section 2.4.1.3). The final three measures were proposed by Mahmmod et al. [77], and only one, the Hierarchical Similarity measure 1 (HSM1), was deemed suitable for their work. In this chapter, when comparing these hierarchical measures, the six measures were tested based on rules, using an example from the Patent data set. Table 5.1 illustrates the similarity between the various nodes in a patent hierarchical tree, as shown in Figure 5.1. This comparison was intended to help illustrate the appropriateness of the hierarchical similarity measures, in terms of the requirements and rules set.

- $Rule_1$: the symmetry rule shows that all of the measures are symmetric except for S_{OGM} whose value depends on the node used as the basis for comparison in the denominator. For example, $S_{OGM}(3939720, Aaen) = 0.333333$; whereas, $S_{OGM}(Aaen, 3939720) = 1$.

- $Rule_2$: Similarity to ancestor is higher than similarity to the ancestor's sibling, S_{GCSM} is the only measure that does not follow this rule, because $S_{GCSM}(Olav, Aaen) = S_{GCSM}(Olav, Aafjes) = 0$.

- $Rule_3$: Similarity to closer ancestor is higher than similarity to distant ancestor, S_{SS} does not follow this rule, since both $S_{SS}(3939720, 3939720) = 1$, and $S_{SS}(3939720, 3945451) = 1$.

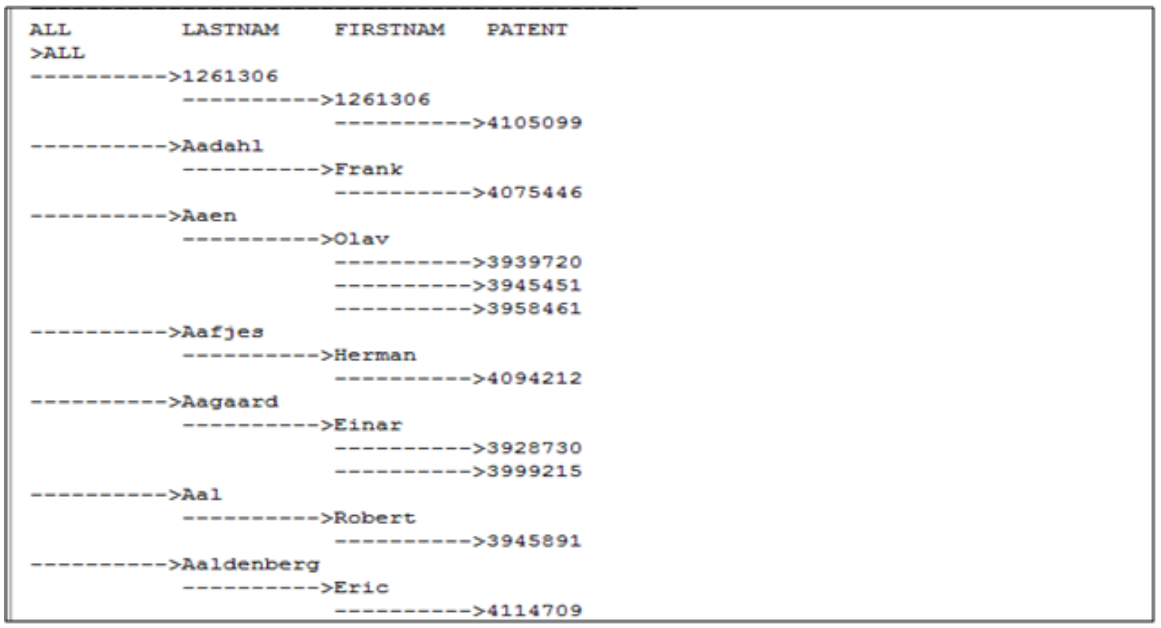


FIGURE 5.1: Example of a Hierarchy Patent Data

- $Rule_4$: lower level siblings are more similar than higher level siblings, all the measures follow this requirement for all measures, such as, $S(3939720, 3945451) \geq S(Aaen, Aafjes)$.

- $Rule_5$: a node should be at least as similar to a sibling at the same level as to a parent of a non-sibling, S_{GCSM} and S_{HSM2} following the rule based on this example $S(3939720, 4094212) \geq S(3939720, Herman) = 0$, although based on [77], the two measures failed. S_{HSM3} failed in following this rule, because $S(3939720, 4094212) < S(3939720, Herman) = 0$.

Based on the example shown in Figure 5.1, the hierarchical similarity rules, S_{HSM1}, and by chance, S_{HSM2}, are the two similarity measures that follow all the rules from among the measures available. Since the distance measures can be derived as a complement of the similarity measures, for the experiments the hierarchical distance measure D_{HSM1}, which is derived from the hierarchical similarity measure S_{HSM1} for the hierarchical data in the database, was used. This hierarchical distance measure was used in combination with other distance functions for the numeric and categorical data types to determine the distances when clustering using the BIRCH algorithm and Equation 2.15.

TABLE 5.1: Similarity Values between Various Nodes in the Tree in Figure 5.1

First Node	Second Node	S_{GCSM}	S_{SS}	S_{OGM}	S_{HSM1}	S_{HSM2}	S_{HSM3}
Aaen	Aafjes	0	0	0	0	0	0.8
Olav	Herman	0	0	0	0	0	0.6
Olav	Aaen	0	1	0.5	0.2	0.666667	0.9
Olav	Aafjes	0	0	0	0	0	0.7
3939720	Aaen	0	0.5	0.333333	0.2	0.5	0.8
3939720	Olav	0.4	1	0.666667	0.4	0.8	0.9
3939720	3939720	1	1	1	0.6	1	1
3939720	3945451	0.666667	1	0.666667	0.4	0.666667	0.8
3928730	3999215	0.666667	1	0.666667	0.4	0.666667	0.8
Einar	Robert	0	0	0	0	0	0.6
3939720	Aafjes	0	0	0	0	0	0.6
3939720	Herman	0	0	0	0	0	0.5
3939720	4094212	0	0	0	0	0	0.4
3939720	Einar	0	0	0	0	0	0.5
3939720	3945891	0	0	0	0	0	0.4
Aaen	3939720	0	0.5	1	0.2	0.5	0.8

As described in Chapter 3, section 3.5, the BIRCH algorithm takes each record and iteratively builds a hierarchical tree of clusters called a CF tree (see Definition 2.1). As the tree is built, each record is inserted into the closest cluster, using the CDF for all attributes. In Figure 5.2, the hierarchical data has to possess the same length for data comparison, to locate the similarity between them in the records. Each character in an attribute on the record is compared with a corresponding character in other records in order to reveal the similarity between them. This is also a way to find the similarity between, for example, the misspelled names of inventors, to group them into one cluster.

5.3 Data Pre-processing

The data sets used in this chapter were the Iris data and the U.S. NBER Patent database. The data source preparation process consisted of loading the data sets from the source to the Microsoft SQL server. The Iris data set is an existing data set, which does not need any reprocessing, while the patent database needs to be cleaned first by removing all the records with missing values, removing excess white-spaces and removing tags, and

```
Num of Entries = 25
{[ Bernstein**  Seymour****     3882149****     1.00   [ 1.0  0.0  0.0 ]      ( 1 )]
[ Phillipps**   Gordon*****     3882151****     1.00   [ 0.0  1.0  0.0 ]      ( 2 )]
[ Newall*****   Christopher     3882151****     1.00   [ 0.0  1.0  0.0 ]      ( 3 )]
[ Seki*******   Shigeru****     3882153****     1.00   [ 0.0  0.0  1.0 ]      ( 4 )]
[ Sato*******   Koji*******    3882153****     1.00   [ 0.0  0.0  1.0 ]      ( 5 )]
[ Henrick****   Clive******    3882154****     1.00   [ 1.0  0.0  0.0 ]      ( 6 )]
[ Siddall****   John*******    3882154****     1.00   [ 1.0  0.0  0.0 ]      ( 7 )]
[ Wharton****   Harry******    3882155****     1.00   [ 1.0  0.0  0.0 ]      ( 8 )]
[ Henrick****   Clive******    3882156****     1.00   [ 1.0  0.0  0.0 ]      ( 9 )]
[ Siddall****   John*******    3882156****     1.00   [ 1.0  0.0  0.0 ]      ( 10 )]
[ Henrick****   Clive******    3882157****     1.00   [ 1.0  0.0  0.0 ]      ( 11 )]
[ Fuchs******   Julius*****    3882160****     1.00   [ 1.0  0.0  0.0 ]      ( 12 )]
[ Gash*******   Virgil*****    3882173****     1.00   [ 1.0  0.0  0.0 ]      ( 13 )]
[ Benedict***   Theodore***    3884642****     1.00   [ 1.0  0.0  0.0 ]      ( 14 )]
[ Shiba******   Keisuke****    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 15 )]
[ Hinata*****   Masanao****    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 16 )]
[ Ohi********   Reiichi****    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 17 )]
[ Sato*******   Arkira*****    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 18 )]
[ Yamasue****   Koutarou***    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 19 )]
[ Kondo******   Tokiharu***    3887380****     1.00   [ 0.0  0.0  1.0 ]      ( 20 )]
[ Shiba******   Keisuke****    3887381****     1.00   [ 0.0  0.0  1.0 ]      ( 21 )]
[ Tsubota****   Hotohiko***    3887381****     1.00   [ 0.0  0.0  1.0 ]      ( 22 )]
[ Tang*******   Arthur*****    3887392****     6.00   [ 1.0  0.0  0.0 ]      ( 23 )]
[ Coggins****   Dolphus****    3887403****     1.00   [ 1.0  0.0  0.0 ]      ( 24 )]
[ Ono********   Minoru*****    3887407****     4.00   [ 0.0  0.0  1.0 ]      ( 25 )]
}
```

FIGURE 5.2: Hierarchical data have the same length

then generating one complete table from different smaller tables. The tables from the data sets used are the Patent, Inventor and Assignee. These small data sets are consolidated and then fed into the BIRCH clustering algorithm by creating one table (374230 data records) containing all the information in the tables being combined. Figure 3.1 exemplifies the patent data table created.

5.4 Clustering and Visualising the Iris Data Set using BIRCH

In order to validate the new method when using the *pq*-gram distance metric in a CDF for a mixed data types, the method was first applied to a simple data set, Iris data. It is an existing known data set that consists of 150 data records and three species. It is used to verify the new method by examining whether the method can produce the same existing classes of Iris data, i.e. to determine how much agreement there is between the new clusters and the original classes. Firstly, the CDF is used without the *pq*-gram metric, then the *pq*-gram distance metric is added to work alongside the CDF. Figures 5.4 and

5.5 produced the clustering of the Iris data using both kinds of CDFs and the BIRCH clustering algorithm.

The results when clustering are visualised using two plotting functions: *cmdscale* and *clusplot* in R. Figure 5.3 visualises the clusters in a multidimensional scale way (MDS). Cluster number 1 is the most divisible, and it is contains the first species of iris flower, called setosa, with the exception of two misclassified setosa's. While the other two clusters in both figures are not suited to separation, because they contain iris versicolor and iris virginica samples in common. Figure 5.4 shows how BIRCH labelled all the setosa's perfectly, but made some mistakes with two other species, selecting far too many flowers for cluster 1. This is because it plots with the *clusplot* function, giving the Iris raw data the function of input, instead of a distance matrix; while the *cmdscale* function requires the matrix as input. When clustering the Iris data, it was easy to use the raw data because it is numeric and does not require transformations for the attributes to be used as input for the *clusplot*. Subsequently, for the Patent data set, the distance matrix was necessary as an input for the *clusplot*, because the patent data attributes are of different types (numeric, hierarchical and categorical). However, the patent raw data can be used as an input to the *clusplot* function in two cases only; first, if the selected data set are all numeric; second, if the raw data is used as a data frame, which is one of the options that the plotting function has as a data input, (see Section 5.8).

In 2009, Park et.al [87] clustered the Iris data using the k-means clustering algorithm and a proposed clustering algorithm named the k-medoids algorithm. The clustering accuracy with the k-means algorithm was 88.7%; whereas, it was 92% with the proposed method. The Iris data was clustered in this chapter using the BIRCH algorithm, and evaluated using a method for estimating accuracy, and by comparing the BIRCH cluster assignments with Iris data with the real Iris classification to ascertain how much agreement there was between them. The accuracy of the data obtained was 88% and this value seems reasonable based on the ranges that belong within maximum range 1, as well as when comparing this with the accuracy involved in clustering the Iris data using the k-means algorithm in [87]. The similarity between the BIRCH cluster assignments and the Iris data with the real Iris classification was also measured using

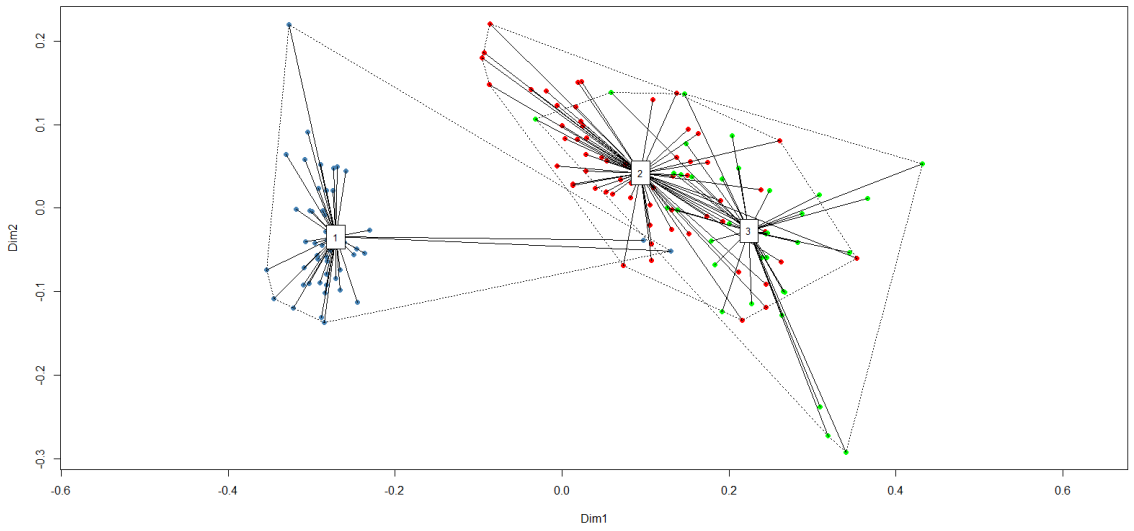


FIGURE 5.3: Multi-dimensional scaling plot for Iris data set using the distance matrix

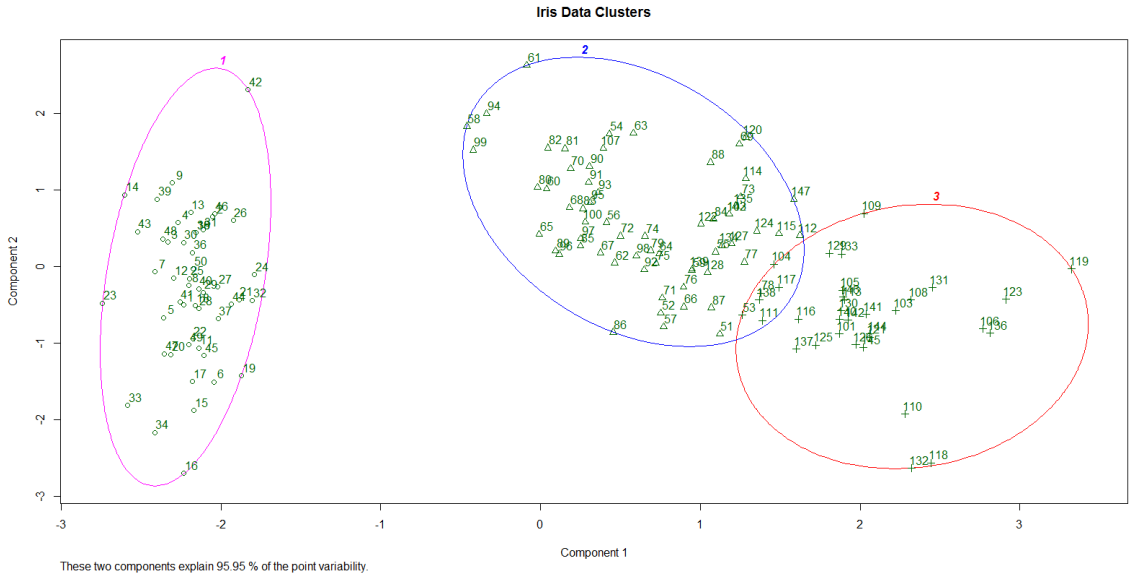


FIGURE 5.4: Plotting Iris data using CDF without pq-gram distance function

validation indices, such as the Rand index and the Jaccard index, which are described in Chapter 3, sections 3.8.3 and 3.8.4. The values obtained for both were 0.8679 and 0.6728, respectively. Figure 5.6 shows the calculation of the accuracy, sensitivity, and specificity of each class in the data.

Although the Iris data consists of numerical attributes only and the CDF is for different types of attributes, the CDF did measure the similarity between the Iris data records such that only the distance functions for the numerical attributes were worked

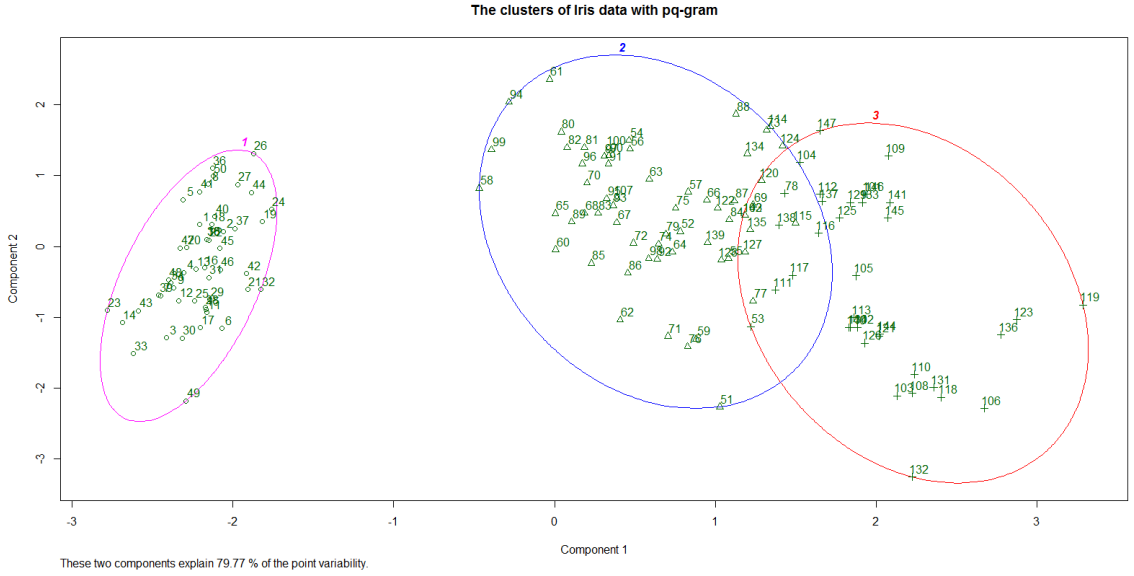


FIGURE 5.5: Plotting Iris data using CDF with pq-gram distance function

```
> confusionMatrix(iris$Species, data$CLUSTER)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0         16        34

Overall Statistics

               Accuracy : 0.88
                 95% CI : (0.817, 0.9273)
    No Information Rate : 0.4267
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.82
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.7500           0.9444
Specificity                 1.0000            0.9767           0.8596
Pos Pred Value              1.0000            0.9600           0.6800
Neg Pred Value              1.0000            0.8400           0.9800
Prevalence                  0.3333            0.4267           0.2400
Detection Rate              0.3333            0.3200           0.2267
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.8634           0.9020
```

FIGURE 5.6: The confusion matrix and statistics for Iris data using CDF without pq-gram

```
> confusionMatrix(data$CLUSTER, iris$Species)
Confusion Matrix and Statistics

          Reference
Prediction    setosa versicolor virginica
  setosa          50          0         0
  versicolor       0         48        14
  virginica        0          2        36

Overall Statistics

               Accuracy : 0.8933
                 95% CI : (0.8326, 0.9378)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.84
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9600           0.7200
Specificity                 1.0000            0.8600           0.9800
Pos Pred Value              1.0000            0.7742           0.9474
Neg Pred Value              1.0000            0.9773           0.8750
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3200           0.2400
Detection Prevalence        0.3333            0.4133           0.2533
Balanced Accuracy           1.0000            0.9100           0.8500
```

FIGURE 5.7: The confusion matrix and statistics for Iris data using CDF with pq-gram

and the others not. This can be achieved by giving the algorithm a tuple record for attribute types (Numerical, Numerical, Numerical, Numerical) corresponding to four numerical attributes (Sepal_Length, Sepal_Width, Petal_Length, Petal_width). For the BIRCH algorithm to cluster the Iris data set, the following parameters were taken: the maximum nodes entries B = 150 nodes, the distance threshold T started at value 0.3 and incremented each time, running by 1.005, until three clusters were obtained without any further changes.

The *pq*-gram distance metric alongside with the combined distance function, was also applied to the Iris data set, and then the distances clustered using the BIRCH clustering algorithm. The results of the clustering showed a slight difference compared to that when using the CDF without the *pq*-gram distance function. The difference was an improvement, based on the new accuracy provided by the new clusters of Iris data, at 0.8933. Figure 5.5 shows the resultant clusters when using the CDF with the *pq*-gram distance metric, and their accuracy is shown in Figure 5.7, which is higher than that in Figure 5.6.

The results obtained using the *pq*-gram distance metric with the CDF for the different data sets show an improvement in performance compared to that when using the *pq*-gram metric only. The new clusters are more reasonable and more similar to the existing classes of the original Iris data set, which are approved by the accuracy provided by clustering the data set and by the hand checking to every data point in the original Iris data set and the new clustered data set.

5.5 Using the New Measure for Patent Data Set

The input data was extracted from the Patent data set as 5-tuple records (LASTNAM, FIRSTNAM, PATENT, CAT, CITY), where the CAT field is a numeric attribute, referring to the category the patent belongs to. CITY is a categorical attribute, and the other three attributes are hierarchical: the last name, first name of the inventor and patent ID number. Hierarchical means the attributes have a hierarchical relationship between each other (implicitly structured like trees). To measure the similar distances between the data records, the CDF, CDF with *pq*-gram and CDF with weighted *pq*-gram were used, which consists of a hierarchical distance function D_{HSM1} for each hierarchical attribute (LASTNAM, FIRSTNAM, PATENT), the Euclidean distance function for numerical attribute (CAT), the histogram of frequencies of the categorical attribute value (CITY) and the *pq*-gram distance function for each input data record.

The construction of the CF tree was based on the threshold $T = 0.5$, which was increased by 1.005 each time (the higher it is, the faster the convergence). The maximum number of entries B, was set at 200, 500 and 1000 data input, to clarify the results and the memory limit (Max. Node Entries * 1024 * 1024). In this example, three clusters were chosen for 200 data records, four clusters for 500 data records and five clusters for 1000 data records. Then the resulted clusters were plotted using the R software function: *clusplot*. After clustering the patent data set using the BIRCH algorithm, the resulting clusters and the distances matrix were saved in two separate (.csv) files, then in R the saved files were processed and plotted.

5.5.1 Clustering the 200 Data Records Using the CDFs and BIRCH Algorithm- Example 1

In this section, the results when clustering the 200 patent data records are displayed in Figures 5.8, 5.9 and 5.10 using the distances matrix produced with the BIRCH algorithm and the plotting function *clusplot*.

Figure 5.8 shows three clusters that resulted when clustering the 200 patent data records using the BIRCH algorithm and the CDF. The data was scaled to project the original data onto directions which maximise variance; the first direction was the first component with maximal variance and the second component had maximal variance perpendicular to the first component. Thus, the two axes represented the first two principal components, which were the linear combination of the original data set. The ellipses shapes are clusters constructed based on the average and covariance matrixes of each group. The clusters are labelled as 1, 2 and 3, and the data point's labels are numbers, representing the numbers in the Patent data records. For example, cluster 3 has 10 records, as shown in Table 5.2. These records are similar to each other in at least one attribute; e.g. record number 1 and 2 are similar to the PATENT code number and the category CAT it belongs to. Records numbered 3, 4 and 5 are similar to each other, sharing the same attributes as well as the CITY name; thus, the three records that are similar to the second record and share the CITY name are in one cluster. Similarly, records number 6 and 7 are similar to each other in LASTNAM, PATENT, CAT and CITY, and similar to the previous records in that CAT number, so they are in same cluster. The last three records share the CAT number with the records in cluster 3. These records or data inputs are close to each other in terms of distance when clustering using BIRCH; thus, they are in one group. Although Table 5.2 shows no similar inventors with different patent numbers, it still groups the data records based on the similarity of other attributes and the closeness of distance between each of the two records.

Table 5.3 shows a number of similar data records in cluster number 1, and the similarity between these data records is greater than that in cluster number 2. In the table, the different patent codes belonging to the same inventor or inventors who have

TABLE 5.2: All the data points (records) of the cluster number 3 in Figure 5.8

Records	LASTNAM	FIRSTNAM	PATENT	CAT	CITY
1	Ono	Minoru	3887407	4	Kodaira
2	Momoi	Toshimitu	3887407	4	Tokyo
3	Nakamura	Junji	3887435	3	Tokyo
4	Miyashiro	Shigeyoshi	3887435	3	Tokyo
5	Okada	Hiroshi	3887435	3	Tokyo
6	Harich	Jakob	3890212	3	Orlando
7	Harich	Franz	3890212	3	Orlando
8	Chetter	John	3890196	4	Lytham-St.-Annes
9	Miller	Richard	3890258	4	Berwyn
10	Bergeron	John	3890202	3	Schenectady

TABLE 5.3: Some of the patent data records of cluster 1 in Figure 5.8 that have the most similarity

Records	LASTNAM	FIRSTNAM	PATENT	CAT	CITY
1	D'Alelio	Gaetano	3890272	1	South Bend
2	D'Alelio	Gaetano	3890274	1	South Bend
3	Arai	Tuhru	3887443	1	Nagoyashi
4	Arai	Shinji	3884790	1	Nagoya
5	Henrick	Clive	3882154	1	Palo Alto
6	Henrick	Clive	3882156	1	Palo Alto
7	Henrick	Clive	3882157	1	Palo Alto
8	Shiba	Keisuki	3887380	1	Minami-Ashigara
9	Shiba	Keisuki	3887381	1	Kanagawa
10	Sato	Koji	3882153	1	Iwaki
11	Sato	Arkira	3887380	1	Minami-Ashigara

the same last name with different written city names, as in records 3 and 4 in the table, are all in one cluster. The records are designated to cluster 1 by their category number (CAT), despite the fact that they differ in some of their other attributes.

In general, it can be seen that the most similar data records in the majority of the attributes are grouped into one cluster. The other clusters may contain dissimilar data records, except for one attribute. That is to say, some clusters are better at attaining similarity between the data records, when compared to the other clusters. For example, in Figure 5.8, cluster number 1 is better at attaining the most similar data records, as shown in Table 5.3, which compares it with cluster number 3, the records for which are shown in Table 5.2.

Since the *pq*-gram distance metric considers the entire database entry record and not just one column of each data record when measuring the similar distances between two data records, the metric is added to the CDF to work alongside D_{HSM1} as an another hierarchical distance metric. Thus, the CDF for the 5-tuple records consists of one distance function for the numerical attribute, one distance function for the categorical attributes and three hierarchical distance functions for the hierarchical attributes plus the *pq*-gram distance function for the entire 5-tuple record

Figure 5.9 shows the impact on the results when adding the *pq*-gram metric to the CDF when clustering the Patent data set. The values for the p and q parameters are chosen as $p = 2$ and $q = 2$ for all the experiments in this chapter (see Chapter 4, Section 4.4.2) since better distances are obtained when the values are small, while the *pq*-gram metric has been applied to THIN data. When comparing Figure 5.9 to Figure 5.8 some overlap in the clusters is displayed. Clusters numbered 3 and 2 shared record numbers 57 and 129, and both belong to the same CATEGORY number. It seems that the *pq*-gram metric with CDF is trying to force some data records to share or move to other clusters which have data records that share some of their attributes. For example, cluster number 3 has two patents in the same category number, which is equal to 4, and cluster number 2 has only one patent in the same category, thus cluster number 2 shares cluster number 3 with the patent with the same category. It is also noteworthy that, despite the fact that there are overlapping clusters shown in Figure 5.9, the clusters may include data records that share similarity in at least one attribute.

To show the impact of one distance function among the others in the CDF, weights were added to the different aspects of the CDF as described in Equation 2.16, while clustering the 200 patent data records into three clusters using the *pq*-gram distance function along with the CDF. The weight values are equally distributed to the different parts of the CDF, with the exception of the part that needs to be investigated. The latter will have the sum of the other parts' weights, which means it has the greatest impact on the calculation process. For example, to give the greater impact to the *pq*-gram distance function along with the CDF, the weights are added as follows: $Dis(Cl_1, Cl_2) = \left[(dis_h \times 0.2) + (dis_h \times 0.2) + (dis_h \times 0.2) + (dis_n \times 0.2) + (dis_c \times 0.2) + (dis_{pqgram} \times 1)\right]^{\frac{1}{2}}$

where dis_h, dis_n and dis_c represent the distance functions for the hierarchical, numerical and categorical attributes, respectively. dis_{pqgram} represents the distance function for the pq-grams of the data records. The weights are added equally to each part of the CDF except for the aspect that needs to be investigated, which is given the largest weight. In the example above, the pq-gram distance function part is given the largest weight. The result when assigning a weight to the pq-gram distance function gives the same clusters as are described in Figure 5.9. In this example, it is evident from the figures that giving a weight to the pq-gram distance function in the CDF did not affect the results when clustering. Also in both figures (5.9 and 5.10), cluster numbers 3 and 2 share records numbered 57 and 129, and both belong under the same CATEGORY number.

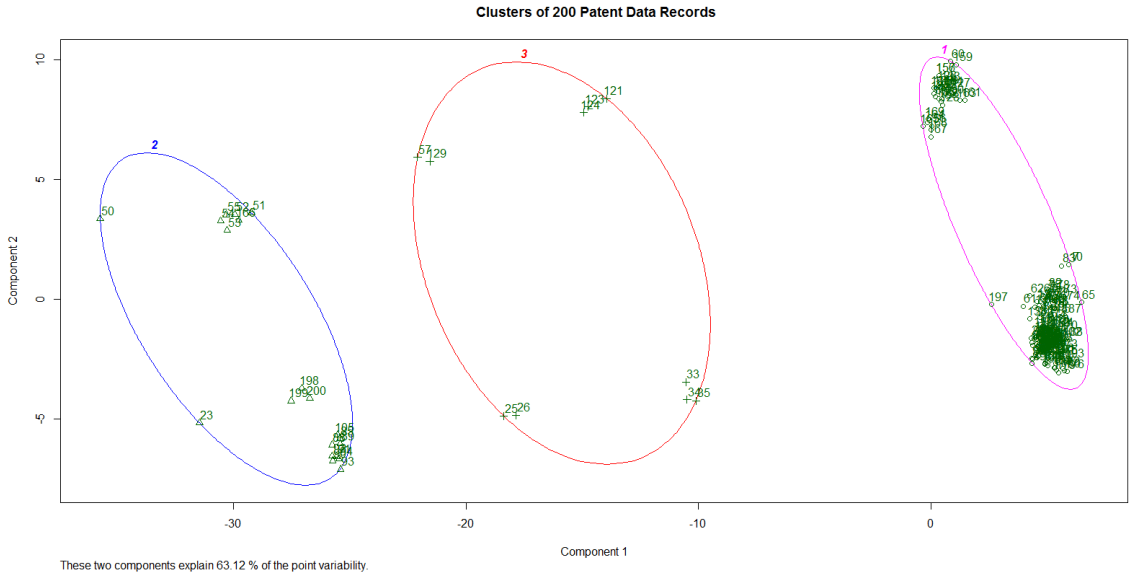


FIGURE 5.8: Clusters of 200 patent data records using CDF and clusplot function

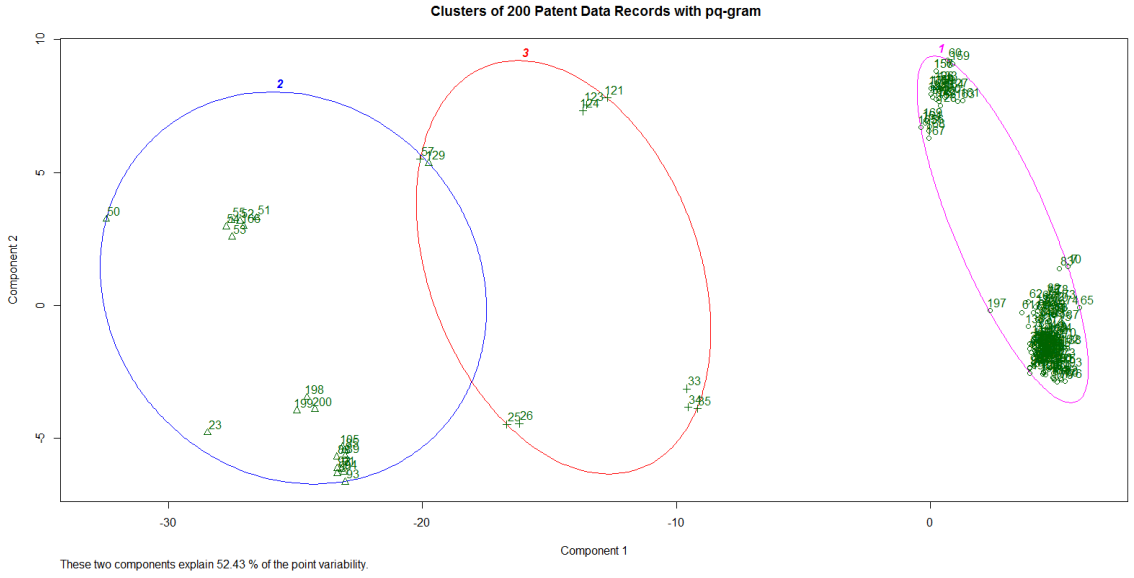


FIGURE 5.9: Clusters of 200 patent data records with 2,2-gram

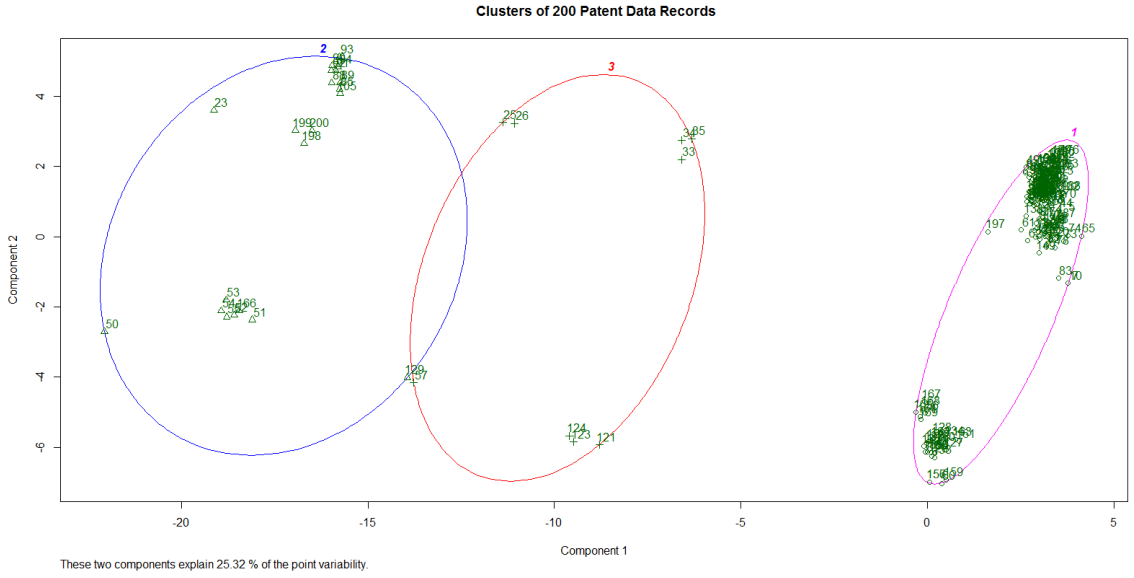


FIGURE 5.10: Clusters of 200 patent data records by giving a weight to the 2,2-gram metric

5.5.2 Clustering the 500 Data Records Using the CDFs and BIRCH Algorithm- Example 2

In this section, another sample of the patent database is clustered to illustrate the role of the *pq*-gram metric when used with the CDF to measure the similarity between the data set records more clearly. The sample comprises 500 data records and is clustered into four clusters, once using the CDF without the *pq*-gram metric as in Figure 5.11 and using the CDF with the *pq*-gram metric as in Figure 5.12.

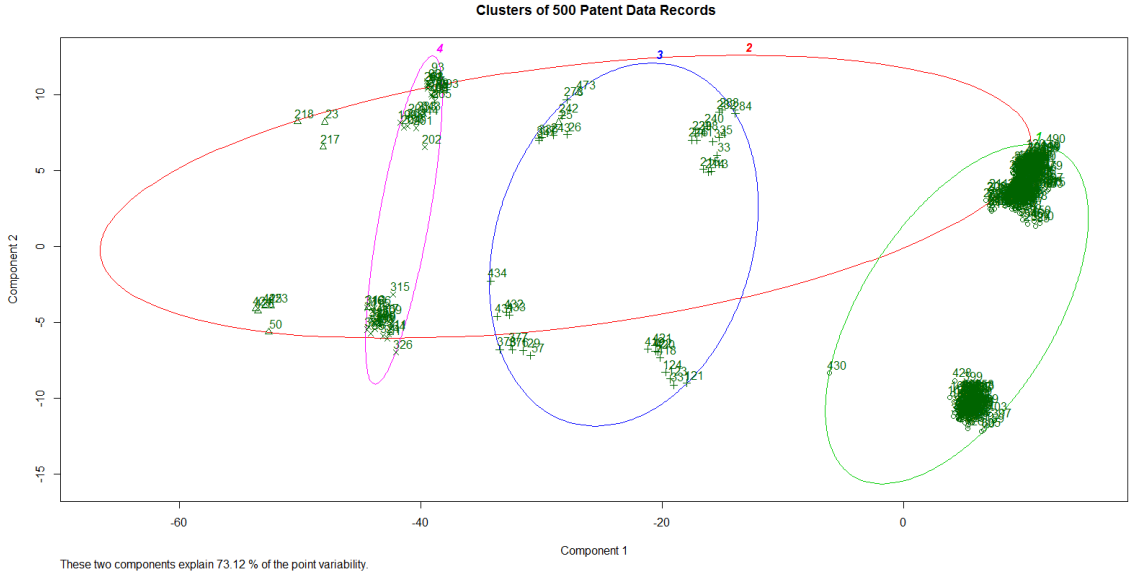


FIGURE 5.11: Clusters of 500 patent data records without pq-gram distance function

The comparison between Figure 5.11 and Figure 5.12 shows that using the CDF with the *pq*-gram metric in this case produces better clustering than without it. In other words, where the *pq*-gram metric is applied, the clusters are well separated and most of the data records inside have relative similarity. For example, cluster number 1 has all the data records belonging to patent category (CAT) number 1, which is the most similar attribute between all the data records, although there are other attributes. Cluster number 2 has all the data records similar in categories number 5 and 6, which are the most similar attributes in the data records, besides other similar attributes like the CITY and LASTNAM. Within the 500 data records, there is only one record with category number 2, and so it appears in cluster number 4 alone. When looking at cluster number

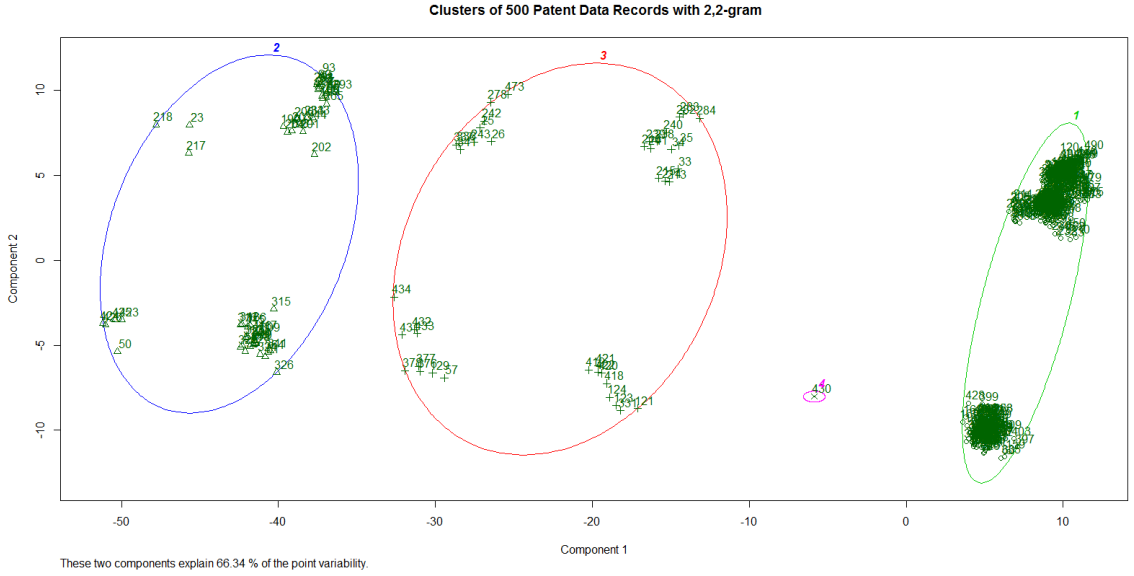


FIGURE 5.12: Clusters of 500 patent data records using BIRCH algorithm and 2,2-gram metric

4, the data record is seen to have some similarity with other data records within other clusters like cluster number 3. Thus, to make this data point more apparent in a cluster with more similarity data records, a weight has been given to the *pq*-gram distance function to lend it greater emphasis when clustering the data. This process enhanced the results and produced Figure 5.13.

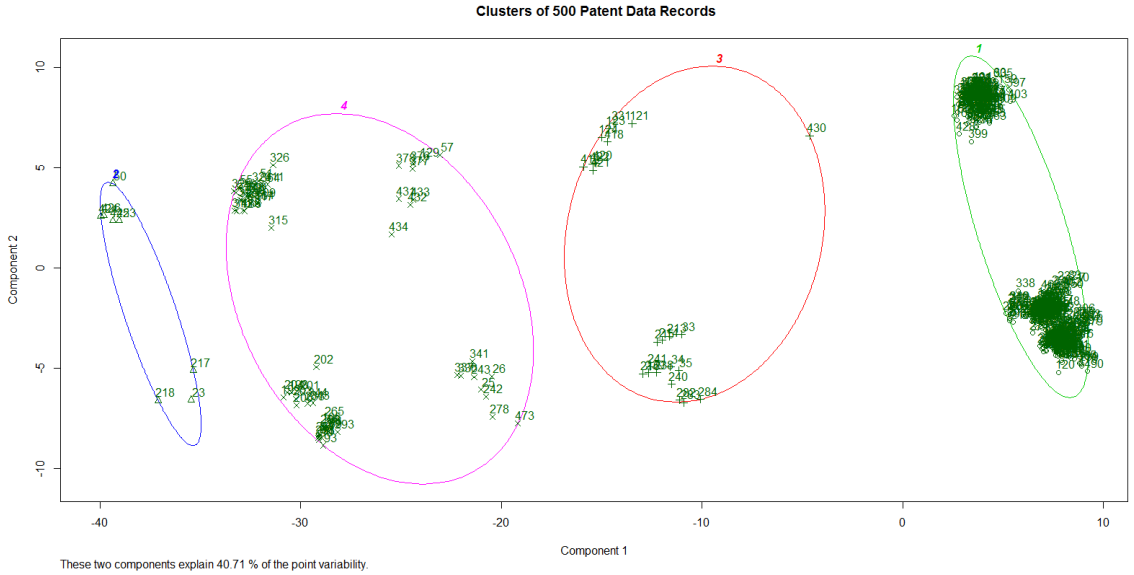


FIGURE 5.13: Clusters of 500 patent data records by giving a weight to the 2,2-gram metric

The clusters of Figure 5.13 show more compactness comparing to those in Figure 5.11 and Figure 5.12. The data point, which was alone in cluster 4, as seen in Figure 5.12 was added to cluster 3, which comprises all the data records for CAT number 3, but shares FIRSTNAM with the other data records. This means that the *pq*-gram distance function, along with the CDF tried to combine this data record with others sharing some similarity. Cluster number 2 consists of data records with category number 6 and a similar patent number.

5.5.3 Clustering the 1000 Data Records Using the CDFs and BIRCH Algorithm- Example 3

In this section, the same process as that described in the previous sections was performed, but using 1000 patent data records. The data was clustered to five clusters as displayed in Figures 5.14 using the distances matrix produced using the BIRCH algorithm and the plotting function *clusplot*. As with Figure 5.8, the clusters are also labelled, but to 1, 2, 3, 4 and 5; and the data point labels are numbers representing the patent data records. Each cluster in the figure contains a number of data records that are very close in terms of distance and similar to at least one of the attributes. In addition, the overlap between cluster number 2 with cluster 3 on the right and cluster 5 on the left arises because some data points in cluster 2 are the same distance from the data points in clusters 3 and 5.

The comparison between Figure 5.15 with Figure 5.14 shows that the CDF with the *pq*-gram metric separates the data records for all patents belonging to category number 1 in cluster number 1; whereas in Figure 5.14 cluster number 1 includes the patents that belong to category numbers 1 and 2. Furthermore, cluster number 3 includes all the data records that have the category attribute 2 and 3 in Figure 5.15, while in Figure 5.14, the same cluster corresponds with cluster number 4, and includes only the patents belonging to category number 3. Regarding the other attributes in the clusters, a variety of attributes appear in each cluster, and not all the data records in a single cluster share the

same attributes, with the exception of the patent category number. Generally, the *pq*-gram distance metric, used in conjunction with the CDF, offers a slight improvement in terms of the approach offered when gathering similar patent data records into one cluster.

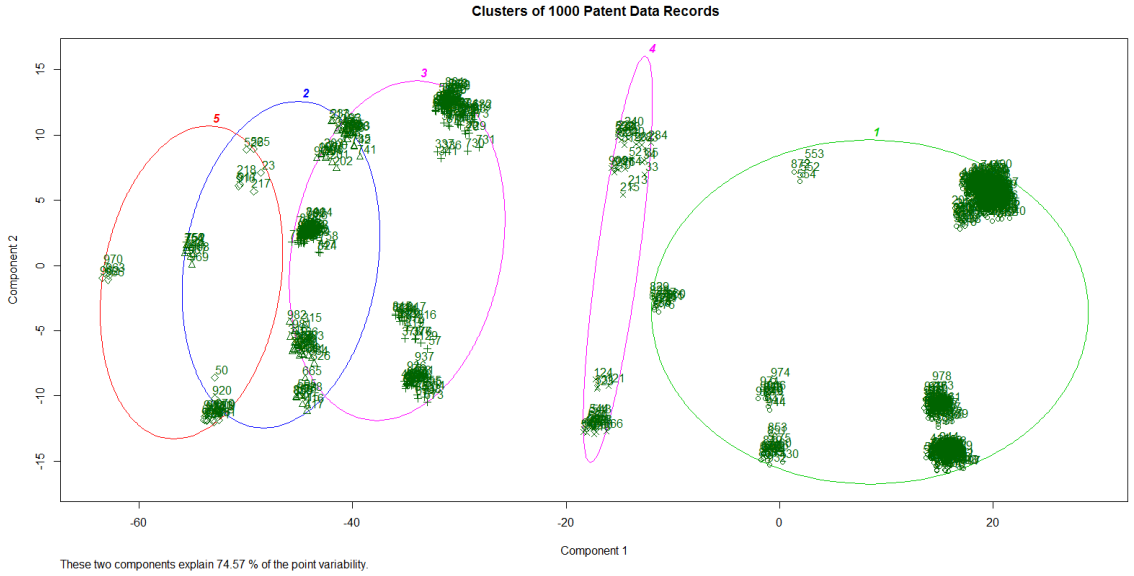


FIGURE 5.14: Plotting 1000 patent data records using clusplot function

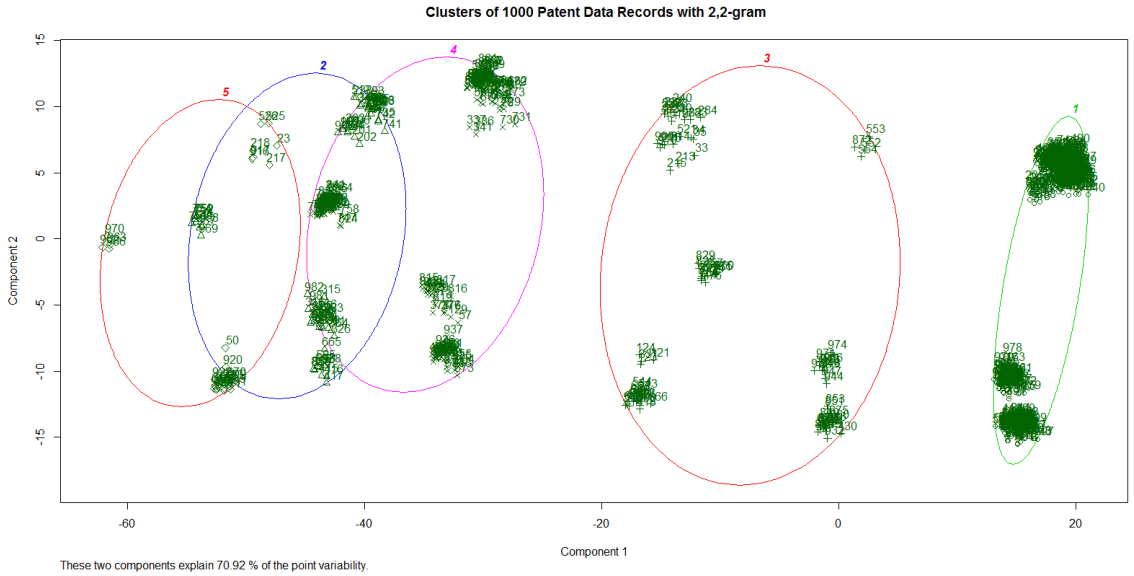


FIGURE 5.15: Clusters of 1000 patent data records with 2,2-gram

In Figure 5.16, the results show four of the clusters are quite well separated while one overlaps them all. That is to say, the CDF with weighted *pq*-gram distance function could not produce better clusters than shown in Figure 5.15.

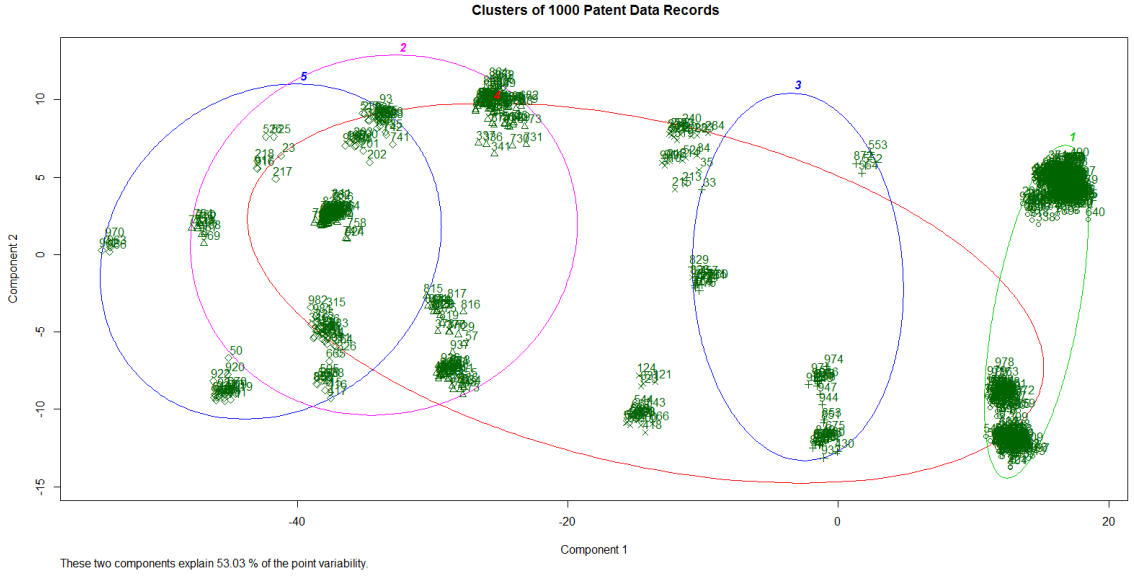


FIGURE 5.16: Clusters of 1000 patent data records by giving a weight to the 2,2-gram metric

5.6 Replacing the Hierarchical Measures by the Pq-gram Distance Function

In this section, the hierarchical measures in the CDF have been replaced by the *pq*-gram metric, to be combined with measures of numerical and categorical data. Practically, the replacement process is done by not considering the hierarchical measures in the CDF, by giving them the weight 0. Thus, the New-CDF will then be a combination of the distance functions of the *pq*-gram distance function for hierarchical data, the Euclidean distance function for numerical data and the histogram of the frequencies of the attribute values for categorical data. The New-CDF was applied to the same examples as detailed in the previous sections to show the differences between their results when using the *pq*-gram along with the CDF or the New-CDF.

5.6.1 Example 1

In this example, the New-CDF is used to measure the distances between 200 patent data records, which are clustered into three clusters using the BIRCH algorithm as shown in Figure 5.17. The three clusters in this figure, when compared to those in Figures 5.8,

5.9 and 5.10, have almost the same data records, but their distribution in each cluster is different. Cluster number 3 has the same records as illustrated in Table 5.2 except for one record (record number 129), which has been moved to cluster number 2, because its *pq*-gram distance value differs from the distance values for the other 9 records. Arguably, the overlap between clusters number 2 and 3 is because the *pq*-gram distance function gives similar distance values to overlapped records.

Thus, it is concluded from these results that by using the weighted *pq*-gram distance function (New-CDF), the aim is to benefit from moving records with the same *pq*-gram distances into one cluster. However, this does not happen every time as will be seen in the next example when clustering the 500 patent data records into 4 clusters.

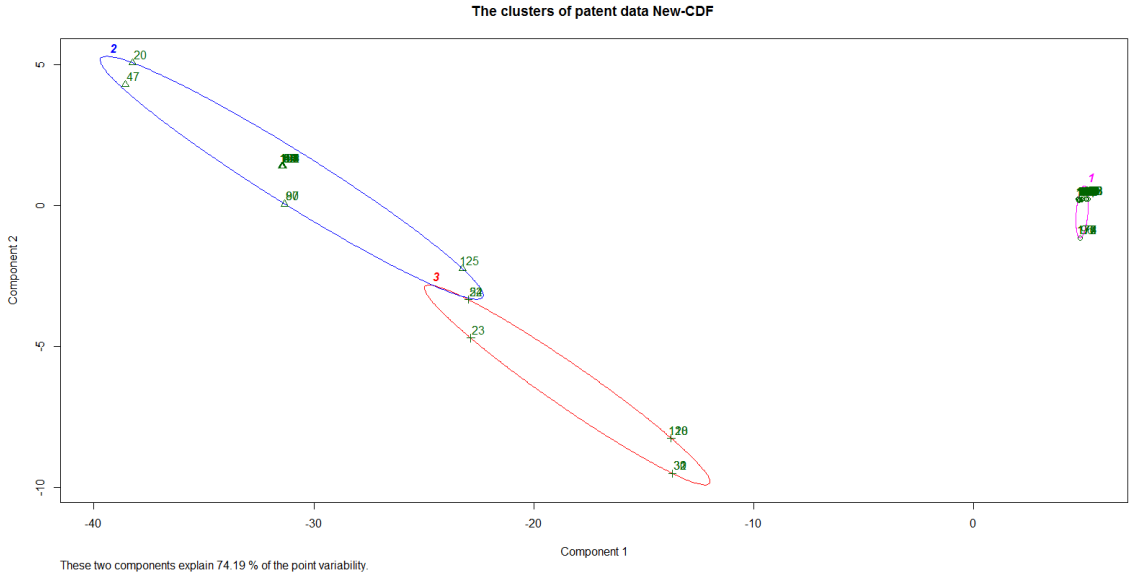


FIGURE 5.17: Clusters of 200 patent data records using the New-CDF

5.6.2 Example 2

In this example, New-CDF is used to measure the distances between 500 patent data records, which are clustered into 4 clusters using the BIRCH algorithm as shown in Figure 5.18. This figure is very different from previous figures, like Figure 5.11 when using the CDF without *pq*-gram, Figure 5.12 using the *pq*-gram along with the CDF and Figure 5.13 using the weighted *pq*-gram with the CDF. The difference is that the 4 clusters are very overlapping, such that it is hard to interpret the data. However,

the basic interpretation could be that, while the *pq*-gram distance function is trying to put all the same records with the same *pq*-gram distances into one cluster, that cluster already shares other similar attributes, but with different *pq*-grams distances, with the other clusters. Thus, this causes the clusters to overlap, showing they have similar data records in common.

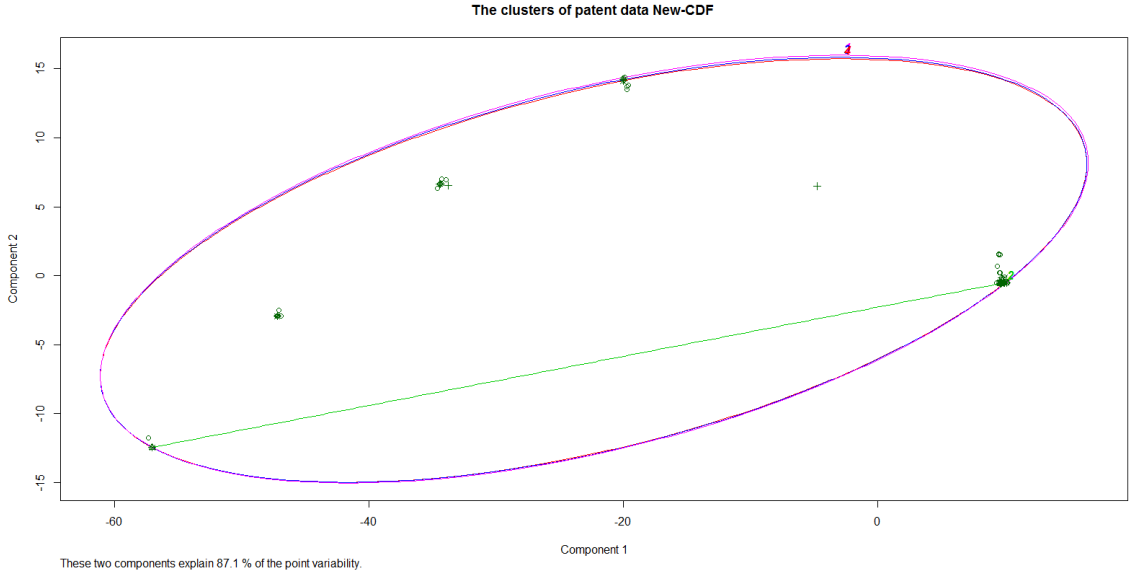


FIGURE 5.18: Clusters of 500 patent data records using the New-CDF

5.6.3 Example 3

In this example, the New-CDF is used to measure the distances between 1000 patent data records, which are clustered into 5 clusters using the BIRCH algorithm (see Figure 5.19). The clusters overlap heavily, which shows the New-CDF could not produce any meaning groups for this example.

5.7 Cluster Validation

In this section, three tables were produced to illustrate the indice values for the results when clustering the 200, 500 and 1000 patent data records in different cases. The indices are the Silhouette and C-index, which are described in Chapter 3, Sections 3.8.1 and

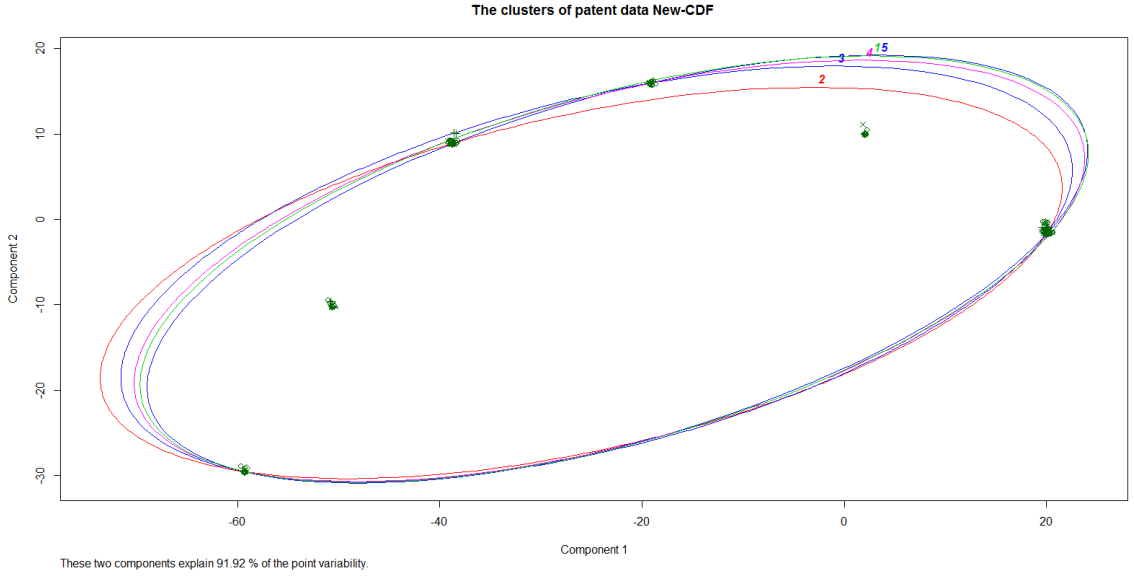


FIGURE 5.19: Clusters of 1000 patent data records using the New-CDF

3.8.2, respectively. Each table shows the index values for the clusters that resulted when clustering the patent data set using the BIRCH algorithm and the CDF without the *pq*-gram distance function; the CDF with the *pq*-gram distance function; and the New-CDF, in which the hierarchical distance functions were replaced by the *pq*-gram distance function.

TABLE 5.4: Silhouette and C-index values for the 3 clusters of 200 patent data records

	The Silhouette Index	C-Index
Using CDF without pq-gram	0.2405	0.0072
Using CDF with pq-gram	0.1858	0.0076
Using the New-CDF	0.2020	0.000028

TABLE 5.5: Silhouette and C-index values for the 4 clusters of 500 patent data records

	The Silhouette Index	C-Index
Using CDF without pq-gram	0.2209	0.0159
Using CDF with pq-gram	0.1808	0.0108
Using the New-CDF	0.2295	0.001

The Silhouette index value for the clusters resulted from using the *pq*-gram distance metric with the CDF, showing a lower compactness of clusters compared to the values when using the other distance functions. Alternatively, the C-index value for the

TABLE 5.6: Silhouette and C-index values for the 5 clusters of 1000 patent data records

	The Silhouette Index	C-Index
Using CDF without *pq*-gram	0.2225	0.0357
Using CDF with *pq*-gram	0.1722	0.0211
Using the New-CDF	0.2105	0.00104

clusters, which resulted from using the New-CDF, shows the lowest value referring to good clustering, despite the lack of interpretability of the figures produced using it.

Since the figures resultant from using the New-CDF were not possible to be interpreted, although the indices value were relatively good, in the next section, those figures will be presented using a different method by using different data input to the *clusplot* function.

5.8 Using a Different Method to Compare the Results

In this section, a different method is used to compare the results of clustering the patent data using the CDF with the *pq*-gram distance function and the New-CDF. The method is to use another type of input data to the *clusplot* function. The function according to [76] can have a matrix or a data frame input, where the latter input is used in this section. While in Section 5.6, the data input to the plotting function was a matrix of distance values resultant from the BIRCH algorithm. For only two samples of the patent data set, 200 and 500 data records, the figures using both CDFs were compared as follows:

5.8.1 The new figures for 200 data records using both CDFs

Figures 5.10 and 5.17 in the previous sections produced different shapes of clusters and different distribution of the clusters data points as well as the x and y axes of the figures. Both figures resulted from plotting distance matrices using the *clusplot* function. The *pq*-gram distance function along with the CDF and the New-CDF produced different distance matrices, which makes the plotting function produce different clusters. Despite

all the above, the clusters of both figures are almost the same in having the same data points in each cluster. Thus in order to show the clusters of both figures in the same shapes, axes and data point distributions, this section illustrates another way of plotting the clusters by using the data input as a data frame to the *clusplot* function.

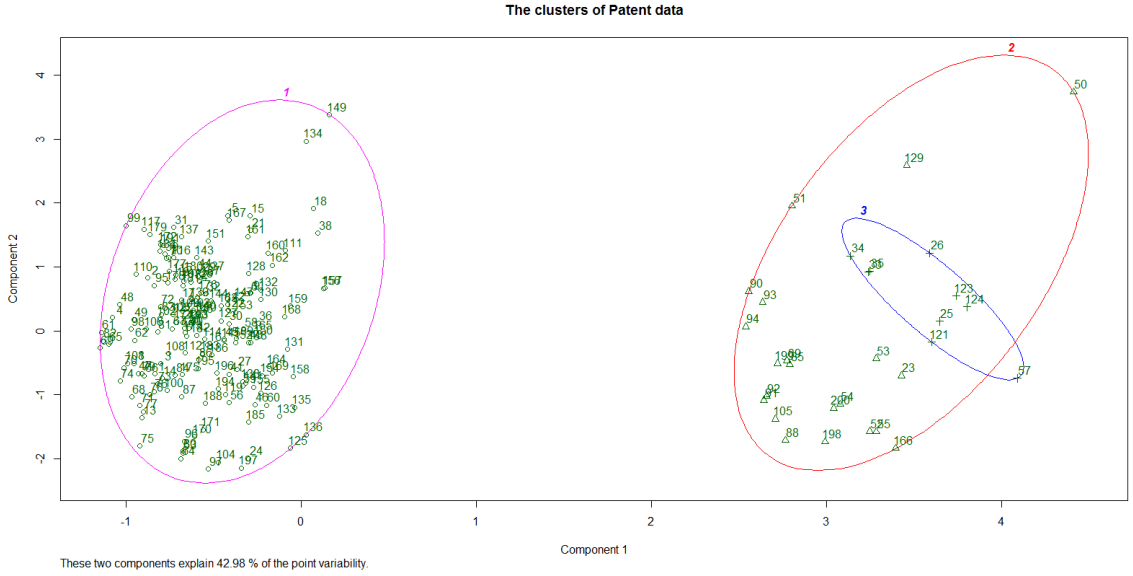


FIGURE 5.20: Clusters of 200 patent data records using the CDF with pq-gram

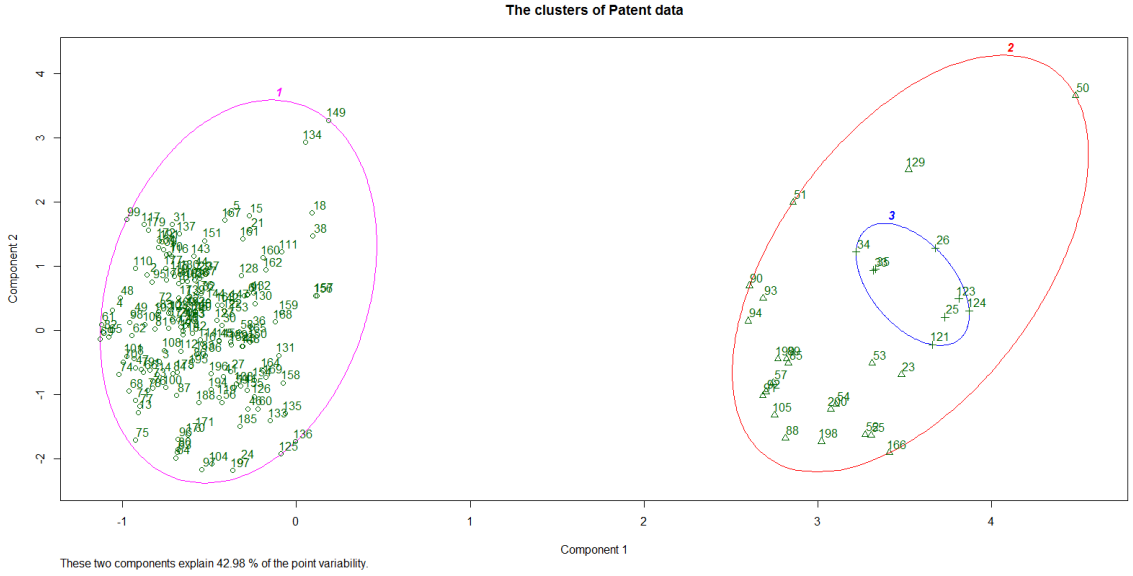


FIGURE 5.21: Clusters of 200 patent data records using the New-CDF

The new results shown in Figures 5.20 and 5.21 can be compared now, since both have the same data point distributions, the same axes and produced reasonable shapes of clusters. Cluster number 1 is exactly the same in both figures, which includes the

patents that are similar in different data attributes but all share the same CAT number. The difference between the figures is in the clusters number 2 and 3, which show an overlapping between almost all the data records except one record in Figure 5.20. This overlapping is because the data records in each cluster are suitable to share the data attributes from both clusters. For example, if the data records were gathered in cluster number 2 because of the CAT number, for instance; and the data records in cluster number 3 because of the *pq*-gram distance values. The overlapping could be because the same data records in cluster number 2 might have the same *pq*-gram distance values that need to be in both clusters.

In general, the New-CDF could not produce better, but almost the same, clusters than that when using the CDF with *pq*-gram distance function. This was the same conclusion when comparing the Figures 5.10 and 5.17 in the previous sections. In other words, the New-CDF has been found not useful for the patent data set, which need to use another data set that might give better clusters, such as the mixed THIN data set.

5.8.2 The New Figures for 500 Data Records Using Both CDFs

In this section, the same process is done as in the previous section, but with 500 data records that are clustered to 4 clusters. Figures 5.22 and 5.23 can be compared as they are similar in the axes limitations and the data record distributions; Both figures are similar in cluster number 1 which has all the data records that are similar in different data attributes and all share the same CAT number of the patents that they belong to that is 1. The other clusters are the difference between the figures, which show the best separable and compact clusters in Figure 5.22. The same reason as in the previous section could be for the overlapping between the clusters and that the data records in each cluster are suitable to share the data attributes from the other clusters.

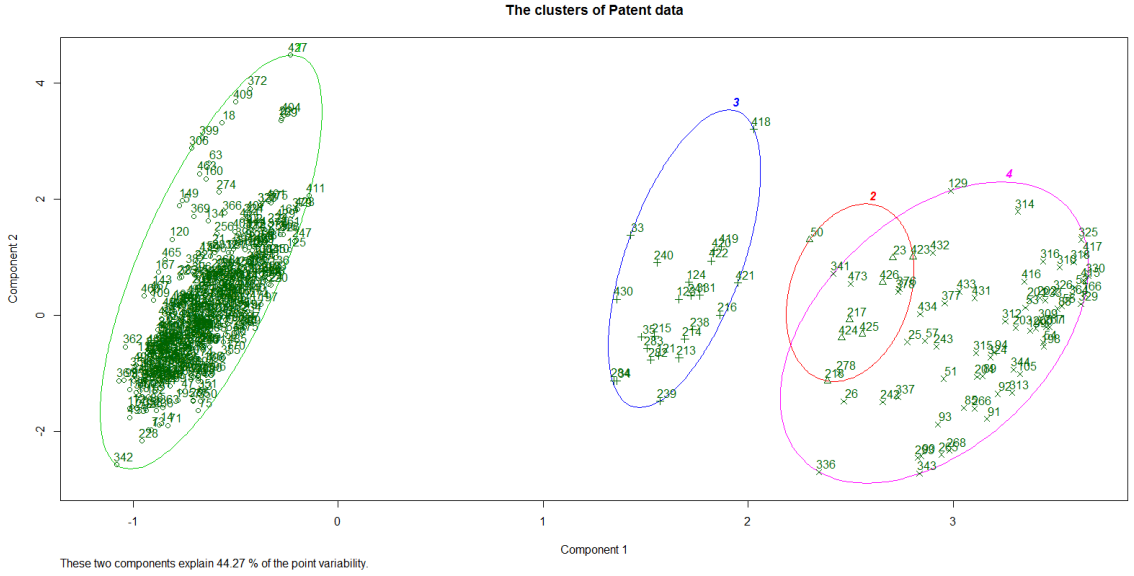


FIGURE 5.22: Clusters of 200 patent data records using the CDF with pq-gram

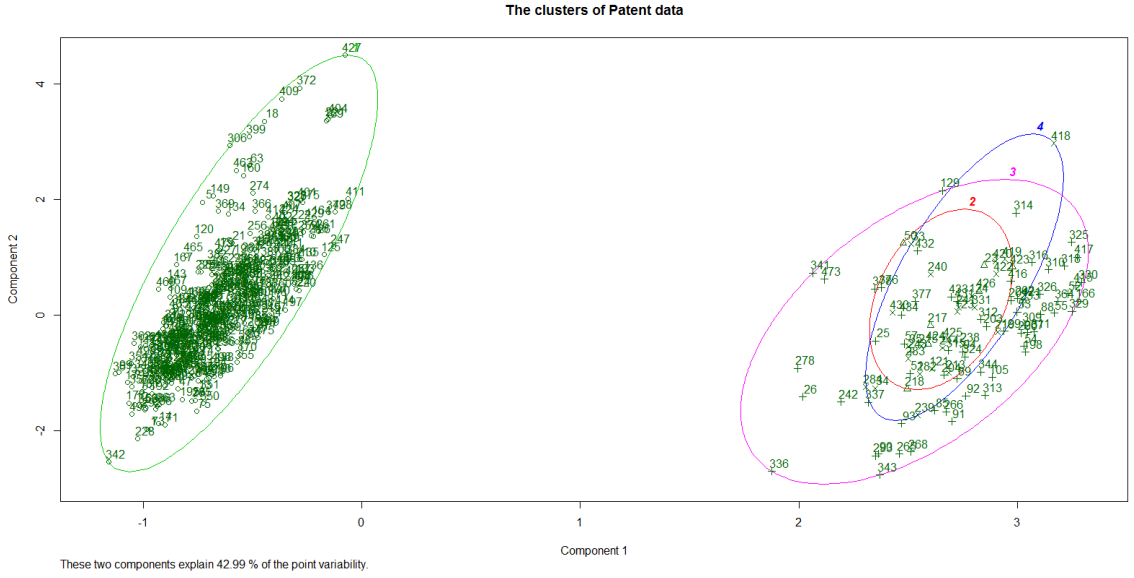


FIGURE 5.23: Clusters of 500 patent data records using the New-CDF

5.9 Discussion

In the previous chapter, the *pq*-gram distance metric was applied to different data sets, non-hierarchical, hierarchical and mixed. The data sets had to be converted to hierarchical structures to suit the metric because the *pq*-gram metric was only designed for hierarchical data [5]. The performance of the metric was not suited to non-hierarchical data sets, and was relatively good for a hierarchical data set. However, the metric did

produce similarity for the mixed THIN data, i.e. the performance was good; although, compared to some other distance metrics it was not the best. Therefore, the performance of the metric has to be improved to deliver better results by investigating different distance measures together for mixed data types' inputs.

The improvement intended pertains to using the *pq*-gram distance metric to measure the hierarchical attributes only, as other distance measures apply for the other data attributes. It was hoped to combine the measures into one distance function to create a new distance measure for measuring the similarity between the data set records. The hierarchical NBER US Patent data set was explored in this chapter by applying the new method. In addition, the simple non-hierarchical Iris data set was explored for the purpose of method validation and verification (for information on both data sets see Chapter 3, Section 3.2 and Section 3.3).

The BIRCH clustering algorithm was also applied to the data sets. It is a well-known and fast hierarchical clustering algorithm for use with large data sets. The algorithm takes each record and iteratively builds a hierarchical tree of clusters called a CF tree (see definition 2.1). As the tree is built, each record is inserted into the closest cluster using the CDF (as described in Chapter 2, Section 2.3.6.1). Before using the *pq*-gram distance metric in the CDF, the distance functions used in this application and combined into one distance function were the hierarchical distance function D_{HSM1} for the hierarchical attributes, the Euclidean distance function for the numerical attributes and a histogram of frequencies for the categorical attributes. Then, the *pq*-gram distance metric was tested twice; in the first test, the *pq*-gram metric worked along with the CDF, i.e. both D_{HSM1} hierarchical measure and the *pq*-gram metric were used to test the hierarchical attributes. In the second test, the hierarchical measures in the CDF were replaced with the *pq*-gram metric and worked in combination with the other non-hierarchical measures to create New-CDF. In both cases, the *pq*-gram metric considered an entire database entry (this is how the *pq*-gram works according to [5]), while the other hierarchical measures considered only one column in each data entry (record).

The simple non-hierarchical Iris data set is used as a validation data set to determine if the BIRCH algorithm could produce classes of Iris (setosa, versiclor and virginica flowers) that would be almost the same as the original using CDF. The data was clustered and the accuracy of the data obtained was 0.88 (see Section 5.4). This value was based on the comparison between the BIRCH cluster assignments, the Iris data and the real Iris classification, to determine the level of agreement between them. Although the accuracy value was not as high as the value when using the classifiers, it seemed to be a good result when clustering the data by assuming the maximum value 1. The new measure produced better results and almost the same classes.

The second data set used in this chapter was the large hierarchical NBER US Patent data set, and the aim of the clustering was to group the data into a number of clusters that might have some similarity; for example, clustering patents by the same inventor and distinguishing them from patents by other inventors. At the beginning, two samples were chosen for clustering using BIRCH (200 and 1000 data records). The 200 data records were clustered into three clusters, and the 1000 data records were clustered into five clusters. The clusters were then plotted using a plotting function in R software *clusplot*, which is a bivariate plot for graphical display. In order to explore the quality of the clusters, two internal indices were employed: Silhouette coefficients and the C-Index. Based on the index values, it was found that some clusters are better than others, and the most similar data records were arranged into one cluster. For example, Figure 5.8 showed three clusters labelled 1, 2 and 3, and cluster number 1 contained more similar data records (see Table 5.3) than the other clusters, such as for patents belonging to the same inventor or inventors who share some similarity in their names or who are from the same city and patent category.

The *pq*-gram distance metric was also used in two ways to find the distances while clustering the patent data sets. An example of 500 patent data records was provided for clustering into four clusters (see Section 5.5.2). The example showed that clustering this data using the *pq*-gram metric along with the CDF (Figure 5.12) produced better clusters than the CDF without the *pq*-gram metric, as shown in Figure 5.11. Figure 5.13 gave much better results when a weight was assigned to the *pq*-gram metric than without

a weight. In both cases when using the *pq*-gram metric along with the CDF, it can be argued that using the metric was useful, as it provided compact and well-separated clusters and there were similar data records in some clusters.

The aim of adding weights to different aspects of the CDF was to discover which attribute would be most strongly emphasised when the CDF is used to calculate the distances (see Equation 2.16). In the examples provided in this chapter, the weighted *pq*-gram distance function did not positively affect the clusters resulting from Figures 5.9 and 5.15, except in the example of clustering the 500 patent data records into 4 clusters. In the latter example, the clusters that resulted from using the CDF along with the weighted *pq*-gram distance function were improved compared to when not using the *pq*-gram metric (see Figures 5.12 and 5.13 in Section 5.5.2). From this it was found that using the weighted *pq*-gram distance function to work alongside the CDF sometimes gives useful outcomes and sometimes not. Therefore, there is no clear answer to the question: Can the *pq*-gram distance metric be combined with the other distance measures (the hierarchical and non-hierarchical) to find the similarity between data records that have different data types, and what are the influences of adding weights to different aspects of the CDF, particularly the *pq*-gram distance function? Thus, the suggestion was to replace the hierarchical distance functions with the *pq*-gram distance function, which would then be combined with the other distance functions for non-hierarchical attributes to investigate if this delivered better outcomes.

The New-CDF was used to measure the similarity between the hierarchical Patent data records, and the results (see Figures 5.17, 5.18 and 5.19) revealed that the New-CDF did not give effectively compacted and separated clusters when compared to the previous results attained when using the *pq*-gram distance function along with the CDF. Thus, because of the good indices values obtained using this measure (as illustrated in Tables 5.4, 5.5 and 5.6), the clusters have to be plotted in another way in order for the figures to be compared. The new plotting was to use the data frame as input to the *clusplot* function instead of the distances matrix attained from clustering the data set using the BIRCH algorithm. New figures were produced and could be compared when using the two CDFs (CDF with pq-gram and New-CDF). Despite all the aforementioned and

based on what was concluded in the previous chapter, the patent data set is often lacking data records regarding something or someone, which affected the *pq*-gram metric performance. The same drawback could explain why the New-CDF did not perform well in this chapter for the large patent data set.

5.10 Conclusion

To conclude, the BIRCH clustering algorithm could produce groups of similar data records with at least one attribute in common. In addition, different versions of the CDFs (CDF, *pq*-gram along with CDF, weighted *pq*-gram along with CDF and New-CDF) performed differently with the large hierarchical Patent data set. However, the result when applying the new measure to the non-hierarchical data set was better than that when using the hierarchical *pq*-gram metric in Chapter 4. Similarly, the performance of the new measure for the hierarchical patent data set was also better, except when New-CDF was used, as the resulting clusters could not be interpreted.

In the next chapter, the new method will be applied to a large mixed data set, the UK medical THIN data set. Since the results and the performance of the new measure were better than the performance of the *pq*-gram distance metric set out in the previous chapter, it is anticipated that the new measure may give better results here also.

Chapter 6

Apply the New Measure to A Mixed Data set

6.1 Introduction

In Chapter 4, the *pq*-gram distance metric was applied to different data sets, including The Health Improvement Network (THIN) data set, by converting the data into a tree-like structure. For the latter data set, the results showed the *pq*-gram distance metric produces good performance, but improvements are required to obtain better results. Thus, Chapter 5 evaluated a new distance measure created by combining the *pq*-gram distance metric and other distance measures for mixed data type records. This new measure was then applied to non-hierarchical and hierarchical only data sets and the results were found to be much better. This provides the motivation to apply an improved distance metric to the mixed data set THIN.

Therefore, the question in this chapter concerns the new measures that are expected to be useful for measuring the similar distances in the mixed THIN data set. Clustering using the BIRCH algorithm and measurements of the similarity distances between data records using the CDFs are applied to the THIN database. Both the CDF with and without the *pq*-gram, as well as New-CDF, are used to ascertain the distances between data records, and the intention is to find groups of similar patients within the medical

data, e.g. those sharing symptoms or side effects, or those records containing specific drugs and other information.

6.2 Data Pre-processing

In this section, information extracted from the THIN data for a number of drugs called DOXEPIN, DESLORATADINE and LEVOFLOXACIN is used. DOXEPIN belongs to the family of Tricyclics that relate to antidepressant drugs. DISLORATADINE is used to treat allergies within the group of Antihistamines, and LEVOFLOXACIN belongs to the Quinolone family, which treats bacterial infections [83]. The drugs sampled are assessed for patients under 18 years old, who are considered children. This group age is chosen because the clinical trails for them are difficult and any algorithm or new method that could work for this age group will be more valuable. The sample size for DOXEPIN after pre-processing is small and contains 1172 data records representing information from 42 patients. While for DESLORATADINE, the number of prescriptions in the chosen age but within 30 days starting from the date of prescription concerns 9949 data records. The sample size chosen for the experiments in this chapter is 1000 data records, representing 148 patients, because this sample can be sufficient to represent the similar distances between patients' data. For the pre-processing of both drugs, DOXEPIN and DESLORATADINE, see Table 4.2.

For the third drug used in this chapter, LEVOFLOXACIN, by doing the same pre-processing as for DOXEPIN and DESLORATADIN, the size of the data by prescriptions is 576 data records in total representing 14 patients aged 0-17 years old. The samples of the THIN data for the three drugs are shown in Appendix A, Figures A.16, A.17 and A.18, respectively. In the Figures, the attributes (FIELD1, FIELD2, FIELD3, FIELD4 and FIELD5) are corresponding to the 5-tuple of the extracted THIN data sets (combid, sex, medcode, age at intake and time of event) for the three drugs used.

The data was clustered using the BIRCH clustering algorithm. First with the CDF for hierarchical, numerical and categorical data, and then with the *pq*-gram distance metric alongside the CDF. The input data records consist of the following data attributes:

combid, sex, medcode, age at intake, and time of event. Combid is the patient identifier (ID) and is hierarchical, medcode is the medical code (READ code) for the symptoms and is also hierarchical, and sex is the categorical data type. Both variables (age at intake) and (time of event) are numeric data types, and refer to age when taking the drug and the time when the medical event occurred after taking the drug within 30 days starting from the date of prescription.

In the next section, the CDF for two hierarchical attributes, one categorical attribute and two numerical attributes (D_{HSM1}, the histogram of the frequencies of the attribute values, D_{HSM1}, Euclidean Distance Function, Euclidean Distance Function) is described as a measure of the distance similarity between patients' data records. For the drug DOXEPIN, the data is clustered into between 5 - 10 clusters to show how the quantity of data grouped each time in the clusters increases, and to show how far the data records are similar to each cluster when using CDF. The Figures for the 5-9 clusters will be shown in Appendix A, while the figure of 10 clusters will be shown in the next section. Figure 6.1 exemplifies the extracted THIN data.

6.3 Distance Calculation and Clustering Using the BIRCH Algorithm for DOXEPIN Drug

6.3.1 The Distance Calculation Using CDF

In this section, the distances between the 1172 data records for the patients who took DOXEPIN were calculated using the CDF, and clustered using BIRCH clustering algorithm. The distance functions were the D_{HSM1} for the hierarchical attributes, the Euclidean distance function for numerical attributes, and the histogram of the frequencies for the attribute values for the categorical attribute. The number of clusters chosen here are 10 and 42 clusters.

The results when clustering the 1172 THIN data records into 5-10 clusters reveal that the smaller the number of clusters the more compact and separable the clusters are,

Results | Messages

	combid	sex	medcode	age_at_intake	description	timeofevt
1	a793901c8	2	7530.11	6165	Tonsillectomy	4
2	a793901c8	2	J085.00	6165	Diseases of lips	0
3	a793901c8	2	AB01.13	6165	Fungal nail infection	7
4	a793901c8	2	H02..00	6165	Acute pharyngitis	5
5	a793901c8	2	H037.00	6165	Recurrent acute tonsillitis	3
6	a793901c8	2	H05z.00	6165	Upper respiratory infection NOS	0
7	a793901c8	2	R006200	6165	[D]Fever NOS	1
8	a793901c8	2	K190z00	6165	Urinary tract infection, site not specified NOS	2
9	a793901c8	2	T804100	6165	Accidental poisoning by paracetamol	2
10	a985600A2	2	SE...00	5871	Contusion (bruise) with intact skin	2
11	a985600A2	2	SK10500	5871	Other nose injuries	2
12	a985600A2	2	1M10.00	5871	Knee pain	8
13	a985600A2	2	4K23.00	5871	Cerv.smear: mild dyskaryosis	8
14	a985600A2	2	7L1L000	5871	Biopsy of lesion of unspecified organ	0
15	a985600A2	2	AD20.00	5871	Pediculus capitis - head lice	6
16	a985600A2	2	L20..11	5871	Spontaneous vaginal delivery	3
17	a985600A2	2	M05..00	5871	Impetigo	6
18	a985600A2	2	N094000	5871	Arthralgia of unspecified site	3
19	a985600A2	2	N13y300	5871	Cervical root syndrome	8
20	a985600A2	2	N327.00	5871	Osteochondritis dissecans	6
21	a985600A2	2	L100.00	5871	Threatened abortion	2
22	a985600A2	2	N094611	5871	Knee joint pain	6
23	a985600A2	2	M240.00	5871	Alopecia	0
24	a985600A2	2	K197.00	5871	Haematuria	6
25	a985600A2	2	K15..00	5871	Cystitis	7
26	a985600A2	2	R020600	5871	[D]Numbness	1
27	a985600A2	2	H05z.00	5871	Upper respiratory infection NOS	1
28	a985600A2	2	E200.00	5871	Anxiety states	1
29	a985600A2	2	E204.11	5871	Postnatal depression	3

FIGURE 6.1: Screenshot of a THIN data sample

as shown in Figures A.1 on page 163 to A.5 on page 165. However, the information about each patient in the data set is only well separated from the other patients' information if the number of clusters increases. For example, in Figure 6.8 on page 136 42 clusters contain information about 42 patients.

In order to show the similar data records in each cluster, an example given, as illustrated in Figure 6.5 on page 135. In the figure, cluster number 8, for instance, has 61 data records, representing information about 2 patients, f992302rM and i982304@w. Both patients have multiple symptoms from the same hierarchy in common, such as 1737 and 1737.11, which refers to the wheezing symptom, 173B.00 and 171..00 which refer to cough symptoms, F501.00, F527.00, F52..00 and F52z.00, which refer to the symptoms for infective otitis externa and H01..00, H02..00, H05z.11, H00..16 and H05z.00, which

all refer to a variety of mild and acute infection symptoms, like Acute sinusitis, Acute pharyngitis and Upper respiratory tract infection. It is possible some clusters might not have good similarity, but in these results each cluster shares some similarities with the data records with regard to at least one attribute. Thus, it is worth noting that some clusters are better than others at sharing similarity between the data records.

The next section presents the same scenario as the previous section, but adds the *pq*-gram distance metric for the hierarchical attributes to the CDF. The 1172 THIN data records were clustered into 5-10 clusters using BIRCH algorithm to discover how many groups the algorithm will suggest as providing a clear classification and to show the effect on the results when adding the *pq*-gram distance metric to the CDF.

6.3.2 Using Pq-gram Metric With the CDF

In this section, the 2, 2-gram metric for hierarchical attributes, along with the measures of other attributes are used in the CDF to find similar distances between the 1172 THIN data records representing 42 patients taking DOXEPIN. The data was clustered into 5-10 clusters using the BIRCH algorithm to show what the clusters have in common and to provide a clear example of how far the new metric influences the resultant clusters. For example, Figures 6.2, 6.3 and 6.4 show how the clusters to the right side of the figures are divided, as the number of the clusters increases such that the patients who have the most similar data records stay in one cluster and non-similar ones remain separate from them. The left side of the figures remains unchanged, since each cluster has either data for only one patient or has the same number of patients, each with similar data records for most attributes. The figures for the 5 and 6 clusters will be shown in the Appendix A. Later, a comparison will be performed to see the differences when using the *pq*-gram metric with and without the CDF. In the examples, the values are p = 2 and q = 2, so that the *pq*-gram is 2, 2-gram.

The results of the clustering in this section with the existence of the 2, 2-gram metric, along with the CDF and without the existence of 2, 2-gram in the previous section show that the clustering of 5-10 clusters gives quite good separation in the clusters.

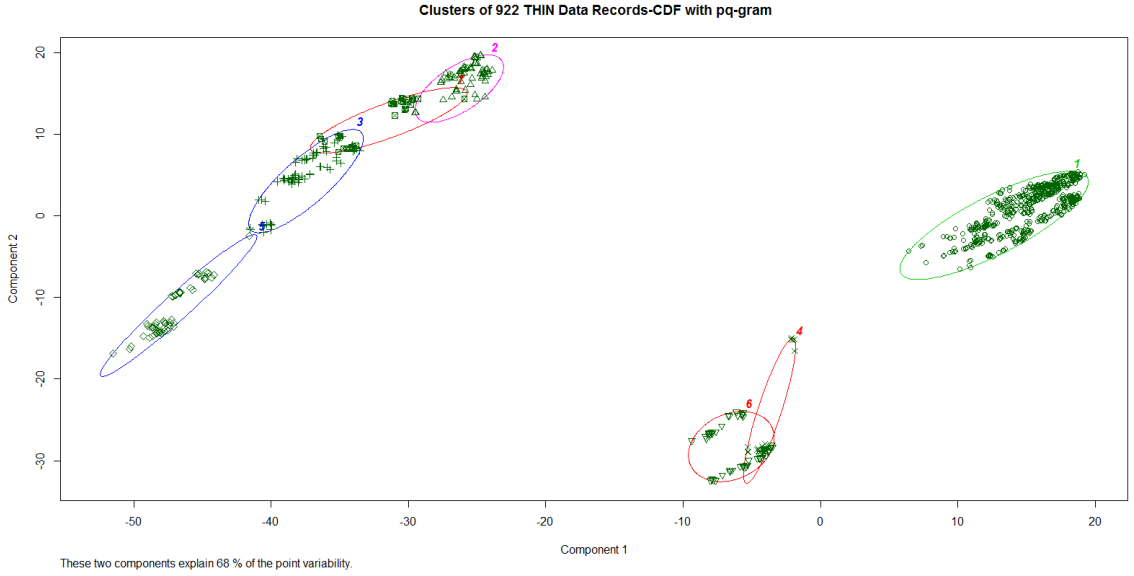


FIGURE 6.2: The 7 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram

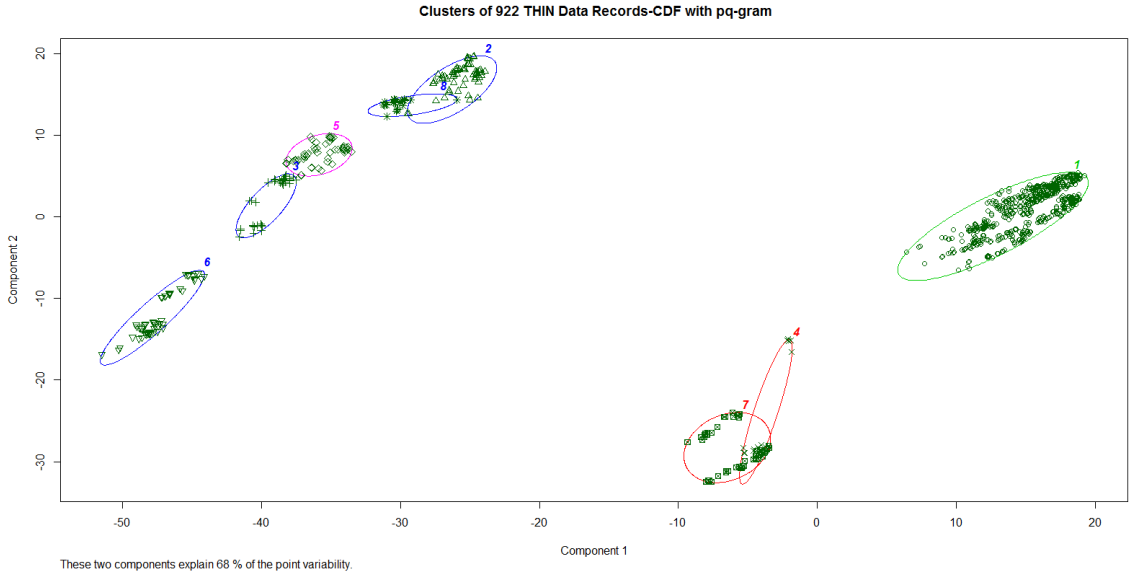


FIGURE 6.3: The 8 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram

However, it is difficult to say which results are better or worse, although based on the index values (see Table 6.1), some difference can be noted. In general, the clustering without the *pq*-gram is slightly tighter than with the *pq*-gram, except that in the last two figures for the 9 and 10 clusters, as in Figures 6.4 and 6.6, where the index values shown are better. The data records that relate to the patients lie in one cluster and are more similar to each other in most of the attributes, for example, cluster number 8 of

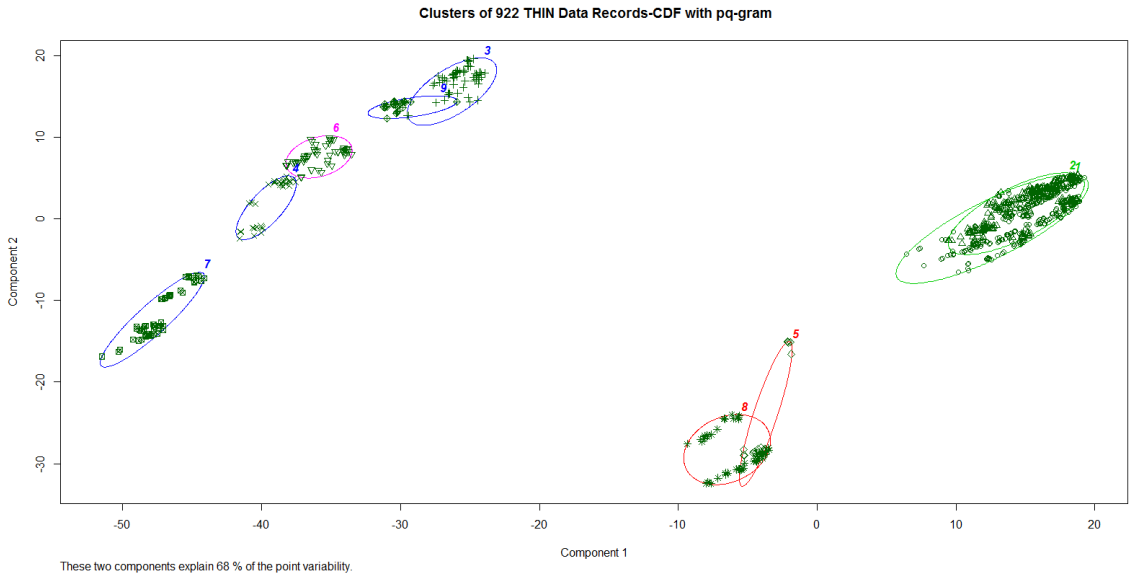


FIGURE 6.4: The 9 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram

Figure 6.6 on page 135.

6.3.3 Adding Weights to the CDF

Using Equation 2.16 this section describes the results when adding weights to the *pq*-gram distance function aspect of the CDF. An example of clustering the 1172 THIN data records into 10 clusters is used by adding the weight to the *pq*-gram distance function, then comparing it to the same example as that in Figure 6.6 on page 135. The difference is that clusters 3, 4, 6, 8 and 10 are better separated in Figure 6.7 than to those in Figure 6.6. Thus, some improvements can be noticed using this comparison and these could relate to the influence of using the weighted *pq*-gram distance function in the CDF.

In the example when clustering the 1172 THIN data records into 42 clusters, the Figures 6.8 and Figure 6.9 were presented. The first figure showed the clustering of the data using the CDF without the 2, 2-gram metric and the second figure showed the clustering of the data using the CDF with the 2, 2-gram metric. The two figures showed only slight difference, but both share the fact that each cluster contains information from only one patient. By adding weights to the aspects of the CDF lending the concentration

TABLE 6.1: Silhouette and C-index values for all the clusters

	Clusters	The Silhouette Index	C-Index
The Combined Distance Function Without 2,2-gram	2	0.3337	0.0192
	3	0.3636	0.0149
	4	0.3448	0.0150
	5	0.3593	0.01649
	6	0.3621	0.02006
	7	0.3192	0.0581
	8	0.3322	0.0628
	9	0.2215	0.07552
	10	0.2353	0.0928
The Combined Distance Function With 2,2-gram	2	0.2760	0.0179
	3	0.3083	0.0140
	4	0.2978	0.0151
	5	0.3126	0.0158
	6	0.3149	0.0196
	7	0.2930	0.0840
	8	0.2986	0.0853
	9	0.3157	0.0829
	10	0.3067	0.0816

to the *pq*-gram distance function, the clustering presents different pictures, as shown in Figure 6.10, comparing to the aforementioned figures. The majority of the clusters are well separated and each one, as in the previous figures, contains the records for just one patient.

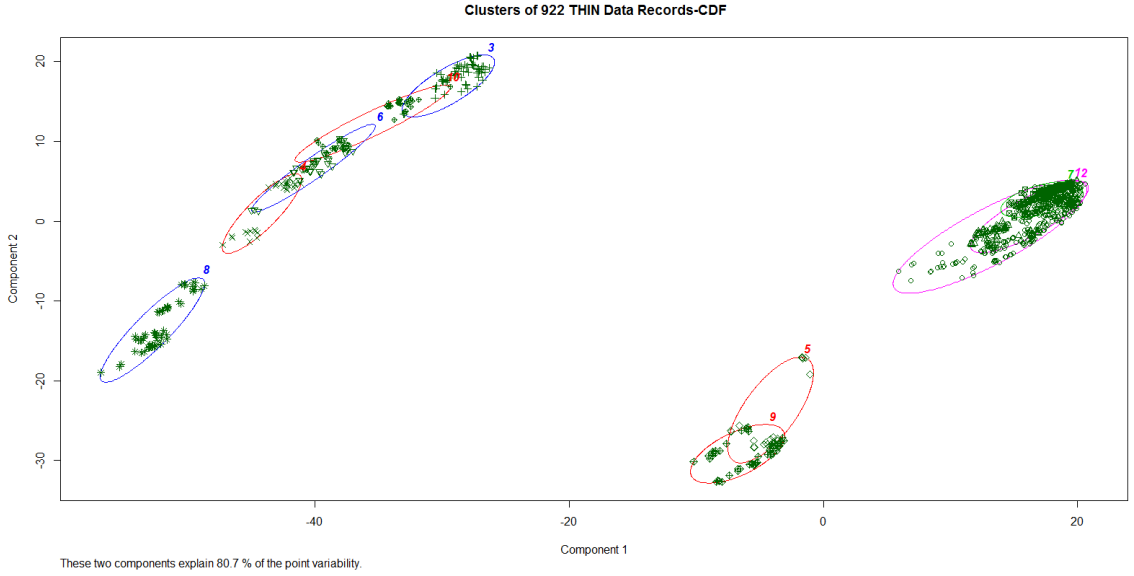


FIGURE 6.5: The 10 clusters of 1172 THIN data records-DOXEPIN drug

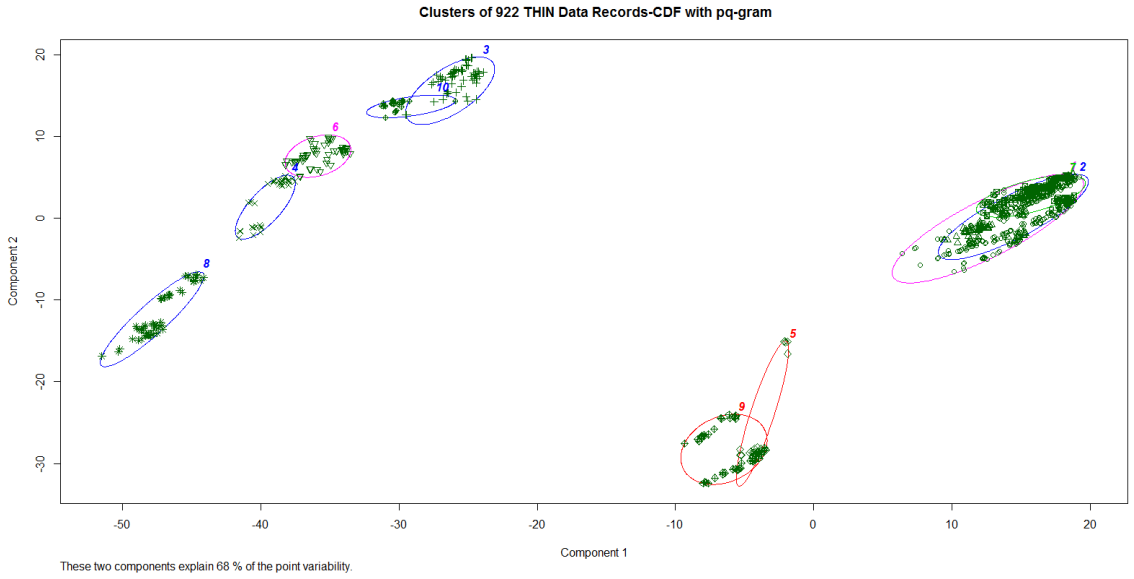


FIGURE 6.6: The 10 clusters of 1172 THIN data records-DOXEPIN drug Using CDF with pq-gram

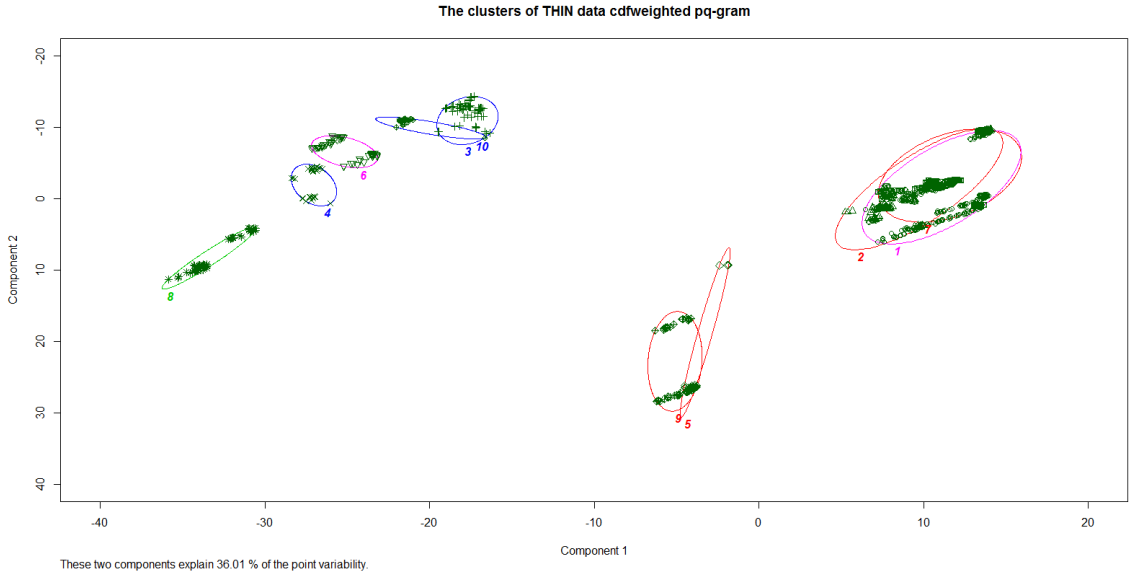


FIGURE 6.7: The 10 clusters of 1172 DOXEPIN drug data using CDF and weighted-pq-gram

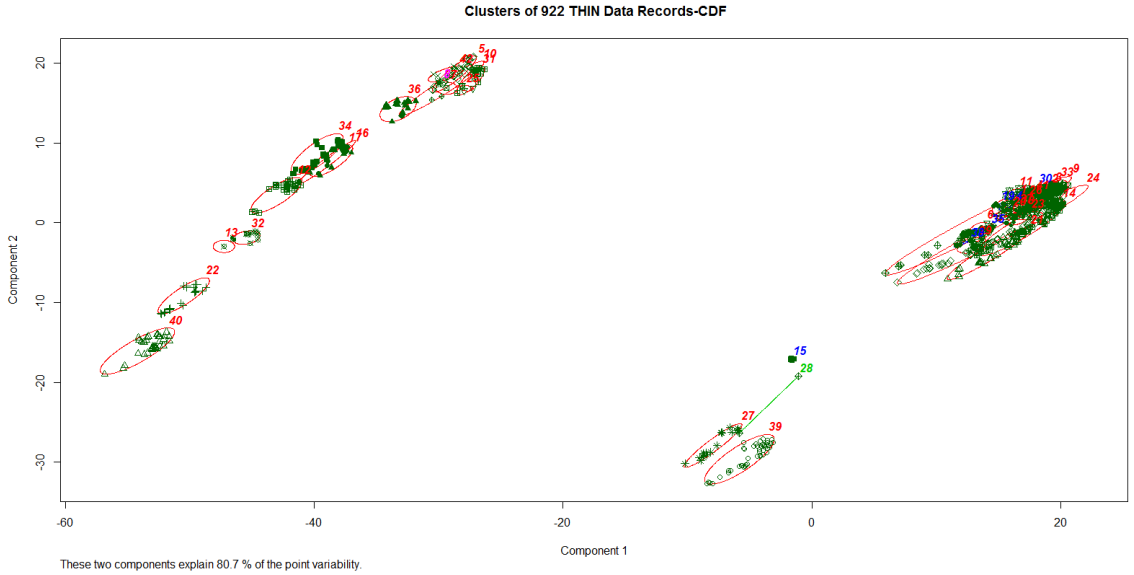


FIGURE 6.8: The 42 clusters of 1172 THIN data records-DOXEPIN drug representing 42 patients

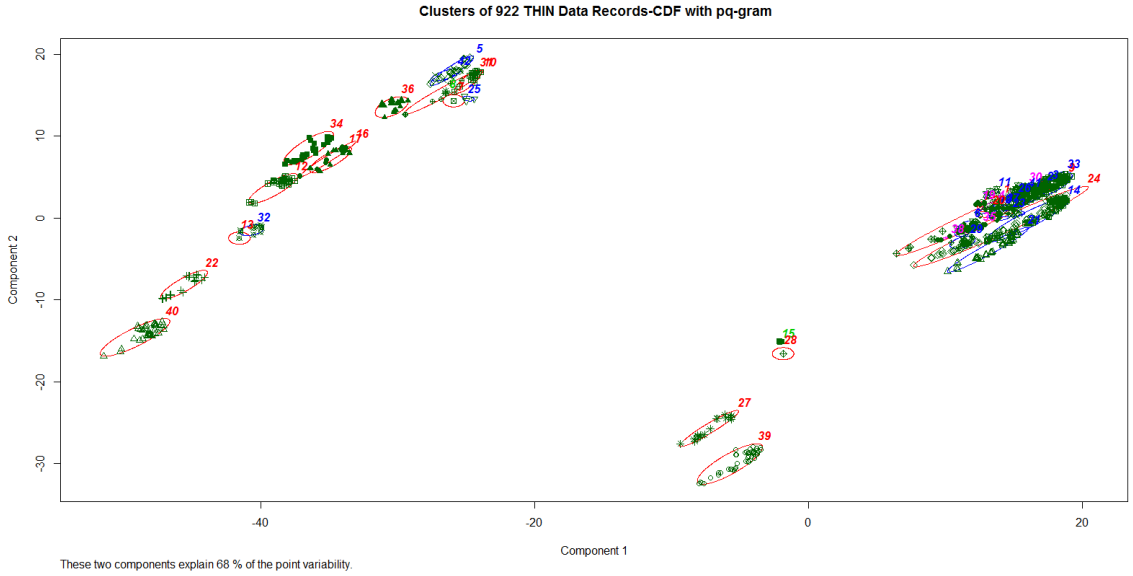


FIGURE 6.9: The 42 clusters of 1172 THIN data records-DOXEPIN drug Using CDF and pq-gram

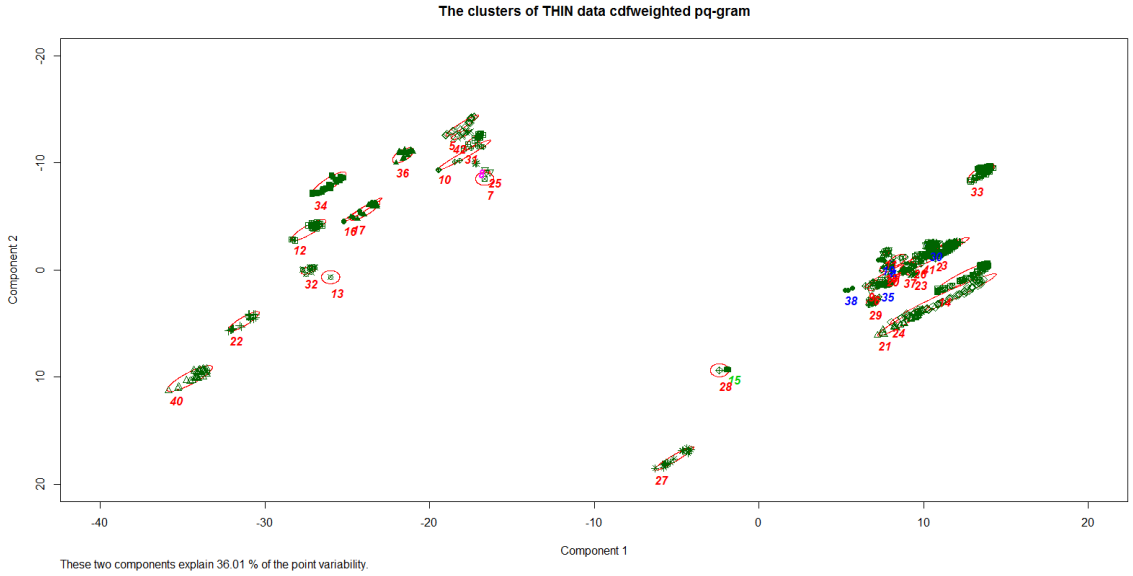


FIGURE 6.10: The 42 clusters of 1172 THIN data records-DOXEPIN drug Using CDF and weighted-pq-gram

6.3.4 Replacing the Hierarchical Measures by the Pq-gram Distance Function

In the previous sections, the role of CDF with the *pq*-gram distance function when measuring the similarity between the data records was slightly better than when not using *pq*-gram. Some clusters were better separated than others using the CDF with the 2, 2-gram distance function, as shown before in Figure 6.6.

This section explores the idea of replacing the hierarchical distance function (D_{HSM1}) with the *pq*-gram distance function and creating a New-CDF using only the *pq*-gram distance function for hierarchical data attributes. The New-CDF comprises the *pq*-gram distance function for hierarchical attributes, the Euclidean distance function for numerical attributes, and the histogram of frequency of the categorical attribute values. The same examples as used when clustering the 1172 data records into 10 clusters and 42 clusters in the previous sections were used here with the New-CDF. The example of clustering the data set to 10 clusters, which is shown in Figure 6.11, using New-CDF shows very overlapped clusters that cannot be interpreted. This gives the first insight that the New-CDF is not able to provide any reasonable clusters, thus more examples for other drugs have been used in this chapter to see if it can produce better clusters.

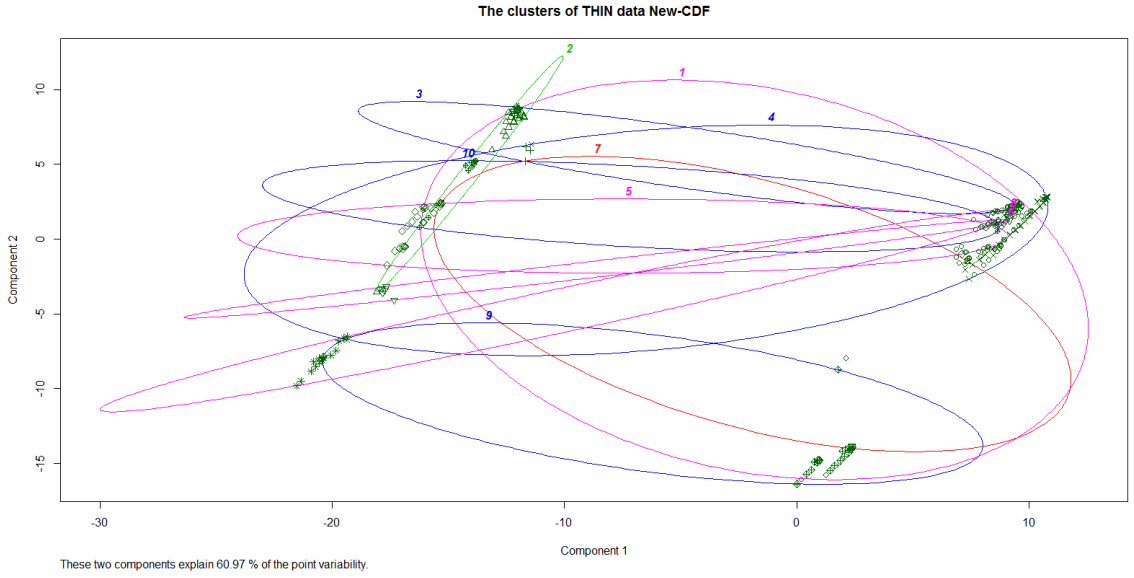


FIGURE 6.11: The 10 clusters of 1172 THIN data records-DOXEPIN drug Using New-CDF

To facilitate comparison between the figures in the previous section and the figures in this section, the figures in the previous section are re-plotted using only two numerical attributes from the data set from the original 1172 data records: the age of the patients, and medical events that occurred within 30 days of taking the drug. Further, the figures resulting when using the New-CDF in this section were plotted over the same period, using the same two numerical attributes data sets, to compare the figures when using the different distance functions (CDF with the *pq*-gram and the New-CDF).

Figures 6.12 and 6.13 were plotted based on two numerical attributes extracted from the 1172 THIN database sample for the DOXEPIN drug: the age of the patients and the timing of any events within 30 days of taking the drug. For both figures, PCA was applied to the data, and the x, y-axes are represented by the first two principal components, which show the underlying structure of the data, i.e. directions where there is the most variance and directions where the data is most dispersed (measured by the standard deviations). In other words, the *clusplot* function has been forced to be plotted based on the principal components resulting from the PCA. Since there are the only two attributes in the sample data set, the x-axis might represent the age of the patients in days for those aged 0-18 years old and the y-axis might represent the date of an event. By using the PCA, the original data points can be deconstructed into eigenvectors and eigenvalues, where the eigenvectors and values exist in pairs: every eigenvector has a corresponding eigenvalue. An eigenvector is the direction, and an eigenvalue is a number that describes how much variance there is in the data in that direction.

The first figure resulted when clustering using the BIRCH algorithm and the *pq*-gram distance function, along with CDF, and the second figure resulted from using the New-CDF. Both figures show identical clusters were produced; e.g. cluster 5, 8 and 9 in the first figure and clusters 7 (corresponding to cluster 5), 8 and 9 in the second figure. The other clusters vary between the two figures; for example cluster number 6 in the first figure and, correspondingly, cluster number 5 in the second figure. Cluster number 6 consists of 3 patients' records, which all share the same attributes of sex and some med-codes; these are either identical or from the same hierarchy, e.g. 8B2D.00, 8B3H.00,

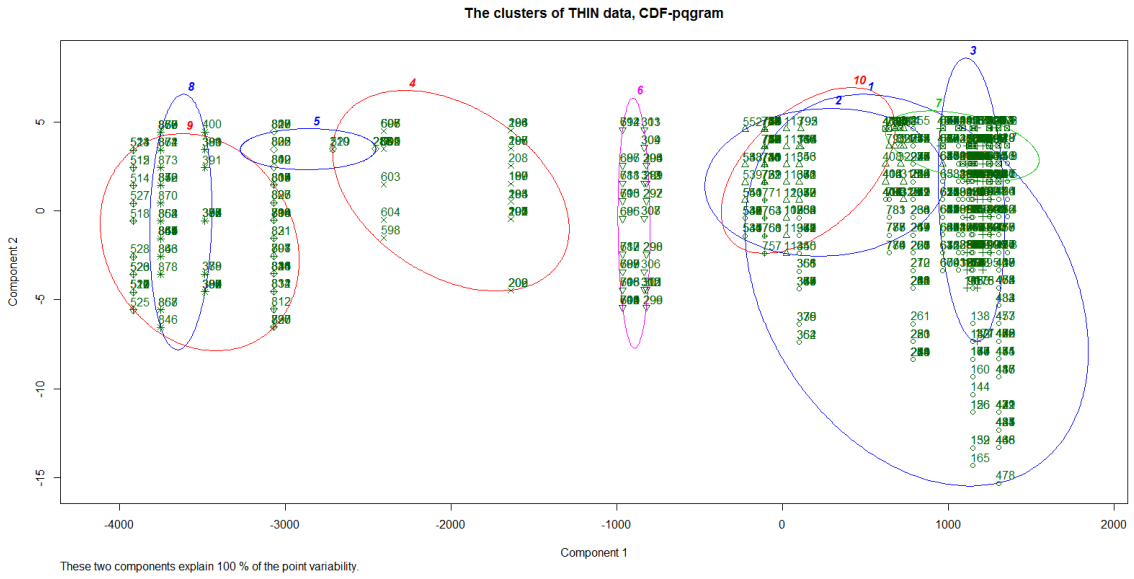


FIGURE 6.12: The 10 clusters of 1172 DOXEPIN drug data using CDF-pq-gram

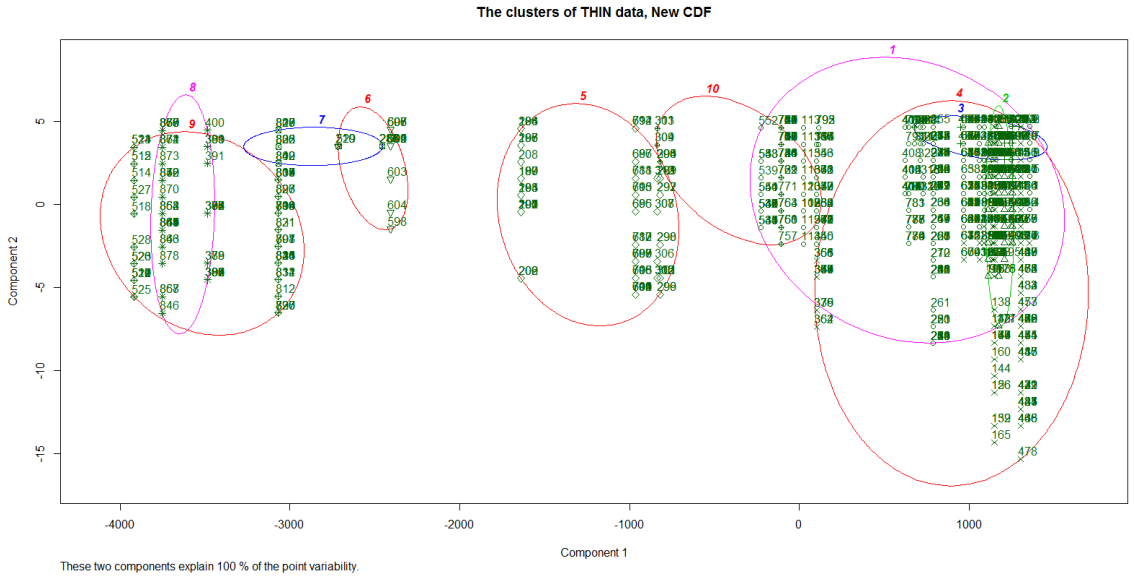


FIGURE 6.13: The 10 clusters of 1172 DOXEPIN drug data using New-CDF

8B41.00 and 8B3V.00. While cluster number 5 adds additional patient records to the other three in cluster 6, these share gender in common with the 8B3H.00 and 8B31400 medcodes. Thus, using New-CDF is useful as it seeks to put patients who have high data similarity into one cluster.

6.4 Distance Calculation and Clustering Using BIRCH Algorithm for DESLORATADINE

This section introduces the example of another drug from the THIN data set; the drug is DESLORATADINE. The distance calculation and clustering using the BIRCH clustering algorithm described here was accomplished using a subset of the drug's data comprising 1000 data records, representing 148 patients aged under 18 years old. For the 1000 data records to be clustered, the CDF and the CDF with *pq*-gram distance function, as well as New-CDF, were applied to measure the similar distances between the data records and cluster them using the BIRCH algorithm. The data records were clustered into 5 clusters and the parameter values given for the *pq*-gram distance metric were $p = 2$ and $q = 2$.

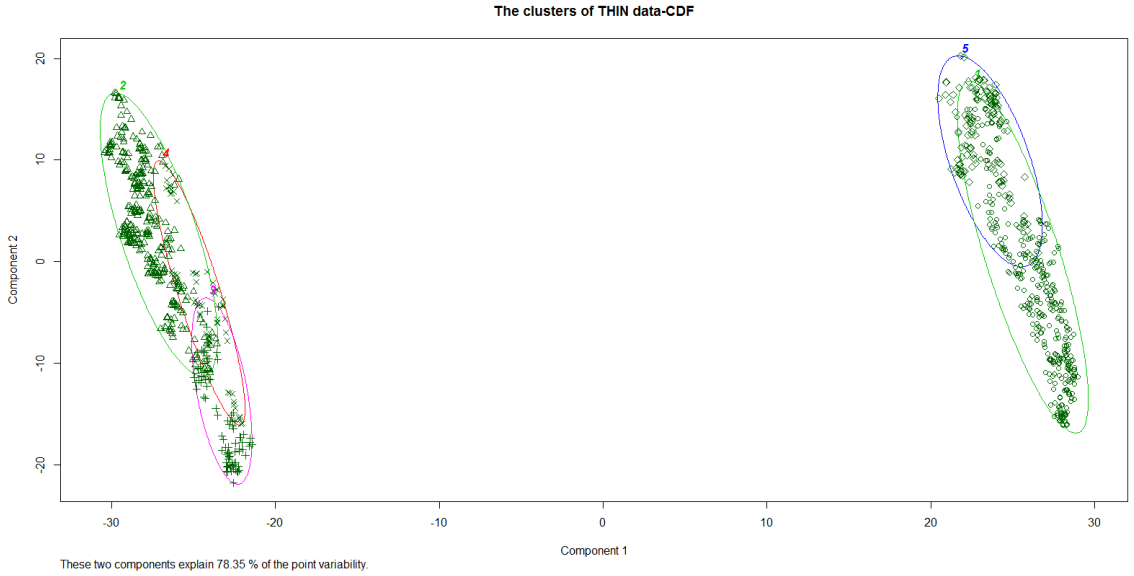


FIGURE 6.14: The 5 clusters of 1000 DESLORATADINE data records-using CDF

In Figure 6.14, 1000 data records were measured with CDF that does not use the pq-gram distance function. According to the figure, clusters numbered 1 and 5 are very close to each other and share considerable information between the data records, while clusters 2, 3 and 4 are closer to each other than the first two clusters. Cluster number 2 contains the data records for 45 patients, cluster number 3 the information for 18 patients and cluster number 4 the data records for 15 patients. In each cluster, the data records have many READ codes in common, as well as some other attributes, such

as the dates when medical events occurred after taking the drug DESLORATADINE. However, the overlap between the clusters means that the records share many similar data attributes. Generally, the five clusters are not well separated, but they do contain a lot of similar data, as well as between the patient records inside each cluster.

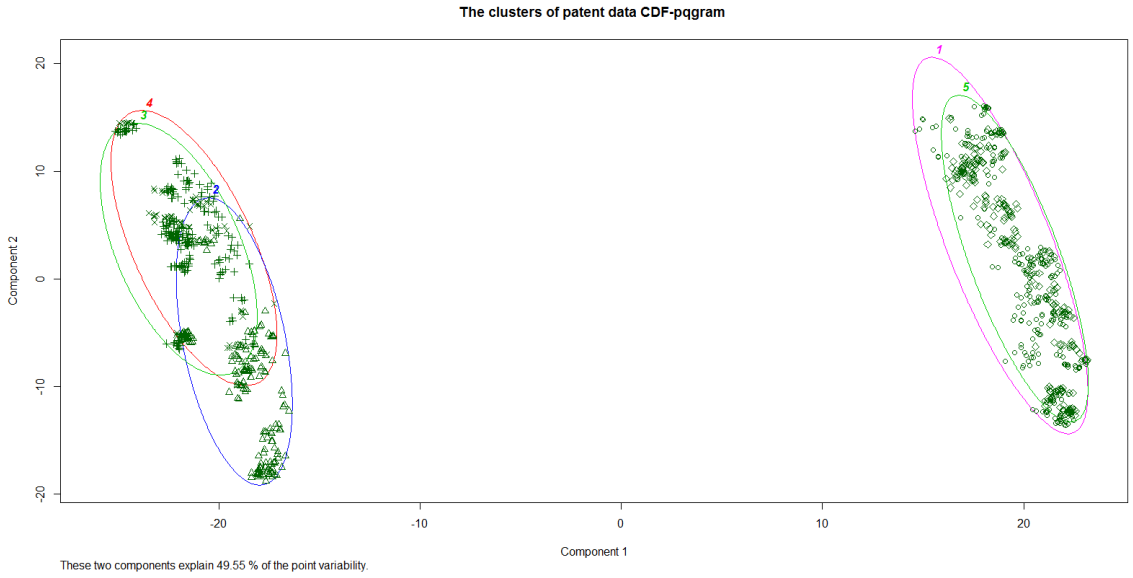


FIGURE 6.15: The 5 clusters of 1000 DESLORATADINE data records using CDF with 2,2-gram

In Figure 6.15, the pq-gram distance function is used along with the CDF to measure the similarity between the same 1000 data records, clustering them into 5 clusters using the BIRCH clustering algorithm. The overlap between the clusters is hard to interpret, since each patient's records are shared between more than one cluster, e.g. patient a6706013B who has four data records distributed between cluster 1 and cluster 5. The explanation for this is that this patient shares some similarity of information with the patients in cluster 1 and some similarity with the patients in cluster 5 . In general, the CDF with the 2, 2-gram cannot provide useful information or effective clustering for such an example.

Figure 6.16 shows the results when clustering the DESLORATADINE data records using New-CDF. In New-CDF, only the *pq*-gram distance metric is used for hierarchical attributes alongside the other non-hierarchical measures, i.e. the hierarchical measures in the CDF have been replaced with the *pq*-gram distance metric to create a new distance function. The replacement was achieved by not considering the hierarchical measure

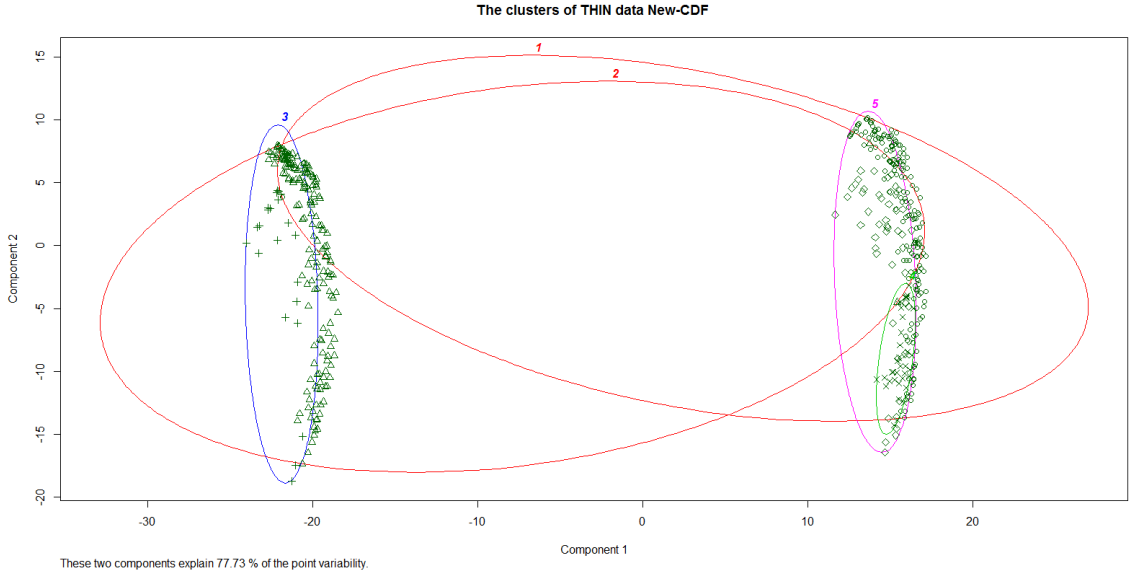


FIGURE 6.16: The 5 clusters of 1000 DESLORATADINE data records using New CDF

while calculating the distances, by giving the weight 0 to those aspects of the CDF, considering only the non-hierarchical measures with the *pq*-gram distance function.

The figure shows some overlapping clusters, contrasting with the previous figure, as seen in clusters 1 and 2. These clusters share some of the *pq*-gram distance values between a number of patients, as well as some symptoms and other attribute values. The *pq*-gram distance function contribution in this calculation led to this overlapping; since some clusters shared the same *pq*-gram distances values.

6.4.1 Using a Different Method to Compare the Results

It could be difficult to compare the results using the CDF with the 2, 2-gram distance function, as in Figure 6.15, and the results from using the New-CDF, as in Figure 6.16, since there is a difference in the positions of the data points and the axes of the figures.

In this subsection, the 5-tuple raw data, as described in Section 6.2, is reduced to only two numerical attributes; that is, the age of the patients and the timing of any medical event that occurred within 30 days after taking DESLORATADINE. Then the PCA is applied to the new data set and the results plotted using the *clusplot* function. The x, y-axes, as in the results reported in the previous section, are represented by the

principal components that display the underlying structure of the data. In other words, the *clusplot* function has been forced to plot based on the principal components resulting from the PCA. Since there are only two attributes in the sample data set, the x-axis represents the age of the patients in days for ages 0-18 years old and the y-axis represents the date of any event. Using the PCA, the original data points are deconstructed into eigenvectors and eigenvalues, where the eigenvectors and values exist in pairs, every eigenvector has a corresponding eigenvalue. The first figure results from clustering using the BIRCH algorithm and the *pq*-gram distance function, along with CDF, and the second figure results from using the New-CDF.

It has been found that it is also difficult for new figures to be interpreted well, because of the very overlapping clusters that were obtained. These are the same as the original figures, which means also that one patient's records are shared between more than one cluster.

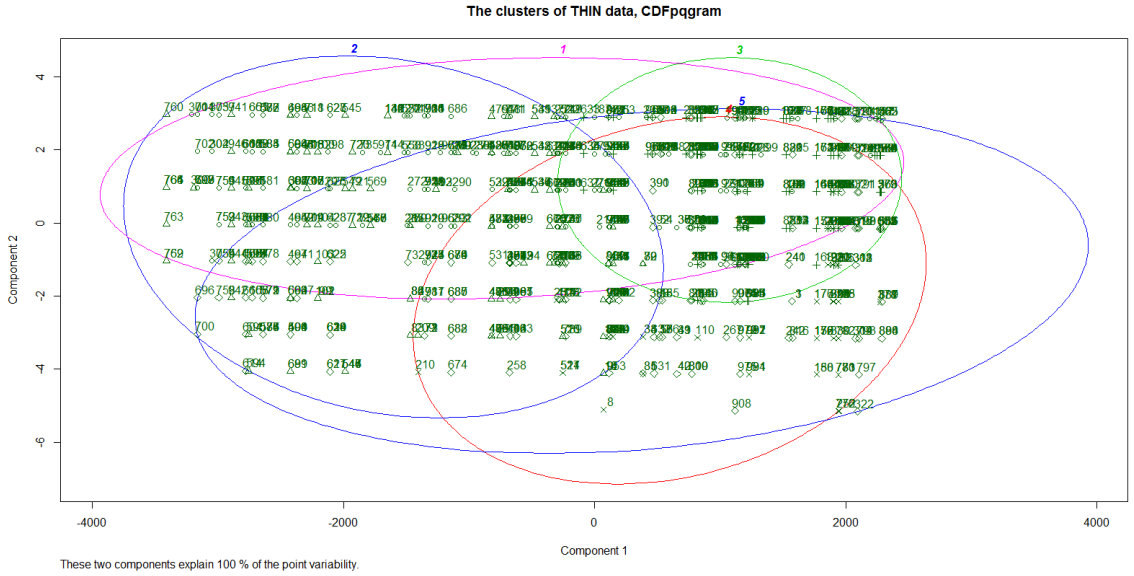


FIGURE 6.17: The 5 clusters of 1000 DESLORATADINE data records using CDF-pq-gram and PCA

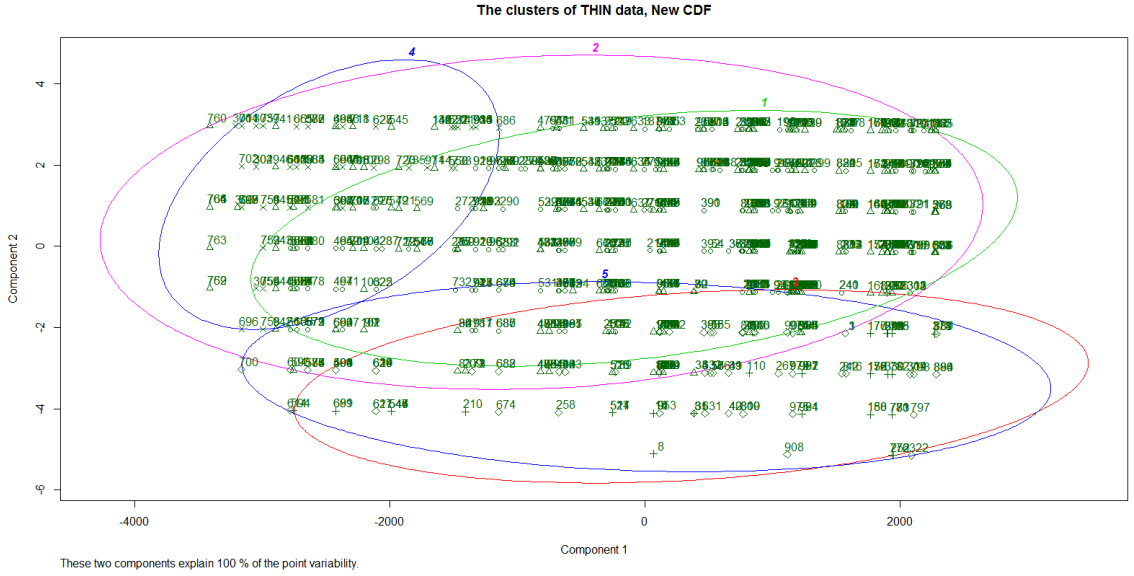


FIGURE 6.18: The 5 clusters of 1000 DESLORATADINE data records using New-CDF and PCA

6.5 Distance Calculation and Clustering Using BIRCH Algorithm for LEVOFLOXACIN

The drug LEVOFLOXACIN has a small sized data set for patients under 18 years old. The sample data covers 14 patients, and this was grouped into 3 clusters using CDF and the BIRCH clustering algorithm. Figure 6.19 shows the three clusters using the CDF without the 2, 2-gram distance function. The clusters are well separated and the information provided about the patients in each cluster shows similarity in most of the attributes, for example cluster number 1 contains six patients all sharing at least one common attribute; e.g. patients a784008Ay, a998401Gi and h981900Rt have symptoms F52..00, F502z00 and F4C0.12, which comes from the same hierarchy, F5 and F, and refers to ear diseases. Cluster number 2 has information about five patients and cluster number 3 that of three patients.

Using the 2, 2-gram distance function along with the CDF produces almost the same clusters as those in the previous figure, as the data contained inside shares the same data records. The difference only emerged in the shapes of the clusters' ellipses, which show slightly different directions based on the average and the covariance matrices of

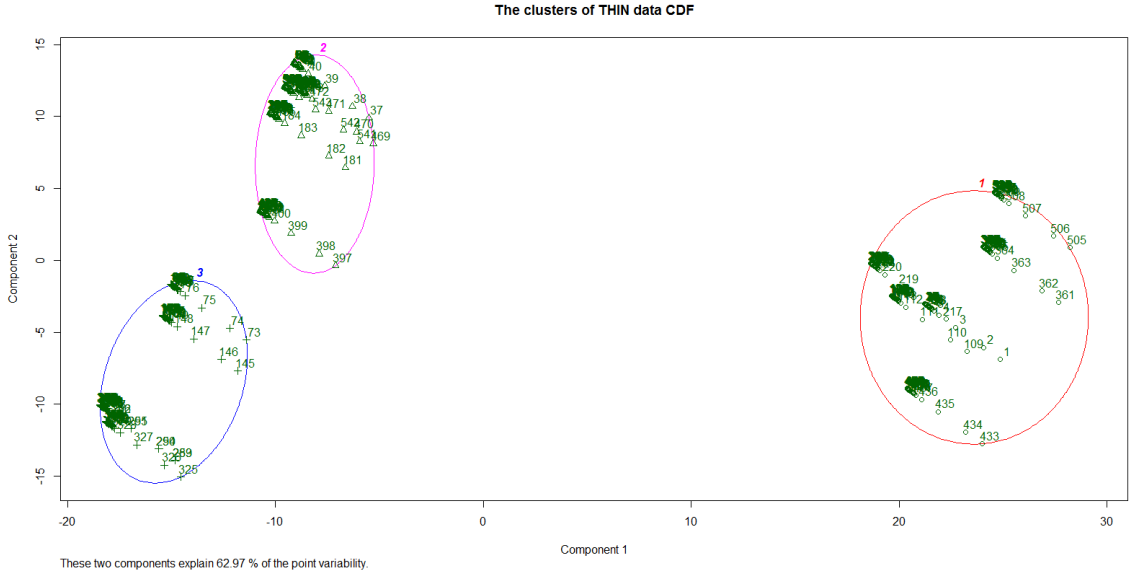


FIGURE 6.19: The 3 clusters of 576 LEVOFLOXACIN data records using CDF

each cluster. The same example shows that same data records for the same patients in each cluster do not change.

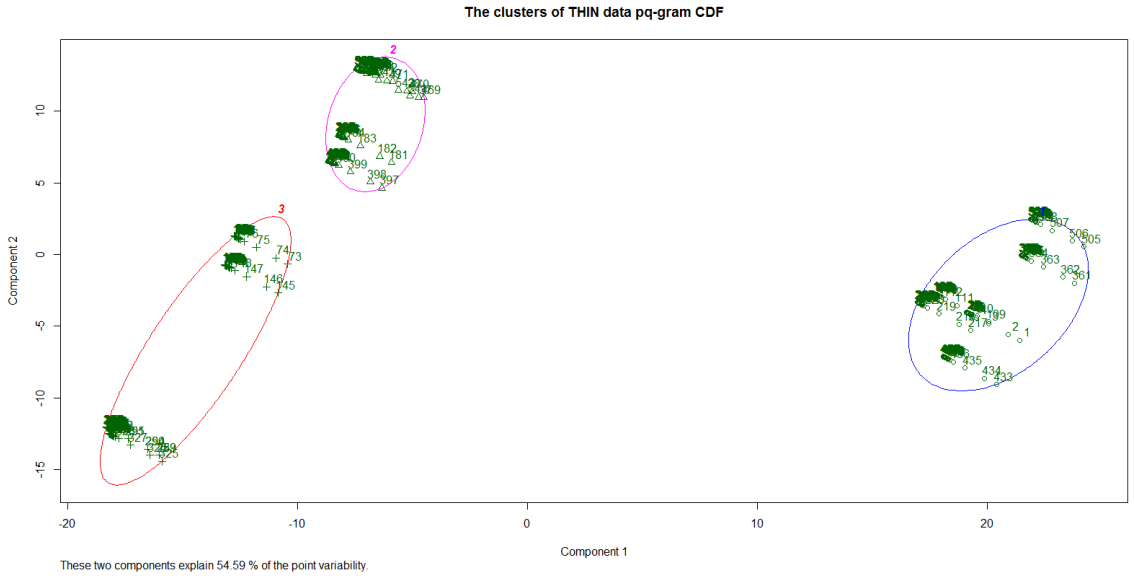


FIGURE 6.20: The 3 clusters of 576 LEVOFLOXACIN data records using CDF with pq-gram

The New-CDF, where the hierarchical measures are replaced by the 2, 2-gram distance function, produced better clusters, as shown in Figure 6.21, when compared with the results from the other drugs' data shown in the previous sections. Cluster number 1 contains the data records of six patients who have some information in common; e.g.

patients a784008Ay and a998401Gi are in the same cluster and share symptoms that come from the same hierarchy F52..00 and F502z00 and refer to ear diseases. In addition, both have symptoms that occurred at the same age for both patients; one example is at age 4246 days i.e. 11 years.

In cluster number 3, the information pertains to three patients who also share some information. Patients a985600ST and a9910039w have the same *pq*-gram distance values, as well as being of the same age when they experienced symptoms from taking the drug. Cluster number 2 contains information about 3 patients, a9856002v, a998400WF and f996702B2, who all share some data; such as, the first two patients sharing a symptom that comes from the same hierarchy (F511.00 and F52z.11) which also refers to ear diseases and specifically the infection otitis media.

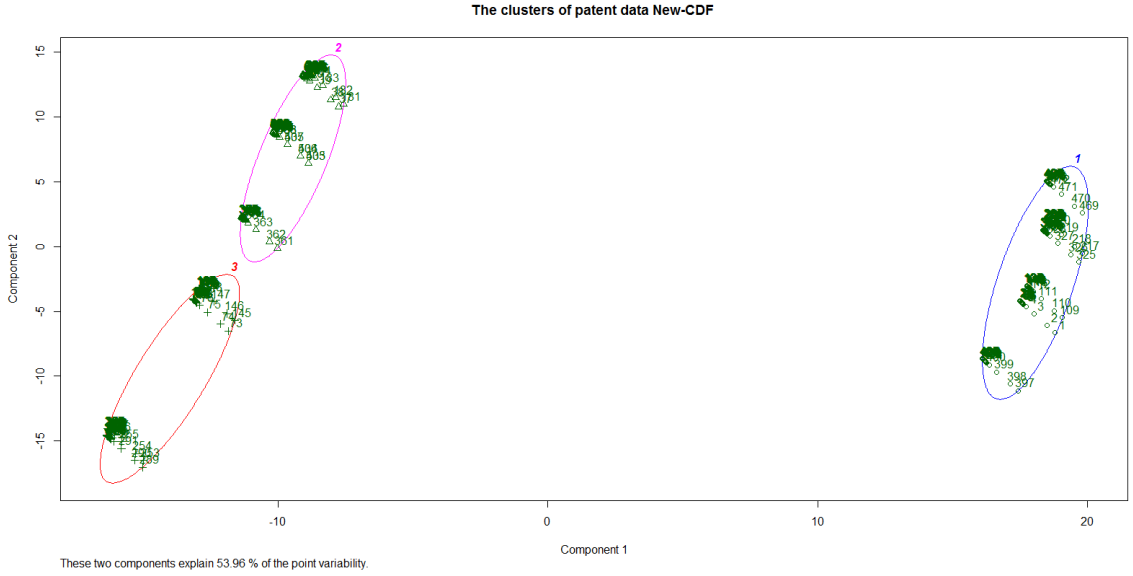


FIGURE 6.21: The 3 clusters of 576 LEVOFLOXACIN data records using New CDF

6.6 The Clusters Validation

The following tables, Tables 6.2, 6.3 and 6.4, illustrate the silhouette and C-index values for the clusters produced by clustering the patients' data records for three different types of drugs. The indices in Table 6.2, for the clusters resulting from clustering the data records for patients who took DOXEPIN, show a high quality of clusters when

TABLE 6.2: Silhouette and C-index values for the 10 clusters of 1172 DOXEPIN data records

	The Silhouette Index	C-Index
Using CDF without pq-gram	0.2353	0.0927
Using CDF with pq-gram	0.3129	0.0157
Using the New-CDF	0.4028	0.0659

TABLE 6.3: Silhouette and C-index values for the 5 clusters of 1000 DESLORATADINE data records

	The Silhouette Index	C-Index
Using CDF without pq-gram	0.3592	0.08004
Using CDF with pq-gram	0.2828	0.0849
Using the New-CDF	0.2288	0.0505

New-CDF is used. A higher quality of clusters than those in the indices in Table 6.4 are shown when using the New-CDF for patients taking the LEVOFLOXACIN drug. In contrast, the index values in Table 6.3 show when the New-CDF is used for measuring the similarity distances between the patients' data records who took the drug DESLORATADINE. These indicate a lower quality compared with those shown in the other tables.

Generally, it can be observed that New-CDF could produce good results when measuring the similarity between data records from subsets of the THIN database. In other words, the new measure can be a useful and suitable method for clustering data from the THIN database, despite the poor quality shown in Figure 6.16, with the subset of data used for the drug DESLORATADINE.

TABLE 6.4: Silhouette and C-index values for the 3 clusters of 576 LEVOFLOXACIN data records

	The Silhouette Index	C-Index
Using CDF without pq-gram	0.44813	0.0457
Using CDF with pq-gram	0.4041	0.0401
Using the New-CDF	0.4659	0.0337

6.7 Summary and Conclusion

In this chapter, a new distance measure was used and applied to the mixed THIN data set to investigate its usefulness with this data set. This was achieved by using the *pq*-gram distance metric with other different distance measures together with different data attributes in one distance function. This is instead of using the *pq*-gram distance metric for all the data attributes, as used in Chapter 4.

Firstly, the new method, the combined distance function CDF, is used without the *pq*-gram distance metric by using a different hierarchical measure for hierarchical attributes named D_{HSM1}, the Euclidean distance metric for the numerical attributes and the histogram of frequencies of the categorical attributes values. The CDF was tested on subsets of the THIN data set to show the difference when using the *pq*-gram distance metric instead of the other hierarchical distance measures. Then, the *pq*-gram distance metric was used along with the CDF, i.e. in combination with the other hierarchical measures D_{HSM1}, to show the impact when adding the new metric. Later, the hierarchical distance functions in the CDF were replaced by the *pq*-gram distance function, such that only the *pq*-gram distance metric was used for the hierarchical attributes, in combination with the other non-hierarchical distance functions, to create a new function called New-CDF.

Both the CDF with the *pq*-gram distance metric and the CDF without it, as well as the New-CDF, were used to measure the distances between data records in terms of similar patients' data, covering symptoms or side effects from a specific drug when using three different subsets of THIN data. The subsets of the data were extracted for those patients taking three kinds of medical drugs: DOXEPIN, DESLORATADINE and LEVOFLOXACIN, and the values of the *pq*-gram parameters used were $p = 2$ and $q = 2$.

For the patients who took the first drug DOXEPIN, 1172 data records were clustered into 5-10 clusters using the BIRCH clustering algorithm in order to show the clusters and their commonalities as the number of clusters increased. The results when clustering this group of THIN data showed that the patients in one cluster shared some

of the same symptoms, gender and some of the same dates in terms of when medical events occurred. For instance, in Figure 6.6 cluster number 3 grouped information for two patients (b990801rU and g9987021J) with the symptoms H17..00 and H05z.00, referring to allergic rhinitis and Upper respiratory infection, respectively. Another two patients were grouped in cluster number 5 and the information they shared related to H170.11 and 171..00 symptoms. The 171..00 symptom for both patients emerged on day 9, so within 30 days of taking the drug. The results also showed that the clusters from 5-10 for both the CDF with the 2, 2-gram and without 2, 2-gram resulted in slight differences. This can be seen from the figures provided, as well as in the index values given in Table 6.1.

Since the differences were not sufficiently clear, a weight was added to the aspects of the CDF and the *pq*-gram distance function. Taking the example of clustering 1172 THIN data entries into 10 clusters, it has been seen that the results reported in Figure 6.7 were slightly better than when no weight was given to the *pq*-gram. This shows the *pq*-gram metric could provide well-separated clusters comprising similar patient information that nevertheless differs in some attributes.

For the group of patients who takes DESLORATADINR, 1000 data records were clustered into 5 clusters using the CDF and the CDF with the *pq*-gram distance function. The difference between the results when using both CDFs was that the use of the *pq*-gram distance function in the calculation led to some overlap between the clusters, since some clusters shared some of the same *pq*-gram distance values. This overlapping influenced the quality of the clusters, as indicated in the tables of indices produced; see Tables 6.2 and 6.3. Using the New-CDF on this subset of THIN data produced more overlapping clusters, and the indices show that using the New-CDF resulted in less compact clusters compared with the other CDFs (see Figure 6.16).

The group of patients that takes LAVOFLOXACIN is a relatively small group of 567 patients; here represented by 14 patients. This data was clustered into 3 clusters using the CDF and CDF with the *pq*-gram distance function as well as New-CDF. For the first two CDFs, the results were almost the same in each cluster, but the quality of the clusters produced by the validity indices used differed. The validity indices, shown

in Table 6.4, indicate that using the CDF with the *pq*-gram distance function produces more compact clusters than when using the CDF without the *pq*-gram metric. The quality of clusters produced using the New-CDF were better than that when using the CDF and CDF with *pq*-gram distance function based on the indice values illustrated in Table 6.4. For this reason, this chapter used another way to compare the figures resulting when using both CDF with the *pq*-gram distance function and the New-CDF. The comparison was made with subsets of data, extracted from the original data set used initially for the three drugs data, based on two numerical attributes. In the previous chapter, the subset of patent data set could not be used in the same way since it consisted of one numerical attribute only. The new data subsets are plotted based on the first principal components when applying the PCA to the new numerical data sets, and the figures are compared for both the CDFs used.

In conclusion, using the different versions of the CDF, the CDF without *pq*-gram, the *pq*-gram along with the CDF, the weighted *pq*-gram along with the CDF and the New-CDF produced different results when using the THIN data (as reported in this chapter) and the US NBER Patent data set (as reported in the previous chapter). The New-CDF was not the better method compared to the other CDFs as it produced very uninterpreted clusters for most of the examples provided in this chapter. The results when using New-CDF to measure the similar distances between the Patent data records in the previous chapter were inferior to the results when using CDF and CDF with the *pq*-gram distance function. In contrast, the New-CDF produced some better similarity for some of the THIN data examples pertaining to patients taking the LEVOFLOXACIN medical drug.

Chapter 7

Conclusions

This thesis focused on developing and designing a new distance measure to assist in the classification of entries in mixed data sets, specifically the THIN data set, by combining an existing hierarchical distance metric with different distance measures. The aim was to use the existing hierarchical distance measure, the *pq*-gram distance metric, to cluster large hierarchical databases to obtain groups of similar data records consisting of mixed data types of attributes. Furthermore, the research aimed to check whether the *pq*-gram distance metric can work for simple non-hierarchical and hierarchical only data sets.

To achieve these objectives, a feasibility study was first performed on the *pq*-gram distance metric using a couple of simple non-hierarchical data sets such as Iris and lenses and a hierarchical only data set named as the NBER US Patent data set. Then subsequently, the metric was applied to mixed subsets of the THIN data set with some other different distance metrics, and the results were compared to show the usefulness of the *pq*-gram distance metric relative to the data set. It was determined that the *pq*-gram metric could not provide similar distances for the simple non-hierarchical data sets and could produce relatively reasonable clusters for the hierarchical only data set when clustering using the *k*-means algorithm. Furthermore, the results of the comparison showed that the metrics could measure the similar distances between the mixed THIN data set records, but the *pq*-gram metric was not the best one among them as shown in Chapter 4, Section 4.4. Thus, it has to be applied in such a way to provide better results

for the mixed THIN data set and group the similar data in one cluster while clustering and distinguish them from data in other clusters.

To overcome the aforementioned issue, a combined distance function (CDF) was produced, which combine the advantages of the hierarchical *pq*-gram distance metric with other traditional distance measures for mixed data sets. The new measure could provide better clusters for the same data sets used and mentioned earlier, while clustering using the BIRCH clustering algorithm, than when only the *pq*-gram metric was applied. Then the CDF was modified to create the New-CDF, which replaced the hierarchical distance functions in the CDF by the *pq*-gram distance function such that the New-CDF consisted of only the *pq*-gram distance function for the hierarchical attributes and other non-hierarchical distance functions for the non-hierarchical data attributes for a mixed data set. The New-CDF could not produce interpretable clusters for the hierarchical only data set, NBER US Patent data set. In contrast, it produced reasonable clusters to some extent for the mixed THIN data set when applied to the subset of patients' data consuming the LEVOFLOXACIN drug, see Chapter 6, Section 6.5.

In order for the new measure to be applied to other data sets, the data set should be hierarchical, i.e. it has to have a hierarchical structure like the NBER US Patent data set or has at least one of the data attributes hierarchical like the READ codes in THIN data set. This is because the new measure includes hierarchical distance functions to measure the distances for the hierarchical attributes in a mixed data set. Although the results when the new measure applied to the Iris data set were better than that when the only *pq*-gram metric was used, as illustrated in Chapter 5, the new measure calculated the distances for the data set such that the numerical distance function in the CDF only worked and the other distance functions were disabled. Therefore, the data set should also be mixed (not only numerical or only categorical) in order for the other distance functions, specifically the hierarchical ones to work on the new measure.

The larger the data set is the better the results can be obtained. In other words, the larger the number of data records that belong to something or someone in the data set, like in THIN, the better chance there is for the new measure to obtain the similar distances between them. For example, the results obtained for the patent data set in

Chapter 5, Section 5.6 were not satisfied and the clusters were not interpretable for all the subsets of the data used. In contrast, the new measure, New-CDF, could provide reasonable separated clusters with the most similar distances between the data records in each cluster for some of the subsets extracted from the THIN data set. For example, the subset of patients' data consuming the LEVOFLOXACIN drug, as illustrated in Chapter 6, Section 6.5. Examples of other hierarchical data set that are worth trying the new method on are the spatial data set, XML data set, and gene expression microarray data.

The data sets for the new measure also need to be prepared very carefully by removing all the data records that have missing values or NULL values to avoid the existing incorrect distances values. For instance, the patent data set was prepared by removing all the data records that have missing values, remove excess white-spaces and remove tags to generate one complete table from the sub-tables of the data set, as shown in Chapter 5, Section 5.3.

In the next section, the main contributions of this work are reported, followed by an explanation of the limitations of those contributions. The aims and research questions are also evaluated, and finally several directions for future research that may be followed to complement this thesis are listed.

7.1 Contributions

This work has resulted in the following contributions.

1. **Comparison of Distance Metrics for Hierarchical Data in Medical Databases**

   Six different distance metrics, namely, Euclidean, Minkowski, Manhattan and Hamming distance metrics, for non-hierarchical data and *pq*-gram and an Edit distance metric for hierarchical data were analysed and applied over two subsets of a mixed THIN database for two drugs taken by patients. To date, a comparison of these distance metrics on the THIN data set, particularly with the *pq*-gram distance metric does not appear to have been performed.

In chapter 4, the second objective of this research was achieved; the goal was to discover if the *pq*-gram distance metric is a useful and suitable measure for the mixed THIN data set when clustering patients. The above distance metrics were considered and applied to the data and their results compared. The data was used in two different structures, a tree-like structure for the hierarchical metrics group and a frequency table or matrix for the non-hierarchical metrics group. For both groups of metrics, the distances were found and normalised. Generally, all the metrics produced good similarity distances between the patients' data. For the first drug, DESLORATADINE, the *pq*-gram distance metric produced some identical distances (the distance equal to 0 between two patients' data) and some additional ones that were similar to each other. While for the second drug, the DOXEPIN, the similarity distances were not found by the *pq*-gram distance metric, as the patients' trees needed to contain more information, and a limited number of data records were associated with the patients consuming this drug. The results were verified after finding all the distances, by considering the entire population of patients for each drug and checking whether the distance metrics discriminated sufficiently when using the k-means algorithm for clustering the patient population.

2. **Improving the pq-gram Metric for Mixed Data Sets** At the beginning of this research, the *pq*-gram distance metric was applied to the non-hierarchical data set, hierarchical and mixed data sets, to test its validity and suitability. The performance of the metric showed that it was not good for non-hierarchical data sets, good enough for the hierarchical data set or suitable for the mixed data set. Thus, the performance of the metric needed to be improved to work better with the mixed THIN data set, such that it could work for hierarchical attributes only. In this contribution, two objectives of this research were achieved; both relating to an improvement in the *pq*-gram distance metric's performance, as suitable for measuring the similarity between mixed data records. First, different distance measures were considered for the hierarchical patent data set, such that the *pq*-gram metric worked only for hierarchical attributes along with other distance measures for other types of attributes in a single data record.

A CDF was created for three types of attributes in the Patent database, with an entry of 5-tuples (hierarchical, hierarchical, hierarchical, numerical, categorical). The CDF in Equation 2.15 used the D_{HSM1} for the hierarchical attribute, the Euclidean distance function for the numerical attribute and the histogram of frequencies of categorical attribute value, as well as the *pq*-gram distance function for the entire entry of 5-tuples to be added to the CDF. Firstly, the CDF was used without the *pq*-gram distance function and the results when clustering the data using the BIRCH algorithm showed good clusters for the hierarchical data set; they were well separated and compact and showed similarity between the data records in each cluster, see Chapters 5, section 5.5 or the patent data set.

By adding the *pq*-gram distance function to work alongside the CDF, the same clusters showed a slight difference in terms of an improvement for the Patent data. Therefore, the idea of adding weight to the aspects of the CDF was proposed as a way to change the influence of a specified attribute during CDF performance. The weighted *pq*-gram distance function, along with the CDF, provided better clusters only in some examples of the patent data set (see Chapter 5, Figure 5.13).

3. **Applying the New Distance Measure to a Large Data Set**

   The last objective of this research was obtained, whereby the *pq*-gram metric was used as a replacement for the other hierarchical measures in the CDF, and as the only measure for the hierarchical attributes to test its suitability for clustering large mixed data sets while clustering using the hierarchical BIRCH algorithm.

   The *pq*-gram distance function working alongside the CDF means it is working as an extra metric added to the CDF, consisting of another distance function for hierarchical measure, D_{HSM1}. In this contribution, the New-CDF consisted of only the *pq*-gram distance function for hierarchical attributes, a distance function for numerical, and a distance function for categorical attributes. Using the same examples for the patent data set, the New-CDF did not give good clustering results. However, it worked for the THIN data, producing good similarities between the data records, such that each cluster, while clustering using BIRCH algorithm, consisted of mostly the same data records as those which each of the two patients

have in common. Thus, it can be concluded that the New-CDF could be useful if used for a suitable hierarchical or mixed data set, that produces some reasonable clusters. For example, THIN data contains lots of records for each patient, which makes it easy to find similarities between records, whereas the Patent data consisted of few data records or only one record for each patent or for each inventor, which makes it more difficult to find the similarities between the records. For example, in Figure 5.18, which shows the four clusters of the 500 patent data records, the records that belong to one inventor can readily be grouped into one cluster. However, many inventors may be grouped into one cluster without sharing any information or data records. Meanwhile, for the THIN data (see Figure 6.21), the data records that locate many patients in one cluster share at least one attribute. This arises because the large trees that the *pq*-gram makes from the many data records belonging to each patient produce a higher probability of matching two patients who share at least one attribute.

7.2 Limitations of the Contributions

There were limitations that affected the work performed, as discussed below in reference to each contribution:

1. While all the distance metrics used for the comparison in Chapter 4 produced reasonable similar distances between the THIN data subsets, the *pq*-gram distance metric was not the best. The data structure used for the non-hierarchical distance metrics was found to be very simple when compared to the tree-like structure used for the *pq*-gram distance metric and other hierarchical distance metrics used. Overall, to compare between distance metrics including the *pq*-gram metric successfully, it is probably better for the data structure to be the same for all the metrics used.

2. Using the *pq*-gram distance metric in the CDF at the beginning did not deliver a sufficient improvement in results when compared to those obtained without it.

The idea of adding a weight to the *pq*-gram distance function aspect in the CDF helped, to some extent, to achieve better outcomes from the clusters when clustering using the BIRCH clustering algorithm. However, the use of the weighted *pq*-gram distance metric in CDF impacted the performance of the algorithm, slowing the running time, with the result that it took longer to acquire the results about the same data sets.

3. Using the New-CDF for different large databases led to different results for clusters, which were good for some databases and not for others. The results obtained using the NBER US Patent database were not good enough when compared to those from the THIN database, as the latter produced richer data records with additional information relating to someone/something in the database.

4. Although the New-CDF produced better results for the THIN database in Chapter 6, when comparing this with the results when using only the *pq*-gram distance metric for the same database in Chapter 4, the results were seen to be more complex. This is because the clustering algorithm used was more complicated and using different distance measures made the results more complicated and difficult to interpret.

7.3 Evaluation of Aims

The aims and research questions stated in Chapter 1 (see Section 1.2) have been addressed as follows:

1. **Can the *pq*-gram distance metric be applied to any data set to identify similarity between objects reliably?**

   This was addressed by applying the *pq*-gram distance metric to simple non hierarchical data sets such as the Lenses data set and the Iris data set. The two data sets then have to be converted to a tree-like structure (hierarchical) to suit the distance metric and to help study the metric's ability to measure the similarity in

simple data sets, as stated in Chapter 4. Since the data sets already contain classified data, the *pq*-gram distance metric is applied to determine if the metric could produce the same similar data objects in each cluster as when clustering using the *k*-means clustering algorithm. The accuracy of the clusters for the Iris data was not very promising (see Section 4.2.2). Regarding the Lenses data set, the data points that represent the *pq*-gram trees were not sufficient for clustering as the number of required clusters was three. Thus, this data set could not be clustered using the *pq*-gram distance metric and the k-means clustering algorithm. Applying the *pq*-gram distance metric to hierarchical NBER US Patent data set showed the performance of the distance metric was good and could be improved if there were lots of data records belonging to the Patents or any attribute of the data set.

2. **What is the performance of the *pq*-gram distance metric compared to other measures when measuring the similarity between the mixed THIN data and is the metric suitable for clustering the entire patient population from the data set?**

   This was tested by comparing between the different distance metrics, including the *pq*-gram distance metric, while applying them to subsets of the THIN data set. All the distance metrics used produced some similarity between the data sets' objects; however, the *pq*-gram distance metric was not the best. By clustering the patients' population using the k-means clustering algorithm, the *pq*-gram distance metric was suitable to some extent and discriminated sufficiently when clustering the patient population (see Chapter 4, from Section 4.4 onwards).

3. **How can the performance of the *pq*-gram metric be improved to measure the similarity between mixed data type records in large databases if it produces poor performance?**

   By examining the different distance measures for hierarchical and non-hierarchical attributes in mixed data type records, a combination between the different distance measures, including the *pq*-gram distance metric for the hierarchical attributes, was tested as a measure of the similarity between the data records of a large hierarchical database, i.e. the NBER US Patent database (see Equation 2.15

in Section 2.4.3). The CDF was applied to different subsets of the database and produced good similarity between data records that have been grouped together into a single cluster when clustering using the BIRCH clustering algorithm (see Chapter 5, Section 5.5.2).

4. **When combining different distance measures, are the new measures useful for hierarchical data sets to discover similarity between data records and are they suitable for clustering large mixed data sets while using the hierarchical BIRCH clustering algorithm?**

   This was addressed in different steps. Firstly, a new distance measure (the CDF) was created to measure similarity among mixed data. Secondly, weights were added to the different aspects of the CDF such that the *pq*-gram distance function has the biggest emphasis by giving it a big weight (see Equation 2.16 in Section 2.4.3.1). Then, the CDF was modified to create a new CDF called New-CDF, created by replacing the hierarchical distance measures in the CDF with the *pq*-gram distance function. The New-CDF was applied to the THIN database to present better similarities between the data set records, and it discriminated sufficiently well when clustering the patient population using the hierarchical BIRCH clustering algorithm (Chapter 6, Sections 6.3, 6.4 and 6.5).

7.4 Future Works

A number of research directions may be followed to improve the work in this thesis further.

- In this work, the data clustering used was hard clustering, which proposes that each data point should belong to only one cluster. To place the data points in the clusters, different distance measures were used and the specific one for a hierarchical data set was the *pq*-gram distance metric, which controls the formation of clusters. To show better association between each data point and a particular cluster, fuzzy clustering can provide advantages by giving the data elements a set

of membership values, making it possible to assign each one to more than one cluster.

- In this work, two internal validity indices were used to assess the clustering results and the quality of the clusters: Silhouette and C-Index. However, one direction for future research could be to investigate other different indices as instruments to verify the compactness of the returned clusters

- Using the same THIN database, it would be possible to extract another subset of the database based on drug dosage attributes, including the hierarchical one named dosecode. Patients' records would then be clustered based on the drug dosage value to determine the association between a medical event and the drug prescribed for the patients. The drug dosage value has additional variables/attributes related to it. These attributes include the prescription quantity (rxQty), which can be used with the daily dosage value (dosgval) to determine the number of days over which a drug was prescribed. The code for each drug dosage is represented in the database as (dosecode), and the prescription days given (rxdays). These attributes may help to cluster patients or drugs to find associations between each medical event and the specified drug. In other words, the information from the data set would then be represented as follows: for each patient who has a specific illness, the drugs that the patient takes for that illness will be found. In addition, for each drug the patient takes, the dosage code and the number of days prescribed and the prescription days would be known. Based on this theory, the information pertaining to the patients would be represented in the new data set and the new method New-CDF applied to find any similarity between the data records. This might help to determine an association between a specific illness and the drug prescribed.

- The New-CDF method can be used on another longitudinal database, such as The Clinical Practice Research Datalink (CPRD). The CPRD is the world's largest database of anonymised, longitudinal primary care medical records, also linking with secondary care data sets. The database has clinical and prescription data and can provide information to support pharmacovigilance (indication, utilisation,

and risk/benefit profiles of drugs) and formal pharmacoepidemiologic studies, including information on demographics, medical symptoms, therapy (medicines, vaccines, devices), and treatment outcomes (see www.cprd.com for more details). The new method should be tested further in contexts where it is most applicable. Specifically, it is best applied to newer databases that share the same characteristics as the THIN database, comprising multiple records. This is because the *pq*-gram distance metric, which measures the similarity between trees, requires large trees to increase the opportunities to identify similarity between two database entries.

Appendix A

Appendix A

A.1 The figures of DOXEPIN drug Using CDF

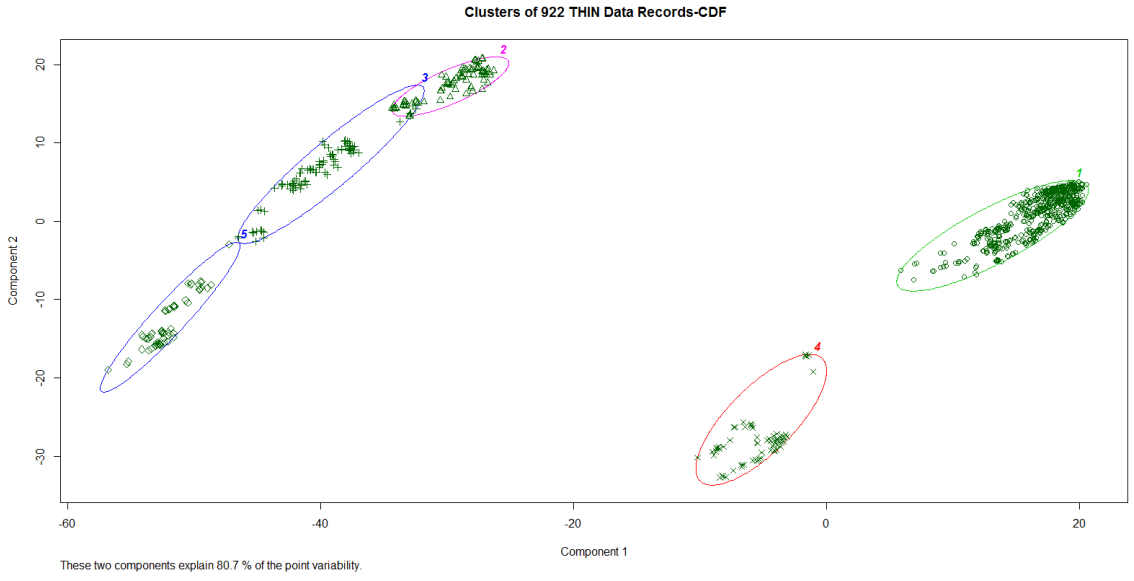


FIGURE A.1: The 5 clusters of 1172 THIN data records-Doxepin drug

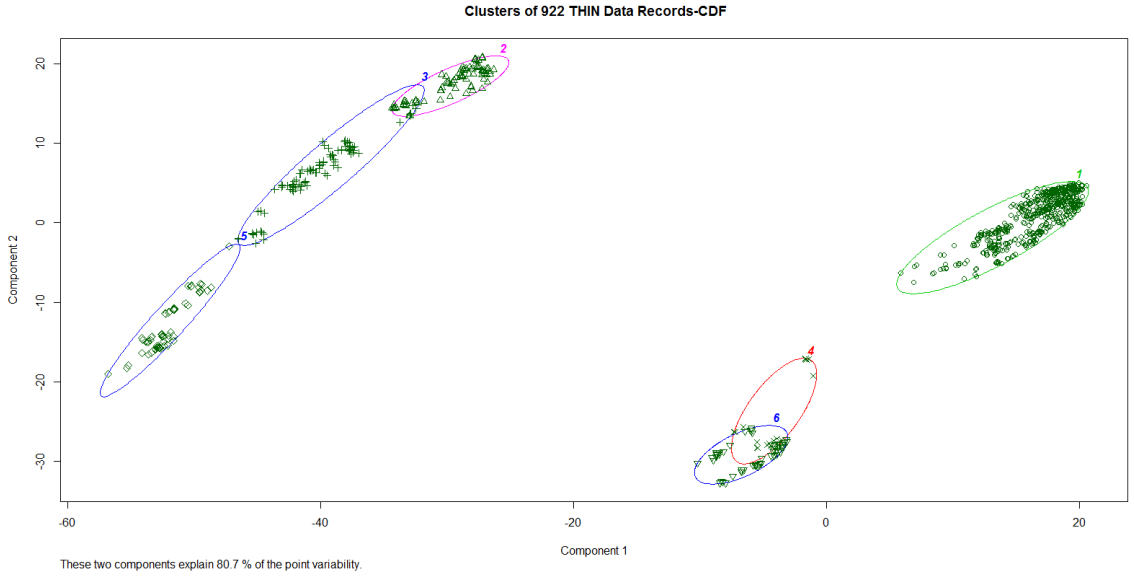


FIGURE A.2: The 6 clusters of 1172 THIN data records-Doxepin drug

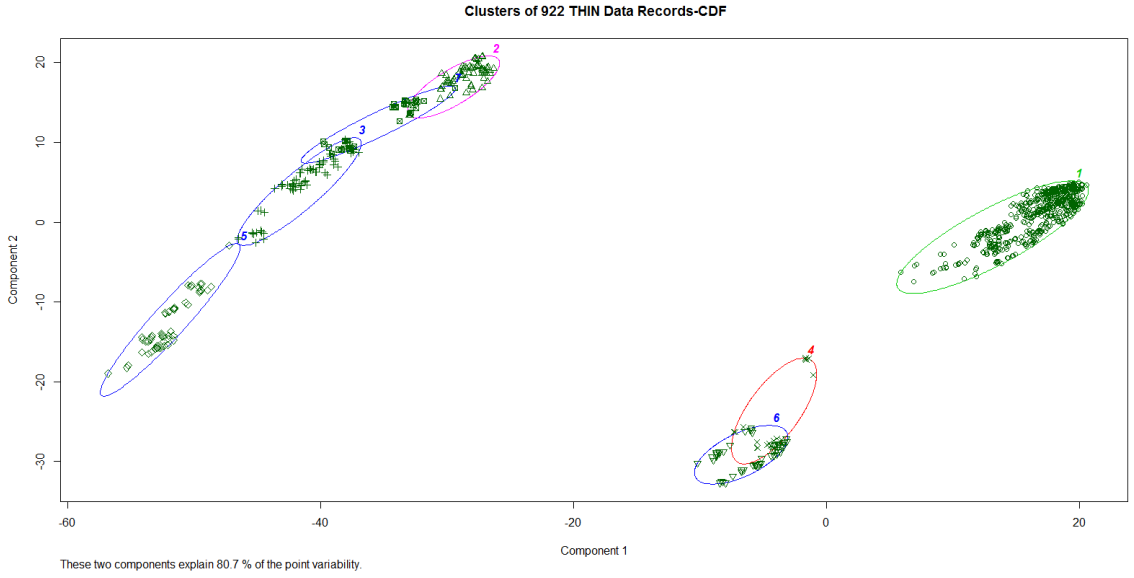


FIGURE A.3: The 7 clusters of 1172 THIN data records-Doxepin drug

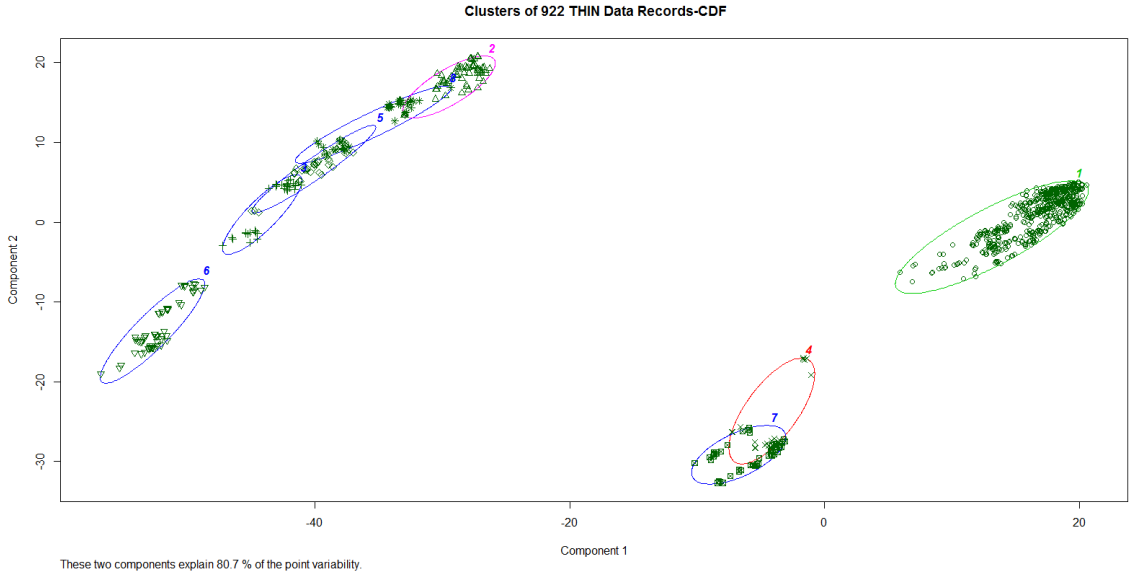


FIGURE A.4: The 8 clusters of 1172 THIN data records-Doxepin drug

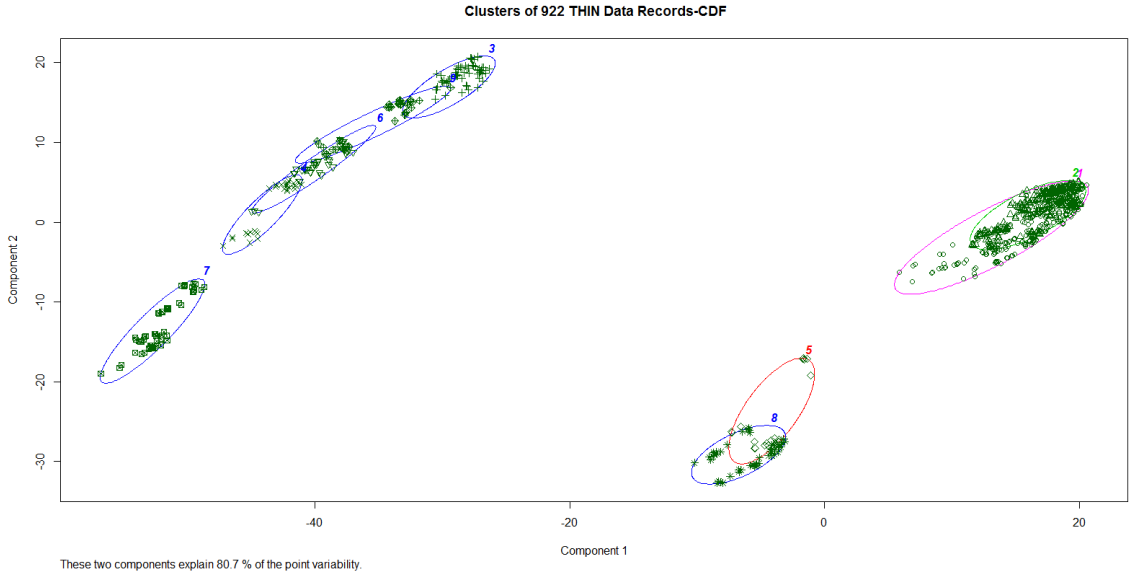


FIGURE A.5: The 9 clusters of 1172 THIN data records-Doxepin drug

A.2 The figures of DOXEPIN drug Using CDF with pq-gram

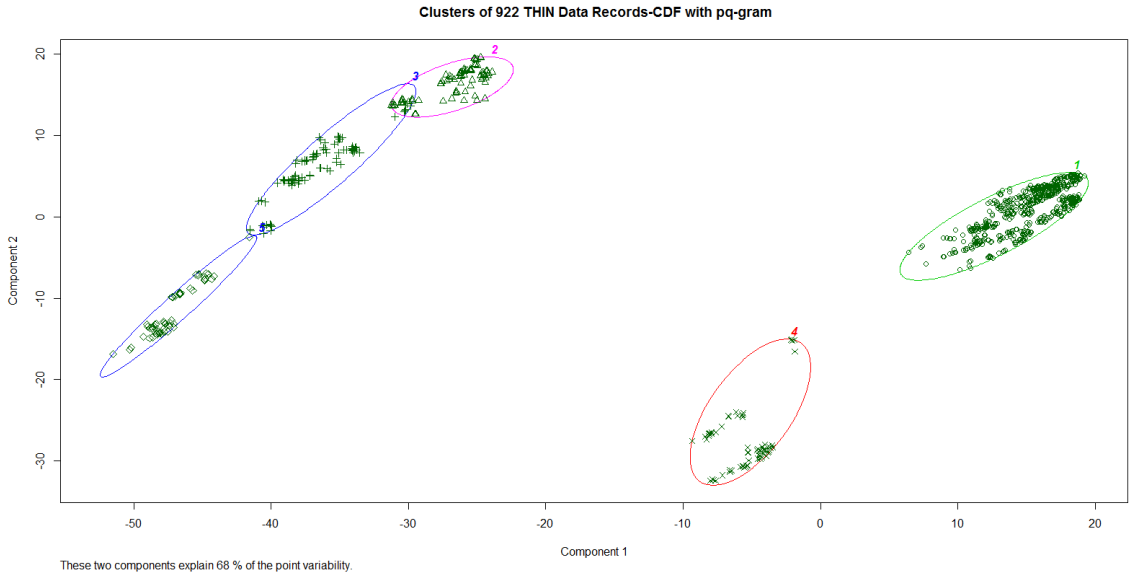


FIGURE A.6: The 5 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram

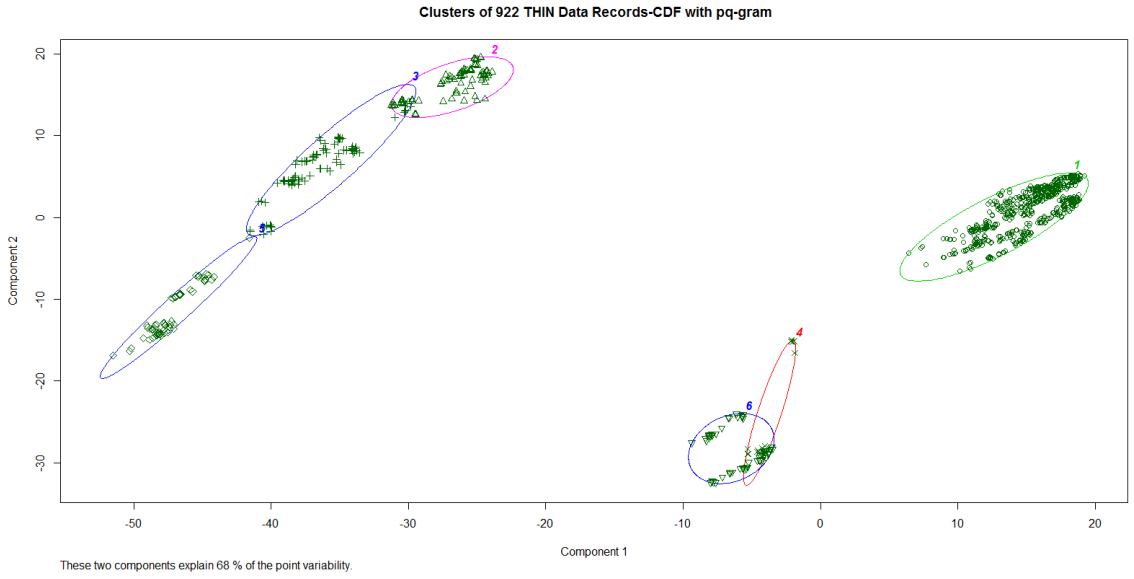


FIGURE A.7: The 6 clusters of 1172 THIN data records-Doxepin drug Using CDF and pq-gram

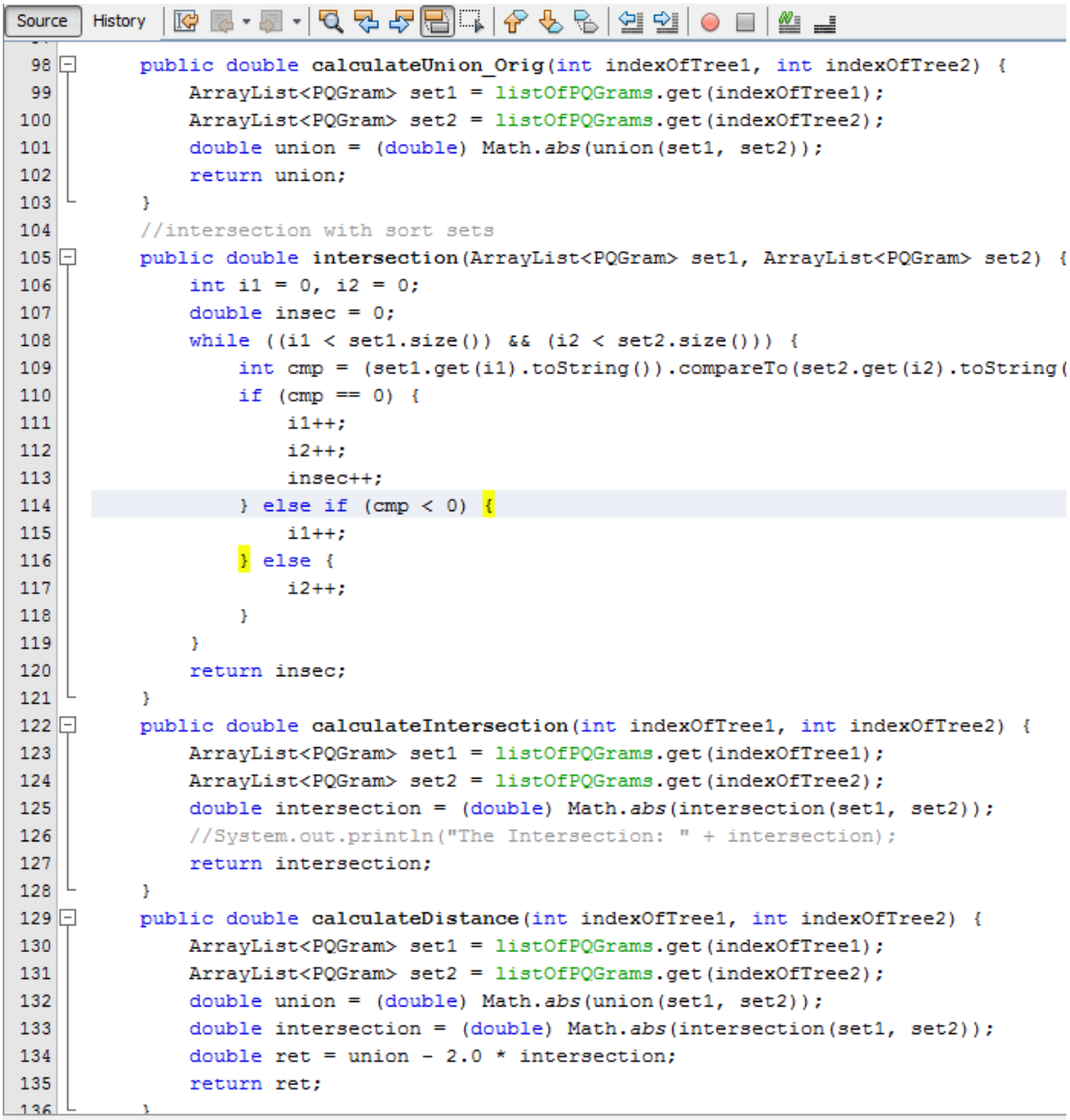
A.3 The software used and examples of the code

The databases were stored in MS SQL server and the necessary tables needed for this work were prepared after cleaning the databases by removing the missing values, remove of excess white-space and remove tags from NBER US Patent database. While for the THIN database, some tables are re-created by removing the non-medical events like the 'ZZZZZ00', the death events and the patients who transferred to another practice and still have the registered details in the investigated practice.

The Java programming language and the R software were used, where the first one used for the clustering process and producing the clusters and the R software is used to plot the resulted clusters using specific plotting function named as *clusplot*.

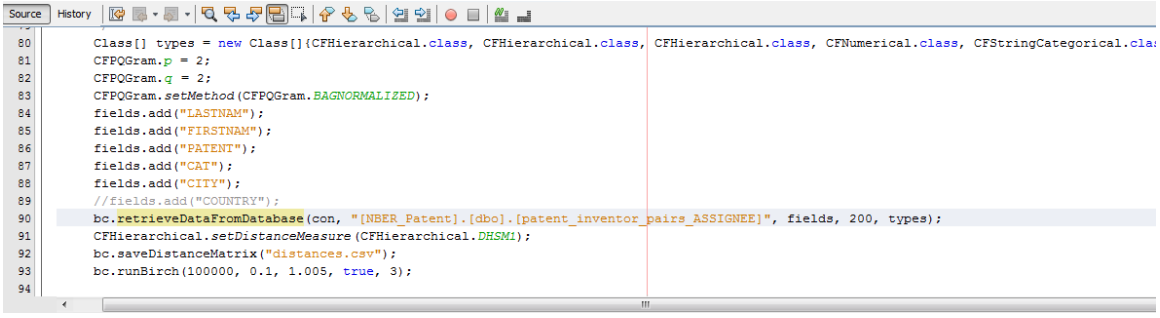
In the Figure A.8, the code of how the *pq*-gram distance calculated is shown. According to the equation 3.11, the *pq*-gram distance is calculated by taking the intersection of two list of *pq*-grams (*pq*-gram profiles), as shown in the first section of the Figure A.8, subtracting the 2 multiplied by the intersection of the same two list of *pq*-grams (the second and third sections in the same figure).

The code shown in Figure A.9 is displaying the steps of running the BIRCH clustering algorithm. It shows how *pq*-gram parameters were chosen and how to use the fields of the data set chosen as well. The last function of the code sets the parameters of the algorithm, which are the maximum number of CF tree nodes, the Threshold T and the number of clusters chosen by the user to be displayed.



```
98      public double calculateUnion_Orig(int indexOfTree1, int indexOfTree2) {
99          ArrayList<PQGram> set1 = listOfPQGrams.get(indexOfTree1);
100         ArrayList<PQGram> set2 = listOfPQGrams.get(indexOfTree2);
101         double union = (double) Math.abs(union(set1, set2));
102         return union;
103     }
104     //intersection with sort sets
105     public double intersection(ArrayList<PQGram> set1, ArrayList<PQGram> set2) {
106         int i1 = 0, i2 = 0;
107         double insec = 0;
108         while ((i1 < set1.size()) && (i2 < set2.size())) {
109             int cmp = (set1.get(i1).toString()).compareTo(set2.get(i2).toString(
110             if (cmp == 0) {
111                 i1++;
112                 i2++;
113                 insec++;
114             } else if (cmp < 0) {
115                 i1++;
116             } else {
117                 i2++;
118             }
119         }
120         return insec;
121     }
122     public double calculateIntersection(int indexOfTree1, int indexOfTree2) {
123         ArrayList<PQGram> set1 = listOfPQGrams.get(indexOfTree1);
124         ArrayList<PQGram> set2 = listOfPQGrams.get(indexOfTree2);
125         double intersection = (double) Math.abs(intersection(set1, set2));
126         //System.out.println("The Intersection: " + intersection);
127         return intersection;
128     }
129     public double calculateDistance(int indexOfTree1, int indexOfTree2) {
130         ArrayList<PQGram> set1 = listOfPQGrams.get(indexOfTree1);
131         ArrayList<PQGram> set2 = listOfPQGrams.get(indexOfTree2);
132         double union = (double) Math.abs(union(set1, set2));
133         double intersection = (double) Math.abs(intersection(set1, set2));
134         double ret = union - 2.0 * intersection;
135         return ret;
136     }
```

FIGURE A.8: The code of how the pq-gram distance calculated



```
80      Class[] types = new Class[]{CFHierarchical.class, CFHierarchical.class, CFHierarchical.class, CFNumerical.class, CFStringCategorical.cla
81      CFPQGram.p = 2;
82      CFPQGram.q = 2;
83      CFPQGram.setMethod(CFPQGram.BAGNORMALIZED);
84      fields.add("LASTNAM");
85      fields.add("FIRSTNAM");
86      fields.add("PATENT");
87      fields.add("CAT");
88      fields.add("CITY");
89      //fields.add("COUNTRY");
90      bc.retrieveDataFromDatabase(con, "[NBER_Patent].[dbo].[patent_inventor_pairs_ASSIGNEE]", fields, 200, types);
91      CFHierarchical.setDistanceMeasure(CFHierarchical.DHSM1);
92      bc.saveDistanceMatrix("distances.csv");
93      bc.runBirch(100000, 0.1, 1.005, true, 3);
94
```

FIGURE A.9: The code of how to run BIRCH algorithm

A.4 Figures for Patent and THIN Data Sets

Results Messages

	PATENT	GYEAR	GDATE	APPYEAR	COUNTRY	POSTATE	ASSIGNEE	ASSCODE	CLAIMS	NCLASS	CAT	SUBCAT	CMADE	CRECEIVE	RATIOCIT	GENERAL
1	3070801	1963	1096	NULL	BE	NULL	NULL	1	NULL	269	6	69	NULL	1	NULL	0
2	3070802	1963	1096	NULL	US	TX	NULL	1	NULL	2	6	63	NULL	0	NULL	NULL
3	3070803	1963	1096	NULL	US	IL	NULL	1	NULL	2	6	63	NULL	9	NULL	0
4	3070804	1963	1096	NULL	US	OH	NULL	1	NULL	2	6	63	NULL	3	NULL	1
5	3070805	1963	1096	NULL	US	CA	NULL	1	NULL	2	6	63	NULL	1	NULL	0
6	3070806	1963	1096	NULL	US	PA	NULL	1	NULL	2	6	63	NULL	0	NULL	NULL
7	3070807	1963	1096	NULL	US	OH	NULL	1	NULL	623	3	39	NULL	3	NULL	0
8	3070808	1963	1096	NULL	US	IA	NULL	1	NULL	623	3	39	NULL	4	NULL	0
9	3070809	1963	1096	NULL	US	AZ	NULL	1	NULL	4	6	65	NULL	0	NULL	NULL
10	3070810	1963	1096	NULL	US	IL	NULL	1	NULL	4	6	65	NULL	3	NULL	0
11	3070811	1963	1096	NULL	US	CA	NULL	1	NULL	4	6	65	NULL	8	NULL	0
12	3070812	1963	1096	NULL	US	LA	NULL	1	NULL	4	6	65	NULL	3	NULL	0
13	3070813	1963	1096	NULL	US	NY	NULL	1	NULL	5	6	65	NULL	2	NULL	0
14	3070814	1963	1096	NULL	US	MN	NULL	2	NULL	267	5	59	NULL	2	NULL	1
15	3070815	1963	1096	NULL	US	CO	NULL	1	NULL	7	5	59	NULL	1	NULL	0
16	3070816	1963	1096	NULL	US	OK	NULL	1	NULL	114	5	55	NULL	4	NULL	0
17	3070817	1963	1096	NULL	US	RI	NULL	2	NULL	114	5	55	NULL	5	NULL	1
18	3070818	1963	1096	NULL	US	IN	NULL	1	NULL	441	6	69	NULL	4	NULL	1
19	3070819	1963	1096	NULL	US	TN	NULL	4	NULL	12	6	63	NULL	0	NULL	NULL
20	3070820	1963	1096	NULL	GB	NULL	NULL	2	NULL	12	6	63	NULL	0	NULL	NULL
21	3070821	1963	1096	NULL	US	IL	NULL	2	NULL	15	6	69	NULL	1	NULL	0
22	3070822	1963	1096	NULL	US	NY	NULL	2	NULL	401	1	12	NULL	4	NULL	0
23	3070823	1963	1096	NULL	US	MI	NULL	1	NULL	401	1	12	NULL	8	NULL	1
24	3070824	1963	1096	NULL	US	IL	NULL	1	NULL	401	1	12	NULL	5	NULL	0
25	3070825	1963	1096	NULL	US	IL	NULL	1	NULL	401	1	12	NULL	7	NULL	1
26	3070826	1963	1096	NULL	US	IA	NULL	1	NULL	401	1	12	NULL	1	NULL	0

FIGURE A.10: The Patent table From Patent Data Set

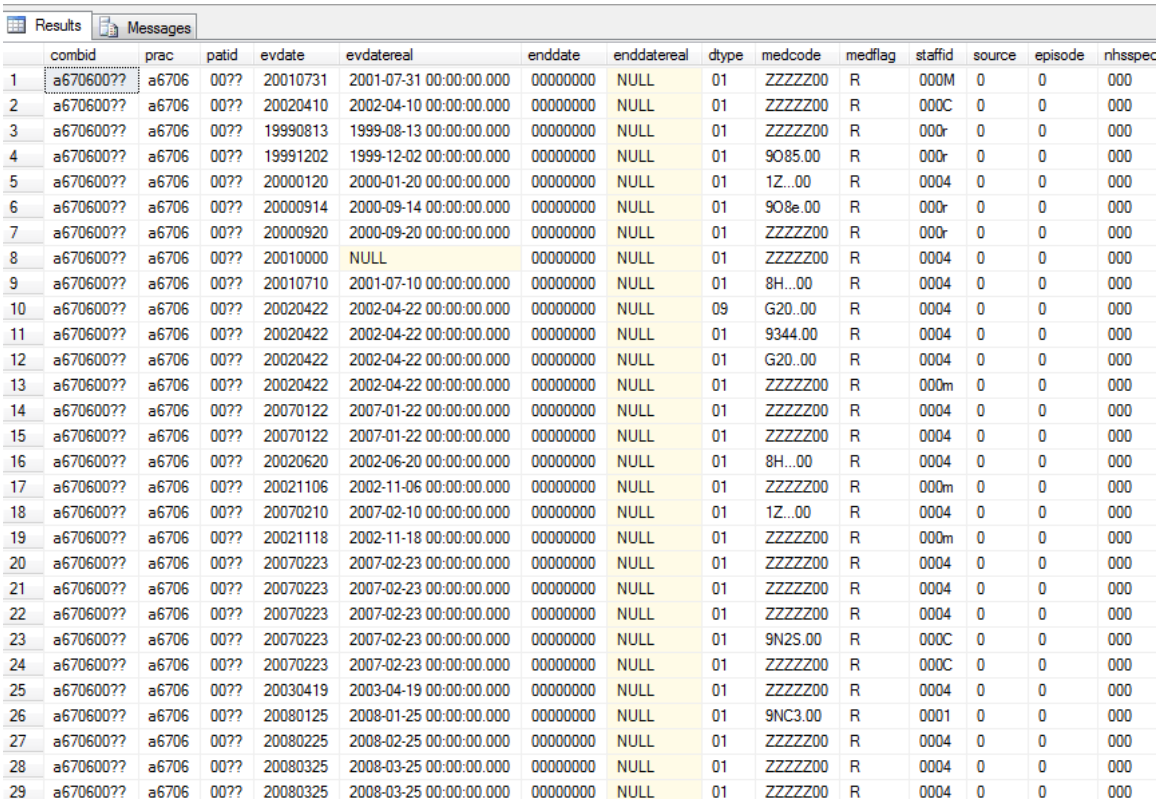
	PATENT	LASTNAM	FIRSTNAM	MIDNAM	MODIFNAM	STREET	CITY	POSTATE	COUNTRY	ZIP	INVSEQ
1	3858241	Durand	Philip	E.	NULL	NULL	Hudson	MA	US	NULL	1
2	3858241	Norris	Lonnie	H.	NULL	NULL	Milford	MA	US	NULL	2
3	3858242	Gooding	Elwyn	R.	NULL	120 Darwin Rd.	Pinckney	MI	US	48169	1
4	3858243	Pierron	Claude	Raymond	NULL	NULL	Epinal	NULL	FR	NULL	1
5	3858243	Jenny	Jean	Paul	NULL	NULL	Decines	NULL	FR	NULL	2
6	3858243	Zuccaro	Robert	NULL	NULL	NULL	Epinal	NULL	FR	NULL	3
7	3858244	Mann	Richard	L.	NULL	P.O. Box 69	Woodstock	CT	US	6281	1
8	3858245	Nate	Michael	A.	II	NULL	New York	NY	US	NULL	1
9	3858245	Mann	Maurice	A.	NULL	NULL	New York	NY	US	NULL	2
10	3858246	Milo	Simcha	NULL	NULL	1500 E. Riverside; Apt. 204 D	Austin	TX	US	78741	1
11	3858247	Bauman	Jack	NULL	NULL	P.O. Box 25310	Los Angeles	CA	US	90025	1
12	3858248	Crowe	Sylvester	L.	NULL	501 W. Madison St.	Franklin	IN	US	46131	1
13	3858249	Howard	Durrell	Unger	NULL	306 Krameria Dr.	San Antonio	TX	US	78213	1
14	3858250	Coglitore	Anthony	NULL	NULL	52 Copley St.	Staten Island	NY	US	10314	1
15	3858251	Vollrath	Walter	J.	NULL	NULL	Sheboygan	WI	US	NULL	1
16	3858251	Vollrath	Richard	J.	NULL	NULL	Sheboygan	WI	US	NULL	2
17	3858251	Vimoche	Paul	R.	NULL	NULL	Newton	WI	US	NULL	3
18	3858251	Turk	Joseph	NULL	NULL	NULL	Sheboygan	WI	US	NULL	4
19	3858251	Gerdes	Paul	E.	NULL	NULL	Sheboygan	WI	US	NULL	5
20	3858251	Singer	Roy	W.	NULL	NULL	Sheboygan	WI	US	NULL	6
21	3858252	Ejchorszt	Olgierd	Z.	NULL	5580 Vesuvian Walk	Long Beach	CA	US	NULL	1
22	3858253	Lauzon	Adrian	A.	NULL	5211 Cartier St.; Apt. 1	Montreal; ...	NULL	CA	NULL	1
23	3858254	Coomes	Edmund	S.	NULL	Rt. 1; Box 78F	Biggs	CA	US	95917	1
24	3858255	Purves	George	T.	Jr.	5337 Shorewood Dr.	Indianapolis	IN	US	46220	1
25	3858256	Beer	Andrew	E.	NULL	25 Sutton Pl.	New York	NY	US	10022	1
26	3858257	Young	Samuel	NULL	NULL	14 Crest Rd.	Danbury	CT	US	6810	1
27	3858258	Stevens	Peter	S.	NULL	708 Strand	Manhattan ...	CA	US	90266	1

FIGURE A.11: The inventor table From Patent Data Set

Results | Messages

	ASSIGNEE	COMPNAME
1	5	"AAA ASSOCIATES, INC."
2	10	"AAA OFFICE COFFEE SERVICE, INC."
3	15	AAA PIPE CLEANING CORPORATION
4	20	AAA PRODUCTS INTERNATIONAL INC.
5	25	"AAA SALES & ENGINEERING, INC."
6	30	AAA STEEL AND ENTERPRISES CORP.
7	35	AA. BROVIG A.S.
8	40	AACCURATE COUNTERS COMPANY
9	50	A. A. COMPUTERIZED SECURITY DOORS 1989 LTD.
10	55	"AADAMS MERCHANDISING, INC."
11	60	A + A DIE CASTING CO.
12	70	A. + A. ENGINEERING CO.
13	75	A + A EQUIPMENT CO.
14	80	A-AFFORDABLE KEY & LOCK INC.
15	85	AA GAGE DIVISION
16	90	A. AHLSTROM CORPORATION
17	95	A. AHLSTROM OSAKEYHTIO
18	100	A. AHLSTROM
19	105	AAI CORPORATION
20	110	"AAKAC INDUSTRIES, INC."
21	115	AAKRONITE DECORATIVES LIMITED
22	120	"AALBA DENT, INC."
23	125	AALBORG BOILERS A/S
24	135	AALBORG VAERFT A/S
25	140	AALCO MANUFACTURING COMPANY
26	145	"AAL ENTERPRISES, INC."
27	150	"AALTO SCIENTIFIC, LTD."

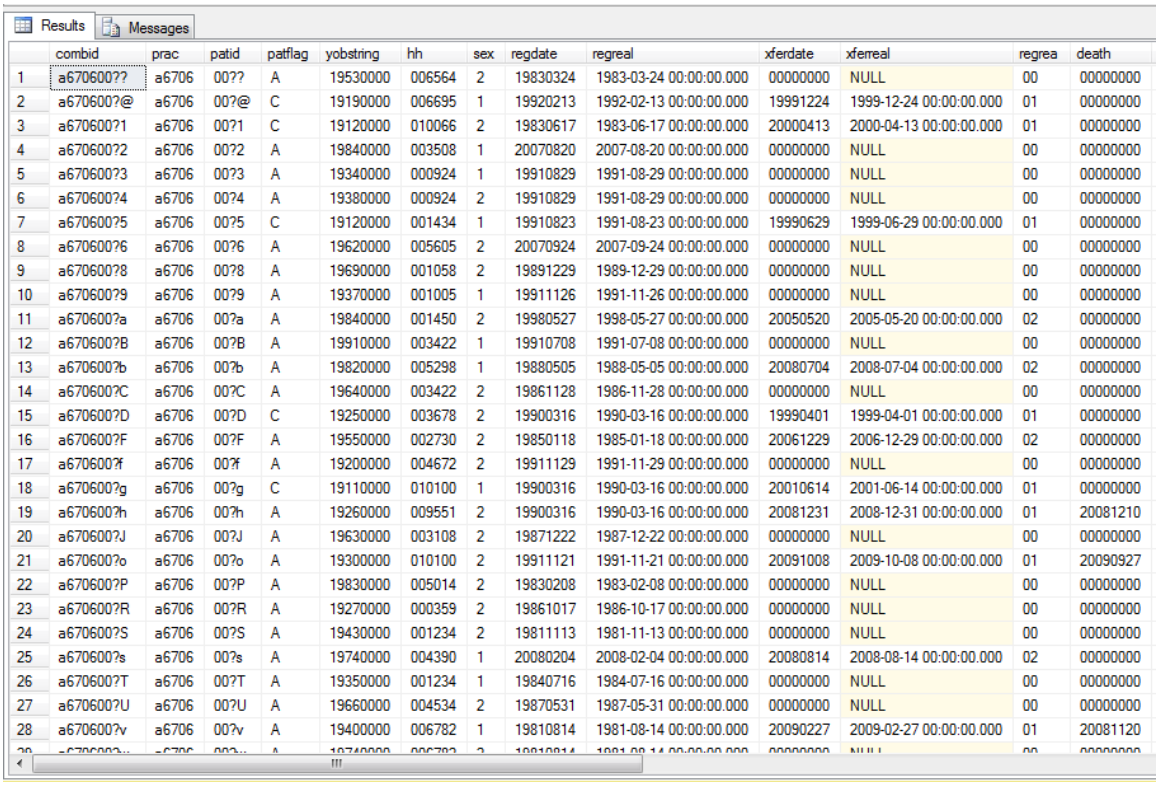
FIGURE A.12: The Assignee table From Patent Data Set



Results Messages

	combid	prac	patid	evdate	evdatereal	enddate	enddatereal	dtype	medcode	medflag	staffid	source	episode	nhsspec
1	a670600??	a6706	00??	20010731	2001-07-31 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000M	0	0	000
2	a670600??	a6706	00??	20020410	2002-04-10 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000C	0	0	000
3	a670600??	a6706	00??	19990813	1999-08-13 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000r	0	0	000
4	a670600??	a6706	00??	19991202	1999-12-02 00:00:00.000	00000000	NULL	01	9O85.00	R	000r	0	0	000
5	a670600??	a6706	00??	20000120	2000-01-20 00:00:00.000	00000000	NULL	01	1Z...00	R	0004	0	0	000
6	a670600??	a6706	00??	20000914	2000-09-14 00:00:00.000	00000000	NULL	01	9O8e.00	R	000r	0	0	000
7	a670600??	a6706	00??	20000920	2000-09-20 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000r	0	0	000
8	a670600??	a6706	00??	20010000	NULL	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
9	a670600??	a6706	00??	20010710	2001-07-10 00:00:00.000	00000000	NULL	01	8H...00	R	0004	0	0	000
10	a670600??	a6706	00??	20020422	2002-04-22 00:00:00.000	00000000	NULL	09	G20..00	R	0004	0	0	000
11	a670600??	a6706	00??	20020422	2002-04-22 00:00:00.000	00000000	NULL	01	9344.00	R	0004	0	0	000
12	a670600??	a6706	00??	20020422	2002-04-22 00:00:00.000	00000000	NULL	01	G20..00	R	0004	0	0	000
13	a670600??	a6706	00??	20020422	2002-04-22 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000m	0	0	000
14	a670600??	a6706	00??	20070122	2007-01-22 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
15	a670600??	a6706	00??	20070122	2007-01-22 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
16	a670600??	a6706	00??	20020620	2002-06-20 00:00:00.000	00000000	NULL	01	8H...00	R	0004	0	0	000
17	a670600??	a6706	00??	20021106	2002-11-06 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000m	0	0	000
18	a670600??	a6706	00??	20070210	2007-02-10 00:00:00.000	00000000	NULL	01	1Z...00	R	0004	0	0	000
19	a670600??	a6706	00??	20021118	2002-11-18 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000m	0	0	000
20	a670600??	a6706	00??	20070223	2007-02-23 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
21	a670600??	a6706	00??	20070223	2007-02-23 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
22	a670600??	a6706	00??	20070223	2007-02-23 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
23	a670600??	a6706	00??	20070223	2007-02-23 00:00:00.000	00000000	NULL	01	9N2S.00	R	000C	0	0	000
24	a670600??	a6706	00??	20070223	2007-02-23 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	000C	0	0	000
25	a670600??	a6706	00??	20030419	2003-04-19 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
26	a670600??	a6706	00??	20080125	2008-01-25 00:00:00.000	00000000	NULL	01	9NC3.00	R	0001	0	0	000
27	a670600??	a6706	00??	20080225	2008-02-25 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
28	a670600??	a6706	00??	20080325	2008-03-25 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000
29	a670600??	a6706	00??	20080325	2008-03-25 00:00:00.000	00000000	NULL	01	ZZZZZ00	R	0004	0	0	000

FIGURE A.13: A screen shot of the medical table contained within the THIN database.



Results Messages

	combid	prac	patid	patflag	yobstring	hh	sex	regdate	regreal	xferdate	xferreal	regrea	death
1	a670600??	a6706	00??	A	19530000	006564	2	19830324	1983-03-24 00:00:00.000	00000000	NULL	00	00000000
2	a670600?@	a6706	00?@	C	19190000	006695	1	19920213	1992-02-13 00:00:00.000	19991224	1999-12-24 00:00:00.000	01	00000000
3	a670600?1	a6706	00?1	C	19120000	010066	2	19830617	1983-06-17 00:00:00.000	20000413	2000-04-13 00:00:00.000	01	00000000
4	a670600?2	a6706	00?2	A	19840000	003508	1	20070820	2007-08-20 00:00:00.000	00000000	NULL	00	00000000
5	a670600?3	a6706	00?3	A	19340000	000924	1	19910829	1991-08-29 00:00:00.000	00000000	NULL	00	00000000
6	a670600?4	a6706	00?4	A	19380000	000924	2	19910829	1991-08-29 00:00:00.000	00000000	NULL	00	00000000
7	a670600?5	a6706	00?5	C	19120000	001434	1	19910823	1991-08-23 00:00:00.000	19990629	1999-06-29 00:00:00.000	01	00000000
8	a670600?6	a6706	00?6	A	19620000	005605	2	20070924	2007-09-24 00:00:00.000	00000000	NULL	00	00000000
9	a670600?8	a6706	00?8	A	19690000	001058	2	19891229	1989-12-29 00:00:00.000	00000000	NULL	00	00000000
10	a670600?9	a6706	00?9	A	19370000	001005	1	19911126	1991-11-26 00:00:00.000	00000000	NULL	00	00000000
11	a670600?a	a6706	00?a	A	19840000	001450	2	19980527	1998-05-27 00:00:00.000	20050520	2005-05-20 00:00:00.000	02	00000000
12	a670600?B	a6706	00?B	A	19910000	003422	1	19910708	1991-07-08 00:00:00.000	00000000	NULL	00	00000000
13	a670600?b	a6706	00?b	A	19820000	005298	1	19880505	1988-05-05 00:00:00.000	20080704	2008-07-04 00:00:00.000	02	00000000
14	a670600?C	a6706	00?C	A	19640000	003422	2	19861128	1986-11-28 00:00:00.000	00000000	NULL	00	00000000
15	a670600?D	a6706	00?D	C	19250000	003678	2	19900316	1990-03-16 00:00:00.000	19990401	1999-04-01 00:00:00.000	01	00000000
16	a670600?F	a6706	00?F	A	19550000	002730	2	19850118	1985-01-18 00:00:00.000	20061229	2006-12-29 00:00:00.000	02	00000000
17	a670600?f	a6706	00?f	A	19200000	004672	2	19911129	1991-11-29 00:00:00.000	00000000	NULL	00	00000000
18	a670600?g	a6706	00?g	C	19110000	010100	1	19900316	1990-03-16 00:00:00.000	20010614	2001-06-14 00:00:00.000	01	00000000
19	a670600?h	a6706	00?h	A	19260000	009551	2	19900316	1990-03-16 00:00:00.000	20081231	2008-12-31 00:00:00.000	01	20081210
20	a670600?J	a6706	00?J	A	19630000	003108	2	19871222	1987-12-22 00:00:00.000	00000000	NULL	00	00000000
21	a670600?o	a6706	00?o	A	19300000	010100	2	19911121	1991-11-21 00:00:00.000	20091008	2009-10-08 00:00:00.000	01	20090927
22	a670600?P	a6706	00?P	A	19830000	005014	2	19830208	1983-02-08 00:00:00.000	00000000	NULL	00	00000000
23	a670600?R	a6706	00?R	A	19270000	000359	2	19861017	1986-10-17 00:00:00.000	00000000	NULL	00	00000000
24	a670600?S	a6706	00?S	A	19430000	001234	2	19811113	1981-11-13 00:00:00.000	00000000	NULL	00	00000000
25	a670600?s	a6706	00?s	A	19740000	004390	1	20080204	2008-02-04 00:00:00.000	20080814	2008-08-14 00:00:00.000	02	00000000
26	a670600?T	a6706	00?T	A	19350000	001234	1	19840716	1984-07-16 00:00:00.000	00000000	NULL	00	00000000
27	a670600?U	a6706	00?U	A	19660000	004534	2	19870531	1987-05-31 00:00:00.000	00000000	NULL	00	00000000
28	a670600?v	a6706	00?v	A	19400000	006782	1	19810814	1981-08-14 00:00:00.000	20090227	2009-02-27 00:00:00.000	01	20081120

FIGURE A.14: A screen shot of the patient table contained within the THIN database.

Results | Messages

	combid	prac	patid	rxdate	rxdatereal	drugcode	therflag	doscode	rxqty	rxdays	private	staffid	rxtype	opno
1	a67060072	a6706	0072	19950202	1995-02-02 00:00:00.000	96305997	Y	0lqP	14.00000	000	N	0007	1	00000.00
2	a67060072	a6706	0072	19950220	1995-02-20 00:00:00.000	96623996	Y	0lqQ	3.000000	000	N	0007	1	00000.00
3	a67060072	a6706	0072	19950220	1995-02-20 00:00:00.000	96632990	Y	0000142	30.00000	000	N	0007	1	00000.00
4	a67060072	a6706	0072	19970517	1997-05-17 00:00:00.000	99452998	Y	0lqS	28.00000	000	N	0005	1	00000.00
5	a67060072	a6706	0072	19970527	1997-05-27 00:00:00.000	99452998	Y	0lqT	56.00000	000	N	0008	1	00000.00
6	a67060072	a6706	0072	19971107	1997-11-07 00:00:00.000	99220998	Y	0007268	6.000000	000	N	0004	1	00000.00
7	a67060072	a6706	0072	19980212	1998-02-12 00:00:00.000	98798989	Y	0lqV	56.00000	000	N	0009	1	00000.00
8	a67060072	a6706	0072	19980415	1998-04-15 00:00:00.000	89344997	Y	0lqW	30.00000	000	N	0006	1	00000.00
9	a67060072	a6706	0072	19980512	1998-05-12 00:00:00.000	98798989	Y	0lqX	56.00000	000	N	0004	1	00000.00
10	a67060072	a6706	0072	19980512	1998-05-12 00:00:00.000	99220998	Y	0007268	6.000000	000	N	0004	1	00000.00
11	a67060072	a6706	0072	19980529	1998-05-29 00:00:00.000	98446990	Y	0lqa	60.00000	000	N	0004	1	00000.00
12	a67060072	a6706	0072	19980529	1998-05-29 00:00:00.000	89344997	Y	0lqZ	90.00000	000	N	0004	1	00000.00
13	a67060072	a6706	0072	19980727	1998-07-27 00:00:00.000	98798989	Y	0lqb	84.00000	000	N	0004	1	00000.00
14	a67060072	a6706	0072	19981105	1998-11-05 00:00:00.000	89344997	Y	0lqc	90.00000	000	N	0004	1	00000.00
15	a67060072	a6706	0072	19981105	1998-11-05 00:00:00.000	99220998	Y	0007268	6.000000	000	N	0004	1	00000.00
16	a67060072	a6706	0072	19981105	1998-11-05 00:00:00.000	98798989	Y	0lqe	84.00000	000	N	0004	1	00000.00
17	a67060072	a6706	0072	19990208	1999-02-08 00:00:00.000	97126996	Y	0lpg	100.0000	000	N	0004	1	00000.00
18	a67060072	a6706	0072	19990226	1999-02-26 00:00:00.000	94749996	Y	0007268	56.00000	000	N	0004	1	00000.00
19	a67060072	a6706	0072	19990226	1999-02-26 00:00:00.000	98798989	Y	0lqg	84.00000	000	N	0005	1	00000.00
20	a67060072	a6706	0072	19990419	1999-04-19 00:00:00.000	98798989	Y	0lpD	84.00000	000	N	0004	1	00000.00
21	a67060072	a6706	0072	19990419	1999-04-19 00:00:00.000	97217998	Y	0007268	100.0000	000	N	0004	1	00000.00
22	a67060072	a6706	0072	19990419	1999-04-19 00:00:00.000	94749996	Y	0007268	56.00000	000	N	0004	1	00000.00
23	a67060072	a6706	0072	19990419	1999-04-19 00:00:00.000	99220998	Y	0007268	6.000000	000	N	0004	1	00000.00
24	a67060072	a6706	0072	19990615	1999-06-15 00:00:00.000	91240998	Y	0007268	84.00000	000	N	0004	1	00000.00
25	a67060072	a6706	0072	19990802	1999-08-02 00:00:00.000	98798989	Y	0lpC	84.00000	000	N	0004	1	00000.00
26	a67060072	a6706	0072	19990802	1999-08-02 00:00:00.000	94542998	Y	0007268	56.00000	000	N	0004	1	00000.00
27	a67060072	a6706	0072	19990809	1999-08-09 00:00:00.000	99849988	Y	0lqm	15.00000	000	N	0004	1	00000.00
28	a67060072	a6706	0072	19990908	1999-09-08 00:00:00.000	98798989	Y	0lpB	84.00000	000	N	0004	1	00000.00

FIGURE A.15: A screen shot of the therapy table contained within the THIN database.

LINE NUM	FIELD1	FIELD2	FIELD3	FIELD4	FIELD5
1	a793901c8	2	7530.11**	6165	4
2	a793901c8	2	J085.00**	6165	0
3	a793901c8	2	AB01.13**	6165	7
4	a793901c8	2	H02..00**	6165	5
5	a793901c8	2	H037.00**	6165	3
6	a793901c8	2	H05z.00**	6165	0
7	a793901c8	2	R006200**	6165	1
8	a793901c8	2	K190z00**	6165	2
9	a793901c8	2	T804100**	6165	2
10	a985600A2	2	SE...00**	5871	2
11	a985600A2	2	SK10500**	5871	2
12	a985600A2	2	1M10.00**	5871	8
13	a985600A2	2	4K23.00**	5871	8
14	a985600A2	2	7L1L000**	5871	0
15	a985600A2	2	AD20.00**	5871	6
16	a985600A2	2	L20..11**	5871	3
17	a985600A2	2	M05..00**	5871	6
18	a985600A2	2	N094000**	5871	3
19	a985600A2	2	N13y300**	5871	8
20	a985600A2	2	N327.00**	5871	6
21	a985600A2	2	L100.00**	5871	2
22	a985600A2	2	N094611**	5871	6
23	a985600A2	2	M240.00**	5871	0
24	a985600A2	2	K197.00**	5871	6
25	a985600A2	2	K15..00**	5871	7
26	a985600A2	2	R020600**	5871	1
27	a985600A2	2	H05z.00**	5871	1
28	a985600A2	2	E200.00**	5871	1
29	a985600A2	2	E204.11**	5871	3
30	a985600A2	2	J16y400**	5871	8
31	a985600A2	2	H00..12**	5871	4

FIGURE A.16: The DOXEPIN sample data

LINE NUM	FIELD1	FIELD2	FIELD3	FIELD4	FIELD5
1	a6706013B	1	1652.00**	5484	5
2	a6706013B	1	1Z...00**	5484	2
3	a6706013B	1	E227311**	5484	5
4	a6706013B	1	M261000*	5484	0
5	a6706015R	2	1683.00**	3984	6
6	a6706015R	2	1957.00**	3984	6
7	a6706015R	2	1972.00**	3984	6
8	a6706015R	2	1B1G.11**	3984	8
9	a6706015R	2	1BK..00**	3984	7
10	a6706015R	2	1C2..00**	3984	7
11	a6706015R	2	1Z...00**	3984	2
12	a6706015R	2	7305011**	3984	2
13	a6706015R	2	C381.00**	3984	5
14	a6706015R	2	F502.00**	3984	7
15	a6706015R	2	F51y000**	3984	6
16	a6706015R	2	F587.00**	3984	6
17	a6706015R	2	F587.11**	3984	6
18	a670601EQ	1	1Z...00**	5531	0
19	a670601o8	1	171..00**	4678	4
20	a670601o8	1	19C..00**	4678	4
21	a670601o8	1	1A5B.00**	4678	5
22	a670601o8	1	1B1G.11**	4678	4
23	a670601o8	1	1Z...00**	4678	2
24	a670601o8	1	F504.11**	4678	4
25	a670601o8	1	H170.11**	4678	4
26	a670601o8	1	M0...00**	4678	5
27	a670601o8	1	M261000*	4678	3
28	a670601o8	1	N245400*	4678	5
29	a670601o8	1	N324400*	4678	3
30	a670601o8	1	SD...00**	4678	5
31	a670601o8	1	SL...15**	4678	4

FIGURE A.17: The DESLORATADINE sample data

A	B	C	D	E	F
LINE NUM	FIELD1	FIELD2	FIELD3	FIELD4	FIELD5
1	a784008Ay	1	F52..00**	83	14
2	a784008Ay	1	F52..00**	1052	14
3	a784008Ay	1	F52..00**	2781	14
4	a784008Ay	1	F52..00**	3871	14
5	a784008Ay	1	F52..00**	4246	14
6	a784008Ay	1	F52..00**	4382	14
7	a784008Ay	1	F52..00**	4456	14
8	a784008Ay	1	F52..00**	4460	14
9	a784008Ay	1	F52..00**	4754	14
10	a784008Ay	1	F52..00**	4833	14
11	a784008Ay	1	F52..00**	4916	14
12	a784008Ay	1	F52..00**	4981	14
13	a784008Ay	1	F52..00**	4995	14
14	a784008Ay	1	F52..00**	5049	14
15	a784008Ay	1	F52..00**	5176	14
16	a784008Ay	1	F52..00**	5211	14
17	a784008Ay	1	F52..00**	5304	14
18	a784008Ay	1	F52..00**	5363	14
19	a784008Ay	1	F52..00**	5416	14
20	a784008Ay	1	F52..00**	5579	14
21	a784008Ay	1	F52..00**	5614	14
22	a784008Ay	1	F52..00**	5623	14
23	a784008Ay	1	F52..00**	5660	14
24	a784008Ay	1	F52..00**	5673	14
25	a784008Ay	1	F52..00**	5684	14
26	a784008Ay	1	F52..00**	5693	14
27	a784008Ay	1	F52..00**	5747	14
28	a784008Ay	1	F52..00**	5799	14
29	a784008Ay	1	F52..00**	5815	14
30	a784008Ay	1	F52..00**	5827	14
31	a784008Ay	1	F52..00**	5841	14

FIGURE A.18: The LEVOFLOXACIN sample data

Bibliography

[1] M. Arevalillo-Herráez, J. Domingo, and F. J. Ferri, "Combining similarity measures in content-based image retrieval," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2174–2181, 2008.

[2] N. Augsten, M. Böhlen, C. Dyreson, and J. Gamper, "Windowed pq-grams for approximate joins of data-centric xml," *The VLDB Journal*, vol. 21, no. 4, pp. 463–488, 2012.

[3] N. Augsten, M. Böhlen, and J. Gamper, "Approximate matching of hierarchical data using pq-grams," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 301–312.

[4] ——, "Approximate Tree Matching with pq-Grams," http://dbresearch.uni-salzburg.at/publications/2005vldb-approximate-matching/slides-screen.pdf, 2005.

[5] N. Augsten, M. Böhlen, and J. Gamper, "The pq -gram distance between ordered labeled trees," *ACM Transactions on Database Systems*, vol. 35, no. 1, pp. 1–36, Feb. 2010.

[6] K. L. B. Kathryn, B. Sergey, "A scalable clustering method based on density," The University of Texas at Dallas, Tech. Rep., 2006.

[7] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, "Adaptive name matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[8] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[9] B. T. Blak, M. Thompson, H. Dattani, and A. Bourke, "Generalisability of the health improvement network (thin) database: demographics, chronic disease prevalence and mortality rates," *Informatics in primary care*, vol. 19, no. 4, pp. 251–255, 2011.

[10] R. E. Bonner, "On some clustering techniques," *IBM journal of research and development*, vol. 8, no. 1, pp. 22–32, 1964.

[11] I. Borg and P. Groenen, "Modern multidimensional scaling: theory and applications," *Journal of Educational Measurement*, vol. 40, no. 3, pp. 277–280, 2003.

[12] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," *red*, vol. 30, no. 2, p. 3, 2008.

[13] P. Bradley, K. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, pp. 1–8, 2000.

[14] A. H. Cannon, L. J. Cowen, and C. E. Priebe, "Approximate distance classification," *Computing Science and Statistics*, pp. 544–549, 1998.

[15] O. Carugo, "Proximity measures for cluster analysis," in *Data Mining Techniques for the Life Sciences*. Springer, 2010, pp. 163–174.

[16] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[17] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *Knowledge and data Engineering, IEEE Transactions on*, vol. 8, no. 6, pp. 866–883, 1996.

[18] W. Chen, "New algorithm for ordered tree-to-tree correction problem," *Journal of Algorithms*, vol. 40, no. 2, pp. 135–158, 2001.

[19] P. Cichosz, *Data Mining Algorithms: Explained Using R*, 3rd ed. John Wiley & Sons, 10 2015, ch. Clusering Model Evaluation.

[20] J. F. Committee and R. P. S. of Great Britain, *British national formulary (bnf)*. Pharmaceutical Press, 2012, vol. 64.

[21] R. Cornish, "Statistics: Cluster analysis," 2007.

[22] T. Dalamagas, T. Cheng, K.-J. Winkel, and T. Sellis, "A methodology for clustering xml documents by structure," *Information Systems*, vol. 31, no. 3, pp. 187–228, 2006.

[23] R. C. De Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering," *Pattern Recognition*, vol. 45, no. 3, pp. 1061–1075, 2012.

[24] L. J. Deborah, R. Baskaran, and A. Kannan, "A survey on internal validity measure for cluster validation," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, pp. 85–102, 2010.

[25] E. D. Demaine, S. Mozes, B. Rossman, and O. Weimann, "An optimal decomposition algorithm for tree edit distance," *ACM Transactions on Algorithms (TALG)*, vol. 6, no. 1, p. 2, 2009.

[26] J. P. Dijcks, "Oracle: Big data for the enterprise," *Oracle White Paper*, 2012.

[27] E. Dimitriadou, S. Dolničar, and A. Weingessel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika*, vol. 67, no. 1, pp. 137–159, 2002.

[28] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[29] D. Dohare and V. S. Devi, "Combination of similarity measures for time series classification using genetic algorithms," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 401–408.

[30] A. W. Edwards and L. L. Cavalli-Sforza, "A method for cluster analysis," *Biometrics*, pp. 362–375, 1965.

[31] D. Edwards, "Statistical analysis of gene expression microarray data." *Biometrics*, vol. 60, no. 1, pp. 287–289, 2004.

[32] EPIC, "THIN Data Guide for Researcher," 2010.

[33] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[34] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[35] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[36] M. M. M. Fuad and P.-F. Marteau, "The extended edit distance metric," in *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*. IEEE, 2008, pp. 242–248.

[37] P. Ganesan, H. Garcia-Molina, and J. Widom, "Exploiting hierarchical domain structure to compute similarity," *ACM Transactions on Information Systems (TOIS)*, vol. 21, no. 1, pp. 64–93, 2003.

[38] E. Garrett-Mayer and G. Parmigiani, "Clustering and classification methods for gene expression data analysis," *DNA Microarrays*, p. 241, 2007.

[39] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[40] D. R. Goldstein, D. Ghosh, and E. M. Conlon, "Statistical issues in the clustering of gene expression data," *Statistica Sinica*, vol. 12, no. 1, pp. 219–240, 2002.

[41] J. C. Gower, "Euclidean distance geometry," *Mathematical Scientist*, vol. 7, no. 1, pp. 1–14, 1982.

[42] R. L. Grant, V. M. Drennan, G. Rait, I. Petersen, and S. Iliffe, "First diagnosis and management of incontinence in older people with and without dementia in primary care: a cohort study using the health improvement network primary care database," *PLoS Med*, vol. 10, no. 8, p. e1001505, 2013.

[43] P. E. Green and V. R. Rao, "A note on proximity measures and cluster analysis," *Journal of Marketing Research*, pp. 359–364, 1969.

[44] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, 2001.

[45] B. H. Hall, A. B. Jaffe, and M. Trajtenberg, "The nber patent citation data file: Lessons, insights and methodological tools," National Bureau of Economic Research, Tech. Rep., 2001.

[46] D. Hanisch, A. Zien, R. Zimmer, T. Lengauer *et al.*, "Co-clustering of biological networks and gene expression data," in *ISMB*, 2002, pp. 145–154.

[47] D. Hassan, U. Aickelin, and C. Wagner, "Comparison of distance metrics for hierarchical data in medical databases," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 3636–3643.

[48] S. Hosangadi, "Distance measures for sequences," *arXiv preprint arXiv:1208.5713*, 2012.

[49] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[50] L. J. Hubert and J. R. Levin, "A general statistical framework for assessing categorical clustering in free recall." *Psychological bulletin*, vol. 83, no. 6, p. 1072, 1976.

[51] P. Jaccard, "The distribution of the flora in the alpine zone," *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[52] J. E. Jackson, *A user's guide to principal components*. John Wiley & Sons, 2005, vol. 587.

[53] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, 1999.

[54] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[55] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 11, pp. 1370–1386, 2004.

[56] T. Jiang, L. Wang, and K. Zhang, "Alignment of trees-an alternative to tree edit," *Theoretical Computer Science*, vol. 807, pp. 137–148, 1995.

[57] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[58] A. Joshi and R. Krishnapuram, "On mining web access logs," DTIC Document, Tech. Rep., 2000.

[59] S. Jun, "A clustering method of highly dimensional patent data using bayesian approach," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 7–11, 2012.

[60] H. T. Kahraman, S. Sagiroglu, and I. Colak, "The development of intuitive knowledge classifier and the modeling of domain dependent data," *Knowledge-Based Systems*, vol. 37, pp. 283–295, 2013.

[61] K. Kailing, H. Kriegel, S. Schönauer, and T. Seidl, "Efficient similarity search for hierarchical data in large databases," *Advances in Database Technology-EDBT 2004*, pp. 676–693, 2004.

[62] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and AlgoriData*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011, ch. Cluster Analysis.

[63] P. N. Klein, "Computing the edit-distance between unrooted ordered trees," in *Algorithms - ESA 98*. Springer, 1998, pp. 91–102.

[64] T. Körner, "Metric and topological spaces," 2010.

[65] F. Kovács, C. Legány, and A. Babos, "Cluster validity measurement techniques," in *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.

[66] C. Kucharik and J. Norman, "Measuring canopy architecture with a multiband vegetation imager (mvi)," in *CONFERENCE ON AGRICULTURAL AND FOREST METEOROLOGY WITH SYMPOSIUM ON FIRE AND FOREST METEOROLOGY*, vol. 22, 1996, pp. 128–130.

[67] C. Kucharik, J. Norman, L. Murdock, and S. Gower, "Characterizing canopy nonrandomness with a multiband vegetation imager (mvi)," *Journal of Geophysical Research: Atmospheres (1984–2012)*, vol. 102, no. D24, pp. 29 455–29 473, 1997.

[68] R. Lai, A. D Amour, A. Yu, Y. Sun, V. Torvik, and L. Fleming, "Disambiguation and co-authorship networks of the us patent inventor database," *Harvard Institute for Quantitative Social Science, Cambridge, MA*, vol. 2138, 2011.

[69] M. L. Lee, L. H. Yang, W. Hsu, and X. Yang, "Xclust: clustering xml schemas for effective integration," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 292–299.

[70] S. Leigh, "A user's guide to principal components," *Technometrics*, vol. 35, no. 1, pp. 83–85, 1993.

[71] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversible.pdf," *CYBERNETICS AND CONTROL THEORY*, vol. 10, no. 8, 1966.

[72] J. D. Lewis, R. Schinnar, W. B. Bilker, X. Wang, and B. L. Strom, "Validation studies of the health improvement network (thin) database for pharmacoepidemiology research," *Pharmacoepidemiology and drug safety*, vol. 16, no. 4, pp. 393–401, 2007.

[73] W. Lian, D. W.-L. Cheung, N. Mamoulis, and S.-M. Yiu, "An efficient and scalable algorithm for clustering xml documents by structure," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 1, pp. 82–96, 2004.

[74] D. Madigan, P. Ryan, S. Simpson, and I. Zorych, "Bayesian methods in pharmacovigilance," *Bayesian Statistics*, vol. 9, pp. 421–438, 2010.

[75] M. Maechler, *Cluster Analysis Extended Rousseeuw et al.*, R CRAN, 2013.

[76] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, M. Studer, and P. Roudier, "Package cluster," 2015.

[77] A. N. Mahmood, C. Leckie, and P. Udaya, "An efficient clustering scheme to exploit hierarchical data in network traffic analysis," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 6, pp. 752–767, 2008.

[78] R. Maruthaveni and S. Sangamithra, "Gathering large databases using ek-means & pca approach," *International Journal of Computer Applications & Information Technology*, vol. Vol. 5, Feb-March 2014.

[79] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 169–178.

[80] T. Medvedev and A. Ulanov, "Company names matching in the large patents dataset," *Hewlett-Packard Development Company, LP*, 2011.

[81] R. T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 14, no. 5, pp. 1003–1016, 2002.

[82] A. Nierman and H. Jagadish, "Evaluating structural similarity in xml documents." in *WebDB*, vol. 2, 2002, pp. 61–66.

[83] R. P. S. of Great Britain *et al.*, *British National Formulary 61*. Royal Pharmaceutical Society, 2011.

[84] N. Ohkura, K. Hirata, T. Kuboyama, and M. Harao, "The q-gram distance for ordered unlabeled trees," in *Discovery Science*. Springer, 2005, pp. 189–202.

[85] J. C. Oxtoby, "Metric and topological spaces," in *Measure and Category*. Springer, 1971, pp. 39–41.

[86] S. Pandit, S. Gupta *et al.*, "A comparative study on distance measuring approaches for clustering," *International Journal of Research in Computer Science*, vol. 2, no. 1, pp. 29–31, 2011.

[87] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[88] M. Pawlik and N. Augsten, "Rted: a robust algorithm for the tree edit distance," *Proceedings of the VLDB Endowment*, vol. 5, no. 4, pp. 334–345, 2011.

[89] G. Pison, A. Struyf, and P. J. Rousseeuw, "Displaying a clustering with clusplot," *Computational statistics & data analysis*, vol. 30, no. 4, pp. 381–392, 1999.

[90] M. K. Rafsanjani, Z. A. Varzaneh, and N. E. Chukanlo, "A survey of hierarchical clustering algorithms," *The Journal of Mathematics and Computer Science (JMCS)*, vol. 5, no. 3, pp. 229–240, 2012.

[91] S. K. Rai and N. Mishra, "Dbcsvm: Density based clustering using support vector machines," 2012.

[92] A. Rajaraman and J. Ullman, *Mining of massive datasets*, 2011, ch. Finding Similar Items, p. 74.

[93] ——, *Mining of massive datasets*, 2011, ch. Clustering, pp. 225–233.

[94] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[95] A. C. Rencher, *Methods of multivariate analysis*, 2003, vol. 492, ch. Cluster Analysis, pp. 451–494.

[96] J. Reps, J. Feyereisl, J. M. Garibaldi, U. Aickelin, J. E. Gibson, and R. B. Hubbard, "Investigating the detection of adverse drug events in a uk general practice electronic health-care database," *arXiv preprint arXiv:1307.1078*, 2013.

[97] J. Reps, J. M. Garibaldi, U. Aickelin, D. Soria, J. E. Gibson, and R. B. Hubbard, "Discovering sequential patterns in a uk general practice database," in *Biomedical and Health Informatics (BHI), 2012 IEEE-EMBS International Conference on*. IEEE, 2012, pp. 960–963.

[98] J. M. Reps, J. M. Garibaldi, U. Aickelin, D. Soria, J. E. Gibson, and R. B. Hubbard, "A novel semisupervised algorithm for rare prescription side effect discovery," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 2, pp. 537–547, 2014.

[99] J. M. Reps, "Detecting adverse drug reactions in the general practice healthcare database," Ph.D. dissertation, University of Nottingham, 2014.

[100] J. M. Reps, J. M. Garibaldi, U. Aickelin, D. Soria, J. Gibson, and R. Hubbard, "Comparison of algorithms that detect drug side effects using electronic healthcare databases," *Soft Computing*, vol. 17, no. 12, pp. 2381–2397, 2013.

[101] M. R. Rezaee, B. P. Lelieveldt, and J. H. Reiber, "A new cluster validity index for the fuzzy c-mean," *Pattern recognition letters*, vol. 19, no. 3, pp. 237–246, 1998.

[102] L. Ribeiro and T. Härder, "Evaluating performance and quality of xml-based similarity joins," in *Advances in Databases and Information Systems*. Springer, 2008, pp. 246–261.

[103] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[104] C. Ros, D. Güllmar, and J. Reichenbach, "A new combined distance measure for the clustering of fiber tracts in diffusion tensor imaging (dti)," *Proc Intl Soc Mag Reson Med. Stockholm, Sweden*, vol. 18, 2010.

[105] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[106] H. Samet, "Applications of spatial data structures," 1990.

[107] ——, "Hierarchical spatial data structures," in *Design and Implementation of Large Spatial Databases*. Springer, 1990, pp. 191–212.

[108] K. U. Schulz and S. Mihov, "Fast string correction with levenshtein automata," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 67–85, 2002.

[109] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing shock graphs," in *null*. IEEE, 2001, p. 755.

[110] S. M. Selkow, "THE TREE-TO-TREE EDITING PROBLEM," vol. 6, no. 6, 1977.

[111] R. Shahid, S. Bertazzon, M. L. Knudtson, and W. A. Ghali, "Comparison of distance measures in spatial analytical modeling for health service planning," *BMC health services research*, vol. 9, no. 1, p. 200, 2009.

[112] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: A review," in *Computational Science and Its Applications–ICCSA 2014*. Springer, 2014, pp. 707–720.

[113] A. Silberschatz, H. Korth, and S. Sudarshan, *Database Systems Concepts*, 5th ed. New York, NY, USA: McGraw-Hill, Inc., 2006.

[114] A. Singh, A. Yadav, and A. Rana, "K-means with three different distance metrics," *International Journal of Computer Applications*, vol. 67, no. 10, pp. 13–17, 2013.

[115] D. Soria, "Novel methods to elucidate core classes in multidimensional biomedical data," Ph.D. dissertation, Computer Science, 2010.

[116] N. Srivastava, V. Mishra, and A. Bhattacharya, "Analyzing the sensitivity of pq-gram distance with p and q," *ACM Transactions on Database Systems (TODS) vol*, vol. 35, p. 4, 2010.

[117] C. H. Suryanto, H. Hino, and K. Fukui, "Combination of multiple distance measures for protein fold classification," in *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*. IEEE, 2013, pp. 440–445.

[118] K.-C. Tai, "The tree-to-tree correction problem," *Journal of the ACM (JACM)*, vol. 26, no. 3, pp. 422–433, 1979.

[119] P.-N. Tan, M. Steinbach, and V. Kumar, "Data mining cluster analysis: Basic concepts and algorithms," 2013.

[120] J. Tantrum, A. Murua, and W. Stuetzle, "Assessment and pruning of hierarchical model based clustering," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 197–205.

[121] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[122] G. Valiente, "An efficient bottom-up distance between trees," in *spire*. IEEE, 2001, p. 0212.

[123] R. Vijay, P. Mahajan, and R. Kandwal, "Hamming distance based clustering algorithm," *International Journal of Information Retrieval Research (IJIRR)*, vol. 2, no. 1, pp. 11–20, 2012.

[124] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[125] J. T. Wang, K. Zhang, G. Chang, and D. Shasha, "Finding approximate patterns in undirected acyclic graphs," *Pattern Recognition*, vol. 35, no. 2, pp. 473–483, 2002.

[126] X. Y. Wang, "Fuzzy clustering in the analysis of fourier transform infrared spectra for cancer diagnosis," Ph.D. dissertation, Citeseer, 2006.

[127] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[128] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005, pp. 99–108.

[129] R. Xu and D. Wunsch, "Survey of clustering algorithms." *IEEE transactions on neural networks, a publication of the IEEE Neural Networks Council*, vol. 16, no. 3, pp. 645–78, May 2005.

[130] K. Zhang and D. Shasha, "Simple Fast Algorithms for the Editing Distance between Trees and Related Problems," *SIAM Journal on Computing*, vol. 18, no. 6, pp. 1245–1262, Dec. 1989.

[131] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *ACM SIGMOD Record*, vol. 25, no. 2. ACM, 1996, pp. 103–114.

[132] ——, "Birch: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.