Banded Pattern Mining For N-Dimensional Zero-One Data

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy

by

Fatimah Binta Abdullahi

Faculty of Science
Department of Computer Science

October 2016

Dedication

To the memory of my Father Abdullahi Momoh.

Acknowledgements

I will like to express my gratitude first and foremost to my primary supersivor Prof Frans Coenen, specially for his support, constant patience and encourangement throughout the past four years. His constructive critism, research ideas, enthusiam and always making time for discussions have made the completeion of my Ph.D possible. I am indeed previledged to have worked with him. I am also gratiful to my second supervisor, Dr Russell Martin, for his suggestions, assistance and valuable comments.

I would like to extend my profound gratitute to my "PhD Advisors": Professor Paul Dunne, Professor Piotr Krysta and Dr Mohammad Khan for providing me with constructive feedback and suggestions at various times. The Department of Computer Science at the University of Liverpool has been an excellent place to conduct research; all staff members and colleagues have been helpful whenever necessary.

I would also like to extend my profound gratitude to the Government of Nigeria, especially the Petroleum Technology Development Fund (PTDF), for their financial support, that enabled me to conduct this research. My deepest thanks also goes to my family and friends in Nigeria for their support, trust, encouragement and prayers.

Last but not the least, I am eternally indebted to my husband for his encouragement and constant support; motivation without which the completion of my PhD would not have been possible.

Abstract

The objective of the research presented in this thesis is to investigate and evaluate a series of techniques directed at identifying banded patterns in zero-one data, starting from 2D data and then increasing the number of dimensions to be considered to 3D and then ND. To this end the term Banded Pattern Mining (BPM) has been coined; the process of extracting hidden banded patterns from data. BPM has wide applicability in areas where the domain of interest can be represented in the form of a matrix holding $1s$ and $0s$. Five BPM algorithms are proposed (and a number of variations) directed at finding bandings in zero-one data: (i) 2D-BPM, (ii) Approximate 3D-BPM, (iii) Exact 3D-BPM, (iv) Approximate ND (AND) and (iv) Exact ND (END) BPM. This thesis describes and discusses each of these algorithms in detail.

The main challenges of BPM are: (i) how best to identify bandings in 2D data sets without the need to consider large numbers of permutations, (ii) how to address situations where there is a possibility of multiple dots being located at individual locations in a ND zero-one (dot) data space of interest and (iii) how best to identify bandings with respect to large ND data sets that cannot be held in primary storage. To address the first issue a banding score mechanism is proposed that avoids the need to consider large numbers of permutations. This has been incorporated into the 2D-BPM algorithm. To address the second issue, the idea was to use a "multiple dot" mechanism; a mechanism in the context of both the approximate and exact BPM algorithms that includes the possibility of some cells in the data space of interest holding more than one dot. To address the third issue, sampling and segmentation techniques were proposed to identify bandings in large ND data sets.

Full evaluations of each of the BPM algorithms are presented. For evaluation purpose the data sets used were categorised as follows: (i) randomly generated synthetic data sets, (ii) UCI data sets and (iii) a specific application; the Great Britain (GB) Cattle Tracing System (CTS) database in operation in GB, from which 5D binary valued data sets were extracted and used. In the latter case the dimensions were: (i) records (number of animal movements), (ii) attributes, (iii) sender easting (x coordinate holding area), (iv) sender northing (y-coordinate holding areas) and (v) time (month). Other Banding application domains that could have been considered include: (i) network analysis, (ii)

co-occurence analysis, (iii) VLSI chip design and (iv) graph drawing. An independent metric, Average Band Width (ABW), was proposed and used to measure the quality of bandings and provide a mechanism for the comparison of BPM algorithms. More specifically, data sets from 2003 to 2006 across four specific counties in GB were used; Aberdeenshire, Cornwall, Lancashire and Norfolk. The reported evaluation indicates that the use of approximate BPM (rather than exact BPM) produces more efficient results in terms of run-time, whilst the use of exact BPM provided promising results in terms of the quality of the bandings produced. The reported evaluation also indicates that a sound foundation has been established for future work with respect to high performance computing variations of the proposed BPM algorithms.

Contents

Dedication i

Acknowledgements ii

Abstract iii

Content v

List of Figures x

List of Tables xiv

List of Algorithms xviii

Notations xix

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Research Motivation . . . 3
- 1.3 Research Questions and Issues . . . 4
- 1.4 Research Methodology . . . 5
- 1.5 Research Contribution . . . 7
- 1.6 Publications . . . 9
- 1.7 Structure of Thesis . . . 11
- 1.8 Summary . . . 12

2 Literature Review and Previous Work 13
- 2.1 Introduction . . . 13
- 2.2 Overview of Banded Pattern Mining . . . 13
  - 2.2.1 Advantages of Banding . . . 14
  - 2.2.2 Banding Application Areas . . . 15
  - 2.2.3 Numerical Analysis . . . 17
  - 2.2.4 Reorderable Matrices . . . 19
  - 2.2.5 Reorderable Pattern . . . 22
  - 2.2.6 Bandwidth Minimisation Problem (BMP) . . . 24
  - 2.2.7 Matrix Seriation . . . 24
  - 2.2.8 Banded Pattern Mining . . . 26

2.2.9 Banded Pattern Mining Summary and Discussion . . . 27
2.3 Review of Selected Banding Algorithms . . . 28
2.3.1 Barycenter (BC) Algorithm . . . 28
2.3.1.1 Fixed Permutation (MBA) Algorithm . . . 30
2.3.1.2 Bi-directional Fixed Permutation (MBA) Algorithm . . . 32
2.4 Data Mining And Knowledge Discovery in Databases (KDD) Process . . . 34
2.4.1 The KDD Process . . . 35
2.4.2 The DM KDD Sub-Process . . . 36
2.4.3 Frequent Item-set Mining (FIM) . . . 36
2.5 Sampling and Segmentation Techniques . . . 37
2.5.1 Sampling Technique . . . 38
2.5.2 Segmentation technique . . . 39
2.6 Evaluating Criteria . . . 39
2.7 Summary . . . 40

3 Evaluation Datasets **42**
3.1 Introduction . . . 42
3.2 Randomly Generated Sythentic Data . . . 42
3.3 University Of California Irvine (UCI) Data sets . . . 43
3.4 Great Britain (GB) Cattle Tracing System . . . 45
3.4.1 CTS Data Set Construction . . . 47
3.4.2 3D CTS data sets . . . 48
3.4.3 5D CTS data sets . . . 49
3.4.4 GB cattle CTS data set For Sampling . . . 51
3.4.5 GB cattle CTS data set For Segmentation . . . 54
3.5 Summary of Data . . . 54

4 2D Banding Mechanism **55**
4.1 Introduction . . . 55
4.2 2D Banding Formalism . . . 56
4.3 2D Banding Score Calculation . . . 57
4.4 2D Global Banding Score Calculation . . . 58
4.5 The 2D Banded Pattern Mining (2D-BPM) Algorithm . . . 61
4.6 A Working Example using the 2D-BPM Algorithm . . . 62
4.7 Evaluation of the 2D-BPM Algorithm . . . 64
4.7.1 Analysis of 2D-BPM algorithm in terms of number of iterations . . . 65
4.7.2 Efficiency of 2D-BPM Algorithm using Synthetic Data sets . . . 65
4.7.3 Run-time Comparison Using UCI Data sets . . . 68
4.7.4 Banding Quality of 2D-BPM algorithm Using UCI Data sets . . . 69
4.7.5 Effectiveness of Banding with respect to Frequent Item-set Mining (FIM) . . . 76
4.8 Summary . . . 78

5 Approximate Banding Mechanism **80**
5.1 Introduction . . . 80
5.2 3D Approximated Banding Formalism and Calculation of Banding Scores 81
5.3 Overview of Approximate 3D Banded Pattern Mining (A3D-BPM) Mechanism . . . 84

5.4 A Working Example Using the A3D-BPM Algorithm 85
5.5 Summary 87

6 Exact Banding Mechanism 90
6.1 Introduction 90
6.2 Formalism and Banding Score Calculation 91
6.3 Maximum Distance Tables 95
6.3.1 M-Table Generation and the MDC Algorithm 96
6.3.2 M-Table Construction Example 98
6.4 The Exact 3D Banded Pattern Mining (E3D-BPM) Algorithm 99
6.4.1 The E3D-BPM Algorithms 99
6.4.2 A Working Example Using the E3D-BPM Algorithms 99
6.5 Evaluation of E3D-BPM Mechanism 108
6.5.1 Comparison Between E3D-BPM And A3D-BPM Algorithms In Term of Run-times 108
6.5.2 Comparison Between E3D-BPM And A3D-BPM Algorithms In Term of Global Banding Score (GBS) Values 109
6.6 Summary 112

7 ND Banded Pattern Mining Mechanisms 114
7.1 Introduction 114
7.2 ND Banding Formalism and Calculation of Banding Score 115
7.3 M-Tables in ND Space 116
7.4 ND Banded Pattern Mining (ND-BPM) Algorithms 117
7.4.1 Approximate ND Banded Pattern Mining (AND-BPM) Algorithm 117
7.4.2 Exact ND Banded Pattern Mining (END-BPM) Algorithm 119
7.4.3 Theoretical Complexity of the ND Banded Pattern Mining (ND-BPM) Algorithm 120
7.5 Evaluations of the ND Banded Pattern Mining (ND-BPM) Mechanism . . 120
7.5.1 Comparison Between END-BPM And AND-BPM Algorithms In Term of Run-times 121
7.5.2 Comparison Between END-BPM And AND-BPM Algorithms In Term of Global Banding Score (GBS) Values 123
7.6 Summary 126

8 Multiple Dot Mechanism 128
8.1 Introduction 128
8.2 Multiple Dot Banding Formalism and Calculation of Banding Score . . . 129
8.3 MD Banded Pattern Mining (MD-BPM) Algorithms 130
8.4 A Worked Example Using the MD-BPM Algorithms 131
8.5 Summary 134

9 Discovering Bandings in Big Data Using Sampling and Segmentation 136
9.1 Introduction 136
9.2 BPM Sampling Technique 137
9.3 BPM Segmentation Technique 139
9.4 Evaluation 140
9.4.1 Efficiency Comparison Using Sampling and Segmentation 142

9.4.2 Effectiveness of MD-EBPM and MD-ABPM Algorithms Using Sampling and Segmentation . . . 147
9.5 Summary . . . 156

10 Statistical Significance Testing Using Gaussian Distributions 157
10.1 Introduction . . . 157
10.2 Overview of Statistical Significance Testing . . . 158
10.3 Normal Distribution Curve Generation . . . 160
10.3.1 Static Dot Density . . . 161
10.3.2 Range of Dot Density Values . . . 163
10.4 Banded Pattern Mining Significance Testing . . . 166
10.5 Summary . . . 166

11 Conclusion and Future Research Works 168
11.1 Introduction . . . 168
11.2 Summary . . . 168
11.3 Main Findings and Contribution . . . 170
11.4 Future Work . . . 173

Reference 176

A Additional 2D-BPM Worked Examples 186
A.1 Introduction . . . 186
A.2 A Worked Example Using 2D-BPM Algorithm . . . 186
A.2.1 A Worked Example 1 . . . 186
A.2.2 A Worked Example 2 . . . 190
A.3 A Worked Example Using MD-BPM Algorithm . . . 194
A.3.1 A Multiple Dots Example 1 . . . 194
A.3.2 A Multiple Dots Example 2 . . . 197

B Additional Sampling Experimental Result 200
B.1 Introduction . . . 200
B.2 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Technique in Terms of Run time (RT) for 12,000 Records (1,000 per month) 201
B.3 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Techniques in Terms of Global Banding Score (GBS) for 12,000 records (1,000 per month) . . . 203
B.4 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Technique in Terms of Run time (RT) for 36,000 Records (3,000 per month) 205
B.5 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Techniques in Terms of Global Banding Score (GBS) for 36,000 Records (3,000 per month) . . . 207

C Additional Experimental Result on Segmentation 209
C.1 Introduction . . . 209
C.2 Effectiveness Results in Terms of GBS Using the Euclidean MD-EBPM Algorithm . . . 209

C.3 Effectiveness Results in Terms of GBS Using the Manhattan MD-EBPM Algorithm 214
C.4 Effectiveness Results in Terms of GBS Using the MD-ABPM Algorithm . 218

D Some Additional Analysis Concerning Number of Iterations to Arrive at a Best Banding 222
D.1 Introduction 222
D.2 Further Analysis of 2D-BPM Algorithm in Terms of Number of Iterations 222
D.3 Further Analysis of MD-BPM Algorithm in Terms of Number of Iterations 223

Illustrations

List of Figures

1.1 $2D$ Banding Example: (a) original matrix and (b) original matrix with the rows and columns reordered to reveal a banding . . . 2
1.2 $3D$ Banding Example: (a) original 3D matrix and (b) original 3D matrix with Dim_1, Dim_2 and Dim_3 reordered to feature a banding . . . 2
1.3 Hierarchical categorization of the BPM algorithms proposed and evaluated in the context of the research presented in this thesis. . . . 8

2.1 Example of a fully banded 2D matrix [56] . . . 15
2.2 System of linear equations ([74]) . . . 18
2.3 Example of a system represented as matrix equation ([74]) . . . 18
2.4 Augmented matrix Figure 2.4(a) and process for deriving a reduced echelon matrix form . . . 19
2.5 Bertin device for matrix construction and reconstruction [116] . . . 20
2.6 The reorderable matrix user interface [116] . . . 20
2.7 (a) original matrix and (b) reordered rows and columns of original matrix to reveal a pattern of interest [56] . . . 23
2.8 Example bandwidth minimisation [116] . . . 24
2.9 Example Matrix Seriation[82] . . . 25
2.10 Example illustrating the Barycenter (BC) Algorithm [116] . . . 29
2.11 Example illustrating the MBA Fixed Permutation (MBA_{FP}) Algorithm [56] . 32
2.12 Example illustrating the MBA Bidirectional Fixed Permutation (MBA_{BFP}) Algorithm [56] . . . 34
2.13 KDD process functional steps [45] . . . 35
2.14 Data Mining Goals [45] . . . 37

3.1 January 2013 GB cattle movement data: (a) vertices (locations) and (b) edges (cattle movements) . . . 46
3.2 Close up of Figure 3.1(a) showing North Wales and part of the west coast of England . . . 47

4.1 Examples of 2D dot configurations featuring: (a) a perfect banding and (b) an alternative banding . . . 56
4.2 Example data configurations: (a) all dots ("worst" GBS of 1) and (b) no dots ("best" GBS of 0) . . . 60
4.3 Permutations for dot marix given in Figure 4.1 . . . 60
4.4 Example of the operation of the 2D-BPM algorithm . . . 63
4.5 GBS values per number of iterations . . . 66

4.6 Recorded run time (seconds) using the 2D-BPM, BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms and a range of data sets of increasing size (10,000 to 100,000 in steps of 10,000) . . . 67
4.7 Recorded run time (seconds) using the 2D-BPM, BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms and a range of data sets of increasing density (10% to 50% increasing in steps of 10%) . . . 68
4.8 Wine raw data set: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm, (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA$_{BFP}$ algorithm . . . 73
4.9 Iris raw dataset: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm, (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA$_{BFP}$ algorithm . . . 74
4.10 Glass raw dataset: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA$_{BFP}$ algorithm . . . 75

5.1 Example of a $3D$ dot Configuration featuring a perfect banding . . . 81
5.2 Example of a $3D$ dot Configuration featuring an alternative banding . . . 82
5.3 Input Data Perspective 1 . . . 87
5.4 Input Data Perspective 2 . . . 87
5.5 Input Data Perspective 3 . . . 87
5.6 Input Data rearranged using A3D-BPM after the first iteration, perspective 1 88
5.7 Input Data rearranged using A3D-BPM after the first iteration, perspective 2 88
5.8 Input Data rearranged using A3D-BPM after the first iteration, perspective 3 88
5.9 Input Data rearranged using A3D-BPM after the second iteration, perspective 1 . . . 89
5.10 Input Data rearranged using A3D-BPM after the second iteration, perspective 2 . . . 89
5.11 Input Data rearranged using A3D-BPM after the second iteration, perspective 3 . . . 89

6.1 Example M-Table . . . 95
6.2 M-Tables for example configurations given in Figures 5.1 and 5.2: (a) Euclidean and (b) Manhattan . . . 96
6.3 The order in which locations are selected when generating M-Tables using Euclidean and Manhattan distance calculation, for the spaces 5×6, 8×6 and 8×5. . . . 100
6.4 Example M-Tables (Euclidean and Manhattan) for the illustration of the operation of the MDC algorithm given in Section 6.3.2 . . . 102
6.5 Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 1 . . . 102

6.6 Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 2 . . . 103
6.7 Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 3 . . . 104
6.8 Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 1 . . . 104
6.9 Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 2 . . . 104
6.10 Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 3 . . . 105
6.11 Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 1 . . . 105
6.12 Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 2 . . . 106
6.13 Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 3 . . . 106
6.14 Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 1 . . . 107
6.15 Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 2 . . . 107
6.16 Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 3 . . . 107
6.17 A 2D space illustrating the distinction between Euclidean and Manhattan distance calculation . . . 111

7.1 Compartive Complexity of END-BPM and AND-BPM algorithms . . . 120
7.2 A 3D space illustrating the distinction between Euclidean and Manhattan distance calculation . . . 124

8.1 Input "Dot matrix" for worked example . . . 131
8.2 Dot matrix after rearrangement of Dim_x (iteration 1) . . . 132
8.3 Dot matrix after rearrangement of Dim_y (iteration 1) . . . 132
8.4 Dot matrix after rearrangement of Dim_x (iteration 2) . . . 133
8.5 Dot matrix after rearrangement of Dim_y (iteration 2) . . . 134

9.1 Sampling Run time results using MD-EBPM and MD-ABPM Algorithms . . . 148
9.2 Segmentation Run time results using MD-EBPM and MD-ABPM Algorithms 149
9.3 GBS values for 2003 comparison of 3D data sets using the Euclidean MD-EBPM Algorithm . . . 150
9.4 2003 GBS values comparison for 4D and 5D data sets using MD-EBPM and MD-ABPM Algorithms . . . 155

10.1 Gaussian or Normal Probability Curve [69] . . . 159

10.2 Standard Normal Distribution Table [88] 159
10.3 three-sigma rule for the normal distribution [105] 160
10.4 Line graphs indicatng the number of times per GBS 162
10.5 Line graphs indicatng the number of times per GBS 165

A.1 Input matrix 187
A.2 Input matrix after rearrangement of Dim_x (iteration 1) 187
A.3 Input matrix after rearrangement of Dim_y (iteration 1 188
A.4 Input matrix after rearrangement of Dim_x (iteration 2 189
A.5 Input matrix after rearrangement of Dim_y (iteration 2 190
A.6 Example matrix 191
A.7 Example matrix after rearrangement of Dim_x (iteration 1) 191
A.8 Example matrix after rearrangement of Dim_y (iteration 1) 192
A.9 Example matrix after rearrangement of Dim_x (iteration 2) 193
A.10 Example matrix after rearrangement of Dim_y (iteration 2) 193
A.11 Dot matrix 194
A.12 Dot matrix after rearrangement of Dim_x (iteration 1) 195
A.13 Dot matrix after rearrangement of Dim_y (iteration 1) 195
A.14 Dot matrix after rearrangement of Dim_x (iteration 2) 197
A.15 Dot matrix after rearrangement of Dim_y (iteration 2) 197
A.16 Input dot matrix 198
A.17 Input dot matrix after rearrangement of Dim_x (iteration 1) 198
A.18 Input dot matrix after rearrangement of Dim_y (iteration 1) 199

D.1 GBS values per number of iterations for the remaining four UCI data sets . . 223
D.2 GBS values versus the number of iterations using the Euclidean MD-EBPM and Manhattan MD-EBPM algorithms 225
D.3 GBS values versus the number of iterations using the MD-ABPM algorithm 225

List of Tables

2.1 Calculation of barycenter values for row 29
2.2 Calculation of barycenter values for columns 29
2.3 Summary of Sampling Methods [24] 38

3.1 Statistical summary of selected UCI data sets 45
3.2 Statistical summary of 3D CTS data sets 50
3.3 Statistical summary of 5D CTS data sets for 2003 50
3.4 Statistical summary of 5D CTS data sets for 2004 51
3.5 Statistical summary of 5D CTS data sets for 2005 51
3.6 Statistical summary of 5D CTS data sets for 2006 52
3.7 Statistical summary of the 16 (sample) 4D CTS data sets 53
3.8 Statistical summary of the 16 (sample) 5D CTS data sets 53

4.1 Calculation of BS values for Dim_x 63
4.2 Calculation of BS values for Dim_y 63
4.3 GBS results obtained using the 2D-BPM algorithm and the comparator algorithms for a range of dot matrices of increasing size 68
4.4 Run-time (RT) Results (seconds) Using UCI data sets. 69
4.5 Quality of banding in terms of GBS using 2D UCI data set (best results presented in bold font). 70
4.6 Quality of banding in terms of MRM using 2D UCI data set (best results presented in bold font). 71
4.7 Quality of banding in terms of Accuracy using 2D UCI data set (best results presented in bold font). 71
4.8 Quality of banding in terms of ABW using 2D UCI data sets (best results presented in bold font). 72
4.9 FIM runtime (seconds) with and without banding using 2D-BPM ($\sigma = 2\%$) . 77
4.10 FIM runtime (seconds) with and without banding using MBA_{BFP} ($\sigma = 2\%$) 77
4.11 FIM runtime (seconds) with and without banding using MBA_{FP} ($\sigma = 2\%$) . 78
4.12 FIM runtime (seconds) with and without banding using BC ($\sigma = 2\%$) 78

6.1 Summary of final GBS values obtained with respect to illustration given in Section 6.2 95
6.2 MDC using Euclidean and Manhattan distance calculations 101
6.3 Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Eastings data sets. 110
6.4 Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Northings data sets 110

6.5 Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Temporal data sets . . . 111
6.6 Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Eastings data sets (best results in bold font). . . . 112
6.7 Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Northing data sets (best results in bold font). . . . 112
6.8 Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Temporal data sets (best results in bold font). . . 113

7.1 Runtime results (seconds) for 2003 5D CTS data sets using: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM 122
7.2 Runtime results (seconds) for 2004 5D CTS data sets using: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM 122
7.3 Runtime results (seconds) for 2005 5D CTS data sets using:: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM 123
7.4 Runtime results (seconds) for 2006 5D CTS data sets using:: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM 123
7.5 GBS results for 2003 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM 125
7.6 GBS results for 2004 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM 125
7.7 GBS results for 2005 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM 126
7.8 GBS results for 2006 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM 126

8.1 Calculation of banding scores for dimension x (iteration 1) 131
8.2 Calculation of banding scores for dimension y (iteration 1) 132
8.3 Calculation of banding scores for dimension x (iteration 2) 133
8.4 Calculation of banding scores for dimension y (iteration 2) 134

9.1 Statistical summary of number of records per segmentation 141
9.2 Sampling and Segmentation Runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Aberdeenshire and Cornwall) 143
9.3 Sampling and Segmentation Runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Lancashire and Norfolk) 144

9.4 Sampling and Segmentation Runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Aberdeenshire and Cornwall) . . . 145
9.5 Sampling and Segmentation Runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Lancashire and Norfolk) . . . 146
9.6 Sampling and Segmentation GBS Result for 2003 to 2006 3D and 4D CTS data set (Aberdeenshire and Cornwall) . . . 151
9.7 Sampling and Segmentation GBS Result for 2003 to 2006 3D and 4D CTS data set (Lancashire and Norfolk) . . . 152
9.8 Sampling and Segmentation GBS Result for 2003 to 2006 5D CTS data set (Aberdeenshire and Cornwall) . . . 153
9.9 Sampling and Segmentation GBS Result for 2003 to 2006 5D CTS data set (Lancashire and Norfolk) . . . 154

10.1 List of GBS Occurrence Counts per data set configuration (static dot density) 161
10.2 Mean and Standard Deviation values extracted from data presented in Table 10.1 (static dot density) . . . 161
10.3 List of GBS Occurrence Counts per data set configuration (ranged dot density) . . . 163
10.4 Mean and Standard Deviation values extracted from data presented in Table 10.3 (ranged dot density) . . . 164
10.5 GBS results with Normal Distribution (static dot density) . . . 166
10.6 GBS results with Normal Distribution (ranged dot density) . . . 167

A.1 Example 1 Calculation of banding scores for dimension x (iteration 1) . . . 187
A.2 Example 1 Calculation of banding scores for dimension y (iteration 1) . . . 188
A.3 Example 1 Calculation of banding scores for dimension x (iteration 2) . . . 189
A.4 Example 1 Calculation of banding scores for dimension y (iteration 2) . . . 190
A.5 Example 2 Calculation of banding scores for dimension x (iteration 1) . . . 191
A.6 Example 2 Calculation of banding scores for dimension y (iteration 1) . . . 192
A.7 Example 2 Calculation of banding scores for dimension x (iteration 2) . . . 193
A.8 Example 2 Calculation of banding scores for dimension y (iteration 2) . . . 194
A.9 Example 3 Calculation of banding scores for dimension x (iteration 1) . . . 195
A.10 Example 3 Calculation of banding scores for dimension y (iteration 1) . . . 196
A.11 Example 3 Calculation of banding scores for dimension x (iteration 2) . . . 196
A.12 Example 3 Calculation of banding scores for dimension y (iteration 2) . . . 197
A.13 Example 4 Calculation of banding scores for dimension x (iteration 1) . . . 198
A.14 Example 4 Calculation of banding scores for dimension y (iteration 1) . . . 199

B.1 Sampling runtime results (seconds) for 3D and 4D CTS data sets using MD-EBPM and MD-ABPM algorithms with M-Tables . . . 201

B.2 Sampling runtime results (seconds) for 5D CTS data sets using MD-EBPM and MD-ABPM algorithms with M-Tables 202
B.3 Sampling GBS result for 2003 to 2006 3D and 4D CTS data set 203
B.4 Sampling GBS result for 2003 to 2006 5D CTS data set 204
B.5 Sampling runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM algorithms with M-Tables 205
B.6 Sampling runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM algorithms with M-Tables 206
B.7 Sampling GBS result for 2003 to 2006 3D and 4D CTS data set 207
B.8 Sampling GBS result for 2003 to 2006 5D CTS data set 208

C.1 GBS results using the Euclidean MD-EBPM algorithm for Aberdeenshire . . 210
C.2 GBS results using the Euclidean MD-EBPM algorithm for Cornwall 211
C.3 GBS results using the Euclidean MD-EBPM algorithm for Lancashire 212
C.4 GBS results using the Euclidean MD-EBPM algorithm for Norfolk 213
C.5 GBS results using the Manhattan MD-EBPM algorithm for Aberdeenshire . 214
C.6 GBS results using the Manhattan MD-EBPM algorithm For Cornwall 215
C.7 GBS results using the Manhattan MD-EBPM algorithm for Lancashire . . . 216
C.8 GBS results using the Manhattan MD-EBPM algorithm for Norfolk 217
C.9 GBS results using the MD-ABPM algorithm for Aberdeenshire 218
C.10 GBS results using the MD-ABPM algorithm for Cornwall 219
C.11 GBS results using the MD-ABPM algorithm for Lancashire 220
C.12 GBS results using the MD-ABPM algorithm for Norfolk 221

List of Algorithms

1 The Barycenter (BC) algorithm 28
2 The Fixed Permutation (FP) MBA Algorithm 31
3 The Bidirectional Fixed Permutation (BFP) MBA Algorithm 33
4 Random Data Generation Algorithm 44
5 The 2D-BPM Algorithm 62
6 The A3D-BPM Algorithm 86
7 MDC Algorithm 97
8 Calculate M-Table Row Algorithm 98
9 The E3D-BPM Algorithm 103
10 The AND-BPM Algorithm 118
11 The END-BPM Algorithm 119
12 BPM with Sampling Algorithm 139
13 BPM with Segmentation Algorithm 140

Notations

The following notations and abbreviations are found throughout the thesis:

DIM	The set of dimensions $\{Dim_1, Dim_2, \ldots, Dim_n\}$.
D	A binary valued data matrix subscribing to DIM.
K	Dimension sizes $\{k_1, k_2, \ldots, k_n\}; \lvert Dim_1 = k_1\rvert, \lvert Dim_2 = k_2\rvert$ and so on.
2D	Two Dimensional.
3D	Three Dimensional.
5D	Five Dimensional.
ND	N-Dimensional.
BS	Banding Score.
$\mathbf{bs}_{i_p}$	Banding score for a particular index p in dimension i.
$\mathbf{bs}_{ij_p}$	Banding score for index p in dimension i with respect to dimension j.
GBS	Global Banding Score.
$\mathbf{GBS}_i$	The GBS for dimension i.
$\mathbf{GBS}_{ij}$	The GBS for dimension i with respect to dimension j.
BPM	Banded Pattern Mining.
MDC	Maximum Distance Calculation.
KDD	Knowledge Discovery in Databases.
DM	Data Mining.
2D-BPM	Two Dimensional Banded Pattern Mining.
3D-BPM	Three Dimensional Banded Pattern Mining.
ND-BPM	N Dimensional Banded Pattern Mining.
$\mathbf{e}_{ij}$	Index e in dimension i currently at position j.
M	The set of maximum distances.
BMP	Bandwidth Minimisation Problem.
A3D-BPM	Approximate Three Dimensional Banded Pattern Mining.
E3D-BPM	Exact Three Dimensional Banded Pattern Mining.
AND-BPM	Approximate N Dimensional Banded Pattern Mining.
END-BPM	Exact N Dimensional Banded Pattern Mining.
MD-BPM	Multiple Dot Banded Pattern Mining.
MD-ABPM	Multiple Dot Approximate Banded Pattern Mining.
MD-EBPM	Multiple Dot Exact Banded Pattern Mining.
MBA	Minimum Banded Augmentation.

MBA$_{FP}$	Minimum Banded Augmentation with Fixed Permutation.
MBA$_{BFP}$	Minimum Banded Augmentation with Bi-directional Fixed Permutation.
BC	Barycenter.
n×m	Size of matrix: n rows and m columns.
G(V,E)	Graph with the vertex set V and edge set E.
CTS	Cattle Tracing System.
UCI	University of California Irvine.
ABW	Average Band Width.
GB	Great Britain.
VLSI	Very Large Scale Integration.
C1P	Consecutive-One Property.
FIM	Frequent Item-set Mining.
MRM	Mean Row Moment.

Chapter 1

Introduction

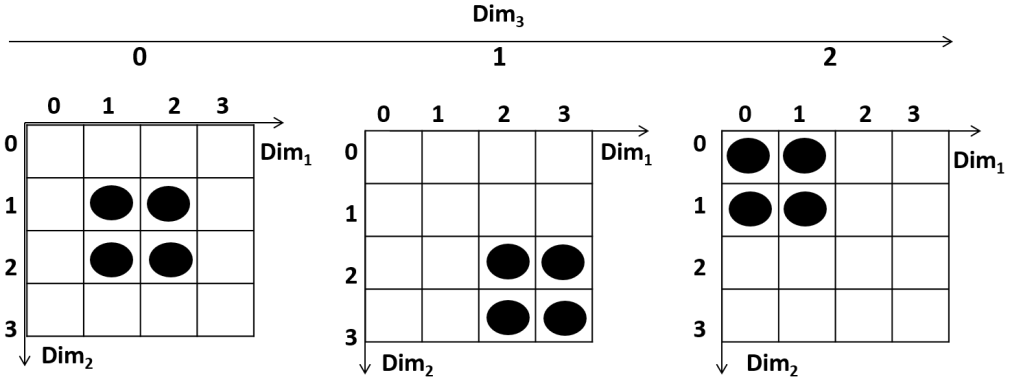
1.1 Overview

The work presented in this thesis is concerned with techniques for identifying "banded patterns" in N-Dimensional (ND) binary valued data. A binary valued data set comprises only ones and zeroes. For ease of understanding, in this thesis, the presence of a one is conceptualized as a dot (a sphere in 3D and a hypersphere in ND). The presence of a zero is conceptualized as the absence of a dot (sphere or hyper-sphere), thus "empty space". Binary valued data occurs frequently in many real world application domains, examples include bioinformatics (gene mapping and probe mapping) [7, 27, 94], information retrieval [12] and paleontology (sites and species occurrences) [10, 106]. A binary valued data set is said to feature a banding if the dimension indexes can be ordered in such a way that the dots are arranged about the leading diagonal. Figures 1.1 and 1.2 depict examples of 2D and 3D banding. The central concerns of this thesis are the mechanisms and processes whereby the dots that feature in a zero-one data set can be effectively and efficiently rearranged so as to reveal a banding, or as close a banding as possible.

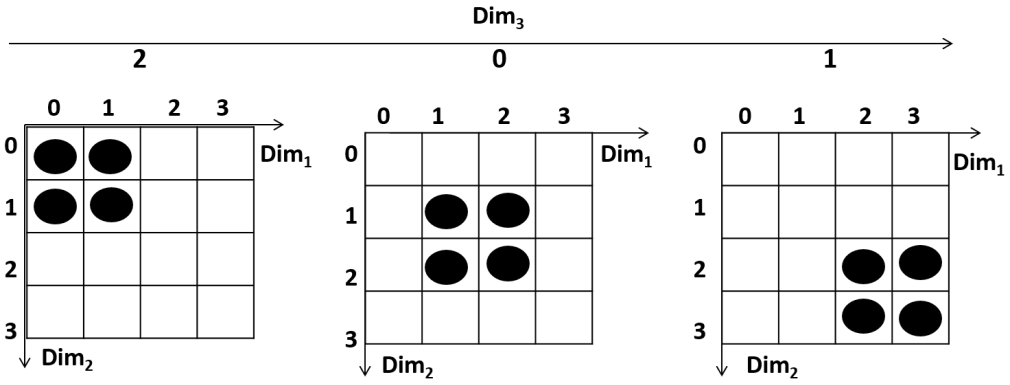
More specifically the work presented in this thesis is concerned with techniques for identifying "banded patterns" in ND binary valued data in the context of data mining. Data mining is primarily concerned with the extraction of hidden, but useful knowledge from data [45]. Data mining combines both statistics and computer science for the purpose of extracting the desired knowledge. As the number and size of electronically generated data sets keeps increasing, the corresponding significance of data mining methods also keeps increasing. Data mining encompasses a number of techniques which, in a very general way, can be categorised in terms of classification, clustering and pattern discovery. The work described in this thesis broadly spans all these techniques in that it is concerned with zero-one data which can be used in any of these contexts, although it can be argued that the discovery of banded patterns falls more within the domain of pattern discovery. Note that the techniques presented for rearranging a given zero-one data set do not change the content of the data, but simply reorders it.

(a) (b)

FIGURE 1.1: 2*D* Banding Example: (a) original matrix and (b) original matrix with the rows and columns reordered to reveal a banding



(a)



(b)

FIGURE 1.2: 3*D* Banding Example: (a) original 3D matrix and (b) original 3D matrix with Dim_1, Dim_2 and Dim_3 reordered to feature a banding

Existing work on identifying bandings in zero-one data [55, 56] has concentrated on the generation and testing of permutations, whilst [92] used barycentric values to identify bandings. The main issue with the identification of banded patterns in this manner is the large number of permutations to be considered (especially in ND), this makes the identification of banding in data a resource intensive enterprise which in turn has lead to existing work being limited to 2D data. Although, using a variety of measures (heuristics), the total number of permutations to be considered can be reduced, thereby

producing a very good approximation solution. In ND, the task of finding bandings is exponentially more challenging than in the case of 2D. To address this issue this thesis presents an alternative solution to the permutation generation and test approach that does not require the generation of permutations but instead operates using the concept of a *Banding Score* (BS). The proposed solution is to iteratively reorder the items in each dimension according to their individual BS until a "best" banding is arrived at, defined in terms of a Global Banding Score (GBS). The thesis also considers additional techniques for the identification of bandings in large data sets; namely sampling and segmentation techniques.

The fundamental idea of the banding score concept presented in this thesis is that the "bandedness" of a data set can be expressed in terms of a Global Banding Score (GBS), a number between 0 and 1. A GBS of "0" will be obtained (if the entire data space is filled with zeros (no dots)) while a GBS of "1" will be obtained (if the entire data space is filled with "ones" (dots)).

It is further proposed that the GBS is calculated by summing and normalizing individual BS associated with individual columns and rows in a given 2D matrix (data set). Individual BS will again be expressed as a number between 0 and 1. The idea is that given a set of column (row) BS, these can be used to reorder the columns (rows) to reveal a banding. Once the rows and columns have been reordered the individual BS values will need to be recalculated, as it is likely that they will have changed as a result of the reordering and a new GBS generated. The expectation is that the new GBS will be "better" than the initial GBS calculated prior to the reordering.

It was also anticipated that the reordering would have to be undertaken over a number of iterations until the GBS value "stabilised". However, the important point to note is that the time complexity of this approach is linear according to the number of columns/rows, as opposed to the non-linear time complexity associated with the permutation generation approaches.

The rest of this introductory chapter is organised as follows. In Section 1.2 the motivation for the research is discussed in further detail. The specific research question and associated research issues are presented in Section 1.3. Section 1.4 outlines the research methodology adopted to address the research question and associated issues, followed in Section 1.5 with a summary of the research contributions. Section 1.6 presents details of published work produced as a result of the described research, followed in Section 1.7 with an outline of the structure of the remainder of this thesis. Finally in Section 1.8 the chapter is concluded with a brief summary.

1.2 Research Motivation

From the foregoing, the main aim of the work presented in this thesis is to investigate and evaluate effective algorithms that will reveal banded patterns in zero-one data (if

they exist) by rearranging the ordering of items within the dimensions. The motivation for the rearranging of zero-one data, so as to reveal a banding, is desirable because:

1. It may be of interest in its own right in that it may enhance the interpretability of the data and or provide a better understanding of the processes whereby the data was generated.

2. It allows for the compression of the data, which may consequently enhance the operation of data mining (and other) algorithms that work with zero-one data.

3. It also allows for the visualisation of the data, which may enhance the visibility of useful information hidden in the data.

Natural interpretations of banded structures include overlapping communities in social networks [106], patterns of species occurring in spatially correlated locations [86] and overlapping roles of genes with respect to various diseases [55].

As noted above, research work on banded data analysis to date has tended mostly to be permutation based and focused on 2D data sets [55, 56, 92] rather than ND data sets. To the best knowledge of the author there is no work on the identification of bandings in ND data. The technical motivation for the research described in this thesis can thus be broadly identified as the desire to develop alternative banding mechanisms that do not consider large numbers of permutations and operate in ND.

To act as a focus for the work a specific application is considered; the Cattle movement Tracing System[1] (CTS) in operation in Great Britain (GB) from which a 5D (records, attributes, eastings, northings and time) binary valued data sets can be extracted. The application motivation was thus a desire to analyse the CTS data so as to provide some insight into cattle movement in GB to support policy makers and other interested parties who may wish to monitor the spread of cattle diseases. Further details of the CTS data base are presented later in this thesis.

1.3 Research Questions and Issues

Given the research motivation presented in Section 1.2 above, the key objective of the work presented in this thesis was to research and investigate effective and efficient mechanisms to identify banded patterns in ND data. This objective can be formulated as a research question as follows:

What are the most appropriate mechanisms and techniques required to identify banded patterns in ND zero-one data spaces in a manner that is both effective and efficient?

The provision of an answer to this research question encompass the resolution of a number of subsidiary questions as follows:

[1] https://www.gov.uk/guidance/animal-identification-movement-and-tracing-regulations.

1. **Mechanisms and Techniques**: What mechanisms and techniques can best be employed to identify a best banding? What are the most suitable techniques for obtaining a best banding?

2. **"Best" Banding**: What is a "best" banding? How is a best banding determined? How is the goodness of a banding measured?

3. **ND Banded Data**: What are the mechanisms that can best be employed to ensure that any proposed banding algorithm will scale up to operate in ND?

4. **Multiple "Dots"**: How is the issue of more than one "1" value (dot) being located at a location (a cell in the matrix of interest holding dots to be reordered so as to achieve a best banding) in a ND data matrix best addressed?

5. **Statistically Significant**: What is the most appropriate mechanism for determining whether a best banding, when identified, is statistically significant or not?

1.4 Research Methodology

To provide an answer to the research question and associated research issues, as described in the previous section, the adopted research methodology was to investigate and evaluate a series of techniques directed at identifying banded patterns in zero-one data starting with 2D data and then increasing the number of dimensions to be considered to 3D and then ND. The initial assumption was that only one dot could be held at each location, this assumption was removed once suitable algorithms had been established. In the context of scalability it was recognised that eventually all the proposed algorithms would no longer be able to operate on a single machine, thus the research methodology included the idea of investigating sampling and segmentation processes compatible with the banded pattern mining concept.

To act as a focus, as noted above, data sets extracted from the Great Britain (GB) Cattle Tracing System (CTS) database were used. This database was selected because: (i) large multi-dimensional dot data sets could be extracted from it and (ii) the analysis of the data would provide an example of the kind of application where ND banding might be usefully employed. Evaluation was conducted predominantly using data extracted from the CTS database. However, in the 2D context evaluation was also conducted using synthetic data sets and a number of benchmark data sets taken from the University of California Irvine (UCI) machine learning repository [18]. Note that the UCI data sets, by default, are all 2D. Although in some cases it might have been possible to contrive higher numbers of dimensions this was not done, and hence the UCI data sets were only used in the 2D context. Where possible, comparisons were made with alternative existing algorithms, although this was again only possible in the 2D context. An independent metric, Average Band Width (ABW), was used for the comparison with existing work because each algorithm, including those presented in this thesis, used different criteria to identify a best banding.

The following eight phase programme of work was adopted:

1. **Representation:** Investigation of mechanisms for conducting the necessary pre-processing with respect to the targeted data sets.

2. **2D Banded Pattern Mining:** Investigation into mechanisms to identify a best banding in 2D data. The intention here was to develop a "benchmark" banded pattern mining algorithm that could be analysed, evaluated and used as the foundation for work conducted in the later stages of the programme of work.

3. **3D Banded Pattern Mining:** Extension of the work on 2D banded pattern mining from Phase 2 above to address banding in 3D. The idea was to consider two alternative approaches: (i) approximate and (ii) exact. The intuition here was that as the number of dimensions under consideration increased the time complexity was also expected to increase; it was conjectured that the use of approximate algorithms might mitigate against this increasing complexity while at the same time producing acceptable bandings.

4. **ND Banded Pattern Mining:** Extension of the work on 3D banded pattern mining from Phase 3 above to address banding in ND, concentrating on example data sets taken from the CTS application.

5. **Multiple Dots:** For phases 2 to 4 the assumption was that locations could only hold one dot each; this is true in the case of data sets where one of the dimensions is record number, but this would not necessarily be the case if a subset of the dimensions within a given data set were considered. Phase 5 was therefore concerned with adapting the algorithms from earlier phases so that the "multiple dots" scenario could be addressed.

6. **Sampling Techniques:** As noted above it was recognized from the start that there would always be data sets whose size was such that they could not be processed in their entirety. Two potential mechanisms for addressing this issue were: (i) sampling and (ii) segmentation. Phase 6 was therefore concerned with sampling, the idea of identifying a banding in a subset of the data set of interest and then applying this to the entire data set. Note that sampling features the possibility of multiple dots at locations, hence Phase 5 was required to preceed Phase 6.

7. **Segmentation Techniques:** Investigation of segmentation technique to address the issue of finding bandings in very large ND data sets which cannot be held in primary storage. The idea here was to conduct bandings sequentially using sequences of data segments taken from a single large ND data set. Note that segmentation also features the possibility of multiple dots at locations.

8. **Statistical Significance:** The final phase of the programme of work was to consider mechanisms whereby the statistical significance of discovered bandings could be ascertained.

With respect to the above a number of Banded Pattern Mining (BPM) algorithms were identified. These can be arranged in a hierarchy as shown in Figure 1.3. In the figure, the leaf nodes indicate individual BPM algorithms while the root and intermediate nodes indicate categories or groupings of BPM algorithm. From the figure the proposed BPM algorithms are grouped as follows: (i) 2D-BPM algorithm, (ii) 3D-BPM algorithms and (iii) ND-BPM algorithms. The 2D-BPM algorithm identifies bandings in two dimensional data, the 3D-BPM algorithms identify bandings in three dimensional data and the ND-BPM algorithm identify bandings in N-dimensional data. The 3D-BPM algorithm category is further sub-divided into: (i) Approximate 3D (A3D) BPM and (ii) Exact 3D (E3D) BPM. The ND-BPM algorithm category in turn is also further divided into: (i) Approximate ND (AND) BPM, (ii) Exact ND (END) BPM and (iii) Multiple dots (MD) BPM. The latter also comprises: (i) Approximate BPM (ABPM) and (ii) Exact BPM (EBPM) variations. Note that for each EBPM algorithm category there is a Euclidean and a Manhattan variation (the significance will become apparent later in this thesis). Although not included in the figure the proposed sampling and segmentation techniques utilise the multiple dots BPM algorithms. The Approximate 3D and ND BPM algorithms, as the name suggests, find approximate bandings (by considering dimension pairings), while the Exact 3D and ND BPM algorithms find exact bandings (by considering the entire data space).

As noted above, in the context of 2D-BPM, comparisons were undertaken with respect to existing work on banded pattern mining [55, 56, 92]. Each of these proposed mechanisms seeks to "optimise" a particular banding parameter. The BPM algorithms presented in this thesis seek to minimise a Global Banding Score (GBS), essentially a composite overall banding score for a given banding. So that a fair evaluation could be undertaken comparison was undertaken in the context of an algorithm independent measure, Average Band Width (ABW). With respect to the evaluation of the proposed banding algorithms in the context of 3D and higher, evaluation was conducted in terms of GBS and run time.

1.5 Research Contribution

The main contributions of the research work considered in this thesis are summarized below. Note that for each item in the summary the chapter or chapters where the contribution is discussed is included in parenthesis. Note also that with respect to the proposed algorithms the reader might find it useful to refer back to Figure 1.3:

1. The concept of a **banding score** that supports the identification of bandings in zero-one data without considering large numbers of permutations (Chapters 4, 5, 6, 7, 8 and 9). This is arguably the most significant contribution of the work.

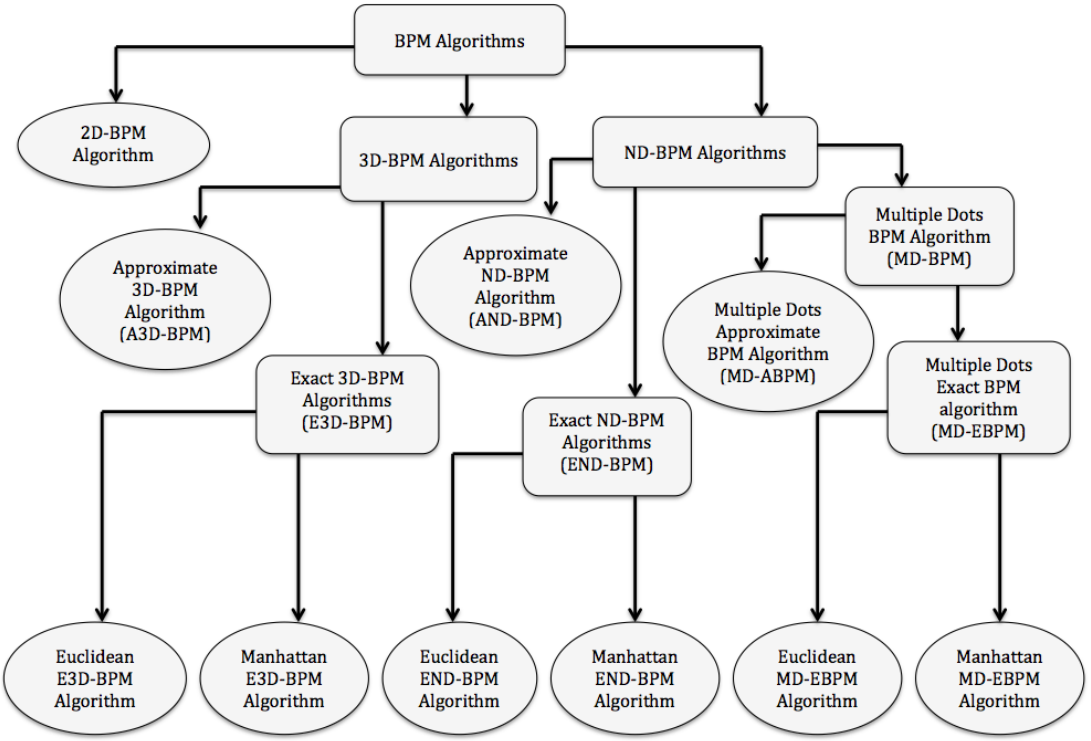


FIGURE 1.3: Hierarchical categorization of the BPM algorithms proposed and evaluated in the context of the research presented in this thesis.

2. The 2D-BPM algorithm for discovering bandings in 2D data sets (Chapter 4).

3. The Approximate 3D (A3D) and Exact 3D (E3D) BPM algorithms, including the Euclidean and Manhattan variations of the E3D-BPM algorithm (Chapters 5 and 6).

4. The Approximate ND (AND) and Exact ND (END) BPM algorithms (Chapter 7).

5. A mechanism for addressing the situation where a location holds multiple dots (Chapter 8) in the context of both approximate and exact BPM (the MD-ABPM and MD-EBPM algorithms).

6. A mechanism for applying bandings to very large data sets using a sampling technique integrated into the banded pattern mining process (Chapter 9).

7. A mechanism for applying bandings to very large data sets using a segmentation technique integrated into the banded pattern mining process (Chapter 9).

8. An independent mechanism, the Average Band Width (ABW) mechanism, for measuring the quality of a banding to support comparison of BPM algorithms (Chapter 4).

9. A mechanism for considering the statistical significance of an identified banding (Chapter 10).

10. Some insights into the CTS database (Chapter 3).

1.6 Publications

Some of the work presented in this thesis has been the subject of a number of refereed publications. These are itemised below. In each case a short description of the paper is included highlighting its significance in the context of the work presented. Where appropriate reference to the chapter where the material appears is also given.

Journal Paper

(a) **Abdullahi, F. B and Coenen, F and Martin, R (2016). Banded Pattern Mining Algorithms in Multi-Dimensional Zero-One Data. "Transactions on Large Scale Data and Knowledge Centered Systems" (TLDKS XXVI), Springer-Verlag Berlin, Heidelberg, pp. 1-31 volume 26 (2016), Special Edition**. Journal article comprising an extended, updated and revised version of conference paper (b) (see below). In this work the approximate BPM and exact BPM (both the Euclidean and Manhattan variations) algorithms were presented. The presented evaluation was conducted using data extracted from the CTS database (as in the case of work described in this thesis).

(b) **Abdullahi, F. B and Coenen, F and Martin, R (2016). Scalable Banded Pattern Mining Algorithms for Big Data. Submitted for refereeing to the IEEE TKDE Journal**. This journal paper summarised the BPM algorithms presented in this thesis, the 2D-BPM and ND-BPM algorithms. Two techniques for processing large data sets were considered, sampling and segmentation. Both were able to identify banding in large data sets, although the segmentation approach was found to to produce better quality bandings. The statistical significance of the bandings produced was also considered and a mechanism founded on the use of Gaussian distribution curves was presented to determine whether the bandings generated using the BPM algorithms were statistical significant or not. Experiments reported in the paper clearly indicated that this was a useful mechanism for determining whether a banding is statistically significant or not. Note that similar experimental results, to those described in the paper, are presented in Chapters 9 and 10.

Conference Papers

(a) **Abdullahi, F. B and Coenen, F and Martin, R (2014). A Novel Approach for Identifying Banded Pattern Mining In Zero-One Data Using Column And Row Banding Score. 10th International Conference on Machine Learning and Data Mining (MLDM'14), Springer-Verlag**

Berlin, Heidelberg, pp. 58-72. St. Petersburg, Russia. 21th-24th July, 2014. Conference paper reporting on some initial work on Banded Pattern Mining (BPM). The paper was the first to propose the banding score mechanism that allowed columns and rows to be rearranged without considering permutations. This mechanism was incorporated into the proposed BPM algorithm to identify banding in 2D data sets. The work was illustrated using a number of UCI data sets. The content of this paper was used as the foundation for the work presented in Chapter 4.

(b) **Abdullahi, F. B and Coenen, F and Martin, R (2014). A Scalable Algorithm for Banded Pattern Mining in Multi-Dimensional Zero-One Data. 16th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'14). Springer-Verlag Berlin Heidelberg, pp. 345-356, Munich, Germany. 1st-5th September, 2014**. In this conference paper, the proposed approximate BPM algorithm for application to 3D zero-one data was presented. The disadvantage of this algorithm (as anticipated) was that it did not necessarily find a best banding but only an approximation best banding because the algorithm did not consider the entire data space when calculating the banding scores (it only consider dimension pairings). The work was illustrated using both UCI Data sets and the CTS database. The work described in this paper provided the foundation for the material presented in Chapter 5.

(c) **Abdullahi, F. B and Coenen, F and Martin, R (2015). Finding Banded Patterns in Data: The Banded Pattern Mining Algorithm. 17th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK'15). Springer-Verlag Berlin, Heidelberg, pp. 95-107, Valencia, Spain. 1st-4th September, 2015**. Conference paper describing the work presented with respect to the Exact BPM algorithm. Two alternative variations were proposed, Euclidean and Manhattan, whereby banding scores could be calculated in the context of 3D data. The reported evaluation was again conducted using the CTS database used previously. The content of this paper features extensively with respect to the work presented in Chapter 6.

(d) **Abdullahi, F. B and Coenen, F and Martin, R (2015). Finding Banded Patterns In Big Data Using Sampling. IEEE International Conference on Big Data (IEEE BigData), pp. 2233-2242. Santa Clara, CA. 29th October - 1st November, 2015)**. This workshop paper was the first to report on technique for identifying bandings in large ND zero-one data sets using a sampling technique. The paper presented a study of the application of the proposed Exact BPM algorithm for large data sets, data sets that were too large to be held in primary storage. The idea presented in the paper was to use a sampling technique whereby the input data was divided into subgroups and records selected from each

subgroup. The study again focused on CTS database. The content of this paper was used in context of the work presented in Chapters 8 and 9.

1.7 Structure of Thesis

The rest of this thesis is organised as follows.

Chapter 2, Literature Review and Previous Work: Presents a literature review of related research and some background material to the work on BPM presented in this thesis. Of note are some comparitor algorithms, namely: (i) the Barycenter (BC) algorithm and (ii) the Minimum Banded Augmentation (MBA) algorithm and its two variations (Fixed Permutation (FP) and Bi-directional Fixed Permutation (BFP)).

Chapter 3, Evaluation Framework: Presents a brief description of the selected data sets used for evaluation purposes. As already noted three categories of data sets were used: (i) randomly generated synthetic data, (ii) benchmark data sets taken from the University of California Irvine (UCI) machine learning repository and (iii) The GB cattle movement CTS database. The latter was used as the main focus for the work described; the former two were used only in the 2D context.

Chapter 4, 2D Banding Mechanism: Introduces the proposed 2D-BPM algorithm which is the foundation for much of the research work presented in this thesis. This is where the concept of a "banding score" is proposed, arguably the main contribution of the work. The chapter includes a worked example of the algorithm. The evaluation presented is with respect to: (i) synthetic data and (ii) UCI data sets. The effectiveness of banding with respect to Frequent Item-set Mining (FIM) is also considered.

Chapter 5, Approximate Banding Mechanism: Introduces the Approximate 3D BPM algorithm (A3D-BPM), the first of the 3D banding mechanism considered in this thesis. The A3D-BPM algorithm operates using dimension pairings rather than the entire data space to calculate approximate (as opposed to exact) banding scores. The chapter includes a worked example of the algorithm.

Chapter 6, Exact Banding Mechanism: Describes the Exact 3D BPM (E3D-BPM) algorithm. The chapter considers a number of alternative ways of calculating exact banding scores, namely: (i) Euclidean and (ii) Manhattan. Again the chapter includes a worked example. The chapter also considers the possibility of pre-calculating parts of the banding score in the interest of reducing the time complexity of the algorithm, an idea referred to as the "M-Table" concept. A particular challenge associated with calculating exact banding scores in 3D (and above) is determining what the maximum distance values are (required for normalization purposes). The chapter thus also presents the Maximum Distance Calculation

(MDC) algorithm for achieving this. The outcomes are reported from a series of experiments undertaken to demonstrate the efficiency and effectiveness of the algorithm in the context of: (i) 3D data sets with and without M-Tables and (ii) the Approximate 3D-BPM algorithm presented in the previous chapter.

Chapter 7, ND Banded Pattern Mining Mechanisms: Presents the Approximate ND (AND) and Exact ND (END) BPM algorithms. These are not significantly different from the 3D BPM algorithms presented in the previous chapter although designed for ND. Of particular note is the operation of the BPM algorithms and the MDC algorithm in the context of ND. The chapter includes a complete comparison of the algorithms in the context of ND data sets extracted from the CTS database.

Chapter 8, Multiple Dots Mechanism: Chapter considers the possibility of locations in the data space holding multiple dots which in turn requires adjustment to the END-BPM and AND-BPM algorithms presented in the previous chapters. Note that the assumption with respect to the foregoing algorithms was that individual locations would hold only one dot.

Chapter 9, Discovering Bandings in Big Data Using Sampling and Segmentation: Presents the two proposed techniques, sampling and segmentation, for identifying bandings in very large data sets (too large to be held in primary storage). The chapter reports on a series of experiments undertaken to illustrate the scalability of the proposed sampling and segmentation techniques in the context of: (i) 3D, (ii) 4D and (iii) 5D data sets extracted from the CTS database.

Chapter 10, Statistical Significance Testing Using Gaussian Distributions: Reports on some ideas considered to determine the significance of identified bandings by considering the generated bandings with respect to the random bandings that can be expected given a Gaussian distribution.

Chapter 11, Conclusion and Future Research: Concludes the thesis with a summary of the work presented, the main findings in terms of the identified research question and subsidiary questions, and some discussion on possible future research directions.

1.8 Summary

In summary, this chapter has provided an overview, and some background, for the research presented in the remainder of this thesis, including details concerning the motivations for the work and the research question and subsidiary questions. It has also provided a brief description of the research methodology and the contributions of the research. In the following chapter, a literature review, intended to provide more detail regarding the background concerning the research described in the thesis, is presented.

Chapter 2

Literature Review and Previous Work

2.1 Introduction

As noted in Chapter 1, the research described in this thesis seeks to establish an effective banding mechanisms that serves to identify banded patterns in N-dimensional zero-one data. This chapter presents a review of the previous work related to the research presented in this thesis. The organisation of the chapter is as follows. Section 2.2 presents a general overview of Bandad Pattern Mining (BPM) in terms of its advantages and disadvantages. A comprehensive review of the domain of BPM is then given in Section 2.3, including the current "state of the art" algorithms. Note that the significance of the latter is that these algorithms were used with respect to the evaluation reported on later in this thesis. A brief overview of the Knowledge Discovery in Databases (KDD) process, and data mining in particular, in the context of banded pattern mining is presented in Section 2.4. Section 2.5 then presents an overview of sampling and segmentation techniques; the reason for their inclusion in this chapter is that work presented later in this thesis, directed at providing mechanisms for applying bandings to very large ND data sets, is founded on ideas concerning sampling and segmentation. To measure the effectiveness of bandings there are a number of metrics that can be used. These are presented in Section 2.6. Finally the chapter is concluded with a summary in Section 2.7.

2.2 Overview of Banded Pattern Mining

As noted in the introduction to this thesis the work described is directed at identifying "bandings" in binary valued data (matrices). An illustration of a fully banded 2D data (matrix) is given in Figure 2.1. Note that given a reasonable complex data set, a perfect banding can typically not be achieved, but some "best" banding is always possible. This section provides an overview of the background to the banded pattern mining concept explored in this thesis.

The concept of banded data has its origins in numerical analysis [74] where it has been used, for example in the context of the resolution of linear equations using the Gaussian elimination method. The concept of bandedness has also been considered in the context of: (i) reorderable matrices to facilitate the graphical analysis (visualisation) of 2D data [17, 16], (ii) the discovery of Reorderable Patterns in 2D data [56, 81], patterns of all kinds that can be revealed by rearranging the data columns and rows,(iii) bandwidth minimisation for the purpose of gaining algorithmic efficiency benefits [25, 26] and (iv) matrix seriation of data to maximise the human visual perception of patterns [81, 10, 111]. In the context of banded pattern mining, the subject of this thesis, the idea was first proposed by Gemma et al. [56]; although the focus here was on minimising the distance of non-zero entries from the main diagonal of a 2D data matrix by considering permutation of the original matrix (ND data matrices were not considered).

The remainder of this section is organised as follows. In Subsection 2.2.1 some general advantages of banding are first considered; this is followed in Subsection 2.2.2 with some discussion of the application of banding. Subsection 2.2.3 then presents an overview of the concept of banding in the context of numerical analysis. Subsection 2.2.4 considers banding in the context of reorderable matrices to support graphical data analysis, while Subsection 2.2.5 considers banding in the context of reorderable patterns. Subsection 2.2.6 then goes on to present the bandwidth minimisation approach to the banding problem. Subsection2.2.7 considers banding in the context of matrix seriation. Previous reported work on banded pattern mining is then considered in Subsection 2.2.8. The section is concluded with a summary in Subsection 2.2.9. Note that what distinguishes the above from the work presented in this thesis is firstly that the above methods typically use the concept of permutations, in some form or another, to identify banded patterns; and secondly that they operate only with respect to 2D data. (because of the resources required with respect to permutation generation). Contrary to the above banding methods, the BPM algorithms proposed later in this thesis use the concept of a banding score to identify bandings, this is less resource intensive and consequently can operate with respect to ND data.

2.2.1 Advantages of Banding

Broadly, banding offers advantages in the context of data interpretation [16] and processing efficiency [19, 41, 8, 89]. More specifically the advantages offered may be summarized as follows.

1. **Data Analysis**: Banding may be indicative of some interesting phenomena which is otherwise hidden in the data and tells us something of significance about the data.

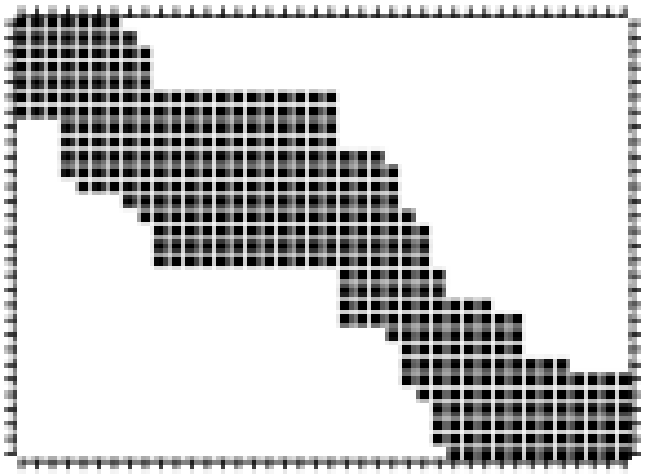


FIGURE 2.1: Example of a fully banded 2D matrix [56]

2. **Algorithm Efficiency**: Working with banded data is seen as preferable from a computational point of view; the computational cost involved in performing certain operations, for example multiplication, falls significantly for banded matrices leading to significant savings in terms of processing time [35].

3. **Data Storage**: Related to 2, when a matrix is banded, only the non-zero entries along or near the diagonal need to be considered. Thus, when using banded storage schemes the amount of memory required to store the data is directly proportional to the bandwidth (the distance of dots from the main diagonal of a matrix). Therefore finding a banding that minimizes the bandwidth is important for both reducing storage space and algorithmic speed-up [93].

4. **Data Ordering**: Banding can enhance the operation of some algorithms that work with zero one data because it imposes an ordering on the data. One example where this advantage can be realised is in the case of frequent itemset mining [1].

5. **Data Visualisation**. The reordering of zero one data provides a useful data visualisation (especially in the case of 2D and 3D data, visualisation becomes more challenging in ND) [16, 17].

2.2.2 Banding Application Areas

Banding has a wide range of applicability in the context of data interpretation of all kinds. This subsection reviews a number application domains where banding has been utilised. The aim, following on from the foregoing subsection, is to provide the reader with a deeper understanding of the benefits of banding. Five applications are considered: (i) Network Data Analytics, (ii) Character Analysis in Literature, (iii) Co-occurence Analysis in Data, (iv) Linguistic Study and (v) Very Large Scale Integration (VLSI). Each is discussed in further detail below.

1. **Network Data Analytics**: Network data analytics is concerned with the representation of the relationship between data entities. For example in a social network the vertices might represent individuals and the edges social interactions between those individuals. Groups of vertices might then represents communities [13]. The banding concept can be used to identify such communities. To do this, a network of interest needs to be represented in the form of an *adjacency matrix* in which the "dots" represent vertices that are connected by an edge. Where we have directed edges the "from nodes" will be listed on one axis and the "to nodes" on the other, in the case of non-directed edges cells are filled by considering edges to be bi-directional. By rearranging the rows and columns of the adjacency matrix a banding can be obtained that features groupings of nodes which in turn will be indicative of communities. A specific example can be found in [55] where an American football network data set from the year 2000 [95] was considered. The vertices in the network were teams while the edges indicated who played who (edges are undirected). The network was thus represented as a 2D, 115×115, adjacency matrix where the rows and columns both represented teams. By identifying bandings in the data it was possible to identify groupings of teams that played against each other in the year 2000.

2. **Character Analysis in Literature**: Character analysis in literature is concerned with identifying characters that frequently appear together. Again an adjacency matrix is constructed with characters listed on the X and Y axes. By banding the matrix groups of characters can be identified that appear together. A specific example can be found in [55] with respect to the work of the French author Victor Hugo. The character adjacency matrix in this case measured 77×77. By applying a banding algorithm to the rows and columns, clusters of characters that co-occur were identified.

3. **Co-occurrence Analysis in Data**: Co-occurrence analysis is concerned with the identification of two different categories of entity that co-exist, for example animal species and geographic locations. The data table (matrix) in this case has one set of entities along the X-axis and another set of entities along the Y-axis. Cells that are filled with a dot indcate a co-existence (co-occurrence) of the referenced entities. A specific example can be found in Juntilla et al. [106], who considered a paleontological application. In this case the rows in the 2D binary valued matrix represented Neolithic sites and the columns fossil genera species. By banding the data Juntilla et al. were able to demonstrate a correlation between certain fossil species and particular Neolithic sites.

4. **Linguistics study**: The objective of banding in the context of linguistic study is to conduct comparisons of specific words used in different geographic locations to get a better understanding of their usage. In this case the 2D matrix comprises words along one axis and geographic locations on the other. Banding then indicates

locations where the same word is used. An example can be found in [55] where 1334 phonological features were considered with respect to a 506 municipalities in Finland. By banding the data a visualisation of the spatial distribution of dialect across different municipalities could be identified.

5. **Very Large Scale Integration (VLSI):** VLSI is concerned with the process of creating an Integrated Circuit (IC) by combining thousands of transistors into a single chip. Koebe and Knochel [76] refers to the application of the banding algorithm in this case as the "block alignment problem", where the problem of designing a VLSI chip layout by finding channels between the circuit component blocks was to be established. However, the assumption here was that the terminal positions are fixed at one end of the block and at the other end the terminals are divided into rearranged cells that minimises the number of crossing terminals.

2.2.3 Numerical Analysis

As noted in the introduction to this section the concept of bandedness has it origin in numerical analysis [8, 60, 113]. Broadly, numerical analysis is concerned with the resolution of all kinds of problems involving "continuous mathematics". This is exemplified by problems involving the numerical solution of systems of m linear equations with n unknowns (Figure 2.2). In other words systems of equations of the form $Ax = b$ where A is an $m \times n$ matrix $[a_{ij}]$ of variable coefficients ($1 \leq i \leq m$ and $1 \leq j \leq n$), b is a "column vector" with m entries (Figure 2.3) and x is "column vector" with n entries (Figure 2.3). The standard numerical analysis method used for resolving such systems of equations is the "Gaussian Elimination" method [19, 48, 87]. Using this method a given system of equations must first be converted into an *augmented matrix* of the system (the augmented matrix is obtained in this case by appending the column vector b to the matrix $[a_{ij}]$). Next the augmented matrix is converted into *echelon matrix* form using row operations. Finally a backward substitution method is used to arrive at the final solution. A matrix is in echelon form if:

1. Rows consisting of entirely zero entries are grouped at the bottom of the matrix.
2. The first non-zero entry of each row is "1" called the leading "1".
3. All the entries below the leading 1 are zeros.

Note that an echelon matrix becomes a *reduced echelon matrix* if the elements above the leading 1 are also all zeros (in a standard echelon matrix only elements below the leading 1 are all zeros).

An example of Gaussian Elimination is presented below, using the system of linear equations shown in Equations 2.1, 2.2 and 2.3. To solve this system using Gaussian Elimination, the system is first converted into an augmented matrix form as shown in Figure 2.4(a). Next the augmented matrix is converted into an echelon matrix by

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ . \\ . \\ . \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{bmatrix}$$

FIGURE 2.2: System of linear equations ([74])

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ . & . & . & . \\ . & . & . & . \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_m \end{pmatrix}$$

FIGURE 2.3: Example of a system represented as matrix equation ([74])

creating zeros below pivot positions in the augmented matrix as shown in Figures 2.4(b), 2.4(c) and 2.4(d). Pivot positions in this case are the elements along the leading diagonal. The echelon matrix is obtained using the row operations: (i) $R2+R1$, (ii) $R3+(-2)R1$, (iii) $R3+(-2)R2$ and (iv) $(1/2)R3$. Next "backward substitution", a method of working the equation backwards is applied to the echelon matrix to arrived at the final reduced echelon matrix, this corresponds to the row operations: (i) $R1+(-2)R3$, (ii) $R2+(-3)R3$ and (iii) $R1+(-3)R2$ to give the reduced echelon matrix form presented in Figures 2.4(e) and 2.4(f). Note that the reduced echelon matrix form of the augmented matrix demonstrates a banded matrix with the non-zero entries about the leading diagonal. Contrary to the Banded Pattern Mining (BPM) algorithm presented in this thesis, the Gaussian Elimination method does not operate by reorganising the rows and columns, but performs arithmetic operations on the rows and columns to arrive at the banded matrix [54]. The Gaussian Elimination method works well in the context of solving systems of equations, but does not easily scale and is unsuited to higher dimensional data matrices.

$$x_1 + 2x_2 + 3x_3 + 2x_4 = -1 \tag{2.1}$$

$$-x_1 - 2x_2 + 2x_3 + x_4 = 2 \tag{2.2}$$

$$2x_2 - 4x_2 + 8x_3 + 12x_4 = 4 \tag{2.3}$$

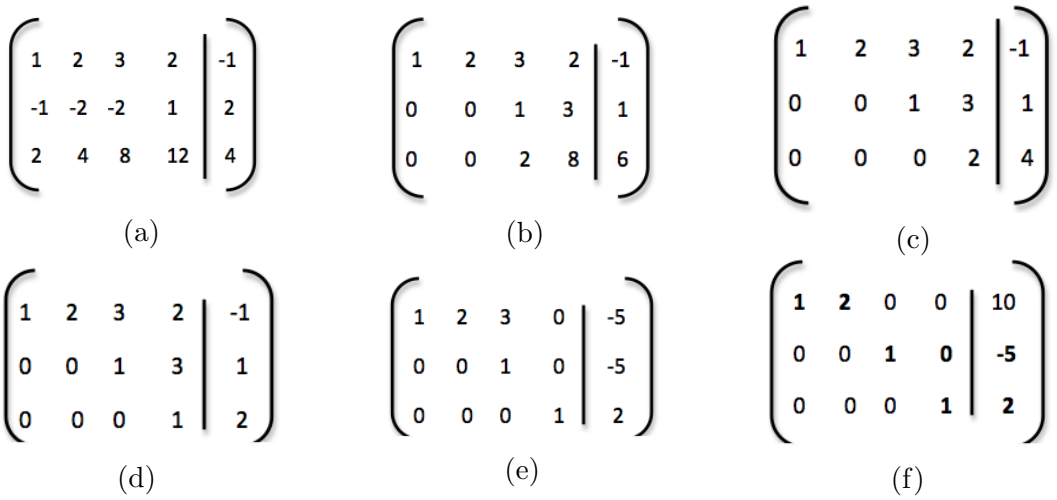


FIGURE 2.4: Augmented matrix Figure 2.4(a) and process for deriving a reduced echelon matrix form

2.2.4 Reorderable Matrices

This section presents the idea of reorderable matrices that support graphical data analysis. A reorderable matrix is a visualisation of 2D tabular data that supports "movement" (swapping) of rows and columns so as to attain a "better view" of the data. The idea is that this "better" view of the data can be obtained after the rows and columns have been rearranged [55]. Note that this does not necessarily mean banding; but if what is meant by a "better view" is banding, then there is clearly a relationship with the concept of banding as presented in this thesis and reorderable matrices. The distinction is that we are interested in automatically finding "patterns" in data that might reveal interesting information, while reorderable matrices are concerned with facilitating experimentation through visual means which might or might not feature banding (although the reordering as envisaged in this thesis will also facilitate visualisation). Note also that the desired visualisation produced using the idea of reorderable matrices is also sometimes augmented using additional mechanisms. For example in [15, 17, 16] symbols such as rectangles or circles were used to represent the data, the symbols had a relative size that reflected the actual data values.

Bertin [16], writing in 1999, used the terms "construction" and "reconstruction" to described respectively, the process of generating a reorderable matrix from tabular data and the process of moving rows and columns. The same terminology will be used in this section.

The history of reorderable matrices in data analysis dates back to the 19th century when Petrie, an English Egyptologist, applied a reordering technique to study archaeological data [81, 91]. Since then a number of reordering methods have been used with respect to a variety of applications. For example Forsyth and Katz [110], in 1951, were the first to introduce the idea of rearranging the rows and columns of "sociomatrices", tabular representations of data collected as part of some sociometric study, so as to

obtain a better presentation of the results of sociometric tests. Brainerd and Robinson in [110], proposed a form of matrix; where the highest values in the matrix were located along the prime diagonal and decreased monotonically when moving away from this diagonal. This matrix become known as the "Robinson Matrix" or (R-Matrix).

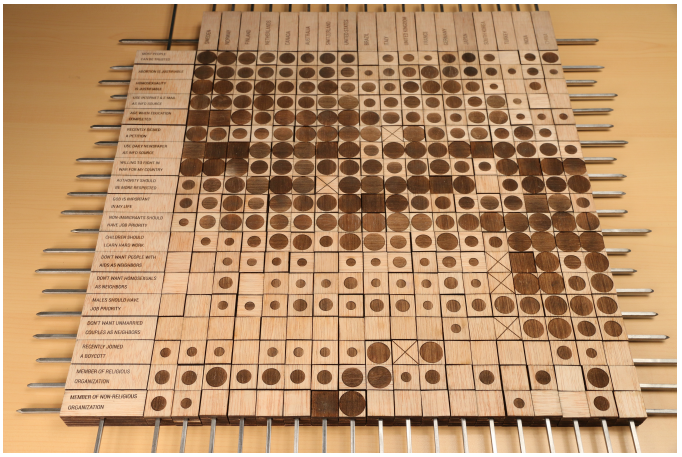


FIGURE 2.5: Bertin device for matrix construction and reconstruction [116]

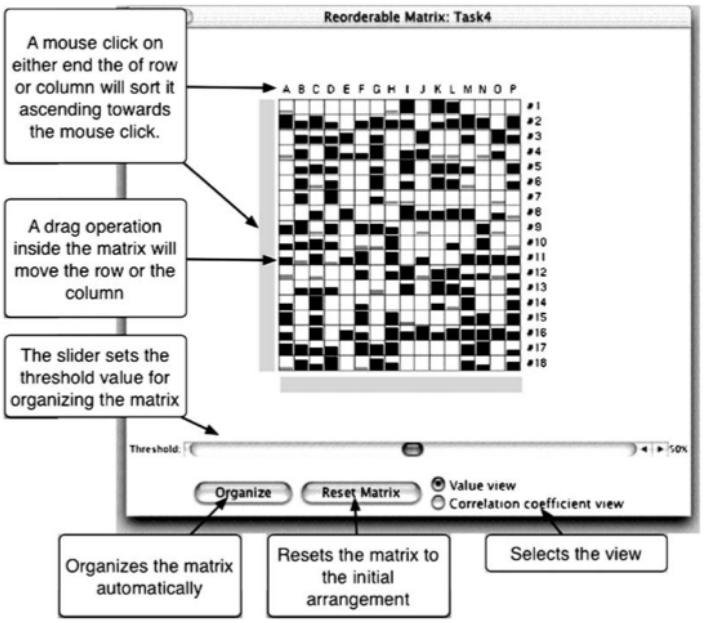


FIGURE 2.6: The reorderable matrix user interface [116]

Bertin [16] identified three reorderable matrix construction methods: (i) manual (without computer usage); (ii) interactive, whereby users manually order matrices within some software environment; and (iii) fully automatic. The latter is thus akin to the banded pattern mining idea presented in this thesis. Each is discussed in some further detail below.

(i) **Manual Method:** The manual approach to reordering matrices was initially conducted as a "paper and pencil" exercise; the matrix in question was redrawn after

each reordering (permutation). The process was then documented by taking a photograph of each reordering step. However, in the mid 1960, Bertins [16] developed a device (Figure 2.5) that rendered the redrawing method unnecessary.

(ii) **Interactive Method:** Interactive methods use computer software to display a given matrix and allow the user to move rows and/or columns. Figure 2.6 shows an example user interface taken from [16]. In [116], the various operations and processes that are typically supported within the context of interactive construction are categorised as follows:

- **Moving operations:** The process of moving rows and columns to new positions in order to potentially identify interesting patterns or to compare adjacent pairs of rows or columns.
- **Threading operations:** The process of "sorting" an entire matrix by moving rows or columns in an attempt to reveal some characteristic of the data.
- **Blocking operations:** The process of "locking down" an area where an interesting pattern is detected in order to avoid accidental change. Note that the entire locked down area can be moved, but only in its entirety. The locked down area can thus be considered to represent a meta-row and column combination.
- **Arranging operations:** The process whereby some predefined arrangement is implemented automatically using software. For example some threading operation (see second bullet point above). Note that the availability of arranging operations tends towards the fully automated mode of reorderable matrix construction (see below).

(iii) **Automatic Method:** The automatic approach to constructing reorderable matrices uses software to entirely automate the desired reordering process. Note that the reordering of binary matrices, as in the case of the bandwidth minimisation problem (see below), is known to be NP-Complete [102]. Makinen and Siirtola in [116] were amongst the first to propose an algorithmic solution to the reorderable matrix problem where the aim was to produce a banding. Since then a number of algorithms have been proposed in order to achieve banding in the context of reorderable matrices. Of note are the 2D Sort and the Barycentric (BC) Algorithms:

- **2D Sort:** The 2D Sort banding algorithm was concerned with sorting a two-dimensional matrix iteratively so as to reveal a banding whereby the non-zero entries, called "black areas" [116] are arranged along the leading diagonal. Although, it should be noted that 2D Sort only performed well with respect to relatively small subset of matrices. The Sort algorithm operated by calculating the weighted sum of rows and columns, and comprised the following five steps:

1. Calculate the weighted sums of rows where the weights are the column positions of each cell.
2. Arrange rows in the matrix in ascending order of the row sum.
3. Calculate the weighted sum of columns where the weights are the row positions of each cell.
4. Arrange columns in the matrix in ascending order of the column sum.
5. Repeat steps 1-to-4 until no further row or column changes occur.

As such the Sort algorithm has some similarities with the banded pattern mining algorithms presented later in this thesis, although the weightings used are calculated in a very different manner and are applicable to ND data.

- **Barycentric (BC) Algorithm:** The BC algorithm was originally developed to support graph drawing where the number of edge "cross-overs" should be minimised. Graphs of interest in this case were translated into adjacency matrices where a "1" entry, indicated an edge between the corresponding vertices, represented by the indicated row and column [42, 72, 78, 116], for bipartite graph layout. As already noted adjacency matrices are akin to the zero-one matrices of interest with respect to this thesis. The BC algorithm operates in a similar manner to the Sort algorithm but using what are referred to as "barycentric" values.

The difference between the 2D Sort and BC algorithm is that the former operates by calculating the weighted row (column) sums of the location indexes of dots within each row (column), while the latter operates by calculating the average of location indexes of dots within each row (column). The BC algorithm has been shown to be much more efficient than the 2D Sort algorithm [116], hence the BC algorithm was used as a comparator algorithm with respect to the evaluation presented later in this thesis. The BC algorithm is therefore discuss in further detail in Section 2.3.1.

2.2.5 Reorderable Pattern

The concept of banding in the context of numerical analysis was discussed in Subsection 2.2.3 and in the context of reorderable matrices in Subsection 2.2.4. This section considers the concept of banding in the context of reorderable patterns. The idea of reorderable patterns is akin to reorderable matrices, the idea is to reorder columns and rows so as to reveals some pattern of interests that may not otherwise have been noticed in the data. The distinction between the idea of reorderable patterns and that of reorderable matrices is that the emphasis is not on visualisation but on pattern discovery. As such the motivation for reorderable patterns can be argued to be the same as that for the work on Banded Pattern Mining (BPM) presented in this thesis; the distinction is that the idea of reorderable patterns is concerned with any pre-prescribed pattern that can be revealed by reordering the columns and rows in a 2D matrix not just banding (it is also not necessarily directed at zero-one data).

Thus pattern discovery in binary 2D matrices, using the idea of reorderable patterns [11, 39], involves the reordering of columns and rows so as to reveal the presence (or otherwise) of some pre-specified pattern P of significance to some end application; for example the pattern might reveal some feature of the data. This can be viewed as a generalisation of the BPM problem of interest with respect to this thesis in the sense that the pattern we are looking for in BPM is dots arranged about the leading diagonal, P in this case would be the locations about the leading diagonal up to a certain distance away. Of course in the context of the domain of reorderable patterns P can be any shape. An example is given in Figure 2.7 where P is a rectangular pattern. Figure 2.7(a) shows the original 2D matrix, while Figure 2.7(b) show the matrix after the rows and columns have been rearranged. The assumption here is that the rectangular shape P is significant with respect to some end application.

The challenge of reorderable patterns is finding correct permutations by which the pattern P is revealed. This is known to be a NP-Complete problem, because it requires a factorial number of permutations of rows and columns to reveal the hidden patterns. A description of a general framework for the discovery of reorderable pattern will be presented next. Consider a pattern P, describing a property (structure) which we wish to find in a binary valued matrix A. The set A' is then the set of all permutations of A that can be obtained by reordering its columns and rows ($A' = \{A_1, A_2, \dots\}$). We say matrix A features pattern P, if $\exists(A_i \in A' \in P)$ such that $P \subseteq A_i$. Given a pattern P, its associated reorderable pattern $R(P)$ will comprise all the matrices in A', that are a subset of P (there may be none).

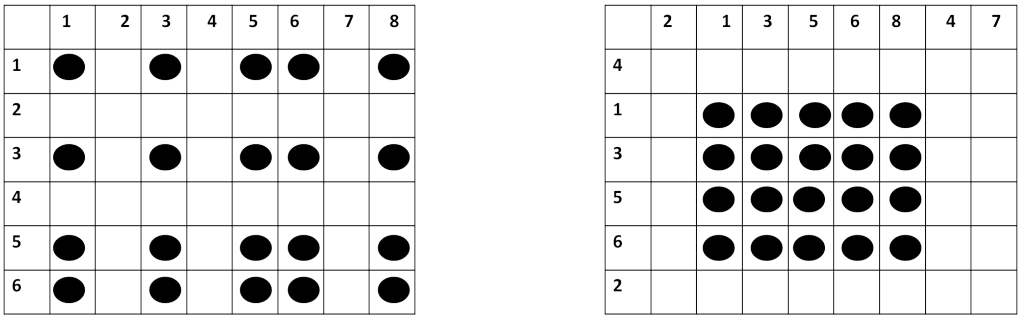


FIGURE 2.7: (a) original matrix and (b) reordered rows and columns of original matrix to reveal a pattern of interest [56]

The idea of Reorderable Patterns, patterns that can be generated by reordering the columns and rows in a 2D matrix is significant with respect to the work presented in this thesis, because it has some similarity with the idea of BPM except that the patterns of interest are not necessarily banded patterns and the data considered does not necessarily have to be zero-one data. The disadvantage of existing approaches to reorderable pattern discovery is that they consider all possible permutations and thus the proposed algorithms are therefore NP-Complete.

2.2.6 Bandwidth Minimisation Problem (BMP)

Bandwidth minimisation [25, 26, 38, 40, 58, 90] is concerned with the process of minimising the bandwidth of the non-zero entries of a sparse 2D matrix by permuting (reordering) its rows and columns such that the non-zero entries form a narrow "band" that is as close as possible to the leading diagonal (see Figure 2.9) [102]. More specifically, given a sparse matrix $A = [a_{ij}]$, the objective is to minimise:

$$\{max|\imath - \jmath| : a_{ij} \neq 0\}$$

An example is given in Figure 2.9 [83, 102, 112]. Thus the idea is to use row-column permutations to transform a given sparse matrix into banded form such that the bandwidth will be as small as possible. This can be achieved, if as many non-zero elements as possible can be arranged along the main diagonal.

The Bandwidth Minimisation Problem (BMP) is a well established combinatorial optimisation problem which originated in the 1950s [28, 35, 103, 53], is similiar to applied mathematics and occurs with respect to many applications in science and engineering [102]. One reported application domain [74] is the computerised structural analysis of steel frameworks, where the reordering of the matrix tends to be beneficial for dealing with what are known as "inversions and determinants". More generally matrix operations with small bandwidth tend to require less space and time. However, it should be noted that Bandwith Minimisation of binary matrices is known to be NP-Complete [35], as it is related to the reordering of binary matrices [93].

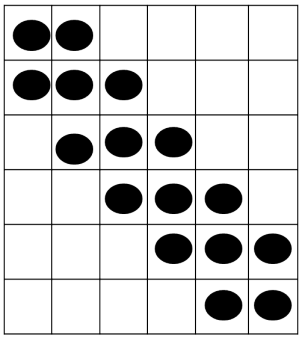


FIGURE 2.8: Example bandwidth minimisation [116]

The BMP is clearly akin to the BPM of concern with respect to this thesis. The distinction is that BMP is directed at the specific objective of minimising bandwidth to aid further processing of the 2D matrices of interest, while BPM is directed at data analysis (a by-product of which happens to be bandwidth minimisation and also visualisation).

2.2.7 Matrix Seriation

Seriation, also known as sequencing, is concerned with rearranging of a set of objects in a linear order so as to reveal structural information. Seriation is an important problem in the field of combinatorial data analysis [9]. However, due to the combinatorial nature

of the problem, the number of possible solutions grows with respect to the problem size (number of objects n) by O(n!). Seriation has been applied to a variety of disciplines, including: (i) archeology and anthropology, (ii) information visualisation, (iii) sociology and sociometry. In archeology, Patrie [52] used the concept to find a chronological order for graves discovered in the Nile area. Here, Cross-Tabulation of graves sites and objects was used. By rearranging the table rows and columns, graves with similar objects were found to be closer to each other (see Figure 2.9).

Seriation as an unsupervised data mining technique that reorders objects into sequence along a one-dimensional continuum so that it best reveals regularity and patterns within the series [82]. Thus the problem is directly related to ranking [50]. The idea is, given a similarity matrix that contain a set of n items, that these items can be ordered along a chain (path) such that the similarity between these items decreases with their distance along the path (that is a total order exists). The idea is to reconstruct the underlying linear ordering using unsorted and possibly noisy, pairwise similarity information. Atkins et al. [10] produced a spectral algorithm that solves the seriation problem exactly in the noiseless case, by showing that for similarity matrices computed from serial variables, the ordering of the eigenvector that corresponds to the second smallest eigenvalue of the Laplacian matrix (the Fiedler vector) matches that of the variables. In practice, this means that performing spectral ordering on the similarity matrix reconstructs the correct ordering provided the items are organized in a chain (path).

The idea of matrix seriation with respect to generating structural information by reordering a set of object in a linear order, so as to maximise the human visual perception of patterns and the overall trend, is significant with respect to the work presented in this thesis, because it has some similarity with the idea of BPM except that the generated patterns of interest are not necessarily banded patterns and the data considered might not necessarily be zero-one data. The disadvantage of the matrix seriation problem solutions is that they considers all possible permutations and thus the problem is NP-Complete.

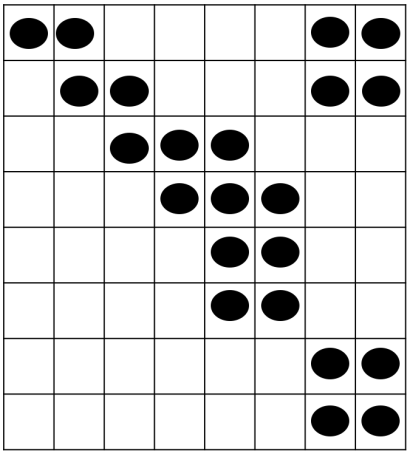


FIGURE 2.9: Example Matrix Seriation[82]

2.2.8 Banded Pattern Mining

There has been some limited previous work directed at Banded Pattern Mining (BPM) as conceived of in this thesis where BPM is defined as the identification of hidden banding in zero-one data. Of particular note in the context of existing work on BPM is the work of Gemma et al. [55] who proposed the Minimum Banded Augmentation (MBA) algorithm.

The MBA algorithm operates by minimising the distance, by considering row and column permutations, of the non zero entries (dots) from the main diagonal of a given 2D matrix. The algorithm considers a series of column permutations to produce a number of *permuted matrices* (ordered matrices). Each column permutation is considered to be fixed whilst row permutations are considered. The algorithm commences by "flipping" zeros and/or ones so that the rows feature what Gemma et al. refer to as the "Consecutive-Ones Property" (C1P). The C1P for a given row is where the $1s$ have a consecutive arrangement. Next the algorithm remove all the "Sperner conflicts" (see Section 2.3.1.2 for further detail). The result is a banding for each permutation. To determine the quality of each bandings, the MBA algorithm uses an accuracy measure (also discussed in further detail in Section 2.3.1.2 below).

To determine the column permutations some heuristical methods were proposed in [56] to determine suitable permutations. These heuristic methods comprised similarity measures for comparing columns and the spectral ordering method for finding a fixed column permutation π [56]. More specifically:

1. **Correlation Similarity:** Heuristic concerned with measuring the similarity between two columns [84]. The aim is to compute a value that will evaluates the strength of the association between the columns. The correlation similarity is defined as follow:

$$\text{CorrelationSimilarity} = (1 + \rho_{a,b})/2$$

   Where $\rho_{a,b}$ is the Pearson coefficient between two columns a and b with value "1" indicating similar columns and "0" indicate non overlapping columns.

2. **Jaccard Coefficient:** An alternative heuristic is to use an overlapping measure computed using the "Jaccard Coefficient" which has its origins in set comparison theory [123, 97]. It is also sometimes referred to as the "Tanimoto measure" [70, 108]. The Jaccard Coefficient is defined by the ratio of the common elements of two columns a and b to the number of all the different columns as follows:

$$\text{JaccardCorrelation} = \frac{|A \cap B|}{|A \cup B|}$$

   Where A and B are the set interpretation for columns a and b. The intuition here is that the Jaccard Coefficient will capture the particularities of a banded structure where it is expected that columns will overlap.

3. **Spectral Ordering:** Heuristic based on the spectral analysis [10, 14, 96, 126] of a similarity graph over columns of a given matrix, where the columns are rearranged so that similar columns are put as close to each other as possible. Thus given a symmetric similarity matrix (a matrix of scores that represent the similarity between columns), the aim is to construct a Laplacian matrix L and to find its eigenvector v that is associated with the second smallest eigenvalue of L [49, 77, 89, 104]. The values v are then sorted to produce the column permutation π [55]. A Laplacian matrix is a matrix defined as: $L = D - A$ (where D is a diagonal matrix and A its adjacency graph) [67].

Two variations of the MBA algorithms have been proposed [56]; the Minimum Banded Augmentation "Fixed Permution" (MBA_{FP}) and the Minimum Banded Augmentation "Bi-directional Fixed Permutation" MBA_{BFP} algorithm. Both algorithms featured the joint disadvantages of: (i) being computationally expensive, and (ii) as consequence, being only applicable to 2D data.

The significance of the MBA_{FP} and MBA_{BFP} algorithms with respect to this thesis is that they were used to compare the operation of the proposed BPM algorithms. The two variations of the MBA algorithm are therefore considered, respectively, in further detail in Sub-sections 2.3.1.1 and 2.3.1.2 below.

2.2.9 Banded Pattern Mining Summary and Discussion

In the foregoing a number of research topics related to banding have been discussed namely: (i) numerical analysis, (ii) reorderable matrices, (iii) reorderable patterns, (iv) bandwidth minimisation, (v) matrix seriation and (vi) banded pattern mining. In the case of numerical analysis the objective was not the banding itself but its usage to solve sets of linear equations. In the case of reorderable matrices the aim was to support the visualisation of 2D data which might be binary (but not necessarily so) and might include banding (but not necessarily so). With respect to reorderable patterns the aim was to determine if predefined geometric patterns exist in a given data set. Again this data set might be binary (but not necessarily so) and the pattern being looked for might be a banding (but again not necessarily so). Bandwidth minimisation is concerned with the efficiency with which 2D matrices can be processed, the banding is not a goal in its own right nor is the objective to discover the nature of any banding that might exist in the input data. With respect to matrix seriation the aim is to reveal the similarity between items in order to maximise the human visual perception of patterns, and the overall trend, which might not necessarily be binary and might not necessarily be bandings. Previous work on banded pattern mining has mostly focussed on the use of row-column permutations. The proposed BPM algorithms have advantages with respect to all of the above. More particularly, because the proposed mechanism does not involve the generation of large numbers of permutations, it is not NP complete; in other words it scales to ND data (unlike the foregoing which, to the best knowledge of the author, were all directed at 2D data).

2.3 Review of Selected Banding Algorithms

This section provides more detail concerning the three algorithms identified above and used for comparison purposes later in the thesis:

1. The Barycenter (BC) algorithm [92].
2. The Minimum Banded Augmentation "Fixed Permutation" (MBA_{FP}) algorithm [55, 56].
3. The Minimum Banded Augmentation "Bidirectional Fixed Permutation" (MBA_{BFP}) algorithm [55, 56].

The BC algorithm is therefore discussed in further detail in Subsection 2.3.1 below and the two MBA algorithms in the following two Sub-section, Subsections 2.3.1.1 and 2.3.1.2.

2.3.1 Barycenter (BC) Algorithm

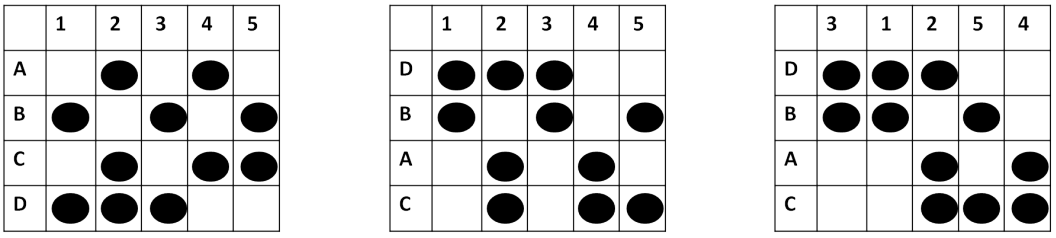
The Barycentric algorithm was introduced in Subsection 2.2.4 above. As noted above the Barycenter (BC) algorithm was originally used with respect to graph drawing algorithms [78], and more recently used to reorder binary matrices [92, 116]. In essence, the Barycentric algorithm finds permutations for both rows and columns such that non-zero entries are as close to each other as possible. It is based on the *barycenter* measure, which is the average position of 1s (dots) in a given column/row. The pseudo code for the Barycenter algorithm is presented in Algorithm 1. The input (Line 1) is a zero-one data set A. The output is a rearranged matrix A (Line 2). The Barycenter algorithm computes the barycenter measure for all rows in A (Lines 4 to 6), then permutates (reorders) the rows in ascending order of the barycenter value (Line 7). The algorithm then transposes the matrix A^T (Line 8) again and iterates until convergence (Line 9).

Algorithm 1: The Barycenter (BC) algorithm

```
1:  Input: An n × m binary matrix A
2:  Output: Permutation of rows and columns of A
3:  loop
4:    for each row i ∈ A do
5:      Compute barycenter for row i
6:    end for
7:    A′ = A with rows rearranged in ascending order of the barycenter measure
8:    A^T = the transpose of A′
9:    Repeat process on A^T until convergence (no further changes)
10:   A = A^T
11: end loop
12: Exit with A
```

A simple example illustrating the operation of the BC algorithm is given in Figure 2.10. Figure 2.10(a) gives the input matrix. As already noted the algorithm first compute

the barycenter for each row i, this is shown in Table 2.1. The rows are arranged in ascending order of barycenter measure as shown in Figure 2.10(b). Next the matrix is transposed and the barycenter values are recalculated for the columns to obtain the results shown in Table 2.2. The columns in the matrix are again rearranged in ascending order of the barycenter measure to produce the configuration shown in Figure 2.10(c). The process is repeated on the next iteration. However in this case the same barycenter measures are produced (indicating that no further changes can be made).



(a) input matrix (b) Rows rearranged (c) Columns rearranged

FIGURE 2.10: Example illustrating the Barycenter (BC) Algorithm [116]

TABLE 2.1: Calculation of barycenter values for row

Index	Weighted sum of dots (index)	Sum of Dots	*barycenter*
1	$(1*2)+(2*4)=10$	$1+2=3$	3.33
2	$(1*1)+(2*3)+(3*5)=22$	$1+2+3=6$	3.67
3	$(1*2)+(2*4)+(3*5)=25$	$1+2+3=6$	4.17
4	$(1*1)+(2*2)+(3*3)=14$	$1+2+3=6$	2.33

TABLE 2.2: Calculation of barycenter values for columns

Index	Weighted sum of dots (index)	Sum of dots	*barycenter*
1	$(1*1)+(2*2)=5$	$1+2=3$	1.67
2	$(1*1)+(2*3)+(3*4)=19$	$1+2+3=6$	3.17
3	$(1*1)+(2*2)=5$	$1+2=3$	1.67
4	$(1*3)+(2*4)=11$	$1+2=3$	3.67
5	$(1*2)+(2*4)=10$	$1+2=3$	3.33

The BC algorithm uses what is referred to as the Mean Row Moment (MRM) to evaluate the quality of the bandings produced. This is calculated as shown in Equation 2.4, where a_{ij} is the jth entry in row (column) i and n is the number of columns (rows)

$$MRM = \frac{\sum_{j=1}^{n} j \cdot a_{i,j}}{\sum_{j=1}^{n} a_{i,j}} \tag{2.4}$$

2.3.1.1 Fixed Permutation (MBA) Algorithm

The MBA algorithms, the MBA_{FP} algorithm described in this Sub-section and the MBA_{BFP} algorithm described in the following Sub-section, have significant similarity. Firstly, in both cases, banding is defined as follows. A matrix is fully banded if there exists a column permutation π and a row permutation κ whereby:

1. For each row/column the dots appear in continuous sequence $[a, b]$ where a and b are the start and end indices; the row/column is thus said to features the consecutive dots property [20, 65, 68, 101, 122, 124, 125, 128].

2. For each pair of rows i and $i + 1$, that feature the consecutive dots property, with sequences $[a_i, b_i]$ and $[a_{i+1}, b_{i+1}]$, $a_i \leq a_{i+1}$ and $b_i \leq b_{i+1}$. In other words the matrix rows should be "stepped". In [55] this requirement is presented in terms of Sperner families of sets. A Sperner family, named after the German mathematician Emmanuel Sperner, is a collection of sets where, for any pair of sets, neither is a proper subset of the other.

Secondly both MBA algorithms operate using a given fixed column permutation π. Thus, at first glance, to find a best banding all column permutations need to be considered. Given m columns there will be $m!$ permutations, However, the requirement of the consecutive dots property, assuming full banding exists, means that some column permutations will not need to be considered. In [55], it is suggested that the concept of an "incompatibility graph" is used, a bipartite graph where the vertices are column index pairs and the edges represent incompatibilities, to generate permutations. Alternatives are suggested in [55].

The above all assumes that a full banding exists, in practice this is often not the case, the solution embedded in both algorithms (but in different ways) is to "flip" zeros to ones (no-dots to dots) and ones to zeroes (dots to non-dots). The MBA_{FP} algorithm considered in this sub-section only flips zeros to ones. Of course this serves to introduce dots that were not in the original data set but it is argued that this is justified where non-dots actually represent "dont knows". Once a "best" banding has been identified the newly introduced dots can be removed to ensure compatibility with the original data set. A good solution is one that minimizes the flips. Note that the row ordering is not changed.

The pseudo code for the MBA_{FP} Algorithm is presented in Algorithm 2. As noted above the pseudo code assumes a given column permutation π. Recall also that the MBA_{FP} only allows 0-to-1 flips. The basic idea behind the Algorithm is to: (i) process the input matrix M so that it features the C1P and (ii) then resolve all Sperner conflicts between rows. The inputs to Algorithm 2 (Line 1) are: (i) a zero-one matrix M and (ii) a fixed column permutation π. To enforce the C1P, all possible $0s$ entries falling between $1s$ for each row in M^{π} (column permutation) (Lines 4 to 6) will be flipped. Next the Sperner conflicts between rows of the given matrix M^{π} are removed (Lines 7

to 13), by ensuring that all row intervals have a pairwise overlapping sequence of rows; that is any two given rows M_i and M_j will form a Sperner family of intervals.

Note that a matrix is said to have "Sperner conflicts", if the rows do not form a Sperner family of intervals: Two rows $M_i = [a, b]$ and $M_j = [a', b']$ with C1P, where $i \neq j$, will form a Sperner family of intervals if they are overlapping such that $a < a'$ and $b' < b$. Since only 0 to 1 flips are allowed, the solution is to extend the row intervals. Here an extension of M_i = [a,b], refers to updating the endpoints of the interval for a new endpoint [a', b'] such that: $a \leq a'$ and $b' \leq b$ or $a' \leq a$ and $b \leq b'$. Note that at every step the algorithm takes a row M_i^π and computes its optimal extension.

This is done by selecting all the super-intervals $M_j^\pi \succ M_i^\pi$ and checking all the potential extensions for M_i^π that could resolve the Sperner conflicts in row i with respect to row j. An extension of M_i^π that will resolves all Sperner-conflicts for that row which can either be: (i) a left-hand side extension to the leftmost $M_j^\pi \succ M_i^\pi$ (Line10 (A)), (ii) a right-hand side extension to the rightmost $M_j^\pi \succ M_i^\pi$ (Line 11 (B)) or (iii) extensions to both the left and right hand sides with combination of two super-intervals (Line 12 (C)), by checking the start point of each $M_j^\pi \succ M_i^\pi$ in combination with the rightmost end point from all other super-intervals $M_\kappa^\pi \succ M_i^\pi$. The algorithm then takes the extension with the fewest transformations for row i.

Algorithm 2: The Fixed Permutation (FP) MBA Algorithm

1: **Input:** An $n \times m$ binary matrix M and column permutation π
2: **Output:** A permutation of κ rows
3: M^π = The Input matrix M with permutation π imposed on it
4: **for each** row $i \in M^\pi$ **do**
5: Flip 0s falling between the first and last 1s
6: **end for**
7: **for each** row $i \in M^\pi$ featuring a consecutive dots sequence $[a, b]$ **do**
8: $C = \{M_j^\pi = [a_j, b_j] |\ M_j^\pi \prec M_i^\pi\}$ row j is contained in row i // conflicting rows
9: Extend $M_i^\pi = [x, y]$ from the following options so that $y - x$ is minimum:
10: (A) $x = min\{a_j | [a_j, b_j] \in C\}$ and y = b
11: (B) $x = a$ and $y = max\{b_j | [a_j, b_j] \in C\}$
12: (C) $x = a_j$ and $y = max\{b_k | [a_k, b_k] \in C, a_k < a_j\}$, for every $M_j^\pi = [a_j, b_j] \in C$ several combinations of a and b
13: **end for**
14: Sort the rows [a,b] of M^π in ascending order of as, resolving ties with ascending order of their bs

A simple example illustrating the operation of the MBA$_{FP}$ algorithm is given in Figure 2.11. Figure 2.11(a) gives the input matrix. As already noted the algorithm first assumes that the column permutation M^π is given before hand, the algorithm then needs to transform "0s" falling between "1s" so that the input matrix features the consecutive-ones relation, this corresponds to flipping one 0 in the first row of the matrix to a 1 entry as shown in Figure 2.11(b). Second the algorithm resolves the Sperner conflicts between the rows. Note that there is a conflict between the second and third row, and this can be resolved by flipping the bottom right "0" entry to a "1"

entry as shown in Figure 2.11(c). The matrix is fully banded for this permutation after making two 0-to-1 flips. The Bidirectional Fixed Permutation MBA_{BFP} variation of the MBA algorithm is discussed next in following subsection.

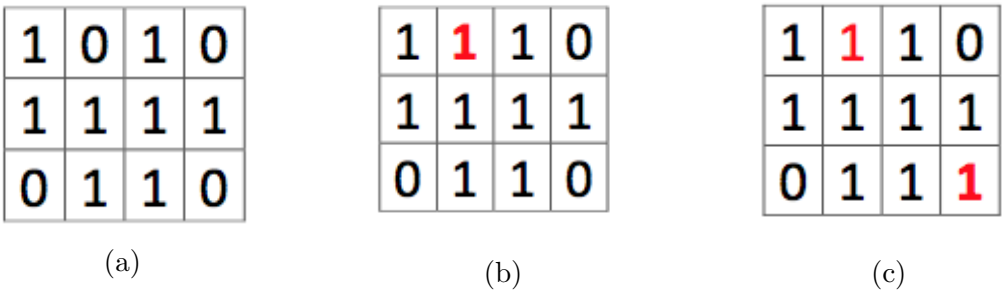


FIGURE 2.11: Example illustrating the MBA Fixed Permutation (MBA_{FP}) Algorithm [56]

2.3.1.2 Bi-directional Fixed Permutation (MBA) Algorithm

The Minimum Banded Augmentation Bidirectional Fixed Permutation (MBA_{BFP}) variation of the MBA algorithm is presented in this sub-section. As before the description is given in terms of a given column permutation π. Note that a good column permutation tends to put similar columns close to each other. As noted above, the Jaccard Coefficient was used as a column similarity measure where the similarity measure returns the value 1 when two columns are similar and 0 for non-ovalapping columns. The spectral ordering in [56], on the other hand, was used to find a fixed column permutation π on matrix M. The distinction between the two variations is that the former (MBA_{FP}) algorithms addresses the CIP only by flipping zero (0) entries to one (1) entries, while the latter (MBA_{BFP}) operates by flipping both zero (0) to one (1) entries and one (1) to zero (0) entries. Given a row in M, to decide whether to flip from a zero to a one or vice-versa, a weighting scheme was used whereby dots are given a "+1" and non-dots a "−1"; the combination of flips that is closest to the sum of the original weightings is then the most desirable. As before the MBA_{BFP} algorithm first adjusts the matrix so that it features C1P and then resolves the Sperner conflicts between the rows. In [55], the MBA_{BFP} algorithm is described as follows [55]; given a binary matrix M, find the minimum number of bidirectional flips (flips from both zero (0) to one (1) entries and one (1) to zero (0) entries) so that M becomes fully banded.

The pseudo code for the MBA_{BFP} algorithm is presented in Algorithm 3: The input (Line 1) is a zero-one matrix M and a column permutation π. The algorithm considers an $n \times m$ matrix M of weight W, where each one entry (dot) will be given a weight of "+1" and each zero entry (no dot), a weight of "−1" (see Equation 2.5 in [56]). The C1P for the bidirectional flips for M^{π}, corresponds to solving the "maximum sub-array problem" on weight W_i^{π}. The objective of solving the maximum sub-array problem is, given an array of numbers, to find the sub-array with the maximum sum of the numbers. Note that it was established that this problem can be solved in Linear time with respect

to the size of the array using the scan-line algorithm [33]. Furthermore, this method returns interval boundaries which are used to solve the C1P on M^π, by setting the fields in M^π to 1 (dot) and others to 0 (no dot) (Lines 4 to 8). Next the algorithm deals with removing the Sperner conflicts between the rows of M^π as described above. Note that additional flips on the rows of M^π are required so that the rows have the Sperner family of intervals property. Let $\tilde{M}$ be the binary matrix M augmented with $M_{ij} = M_i^\pi \backslash M_j^\pi$, for every two rows $M_i^\pi \subset M_j^\pi$, M will be fully banded if and only if $\tilde{M}$ has C1P (see proof in [55]). To remove all Sperner conflicts (Lines 9 to 16) between the row intervals in M^π, the algorithm will go through all the extra rows described in $\tilde{M}$, thus solving the maximum sub-array problem on the rows. Lastly, additional flips are required on the extra rows to establish a C1P. Finally, the rows in M^π are updated according to the changes made over $\tilde{M}$ to get a banded matrix.

Algorithm 3: The Bidirectional Fixed Permutation (BFP) MBA Algorithm

1: **Input:** An $n \times m$ zero-one matrix M and a column permutation π
2: **Output:** A permutation κ of rows
3: M^π = The Input matrix M with column permutation π imposed on it
4: **for each** row $i \in M^\pi$ **do**
5: Let the weight vector for row i on matrix M be W_i^π
6: Let the solution to the maximum consecutive subarray on W_i^π be [a,b]
7: Update M_i^π = [a,b]
8: **end for**
9: **for each pair of** row $i, j \in M^\pi$ **do**
10: **if** $M_i^\pi \subset M_j^\pi$ **then**
11: Let $M_i^\pi \backslash M_j^\pi$ = A
12: Let the weight vector for A be W_A
13: Let the solution to the maximum consecutive subarray on W_A be [a,b]
14: Update M_i^π so that it preserve $M_j^\pi \backslash M_i^\pi = [a, b]$
15: **end if**
16: **end for**
17: Sort the rows [a,b] of M^π in ascending order of as, resolving ties with ascending order of their bs

It is note worthy that the MBA algorithms uses an accuracy (Acc) measure to evaluate the performance of the banding produced and this is calculated as shown in Equation 2.6; where (i) TP (true positives) is the number of $1s$ entries in the rows (or columns) that remained unchanged, (ii) TN (true negatives) is the number of $0s$ entries in the rows (or columns) that remained unchanged, (iii) FP (false positives) is the number of 0 entries that have been transformed into a 1, and (iv) FN (false negatives) is the number of $1s$ entries that have been transformed to a 0.

$$W(ij) = \begin{cases} +1 & \text{if } M_{ij} = 1 \\ -1 & \text{if } M_{ij} = 0 \end{cases} \tag{2.5}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

A simple example illustrating the operation of the MBA_{BFP} algorithm is given in Figure 2.12. Figure 2.12(a) gives the input matrix. As before the MBA_{BFP} algorithm first assumes the column permutation M^{π} is given before hand, the algorithm then needs to transform the matrix so that it features the C1P for each row. In this example, this corresponds to flipping the second "0" entry in the first row to a "1" entry as shown in Figure 2.12(b). Secondly, the MBA_{BFP} algorithm resolves the Sperner conflicts between the second, third and fourth rows, this will be resolved after flipping the last "1" entry in the second row to a "0" entry as shown in Figure 2.12(c).

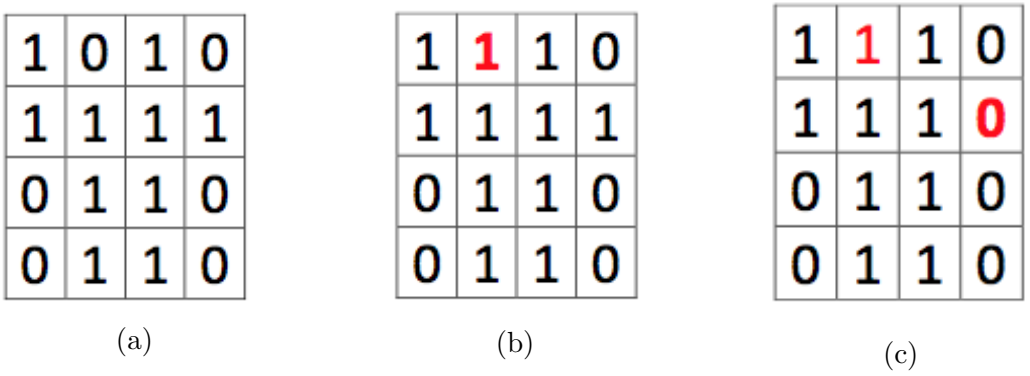


FIGURE 2.12: Example illustrating the MBA Bidirectional Fixed Permutation (MBA_{BFP}) Algorithm [56]

2.4 Data Mining And Knowledge Discovery in Databases (KDD) Process

Given that the number of large data sets that are electronically available keeps increasing, the assumption is that there is an increasing amount of valuable hidden knowledge within this data. The suggestion is that the discovery of this knowledge may be useful to decision makers and stakeholders. At its simplest the data is stored in relational databases. However, query languages like SQL (Structured Query Language) are not well suited to the discovery and extraction of the hidden knowledge that is believed to exist, such as relationships and/or patterns. The identification of such knowledge requires alternative more sophisticated tools and mechanisms; this is the domain of Knowledge Discovery in Databases (KDD). Banded patterns, as presented in this thesis, are a form of hidden knowledge. Hence, the material presented in this thesis is considered to fall within the domain of KDD.

The term Knowledge Discovery in Databases (KDD) and Data Mining (DM) have been used interchangeably to describe the overall process of extracting or discovering useful and meaningful information from data. However, in this thesis, and in line with many other authors, the definition presented in [45] is used; KDD is the overall process

of discovering useful information and knowledge (banded patterns with respect to this thesis) from data, while DM is the sub-process within the overall KDD process where data discovery takes place (hence banded pattern mining).

KDD integrates a number of processes, from raw data preparation prior to the application of DM to final result visualisation. Figure 2.13 shows a schematic of the KDD process as suggested by [23, 45, 85]. With reference to the figure each step is described in further detail in Sub-section 2.4.1. Sub-section 2.4.2 is then directed at the DM stage in particular. One of the motivations presented earlier in this thesis (see Subsection 2.2.1) was the conjecture that banding enhances the efficiency of certain data mining operations. To demonstrate this, later in this thesis, a particular data mining approach known as Frequent Itemset Mining (FIM) is considered. So that the reader has the appropriate background knowledge concerning this DM technique the FIM process is presented in Subsection 2.4.3.

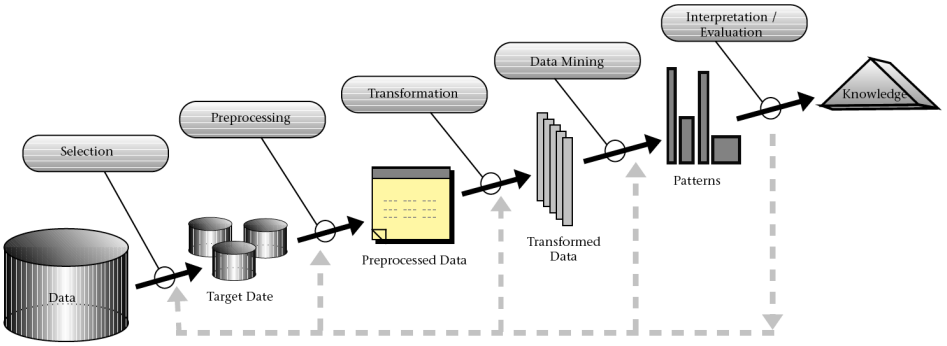


FIGURE 2.13: KDD process functional steps [45]

2.4.1 The KDD Process

The KDD process as noted above, encompasses a number of stages as follows

1. **Selecting and Understanding the application domain:** The first stage is to define the KDD problem's scope and boundaries and to develop an understanding of the application domain and the relevant prior knowledge. During this stage the goals of the KDD exercise, from the end user point of view, are identified.

2. **Data Cleaning and Preprocessing:** This second stage is concerned with data preparation. This stage comprises operations such as: the removal of noise or outliers and the application of strategies for handling missing data.

3. **Data Reduction and transformation:** Often we do not require all the data, or cannot process all the data. This stage thus involves finding the most useful features to represent the data (feature selection). Dimensionality reduction methods

are sometimes used to reduce the effective number of variables under consideration. The data to be mined is also sometimes not in a format to which DM can be applied and thus will need to be transformed into an appropriate format.

4. **Data Mining:** The data mining stage involves the actual searching for the hidden knowledge of interest. For example patterns, classification rules, decision trees, regression models or cluster configurations.

5. **Evaluation:** The final stage of the KDD process is the analysis of the data mining results obtained. This may include the use of visualisation technique to help analysts decide the utility of the extracted knowledge. The bandings identified with respect to the work presented in this thesis may be argued to be a form of visualisation.

The above process is equally applicable to banded pattern mining, as presented in this thesis, although some of the stages required little attention. Typically there is no need for "removal of noise or outliers" and typically there is no "missing data" (Stage 2). Similarly there is also no need for data reduction although the data does need to be transformed into a zero-one format. How this was done with respect to the data sets used for the evaluation reported on later in this thesis is described in the following chapter, Chapter 3. The following subsection, Subsection 2.4.2, considers the Data Mining stage (Stage 4) in more detail because of its significance with respect to this thesis.

2.4.2 The DM KDD Sub-Process

Data Mining (DM) is defined as the application of specific algorithms for extracting patterns from data [23, 45, 85]. As noted above, it is the central element of the KDD process. In [45], the goals of data mining are summarised using a figure, this figure has been reproduced here; Figure 2.14. From the figure we can identify two high level goals: (i) verification and (ii) discovery. Verification is directed at validating certain hypotheses and discovery at finding patterns in data. The discovery goal is further subdivided into: (i) prediction and (ii) description. Prediction is concerned with the discovery of patterns indicative of future behaviour, while description is concerned with patterns that can be used to represent facets of data. Thus, a DM activity such as classification is associated with the prediction goal while activities such as clustering and frequent item set mining (discussed further in Subsection 2.4.3 below) are associated with the description goal. The banded pattern mining proposed in this thesis is also concerned with the description goal.

2.4.3 Frequent Item-set Mining (FIM)

Frequent Itemset Mining (FIM) is concerned with finding patterns in (typically 2D) binary data sets [2, 71, 127]. As noted above, the significance with respect to the work presented in this thesis is that the operation of FIM algorithms, with and without banded

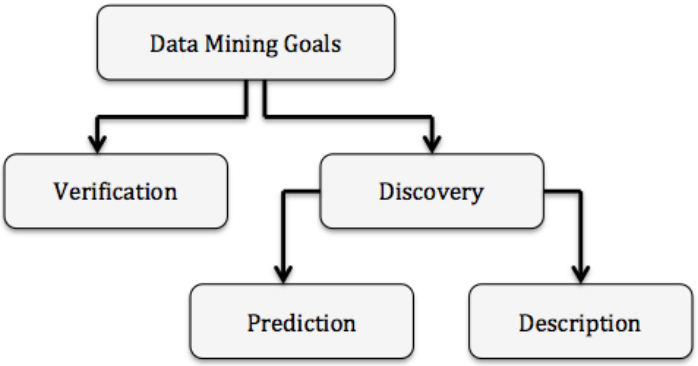


FIGURE 2.14: Data Mining Goals [45]

data, can be used as an indictor of one of the claimed advantages of banding. Namely that banding introduces efficiency savings with respect to some algorithms that operate using binary valued matrices. This section describes the FIM algorithm used for such comparison as reported on later in this thesis.

FIM has been widely reported in the literature typically in the context of transaction data [1, 2, 22, 59]. Transaction data is data exemplified by supermarket basket data. Frequency in this context is defined in terms of an occurrence count which must be greater than a given threshold σ expressed in terms of a percentage of the number of records in the data set under consideration. There are a great many FIM algorithms that can be adopted, well known examples are Apriori [80] and FPgrowth [21]. For the evaluation reported later in this thesis any existing algorithm could have been used, however, the Total From Partial (TFP) FIM algorithm [4, 30] was selected because the source code for this algorithm was readily available to the author.

The TFP algorithm is an established Frequent Pattern Mining (FPM) algorithm that utilises the concept of a set enumeration tree structure called a P-tree (Partial support tree) for fast lookup purposes and a second tree to hold support values called the T-tree (Total tree) [5, 30]. TFP is itself an extension of an earlier algorithm called Apriori-T which used only a T-tree. The T-tree is described as a reverse set enumeration tree, and is argued to offer advantages in terms of time and storage effeciency when generating frequent patterns [32, 51]. Details of the TFP algorithm, together with further details concerning the P-tree and T-tree can be found in [4, 31].

2.5 Sampling and Segmentation Techniques

This section presents an overview of data sampling and segmentation. The significance is that sampling and segmentation are used later in this thesis as mechanisms for processing very large data sets (data sets that cannot be held in primary storage). Subsection 2.5.1 below considers sampling, while Subsection 2.5.2 considers segmentation.

2.5.1 Sampling Technique

Sampling technique is concerned with the selection of a representative subset of a given data set [24]. The term is often used in connection with market analysis where the view of a "sample" of a population is sought so as to estimate the characteristics of a population. In other words, given knowledge of a sample, inferences can be made regarding a whole population. An example can be found in [79], where a sampling approach was used to obtain data from homeless communities in order to estimate the prevalence of mental disorder and assess the services needed by those communities. Further examples can be found in [66, 37, 121]. A review of sampling techniques for market analysis is presented in [24]. A summary of these techniques is presented in Table 2.3 taken from [24], including advantages and disadvantages.

Table 2.3: Summary of Sampling Methods [24]

Sampling Methods	Method Summary	Advantages	Disadvantages
Simple Random	A sampling method where each records in a data set under consideration has an equal chance of being selected.	• Easy to generate. • Avoids bias.	• If data set is large method becomes impracticable. • Poor Representation of overall data set.
Systematic Selection	Method used with respect to stream data.	• More precise than random sampling. • Good coverage of data set.	• Biased. • Under or Over representation of data set likely.
Stratified Sampling	Method whereby the data set is divided into subgroups (strata) and samples are randomly taken from each subgroup.	• Provides highly representative sample. • Correlations and comparisons can be made. • Different sampling approaches can be applied to different stratum.	•Number of subgroups must be predetermined. • Sampling frame has to be prepared seperately for each stratum.
Cluster Sampling	Method whereby a clustering algorithm is applied and representaive samples selected from each cluster.	• Reduction in cost of preparing a sampling frame. • Systematic. • Suited to very large data sets.	• More complicated than other sampling methods.

In the context of data mining, as already noted above, sampling is used where the data set (population) under consideration is too large to be processed as a whole, by processing a subset; the results are then assumed to be representative of the entire data

set. For this assumption to hold the selected subset must be as representative of the data as a whole as possible. There are various techniques which can be adopted to achieve this [24]: (i) Simple Random Sampling, (ii) Systematic Selection, (iii) Stratified Random Sampling and (iv) Cluster Sampling.
With respect to the work presented later in the thesis, a stratified sampling technique was adopted, whereby the data is subdivided into subgroups; k records were then randomly selected from each subgroup. The nature of this stratification will become clearer later in this thesis in Chapter 9. The resulting sample was then used for banded pattern mining and the identified banding applied to the entire data set. The reason for adopting the stratified sampling technique was that it was considered to provide for a more representative sample of the entire data set than the other sampling methods listed above.

2.5.2 Segmentation technique

Segmentation in the context of data mining is a concerned with dividing a given data set into a collection of "chunks" called *segments* and then analysing each segment individually (or only selected segments). Segmentation is an alternative technique to sampling for mining large data sets. Sampling has the principal disadvantage that it can never be guaranteed to be representative of the entire data set from which it is drawn. There is also a general view in the data mining community that wherever possible the entire data set should be taken into consideration to ensure good data mining results. Segmentation offers a solution to these criticisms of sampling. In addition segmentation lends itself to parallelisation/distribution. The challenges of segmentation are: (i) how best to divide the data up so each segment has a similar representative chunk of the data and (ii) how to combine the data mining results (contradictory data mining outcomes may be obtained with respect to different segments). The work described in this thesis adopted the following: (i) segmentation technique and combination using the best GBS and (ii) segmentation technique and combination using the most frequent configuration to identify a best banding.

It should also be noted that the term segmentation is much used in scientific literature but often in alternative contexts to that considered in this thesis. For example the term is frequently used in the context of marketing to identify potential "client segments" (consumers or businesses) [114, 61] and in image analysis to isolate objects of interest in images [115]. However, marketing and image segmentation is different from data segmentation, and thus not of interest with respect to this thesis.

2.6 Evaluating Criteria

This section discusses the evaluation metrics used to measure the quality of the bandings produced by the banding algorithms proposed later in this thesis and the comparator algorithms. The latter was undertaken in two manners:

1. Comparison of the quality of the generated banding with respect to the alternative banding algorithms and variations of such algorithms.

2. Comparison of the effectiveness of the generated bandings, with respect to the alternative banding algorithms and variations there of, in terms of efficiency with respect to established mechanisms that manipulated large collections of binary valued data; namely Frequent Item-Set Mining (FIM).

The first was directed at the quality of the bandings produced. While the second was directed at investigating the suggested advantage that the proposed banding mechanism can provide in terms of the run time efficiency with respect to various mechanisms for manipulating binary valued matrices. More specifically, as already noted, FIM was used. The metric used in this case was runtime.

In terms of the quality of the bandings produced, as noted earlier in Chapter 1, the algorithms proposed in this thesis seek to minimise the concept of a Global Banding Score (GBS), one of the main contributions of this thesis. When comparing the proposed algorithms with the BC, MBA_{BFP} and MBA_{FP} algorithms, that do not seek to minimise GBS, it seemed unfair to do this in terms of GBS. Recall from earlier in this chapter that the BC algorithm uses Mean Row Moment (MRM); whilst the MBA algorithms uses accuracy (acc). Consequently, for the purposes of conducting comparisons, an independent measure was proposed and adopted with respect to the evaluation presented later in this thesis; namely the Average Band Width (ABW).

The ABW measure is the average distance of dots from the diagonal measured according to the distances of the normals from the diagonal to each dot (Equation 2.7, where D is the set of dots (with each dot defined in terms of a set of cartesian coordinates) and $maxABW$ is the maximum possible ABW value given a particular data matrix size.

$$ABW = \frac{\sum_{i=1}^{i=|D|} distance\ d_i\ from\ leading\ diagonal}{|D| \times maxABW} \tag{2.7}$$

2.7 Summary

This chapter has presented a general background to the concept of banded patterns so as to provide the reader with an appropriate level of background knowledge with respect to the work presented later in this thesis. This chapter commenced by considering the advantages/disadvantages of banding and some example application domains. More specifically banded patterns were discussed in the following contexts: (i) numerical analysis, (ii) reorderable matrices, (iii) reorderable patterns, (iv) bandwidth minimisation and, of course, (v) banded pattern mining. This was followed by a general overview of Knowledge Discovery in Databases (KDD), and Data Mining (DM) in particular, so as to place the proposed banded pattern mining in the context of KDD. Finally, the criteria used to evaluate the performance and the significance of bandings was presented. Based

on the literature review presented in this chapter, it can be noted that the challenge of the reported relevant work on banded patterns has been: (i) the generation and testing of large numbers of permutations and (ii) that the algorithms only operate in 2D. However, this is not the case with respect to the proposed BPM algorithms presented later in this thesis. As noted previously in the introductory chapter to this thesis one of the research issues to be addressed by the work described in this thesis is to investigate effective algorithms that avoid the need to consider large numbers of permutations and operate in ND data. The work described in this thesis proposes the Banded Pattern Mining series of algorithms which is based on the concept of banding scores. The next chapter introduces the data sets used in this thesis for evaluation purposes.

Chapter 3

Evaluation Datasets

3.1 Introduction

This chapter describes the data sets used for evaluation purposes with respect to the work presented in this thesis. These data sets can be categorised as follows: (i) randomly generated sythentic data sets, (ii) UCI data sets and (iii) cattle movement data sets. The first two categories comprised 2D data sets only, to allow comparison with existing algorithms, while the third category comprises data sets of higher dimension. Amongst the UCI data sets, some were selected because they are frequently used within the data mining community and others so that a good spread of different sized (in terms of numbers of rows and columns) data sets, with different "densities", could be considered. For the cattle movement data sets, the data sets were extracted from the GB cattle movement database; they were specifically constructed by the author for the purpose of conducting the desired evaluation presented later in this thesis. In the case of the UCI and cattle movement data sets it was necessary to adopt a mechanism whereby the data could be converted into a binary (dot) format.

The rest of this chapter is organised as follows: the synthetic data sets are introduced in Section 3.2, the UCI data sets in Section 3.3 and the cattle movement data sets in Section 3.4. Section 3.5 then concludes the chapter with a summary of all the data sets employed in this thesis.

3.2 Randomly Generated Sythentic Data

The synthetic data sets were used specifically to evaluate the 2D-BPM algorithm proposed later in this thesis (Chapter 4). The data sets were generated using the random data generator proposed in [29][1]; this was originally intended to produce data sets for use in evaluating Association Rule Mining (ARM) algorithms but is equally applicable in the context of banded pattern mining. Note that the random data generator software could have been extended to generate ND data; however the reason for not doing this

[1]The LUCS-KDD Data generator software is available at *http:// www.csc.liv.ac.uk /∼frans /KDD /Software / LUCS_KDD_DataGen_Generator.html*

was that the 2D case is a special case of the ND case. The inputs to the data generator were: (i) the desired number of attributes (columns) m, (ii) the desired number of records (rows) n and (iii) the required density d. Density (d) is the percentage of cells that contain dots in the required data set.

Algorithm 4 illustrates the synthetic data generation process. The inputs are the desired n, m and d values. The output is a collection of Dots D. Dots are allocated to cells $\langle i, j \rangle$ in a weighted random manner. The weighting could simply be done according to d but this would mean each column would be likely to have the same number of dots, an unbalanced distribution of dots was preferred because this was considered to be more "realistic". So that an unbalanced distribution is achieved each column is allocated a probability value weighted by the desired density. This is done in (Lines 7 to 13) of the algorithm. We commence by defining a set P of size m to hold the column probability values (Line 7). For each item p_i in P we allocate a random value between 0 and 100 (Line 9). We use a range of 0 to 100 for the probability values, rather than the more usual range of 0.0 and 1.1, because the value for d is presented as a percentage. We then adjust each value for p_i so that it is weighted by d (Lines 9 and 11). How the adjustment is done depends on whether d is greater or less than 50. Note that if $d = 50$, there is no need for any adjustment. We then (Lines 14 to 21) consider each cell in the data matrix in turn, and for each cell, generate another random number r between 0 to 100. We then compare r with the appropriate p_i value, if $r < p_i$ the cell reference is added to the set D. On completion, we have a collection of dots within a data matrix where the density approximates to d (the exact desired density is unlikely to be achieved because of the random elements included in the process) and the number of dots is not balanced across the columns. Data sets generated in this manner were labeled using the following format $\{n \ldots, m \ldots, d \ldots\}$. For example the label $n100$, $m20$, $d50$ indicates a data set where $n = 100$, $m = 20$ and $d = 50$. Further details concerning individual synthetic data sets used with respect to the evaluations reported on later in this thesis are presented as appropriate.

3.3 University Of California Irvine (UCI) Data sets

This section briefly reviews the UCI data sets used with respect to the evaluations presented later in this thesis. The UCI machine learning data repository [18] was created in 1987, as an archive for benchmark data sets for use by the data mining and machine learning community. Twelve data sets were selected from this repository in such a way that they collectively featured a range of column sizes (m) and row sizes (n). In each case the data was discretised/normalised using the LUCS-KDD (Liverpool University Computer Science - Knowledge Discovery in Data) DN (Discretisation/Normalisation) software [29]. Note that the reason for discretising and normalising the twelve selected data sets used in this thesis was that they were continuous valued data sets and as such

Algorithm 4: Random Data Generation Algorithm

Input:
n = Number of rows.
m = Number of columns.
d = Density.
Output:
D = Collection of dots
$P = \{p_0, p_1, \ldots, p_{m-1}\}$ Set of m column probabilities
loop
  for $i = 1$ **to** $i = m$ **do**
    p_i generate random value between 0 and 100
    if $(d < 50)$ **then**
      $p_i = p_i - (\frac{p_i * d}{50})$
    else
      if $(d > 50)$ **then**
        $p_i = p_i + (\frac{(100 - p_i) * (d - 50)}{50})$
      end if
    end if
  end for
  for $i = 0$ **to** $i = m$ **do**
    for $j = 0$ **to** $j = n$ **do**
      r = random number between 0 and 100
      if $(r < p_i)$ **then**
        $D = D \cup \langle i, j \rangle$
      end if
    end for
  end for
end loop

required preprocessing into the desired zero-one format. Note also that there are very few zero-one benchmark data sets available in the UCI collection.

According to [29], discretisation is the process of categorizing continuously valued data attributes into sub-ranges such that each sub-ranges is identified by a unique integer label (column number). Normalisation on the other hand is the process of converting data attributes with nominal values into unique integer label/column formats. Discretisation can be conducted in two manners:

1. **Equal Size Discretisation (ESD)** where the "dots" are equally distributed across a number of sub-ranges; each sub-range defined so that it holds approximately the same number of dots.

2. **Equal Width Discretisation (EWD)** where the ranges are all of equal length, which in turn usually means that the dots will not be equally distributed across the sub-ranges.

With respect to the UCI data sets EWD was used.

The LUCS-KDD-DN software was originally developed to convert data files available in the UCI data repository into a binary format suitable for use with Association Rule Mining (ARM) software. However the software could clearly equally well be used with respect to other application domains that require binary valued (zero-one) data such as the banded pattern mining application domain of interest with respect to this thesis.

Some statistical information regarding the selected UCI evaluation data sets is given in Table 3.1. In the table the data sets are listed in order of n (number of records). In each case the density value was calculated using equation 3.1.

TABLE 3.1: Statistical summary of selected UCI data sets

Name	Num Records (n)	Num. Columns (m)	Num. Dots	Density (d)
Lympography	148	59	2812	32.20
Hepatitis	155	56	3100	35.71
Wine	178	68	2492	20.59
Heart	303	52	4242	26.92
HorseColic	368	85	8464	27.06
Annealing	898	73	35,022	53.42
Mushroom	8124	90	186,852	25.56
Waveform	5000	101	110,000	21.78
PenDigits	10992	89	186,864	19.10
LetRecognition	20000	106	340,000	16.04
ChessKRvK	28056	58	196,392	12.07
Adult	48842	97	732,630	15.46

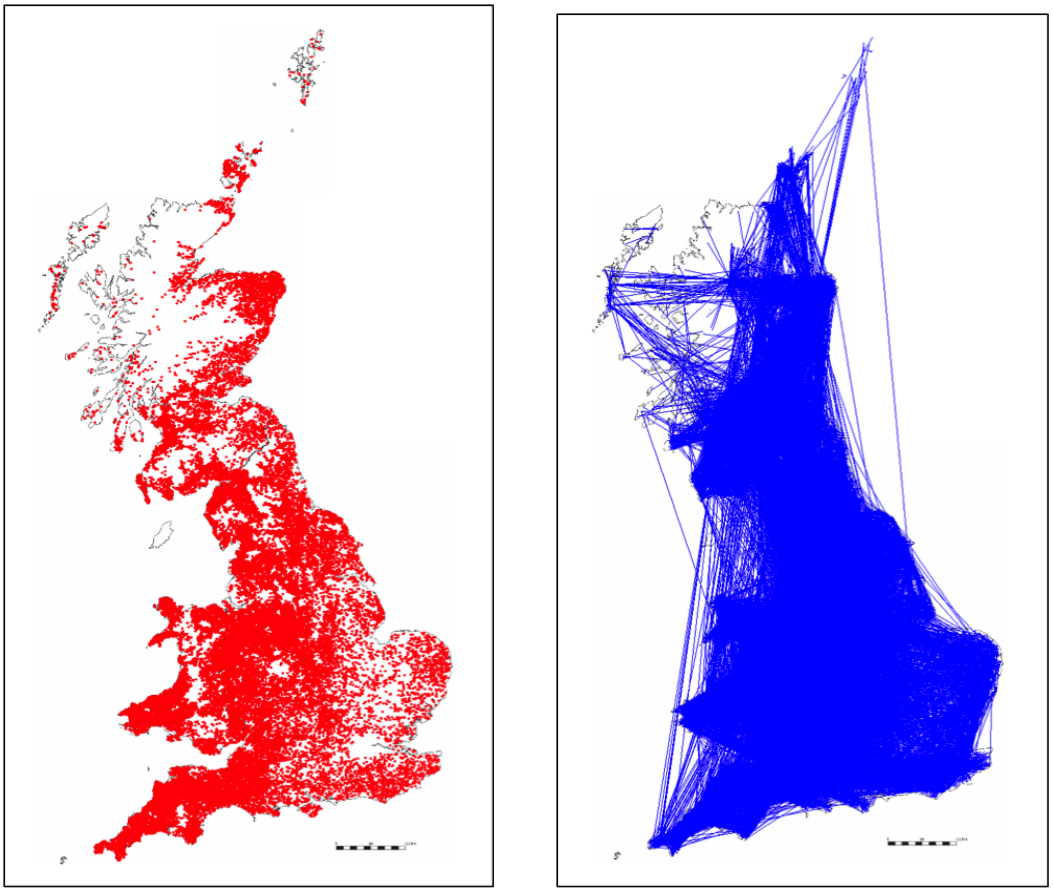
$$D = \frac{Num.\ Dots}{n \times m} \times 100 \tag{3.1}$$

A feature of the UCI data sets is that they are all 2D; the columns represent specific attribute and the rows records. Once discretised the columns represent attribute-values (or in some cases ranges of attributes). Whatever the case the situation where a "cell" may have more than one dot will not arise. It can also be noted that rearranging the record and attribute ordering will not adversely affect the information contained in the data sets in any way.

3.4 Great Britain (GB) Cattle Tracing System

The randomly generated and UCI discretised data sets were all 2D in nature. This section introduces the ND data sets extracted from the database associated with the Great Britain (GB) Cattle Tracing System (CTS). The database is maintained by the UK Department for Environment, Food and Rural Affairs (DEFRA) and records all

the movements of cattle registered within, or imported into GB. The CTS database has been previously studied by a number of authors [62, 98, 109], but not in the context of banding; however the CTS data provides a good example of a large multi-dimensional (ND) data set. Overall the data set was conceptualised as comprising five dimensions: (i) records, (ii) attributes, (iii) "Eastings" (x coordinates of holding areas), (iv) "Northings" (y coordinates of holding areas) and (v) Time. In its raw form each record represents a single animal moved; however, so as to make the data more manageable, some pre-processing was applied so as to collapse records that were identical except for the ID of the animal moved (an extra attribute,"number of animals moved" was added to compensate). The CTS data sets can be viewed in terms of a graph where the vertices represent holding areas and the edges cattle movements. To get a better appreciation of this graph conceptualisation of the CTS data, Figure 3.1(a) shows the vertices (holding areas where cattle were either moved from or moved to), and 3.1(b) the associated edges, with respect to data for the month of January 2013. It is interesting to note from Figure 3.1(a) that the shape of GB can be clearly identified. Figure 3.2 presents a close up of a section of the map given in Figure 3.1(a) that features North Wales and parts of the north-west coast of England.



(a) (b)

FIGURE 3.1: January 2013 GB cattle movement data: (a) vertices (locations) and (b) edges (cattle movements)

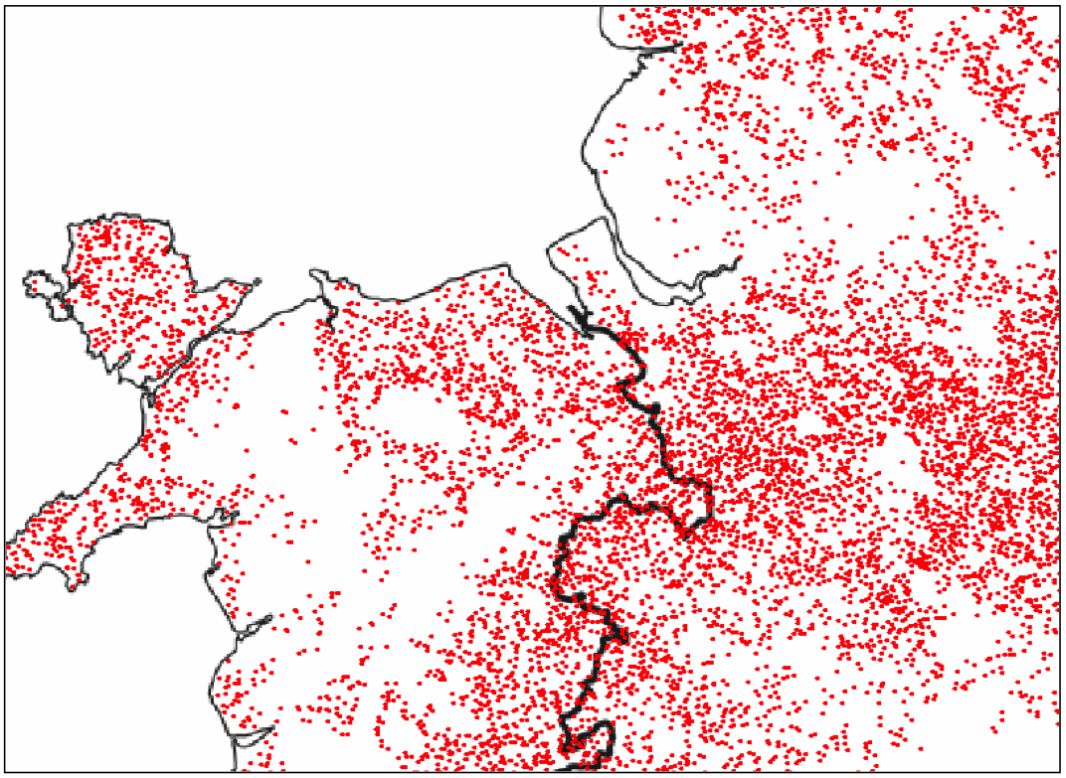


FIGURE 3.2: Close up of Figure 3.1(a) showing North Wales and part of the west coast of England

For the experimental analysis reported on later in this thesis, data sets from 2003 to 2006 across four specific counties were extracted, namely: Aberdeenshire, Cornwall, Lancashire and Norfolk (these counties were selected because they gave a good geographical distribution). More specifically the CTS data sets used with respect to the evaluations reported on later in this thesis can be broadly divided into: (i) 3D data sets, (ii) 5D data sets, (iii) data sets used in the context of sampling and (iv) data sets used in the context of segmentation (the terms sampling and segmentation were introduced in Section 2.5 of Chapter 2).

The rest of this section is organised as follows. Subsection 3.4.1 describe the CTS data construction. Subsection 3.4.2 introduces the 3D CTS data sets, while Subsection 3.4.3, presents the 5D CTS data sets. Subsection 3.4.4 then, considers the GB cattle movement CTS data sets used in the context of sampling and Subsection 3.4.5 the CTS data sets used in the context of segmentation.

3.4.1 CTS Data Set Construction

The CTS database describes individual cattle movement, but typically cattles are moved in batches therefore records describing cattle movement that occur on the same day, with respect the same sender and receiver location and the same breed of the same cattle were grouped together. To this end additional attribute the number of cattle moved was added. Thus each record described the movement of a number of animals of the

same breed and gender, on the same day from specific sender location to specific receiver location. The receiver and sender location attributes referenced using the eastings and northings coordinate system used by the Ordinance Survey of Great Britain (OSGB). Thus as noted above, the data could be conveniently referenced to five dimensions: (i) records (features the sender location,receiver location and the animal moved), (ii) attributes (individual attributes in the records), (iii) sender easting (x- coordinate holding area), (iv) sender northing (y- coordinate holding area) and (v) time. The attribute values were discretised/ normalised using the LUCS-KDD-DN Software [29][2]; this was originally intended to convert data files available in the UCI data repository [29] into a binary format suitable for use in Association Rule Mining (ARM) algorithms but is equally applicable in the context of banded pattern mining.

The LUCS-KDD-DN software operate by considering the column attributes as either numeric or nominal as follows:

(a) **Numeric Attributes**

1. Divide range of attributes into N discrete sub-ranges where the range is less than or equal to 100.
2. For each sub-range,the number of records are counted with respect to the available classes.
3. For each divisions dominant classes are determined with reference to the nearest neighbouring classes.
4. Identical sub-ranges with dominant classes are then combined to form a set of divisions.
5. The number of divisions can either be merged by combining the probability of the resulting dominant classes or selecting the highest probabilty.

(a) **Nominal Attributes**

1. Divide range of attributes into N discrete divisions.
2. Count the number of records that falls into this division with respect to the available classes.
3. Dominant classes for each divisions are identified if they exist otherwise they are removed.
4. Merging the divisions with the joint probability in excess of 90%.

3.4.2 3D CTS data sets

This sub-section reviews the 3D data sets extracted from the CTS database. In total 48 data sets were obtained covering the years 2003, 2004, 2005 and 2006 and divided into three equal sized groups. For the first group the dimensions were: (i) Records, (ii)

[2]The LUCS-KDD-DN software is available at *http:// www.csc.liv.ac.uk /∼frans /KDD /Software / LUCS_KDD_DN/lucs-kdd_DN.html*

Attributes and (iii) Eastings. For the second group the dimensions were: (i) Records, (ii) Attributes and (iii) Northings, whilst for the third group the dimensions were: (i) Records, (ii) Attributes and (iii) Time. The term record, as used here, describes the movement of a number of animals of the same breed and gender, on the same day, from a specific sender location to a specific receiver location. The motivation for the 3D categories was that they featured more than two dimensions, and hence could be used to assess the proposed banding mechanisms in the context of a higher number of dimensions than 2D, while still allowing for the visualisation of the outcomes. The values for the Eastings and Northings dimensions represented the easting or northing associated with the sender location (holding areas), whilst the temporal dimension respresented the months in a year. The Eastings and Northings dimensions were discretised into ten sub-ranges, whilst the temporal dimension was discretised into twelve sub-ranges using EWD. The attribute dimension comprised the following individual attributes: (i) animal gender, (ii) animal age, (iii) cattle-beef, (iv) cattle-dairy, (v) sender location in terms of Eastings and Northings, (vi) sender location type, (vii) receiver location type, (viii) receiver location in terms of Eastings and Northings and (ix) the number of cattle moved where appropriate the attribute values were discretised/normalised in the same manner as described for the UCI data sets above using Equal Size Discretisation with a maximum of five ranges (using the LUCS-KDD_DN Software). In the case of the cattle gender; cattle-beef and cattle-dairy attributes only two attributes were required: male/female, (yes/no) and (yes/no). Tables 3.2 presents a statistical overview of the 3D CTS data sets. With respect to the table the following should be noted:

1. The statistics for each group of 3D data sets are the same, so are presented as a single table.
2. The number of values for the attribute dimension is not constant as the four counties considered had different numbers of possible attribute values for the holding area type and cattle breed type attributes.
3. The data sets are sparser than the 2D data sets.

It should also be noted that the 3D CTS data sets did not feature multiple dots.

3.4.3 5D CTS data sets

This section introduces the 5D data sets extracted from the CTS database. In total 64 data sets were identified covering the four quarters of the years 2003, 2004, 2005 and 2006 and the four counties used previously (thus $4 \times 4 \times 4 = 64$). The dimensions were: (i) Records, (ii) Attributes, (iii) Eastings, (iv) Northings and (v) Time (in months). As before; the Eastings and Northings represented the eastings and northings associated with the sender location (holding areas), and were discretised into ten sub-ranges using EWD. The temporal dimension was divided into 3 intervals such that each interval represented a month (recall that each data set represented a quarter). The attribute

Table 3.2: Statistical summary of 3D CTS data sets

Counties	Years	# Recs.	# Atts.	# Eings. /Nings.	Num. Dots	Density (d)
Aberdeenshire	2003	178172	83	10	1,781,720	12.05
	2004	173612	83	10	1,736,120	12.05
	2005	157033	83	10	1,570,330	12.05
	2006	236206	83	10	2,362,060	12.05
Cornwall	2003	170243	86	10	1,702,430	11.63
	2004	169053	86	10	1,690,530	11.63
	2005	154569	86	10	1,545,690	11.63
	2006	167281	86	10	1,672,810	11.63
Lancashire	2003	167919	80	10	1,679,190	12.50
	2004	217566	82	10	2,175,660	12.50
	2005	157142	80	10	1,571,420	12.50
	2006	196292	80	10	1,962,920	12.50
Norfolk	2003	46977	83	10	469,770	12.05
	2004	46246	83	10	462,460	12.05
	2005	35914	83	10	359,140	12.05
	2006	45150	83	10	451,500	12.05

dimension comprised the same components as in the case of the 3D CTS data sets. As before, continuous values other than the Eastings and Northings were discretised using a maximum of five ranges (again using the LUCS-KDD_DN software). Some statistics concerning the 5D CTS data sets are presented in Tables 3.3, 3.4, 3.5 and 3.6. Note that the number of values for the Easting, Northing and Time dimensions are constant across all the data sets.

Table 3.3: Statistical summary of 5D CTS data sets for 2003

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time	Num. Dots	Density (d)
Aberdeenshire	Q1	42962	98	10	10	3	386,658	9.18
	Q2	46187	101	10	10	3	415,683	8.91
	Q3	41181	104	10	10	3	370,629	8.65
	Q4	47842	107	10	10	3	430,578	8.41
Cornwall	Q1	40501	101	10	10	3	364,506	8.91
	Q2	39626	104	10	10	3	356,634	8.65
	Q3	40226	107	10	10	3	362,034	8.41
	Q4	49890	110	10	10	3	449,010	8.18
Lancashire	Q1	34325	97	10	10	3	308,925	9.27
	Q2	40926	100	10	10	3	368,334	9.00
	Q3	45765	103	10	10	3	411,885	8.73
	Q4	47392	106	10	10	3	426,528	8.49
Norfolk	Q1	11280	98	10	10	3	101,520	9.18
	Q2	14557	101	10	10	3	131,013	8.91
	Q3	9460	104	10	10	3	85,140	8.65
	Q4	11680	107	10	10	3	105,120	8.41

TABLE 3.4: Statistical summary of 5D CTS data sets for 2004

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time	Num. Dots	Density (d)
Aberdeenshire	Q1	43900	98	10	10	3	395,100	9.18
	Q2	43221	101	10	10	3	388,989	8.91
	Q3	38429	104	10	10	3	345,861	8.65
	Q4	47995	107	10	10	3	431,955	8.41
Cornwall	Q1	40126	101	10	10	3	361,134	8.91
	Q2	38226	104	10	10	3	344,034	8.65
	Q3	38751	107	10	10	3	348,759	8.41
	Q4	51950	110	10	10	3	467,550	8.18
Lancashire	Q1	53976	97	10	10	3	485,784	9.27
	Q2	54326	100	10	10	3	488,934	9.00
	Q3	53926	103	10	10	3	485,334	8.73
	Q4	65694	106	10	10	3	591,246	8.49
Norfolk	Q1	11701	98	10	10	3	105,309	9.18
	Q2	12993	101	10	10	3	110,637	8.91
	Q3	9290	104	10	10	3	83,610	8.65
	Q4	12262	107	10	10	3	110,358	8.41

TABLE 3.5: Statistical summary of 5D CTS data sets for 2005

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time	Num. Dots	Density (d)
Aberdeenshire	Q1	41086	98	10	10	3	369,774	9.18
	Q2	41317	101	10	10	3	371,853	8.91
	Q3	30635	104	10	10	3	275,715	8.65
	Q4	43995	107	10	10	3	395,955	8.41
Cornwall	Q1	40226	101	10	10	3	362,034	8.91
	Q2	38076	104	10	10	3	342,684	8.65
	Q3	31301	107	10	10	3	281,709	8.41
	Q4	44986	110	10	10	3	404,874	8.18
Lancashire	Q1	45526	97	10	10	3	409,734	9.27
	Q2	38676	100	10	10	3	348,084	9.00
	Q3	30351	103	10	10	3	273,159	8.73
	Q4	42591	106	10	10	3	383,319	8.49
Norfolk	Q1	8557	98	10	10	3	77,013	9.18
	Q2	10549	101	10	10	3	94,941	8.91
	Q3	7066	104	10	10	3	63,594	8.65
	Q4	9742	107	10	10	3	876,78	8.41

3.4.4 GB cattle CTS data set For Sampling

Later in this thesis (Chapter 9) a number of technique are considered whereby very large data sets, data sets that cannot be held in primary storage, can be banded. The techniques considered fall into two categories according to the adopted paradigm for

Table 3.6: Statistical summary of 5D CTS data sets for 2006

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time	Num. Dots	Density (d)
Aberdeenshire	Q1	54196	98	10	10	3	487,764	9.18
	Q2	56878	101	10	10	3	511,902	8.91
	Q3	56026	104	10	10	3	504,234	8.65
	Q4	69108	107	10	10	3	621,972	8.41
Cornwall	Q1	38276	101	10	10	3	344,484	8.91
	Q2	41099	104	10	10	3	369,891	8.65
	Q3	40601	107	10	10	3	365,409	8.41
	Q4	47305	110	10	10	3	425,745	8.18
Lancashire	Q1	41176	97	10	10	3	370,584	9.27
	Q2	48601	100	10	10	3	437,409	9.00
	Q3	51151	103	10	10	3	460,035	8.73
	Q4	55362	106	10	10	3	498,258	8.49
Norfolk	Q1	9659	98	10	10	3	86,931	9.18
	Q2	13707	101	10	10	3	123,363	8.91
	Q3	8945	104	10	10	3	80,505	8.65
	Q4	12839	107	10	10	3	115,551	8.41

handling the large data sets: (i) sampling and (ii) segmentation. This subsection presents the CTS evaluation data sets used in the context of the sampling techniques considered later in this thesis (the data sets used in the context of segmentation are considered in the following subsection). When conducting sampling, as the name applies, only a sample of the available data is considered and banding performed on this sample. However, this needs to be done in the context of a reference dimension which is not included in the banding exercise. In most cases it makes sense to use the record dimension (if such a dimension exists with respect to the application of interest). In other words given a ND sample data set, banding is considered in terms of (N-1)D; the resulting banding is then imposed on the remainder of the data set; it would therefore not make sense to reorder the records in the sample. It should be noted that when one dimension is left out the remaining data space can feature multiple dots (hence the necessity for the proposed banding algorithms to also be able to operate in the context of multiple dots).

For the evaluation of the proposed sample based banding, 3D, 4D and 5D data sets were extracted from the CTS database (although in each case, for the reason noted above, banding was applied to $N-1$ dimensions). The 3D and 5D data sets were those presented above with the distinction that for the 5D data sets the time dimension was divided into 12 (one month) values. The reason this was not done previously (quarters were considered divided into 3 individual month values) was because of the resource overload that this would have entailed if the data set was considered in its entirety; hence the need for sampling (or segmentation). The dimensions for the 4D data sets comprised: (i) Records, (ii) Attributes (the same attribute set as used previously), (iii) Eastings and (iv) Northings. Sixteen 4D data sets were extracted for the years 2003,

2004, 2005 and 2006, and the four counties of: Aberdeenshire, Cornwall, Lancashire and Norfolk (the same years and counties as used with respect to the 3D and 5D data sets).

A statistical summary concerning the 3D data sets was presented previously in Table 3.2. Similar summaries are presented with respect to the 4D and 5D "sampling" data sets in Tables 3.7 and 3.8.

Table 3.7: Statistical summary of the 16 (sample) 4D CTS data sets

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	Num. Dots	Density d
Aberdeenshire	2003	178172	95	10	10	1,603,548	9.47
	2004	173612	95	10	10	1,562,508	9.47
	2005	157033	95	10	10	1,413,297	9.47
	2006	236206	95	10	10	2,125,854	9.47
Cornwall	2003	170243	98	10	10	1,532,187	9.18
	2004	169053	98	10	10	1,521,477	9.18
	2005	154569	98	10	10	1,391,121	9.18
	2006	167281	98	10	10	1,505,529	9.18
Lancashire	2003	167919	94	10	10	1,511,271	9.57
	2004	217566	94	10	10	1,958,094	9.57
	2005	157142	94	10	10	1,414,278	9.57
	2006	196292	94	10	10	1,766,628	9.57
Norfolk	2003	46977	95	10	10	422,793	9.47
	2004	46246	95	10	10	416,214	9.47
	2005	35914	95	10	10	323,226	9.47
	2006	45150	95	10	10	406,350	9.47

Table 3.8: Statistical summary of the 16 (sample) 5D CTS data sets

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time
Aberdeenshire	2003	178172	95	10	10	12
	2004	173612	95	10	10	12
	2005	157033	95	10	10	12
	2006	236206	95	10	10	12
Cornwall	2003	170243	98	10	10	12
	2004	169053	98	10	10	12
	2005	154569	98	10	10	12
	2006	167281	98	10	10	12
Lancashire	2003	167919	94	10	10	12
	2004	217566	94	10	10	12
	2005	157142	94	10	10	12
	2006	196292	94	10	10	12
Norfolk	2003	46977	95	10	10	12
	2004	46246	95	10	10	12
	2005	35914	95	10	10	12
	2006	45150	95	10	10	12

3.4.5 GB cattle CTS data set For Segmentation

This section briefly introduces the data sets used for evaluating the segmentation banding techniques proposed later in this thesis as an alternative to sampling (see above). Using segmentation local bandings are calculated according to individual segments and then combined to form a global banding for the entire data set. As in the case of the sampling technique considered above, the banding is conducted according to a reference dimension. Again, in most cases it makes sense to use the record dimension (dimension that features the details concerning the sender location, receiver location and the animal moved); provided such a dimension exists given a particular application. The data sets used for the evaluation were the same as those used with respect to the evaluation of the sampling technique considered above. Namely the 3D, 4D and 5D CTS data sets as summarised in Tables 3.2, 3.7 and 3.8.

3.5 Summary of Data

This chapter has described the data sets used for evaluating the proposed Banded Pattern Mining algorithms presented later in this thesis. The presented evaluation data sets were split over three categories: (i) randomly generated synthetic data sets, (ii) UCI data sets and (iii) the GB cattle movement CTS data sets. The first two comprised only 2D data sets. The latter was divided into four further sub-categories: (i) 3D, (ii) 5D, (iii) sampling and (iv) segmentation. In each case, where necessary the attribute values were discretised and normalised to form the desired binary (zero-one) value data sets. The next chapter describes the proposed 2D Banded Pattern Mining (2D-BPM) algorithm, the first of the BPM algorithms considered in this thesis.

Chapter 4

2D Banding Mechanism

4.1 Introduction

The chapter considers the concept of Banded Pattern Mining (BPM) in the context of 2D zero-one data, a special case of ND for the case of simplicity. Recall that the fundamental idea is to rearrange the rows and columns of a given 2D data matrix so that the dots (the non-zero entries) are located about the leading diagonal. Recall also that in Chapter 2, Previous Work, a number of alternative banding algorithms, that have been previously proposed, were described. These algorithms were also directed at identifying bandings in binary data, but tended to operate either by considering permutations or using an alternative mechanism to the banding score mechanism considered in this thesis; thus in a manner different to that presented in this thesis. The suggested significant disadvantage of these alternative banding algorithms was that they were computationally expensive. Thus, anything that can be done to address the computational overhead associated with these existing banding algorithms will be beneficial. This chapter introduces the novel concept of "banding scores" and demonstrates how this concept can be incorporated into a 2D Banded Pattern Mining algorithm, the 2D-BPM algorithm, for extracting banded patterns from 2D data.

The rest of this chapter is arranged as follows. A formalism for BPM in 2D is presented in Section 4.2. Section 4.3 then discusses the process for calculating 2D Banding Scores (BS), whilst Section 4.4 presents the process for calculating 2D Global Banding Scores (GBS); the distinction between BS and GBS will become clear later in the chapter. Section 4.5 presents the 2D-BPM algorithm, while Section 4.6 considers a worked example illustrating how the 2D-BPM algorithm operates. Section 4.7 then reports on the evaluation conducted with respect to the operation of the proposed 2D-BPM algorithm. Finally, in Section 4.8, the chapter is concluded with a brief summary of the main findings.

4.2 2D Banding Formalism

Given a 2D data set the "space" (matrix) can be conceptualised as comprising a $k_1 \times k_2$ grid where k_1 is the size of the 'x' dimension (Dim_x) and k_2 is the size of the 'y' dimension (Dim_y). We can think of Dim_x as comprising columns and Dim_y as comprising rows. With respect to the 2D data sets used for evaluation purposes later in this chapter the columns represent attributes and the rows records. Each individual grid square can then hold a 1 or a 0. However, it should be recalled that in this thesis "ones" are represented by dots and "zeroes" by empty grid squares (as illustrated in Figures 4.1(a) and 4.1(b)). Note that with respect to all 2D grids presented in this chapter the origin is always in the top left-hand corner.

Thus (in 2D), each dot can be defined by a coordinate pair $\langle x, y \rangle$ where $0 \leq x \leq k_1$ and $0 \leq y \leq k_2$. Therefore, a $2D$ data set D, can be considered to comprise a set of m dots, $D = \{d_1, d_2, ..., d_m\}$ such that each d_i is represented by a pair of coordinates $\langle i, j \rangle$ where $i \in Dim_x$ and $j \in Dim_y$. Thus in the case of the configuration given in Figure 4.1(a), we have $D = \{d_1, d_2, d_3\} = \{\langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle\}$, and with respect to Figure 4.1(b), $D = \{d_1, d_2, d_3\} = \{\langle 0, 2 \rangle, \langle 1, 1 \rangle, \langle 2, 0 \rangle\}$. The banding problem is then to rearrange the ordering of the indexes in Dim_x (the column/attribute numbers) and the indexes in Dim_y (the row/record numbers) thereby achieving a "best" banding (the expectation was that in many cases, a perfect banding would not exist; hence the objective was to find a "best" banding). Figure 4.1(a) presents an example of a perfect 2D banding, where dots are arranged along the leading diagonal, of the form we are looking for; Figure 4.1(b) presents an example of an alternative 2D banding, which could also be argued to be perfect but is not of the form we are looking for. We indicate a particular index i in Dim_x using the notation Dim_{x_i}, and a particular index j in Dim_y using the notation Dim_{y_j}.

(a) (b)

FIGURE 4.1: Examples of 2D dot configurations featuring: (a) a perfect banding and (b) an alternative banding

For the discussion regarding BS calculation presented in the next section, when considering the indexes in Dim_x we wish to only consider the Dim_y coordinates and vice versa. Thus when considering the set of dots $Dots_{x_i}$ associated with index i in Dim_x, we wish to consider only the relevant Dim_y coordinates in $Dots_{x_i}$, thus the set $C_{i_y} = \{y_1, y_2, \dots\}$. Conversely when considering the set of dots $Dots_{y_j}$ associated with

index j in Dim_y we wish to consider only the relevant Dim_x coordinates in $Dots_{y_j}$, thus the set $C_{j_x} = \{x_1, x_2, \dots\}$. It may ease understanding to note that C_{i_y}, in the context of frequent itemset mining [1, 2], is referred to as a Transaction ID list or a TID list.

Definition 4.1. Perfect Banding: A zero-one matrix A can be "perfectly banded" if there exist a permutation of columns $\{1, 2, \dots, m\}$ and rows $\{1, 2, \dots, n\}$ such that: (i) for every element in C_j, the values occur consecutively at row indexes $\{i_k, i_{k+1}, i_{k+2}, \dots\}$ and the "starting index" for C_j is less than or equal to the starting index for C_{j+1}; and (ii) for every element in R_i the values occur consecutively at column indexes $\{j_k, j_{k+1}, j_{k+2}, \dots\}$ and the starting index for R_i is less than or equal to the starting index for R_{i+1}.

4.3 2D Banding Score Calculation

This section presents a mechanism for calculating Banding Scores (BS). A simple mechanism for calculating BS, given a set of dots $Dots_{x_i}$ associated with index i in Dim_x, is given in Equation 4.1:

$$BS_{x_j} = \sum_{n=1}^{n=|C_{i_y}|} c_n \tag{4.1}$$

where C_{i_y} is the set of y-coordinates associated with $Dots_{x_i}$ ($|Dots_{x_i}| \equiv |C_{i_y}|$). Similarly, given a set of dots $Dots_{y_j}$, associated with index j in Dim_y a BS can be calculated using Equation 4.2:

$$BS_{y_j} = \sum_{n=1}^{n=|C_{j_x}|} c_n \tag{4.2}$$

where C_{j_x} is the set of x-coordinates associated with $Dots_{y_j}$ ($|Dots_{y_j}| \equiv |C_{j_x}|$).

However, we would like to normalise these banding scores so that the banding score for any column x_j or row y_i is 1. The reason being that by insisting that the banding score is limited to values between 0 and 1 comparisons can be made as to the quality of bandings produced with respect to different data sets that feature different numbers of dimensions, different dimension sizes and different number of dots. Thus:

$$BS_{x_j} = \frac{\sum_{n=1}^{n=|C_{i_y}|} c_n}{\sum_{n=1}^{n=|C_{i_y}|} k_2 - n + 1} \tag{4.3}$$

$$BS_{y_j} = \frac{\sum_{n=1}^{n=|C_{j_x}|} c_n}{\sum_{n=1}^{n=|C_{j_x}|} k_1 - n + 1} \tag{4.4}$$

With respect to the above two equations recall, from Section 4.2 above, that k_1 is the maximum size for dimension Dim_x and k_2 is the maximum size for dimension Dim_y respectively. It should also be noted that, with respect to the divisions featured in

Equations 4.3 and 4.4 that the *dividend* is the sum of the distances that the dots are from the origin while the *divisor* is the sum of the maximum distances that the dots can be from the origin (given a particular index under consideration).

Referring back to Figures 4.1(a) and 4.1(b), using Equations 4.3 and 4.4, the BS for the two configurations will be calculated as follows:

$$BS_{x1} = \frac{1}{3} = 0.33 \qquad BS_{x2} = \frac{2}{3} = 0.67 \qquad BS_{x3} = \frac{3}{3} = 1.0$$

$$BS_{y1} = \frac{1}{3} = 0.33 \qquad BS_{y2} = \frac{2}{3} = 0.67 \qquad BS_{y3} = \frac{3}{3} = 1.0$$

and

$$BS_{x1} = \frac{3}{3} = 1.0 \qquad BS_{x2} = \frac{2}{3} = 0.67 \qquad BS_{x3} = \frac{1}{3} = 0.33$$

$$BS_{y1} = \frac{3}{3} = 1.0 \qquad BS_{y2} = \frac{2}{3} = 0.67 \qquad BS_{y3} = \frac{1}{3} = 0.33$$

Thus, using the above, to achieve a best banding the BS need to be ordered, from the origin, in ascending order.

4.4 2D Global Banding Score Calculation

In Section 1.1, it was suggested that, we could now simply sum the individual banding scores to obtain an overall average global banding score for a particular configuration:

$$GBS = \frac{GBS_x + GBS_y}{2} \tag{4.5}$$

where GBS_x is the GBS for Dim_x (calculated using Equation 4.6 presented below), and GBS_y is the GBS for Dim_y (calculated using Equation 4.7 below). Note that in Equation 4.5 we divide by 2 so as to normalise the total GBS (we have two dimensions).

$$GBS_x = \frac{\sum_{j=1}^{j=k_1} BS_{x_j}}{k_1} \tag{4.6}$$

$$GBS_y = \frac{\sum_{j=1}^{j=k_2} BS_{y_j}}{k_2} \tag{4.7}$$

where the individual BS are calculated using Equations 4.3 and 4.4 as discussed in the previous section.

However, using the above, would mean that the GBS for the configuration presented in Figure 4.1(a) would be the same as that for the configuration presented in Figure 4.1(b) (which features an entirely different kind of banding). Not the desired result. Thus we need to weight the columns and rows as well. The columns and rows can be weighted as follows:

$$GBS_x = \frac{\sum_{j=1}^{j=k_1} BS_{x_j} \times (k_1 - j + 1)}{\sum_{j=1}^{j=k_1} j} \tag{4.8}$$

$$GBS_y = \frac{\sum_{j=1}^{j=k_2} BS_{y_j} \times (k_2 - j + 1)}{\sum_{j=1}^{j=k_2} j} \tag{4.9}$$

Referring back to Figures 4.1(a) and 4.1(b) the GBS for the two configurations, calculated using Equations 4.8, 4.9 and 4.5 respectively, will now be as follows:

$$GBS_x = \frac{0.33 \times (3-1+1) + 0.67 \times (3-2+1) + 1.0 \times (3-3+1)}{1+2+3} = \frac{3.33}{6} = 0.56$$

$$GBS_y = \frac{0.33 \times (3-1+1) + 0.67 \times (3-2+1) + 1.0 \times (3-3+1)}{1+2+3} = \frac{3.33}{6} = 0.56$$

$$GBS = \frac{0.56 + 0.56}{2} = 0.56$$

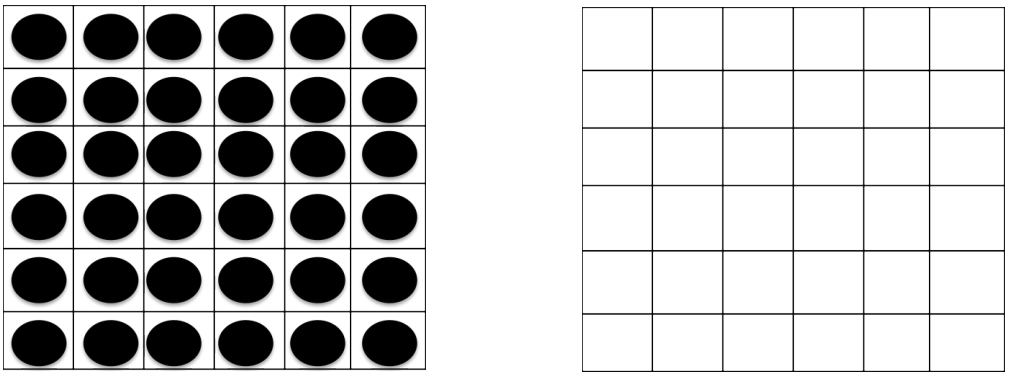
and

$$GBS_x = \frac{1.0 \times (3-1+1) + 0.67 \times (3-2+1) + 0.33 \times (3-3+1)}{1+2+3} = \frac{4.67}{6} = 0.78$$

$$GBS_y = \frac{1.0 \times (3-1+1) + 0.67 \times (3-2+1) + 0.33 \times (3-3+1)}{1+2+3} = \frac{4.67}{6} = 0.78$$

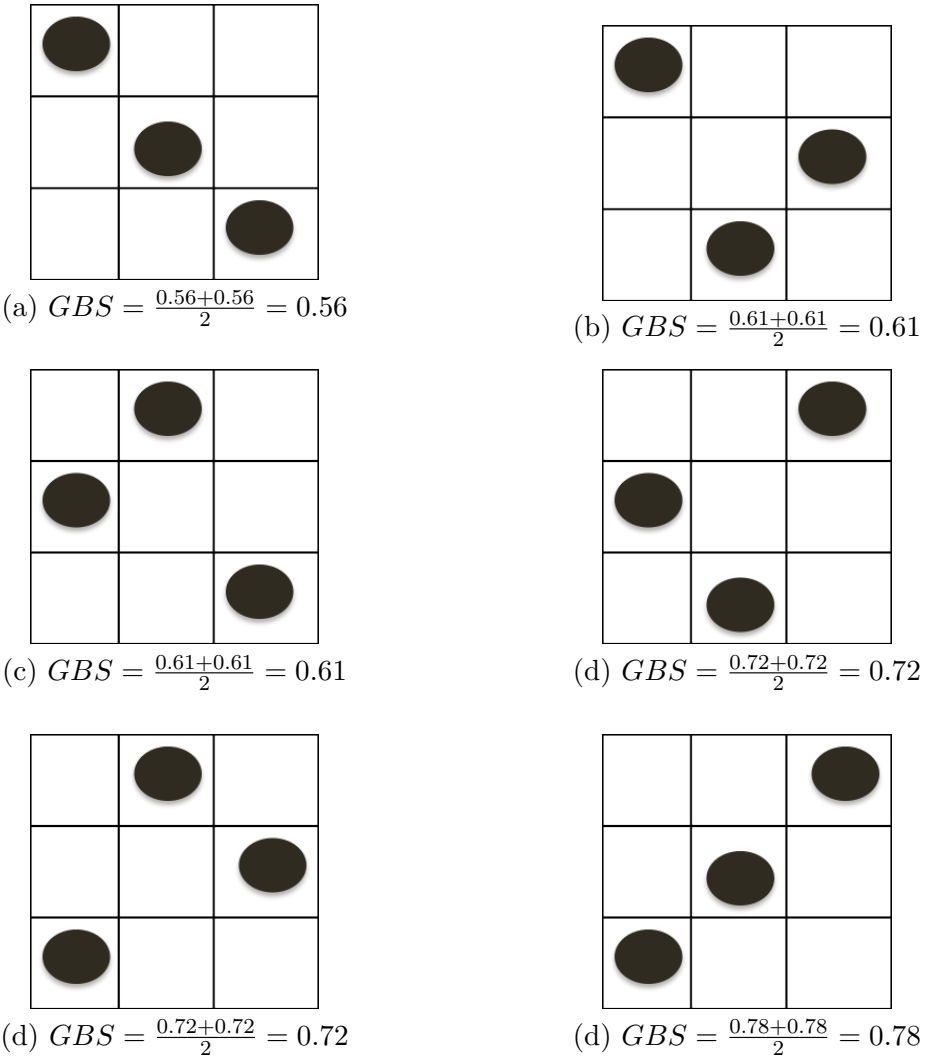
$$GBS = \frac{0.78 + 0.78}{2} = 0.78$$

From the above, it can be seen that the GBS calculation serves to distinguish between the two configurations. Note also that using these equations a best banding is achieved when the total GBS is minimised using Equations 4.8 and 4.9. With respect to the configurations shown in Figures 4.1(a) and 4.1(b) global banding scores of 0.56 and 0.78 were obtained respectively. However, it is worth noting that a global banding score of "1" will only be obtained if the entire data space is filled with "ones" (dots), and a global banding score of "0" will only be obtained if the entire data space is filled with zeros (no dots); unlikely situations in practice (Figures 4.2(a) and 4.2(b)).
Returning to the configuration given in Figure 4.1(a), this configuration can be permuted in six different ways. The different configurations are illustrated in Figure 4.3, together with their GBS. From Figure 4.3 it can clearly be seen that the GBS values serve to differentiate between the different configurations. It can also be seen that if wish to identify a "best" banding score we need to minimise the GBS value.



(a) (b)

FIGURE 4.2: Example data configurations: (a) all dots ("worst" GBS of 1) and (b) no dots ("best" GBS of 0)



(a) $GBS = \frac{0.56+0.56}{2} = 0.56$

(b) $GBS = \frac{0.61+0.61}{2} = 0.61$

(c) $GBS = \frac{0.61+0.61}{2} = 0.61$

(d) $GBS = \frac{0.72+0.72}{2} = 0.72$

(d) $GBS = \frac{0.72+0.72}{2} = 0.72$

(d) $GBS = \frac{0.78+0.78}{2} = 0.78$

FIGURE 4.3: Permutations for dot marix given in Figure 4.1

4.5 The 2D Banded Pattern Mining (2D-BPM) Algorithm

This section presents the 2D-BPM algorithm for identifying bandings in 2D data sets using the banding score concept presented above. The algorithm operates by iteratively rearranging the column and row indexes until the GBS is minimised. The pseudo code for the algorithm is presented in Algorithm 5. The inputs (Lines 1 and 2) are: (i) a value for k_1, the size of Dim_x (thus the maximum 'x' index) from which an index list Dim_x is calculated for dimension x, (ii) a value for k_2, the size of Dim_y (thus the maximum 'y' index) from which an index list Dim_y is calculated for dimension y, (iii) a dots (zero-one) matrix D measuring $k_1 \times k_2$ and (iv) a counter. The counter is used to set a maximum on the number of iterations. The output is a reordered matrix D that features a "best" banding. The algorithm proceeds in an iterative manner. Initially the GBS_{sofar} value is set to 1.0 because using the identified equations we wish to minimise the GBS value to find a "best" banding. On each iteration the algorithm sequentially rearranges the Dim_x and Dim_y indexes according to their banding scores (bs_{index}) values. The process is continued until a minimum value for GBS_{sofar} is reached or the input counter reaches 0. More specifically, on each iteration, the banding score for each index i in Dim_x is calculated (Line 10). The index list Dim_x is then rearranged in ascending order of BS_{xi} to produce Dim'_x (Line 12). The matrix D is then rearranged accordingly to give D' (Line 13) and a GBS value for the x-dimension calculated (GBS'_x), using Equation 4.8 (Line 14). The same process is then followed for Dim_y so as to produce Dim'_y (Lines 15 to 18). The matrix D' is rearranged to give D'' (Line 19) and a GBS value calculated for the y-dimension ($GBS'y$) using Equation 4.9 (Line 20). A new global banding score is then calculated using GBS'_x and GBS'_y to give GBS_{new} (Line 21). If GBS_{new} is greater than or equal to the previously recorded GBS value the loop is exited (Lines 22 to 24) and we return D and the GBS value (this will not be the case on the first iteration so the algorithm will always iterate at least twice). Otherwise the counter is decremented, and we assign D'' to D, Dim'_x to Dim_x, Dim'_y to Dim_y and GBS_{new} to GBS_{sofar} (Line 25) and repeat. On the start of each iteration the counter is tested, if it has reached zero, the loop is exited (Lines 6 to 8). Although not shown in Algorithm 5 the implementation of the algorithm is such that the loop is also exited if no changes (index rearrangements) are made.

With respect to parallising the 2D-BPM algorithm presented in this section, the idea might be to divide the problem into a set of of sub-problems that can be solved concurrently, where each sub-problems can be assigned to a processing element and consequently the sub-processes can be executed simultenously. A Hadoop map reduce framework (or similar) can be adopted to process large data sets, splitting them into subsets and processing each subset on a different processor and combining the results obtained. In the future work section presented at the end of this thesis, it is suggested that a fruitful avenue for further work is the parallelisation of the proposed BPM algorithms.

Algorithm 5: The 2D-BPM Algorithm

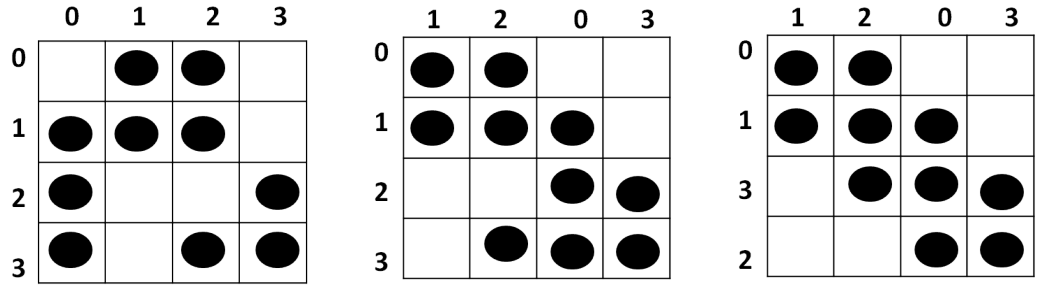
Input: k_1 ($Dim_x = \{0, 1, \ldots, k_1\}$), k_2 ($Dim_y = \{0, 1, \ldots, k_2\}$)
D, a dots 2D data set subscribing to Dim_x and Dim_y, counter
Output: the matrix D rearranged so that the columns and rows serve to minimize GBS
$GBS_{sofar} = 1.0$
loop
  if ($counter == 0$) **then**
    break
  end if
  for all $index \in Dim_x$ **do**
    bs_{x_i} = Banding score for current index using Equation 4.3
  end for
  Dim'_x = Rearranged Dim_x in ascending order according to bs_{index} for Dim_x
  $D' = D$ rearranged according Dim'_x
  GBS_x = Global banding score for Dim'_x using Equation 4.8
  for all $index \in Dim_y$ **do**
    bs_{index} = Banding score for current index using Equation 4.4
  end for
  Dim'_y = Rearranged Dim_y in ascending order according to bs_{index} for Dim_y
  $D'' = D'$ rearranged according Dim'_y
  GBS_y = Global banding score for Dim'_y using Equation 4.9
  GBS_{new} = Overall Global banding score using Equation 4.5
  if ($GBS_{new} \geq GBS_{sofar}$) **then**
    $break$
  else
    $GBS_{sofar} = GBS_{new}$, $Dim_x = Dim'_x$, $Dim_y = Dim'_y$, $D = D''$
  end if
  $counter = counter - 1$
end loop
Exit with D and GBS

4.6 A Working Example using the 2D-BPM Algorithm

To assist in the understanding of the operation of the 2D-BPM algorithm as presented above this section presents a working example. Let us assume a 2D dot matrix measuring 4×4 and configured as shown in Figure 4.4(a) (recall that the origin is in the top left hand corner). Thus $k_1 = 4$ and $k_2 = 4$ and:

$$D = \{\langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,4\rangle, \langle 4,3\rangle, \langle 4,4\rangle\}.$$

As noted above the 2D-BPM algorithm commences by considering the x-dimension first, the calculated banding scores are shown in Table 4.1; the sequence of banding scores is $BS_x = \{1.00, 0.43, 0.78, 1.00\}$. We thus rearrange the indexes in Dim_x in ascending order of BS. Note that (not shown in Algorithm 5) in the case where two or more elements have the same score the ordering is conducted so that the index associated with the largest number of dots is nearest to the centre of the data space, and so on. The result is as shown in Figure 4.4(b).



(a) Raw Data (b) After rearrangement of Dim_x (c) After rearrangement of Dim_y

FIGURE 4.4: Example of the operation of the 2D-BPM algorithm

TABLE 4.1: Calculation of BS values for Dim_x

Index	Dist from origin	Max. dist. from origin	*bs*
1	$2+3+4=9$	$2+3+4=9$	1.00
2	$1+2=3$	$3+4=7$	0.43
3	$1+2+4=7$	$2+3+4=9$	0.78
4	$3+4=7$	$3+4=7$	1.00
		Total	3.21

Considering dimension y next, the BS are calculated as shown in Table 4.2. This produced the set of banding scores $BS_y = \{0.43, 0.67, 1.00, 1.00\}$. Thus in this case the indexes in Dim_y are more or less already arranged in ascending order of BS. We only need to swap the last two indexes so that the element with the greater number of dots is nearer the centre of the data space. The result is as shown in Figure 4.4(c). The GBS_x and GBS_y values are then calculated as follows:

TABLE 4.2: Calculation of BS values for Dim_y

Index	Dist from origin	Max. dist. from origin	*bs*
1	$1+2=3$	$3+4=7$	0.43
2	$1+2+3=6$	$2+3+4=9$	0.67
3	$3+4=7$	$3+4=7$	1.00
4	$2+3+4=9$	$2+3+4=9$	1.00
		Total	3.10

$$GBS_x = \frac{(1.0 \times 4) + (0.43 \times 3) + (0.78 \times 2) + (1.0 \times 1)}{1+2+3+4} = \frac{7.85}{10} = 0.79$$

$$GBS_y = \frac{(0.43 \times 4) + (0.67 \times 3) + (1.0 \times 2) + (1.0 \times 1)}{1+2+3+4} = \frac{6.73}{10} = 0.67$$

The overall global banding score (GBS) is:

$$GBS = \frac{0.79 + 0.67}{2} = 0.73 \tag{4.10}$$

The process is repeated on the next iteration. However in this case the same overall GBS value is produced (indicating that a best banding has already been arrived at). The rearranged dot matrix is as follows (Figure 4.4(c)):

$$D' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,4\rangle, \langle 4,3\rangle, \langle 4,4\rangle\}.$$

4.7 Evaluation of the 2D-BPM Algorithm

This section reports on the evaluation of the proposed 2D-BPM algorithm. The evaluation was conducted using the data sets described in Chapter 3. The objectives of the evaluation were as follows:

1. **Number of Iterations:** To analyse the operation of the 2D-BPM algorithm in terms of the number of iterations required to arrive at a banding.
2. **Efficiency using synthetic data:** To compare the efficiency of the 2D-BPM algorithm with the established BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms, in terms of the size of the data sets (number of rows and columns, and density), using synthetic 2D data sets.
3. **Efficiency using UCI data:** To compare the operation of the 2D-BPM algorithm with the established BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms, in terms of efficiency using 2D data sets taken from the UCI machine learning repository.
4. **Quality of bandings using UCI data:** To compare the operation of the 2D-BPM algorithm with the established BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms, in terms of the quality of the bandings produced using 2D data sets taken from the UCI machine learning repository.
5. **Frequent Itemset Mining:** To illustrate the advantages that can be gained using banding with respect to a standard dot (zero-one) algorithm, namely Frequent Item Set Mining (FIM).

.

Each of these objectives are considered in the following five subsections (subsections 4.7.1 to 4.7.5). All the proposed BPM algorithms were implemented using the JAVA programminig language. All the reported experiments were conducted using a 2.7 GHz Intel Core i5 with 16 GB 1333 MHz DDR3 memory, running OS X 10.8.5 (12F45).

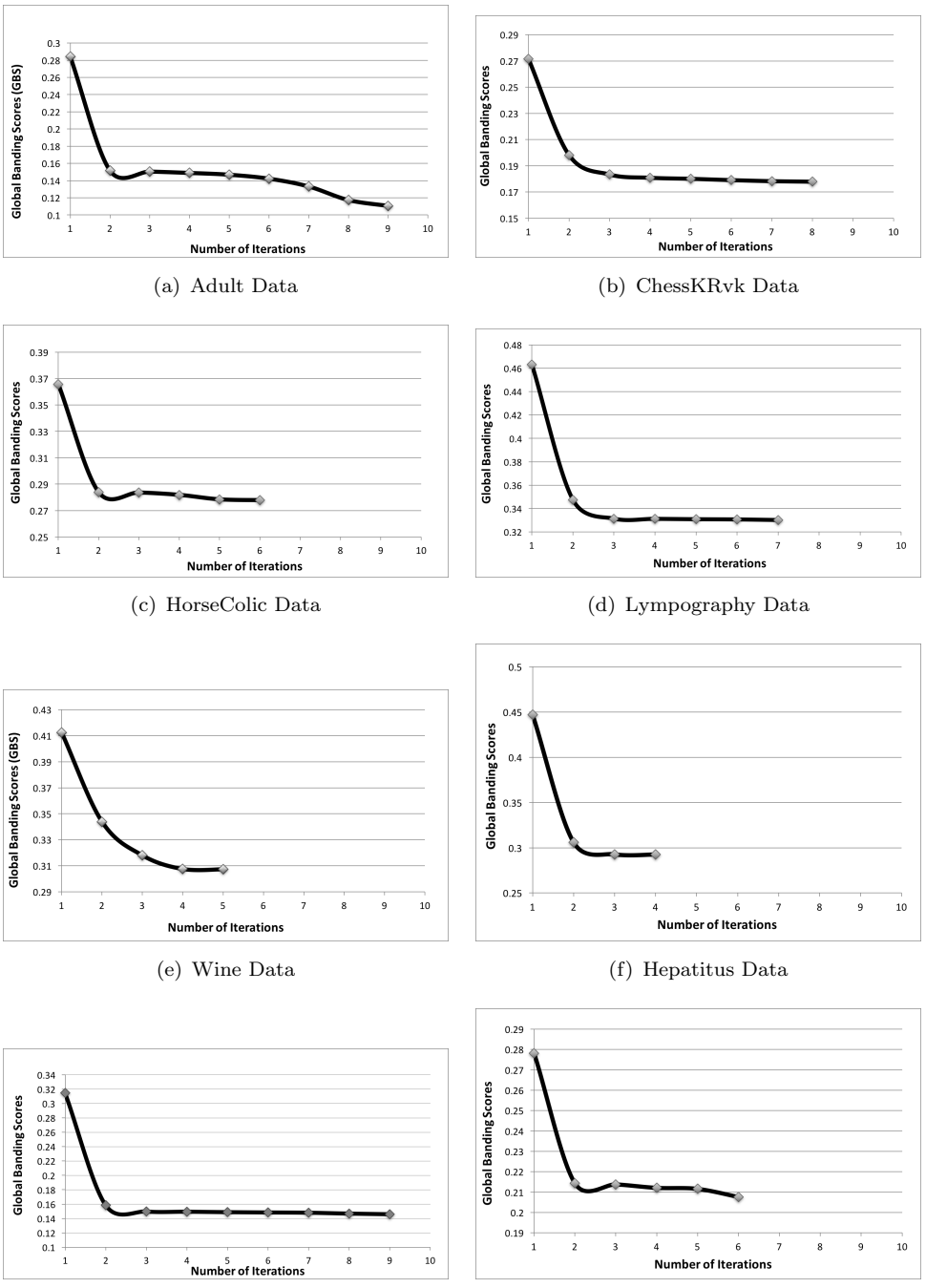
4.7.1 Analysis of 2D-BPM algorithm in terms of number of iterations

To determine the nature of the operation of the 2D-BPM algorithm, in terms of the number of iterations required to arrive at a "best" banding, a sequence of experiments was conducted using the selected UCI data sets introduced in Chapter 3. In each case, on each iteration, the *GBS* value was recorded; a maximum number of iterations counter value of 10 was used. Note that with respect to the competing approaches the generate and test mode of the operation of these systems does not feature iteration. The significance of the experiments considered in this subsection was to demonstrate how the 2D-BPM algorithm progresses over the iterations. The results for eight of the UCI data sets are shown in the plots given in Figure 4.5 where the x axis represents the number of iterations and the y axis the GBS values (plots for the remaining four data sets are given in Appendix D). From the graphs it can be seen, as expected, that GBS values improve (approach 0) as the 2D-BPM algorithm progresses. Closer inspection of the figure indicates that the gain in GBS shows that significant improvement occurs in the first few iterations, between the first two. It can also be seen that the 2D-BPM algorithm always stops before the counter decreases from 10 to 0 (the maximum number of permitted iterations), this is because a best GBS has been found prior to the counter reaching 0. Note that similar results were also obtained for the remaining four UCI data sets, although the associated plots have not been included here because of their similarity to those shown in Figure 4.5 (they are given in Appendix D). Given the results obtained it was concluded that the most appropriate counter value was 10; and consequently this value was used with respect to the remainder of the experiments reported on in this thesis.

4.7.2 Efficiency of 2D-BPM Algorithm using Synthetic Data sets

This subsection presents the results obtained from the comparative analysis of the operation of the proposed 2D-BPM algorithm with respect to the BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms using synthetic data sets of varying size and density. For the experiments synthetic data sets were used because this allowed for the specification of parameters. More specifically the data sets were generated using the LUCS-KDD data generator described in Section 3.2 of Chapter 3 [29]. Note that the generated data sets featured equal numbers of rows and columns; the reason being that this was a convenient way of generating data sets that incrementally included more dots. Two sets of experiments were conducted:

1. **Matrix Size:** Experiments using a sequence of ten randomly generated synthetic data sets of increasing numbers of cells from approximately 10,000 to 100,000 increasing in steps of approximately 10,000. Approximate because using the LUCS-KDD data generator described, the number of cells could only be specified in terms of number of rows and columns. More specifically dot matrices of the following sizes were generated: (i) 100×100, (ii) 141×141, (iii) 173×173, (iv) 200×200,



(a) Adult Data

(b) ChessKRvk Data

(c) HorseColic Data

(d) Lympography Data

(e) Wine Data

(f) Hepatitus Data

(g) Annealing Data

(h) PenDigits Data

FIGURE 4.5: GBS value per number of iterations obtained using eight of the UCI data sets and the 2D-BPM algorithm

(v) 224×224, (vi) 245×245, (vii) 265×265, (viii) 283×283, (ix) 300×300 and (x) 316×316. A dot density of 10% was used (in other words, on average 10% of the cells in each row/column contained a dot).

2. **Density:** Experiments using a sequence of five randomly generated synthetic data of increasing dot density from 10% to 50% increasing in steps of 10%. A data matrix of size 100×100 was generated in each case.

Figure 4.6 shows the runtime results obtained in the context of dot matrices of increasing size. In the figure the x-axis represents data set size, and the y-axis the recorded run-times (seconds). In the figure the colour coding was used simply for ease of comparison, it has no other significance. From the figure, as expected, it can be seen that there is a clear correlation between data set size and run-time; as the data set size increased the processing time also increased with respect to all the algorithms considered. However what is significant with respect to the figure, is that the proposed 2D-BPM algorithm out performed all the other algorithms because it obviates the need for the generation of large numbers of permutations. Note that with respect to Figure 4.6, it is noteworthy that the 2D-BPM algorithm required less processing time. The reason being, as noted aboved, that 2D-BPM did not need to generate large numbers of permutations.

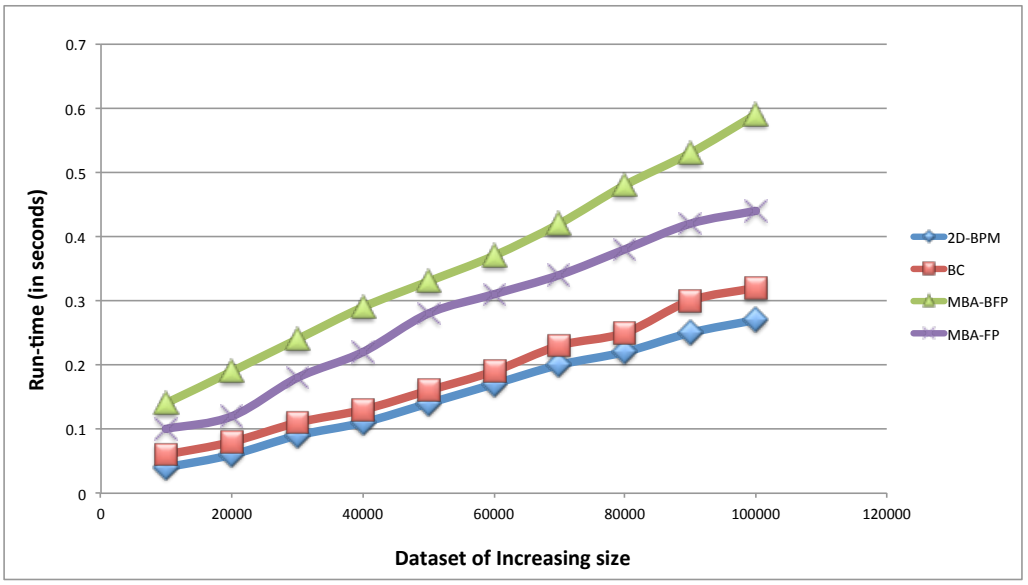


FIGURE 4.6: Recorded run time (seconds) using the 2D-BPM, BC, MBA_{BFP} and MBA_{FP} algorithms and a range of data sets of increasing size (10,000 to 100,000 in steps of 10,000)

Figure 4.7 shows the runtime results obtained in the context of dot matrices of increasing density. In the figure the x-axis represents density, while the y-axis records run-time (seconds). From the figure it can be seen, again as expected, that the run time increased with dot density. The figure also again demonstrates that the proposed 2D-BPM algorithm is faster than the comparator algorithms considered.

For completeness Table 4.3 shows the GBS values obtained (a more detailed comparative study of the effectiveness of the proposed 2D-BPM algorithm is given in Sub-section 4.7.4). The table also gives: (i) the number of columns (attributes values after discretisation) for each dataset, (ii) the number of rows (records) for each data set, and (iii) the approximate total number of cells. GBS values are shown before any banding has taken place, for the proposed 2D-BPM algorithm, and for the three comparator algorithms considered (for each data set the best recorded GBS value is given in bold font). From

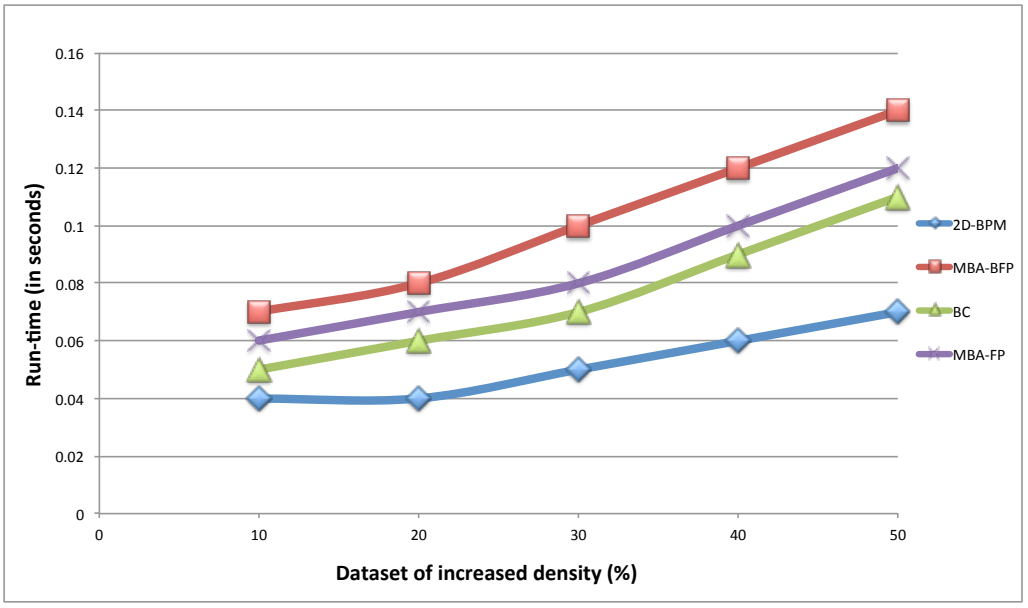


FIGURE 4.7: Recorded run time (seconds) using the 2D-BPM, BC, MBA_{BFP} and MBA_{FP} algorithms and a range of data sets of increasing density (10% to 50% increasing in steps of 10%)

the table, it can be seen that the proposed 2D-BPM algorithm performed well compared to the other established algorithms, in terms of GBS, regardless of the size of the data matrix considered.

TABLE 4.3: GBS results obtained using the 2D-BPM algorithm and the comparator algorithms for a range of dot matrices of increasing size

Data sets	# Rows	# Cols.	Apprx # Cells	GBS				
				Bef. Band.	2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Syn1	100	100	10,000	0.5275	**0.4470**	0.4776	0.5059	0.4880
Syn2	141	141	20,000	0.5044	**0.4609**	0.4718	0.5183	0.4929
Syn3	173	173	30,000	0.5328	**0.4644**	0.4852	0.5066	0.5075
Syn4	200	200	40,000	0.5252	**0.4798**	0.4922	0.5099	0.5019
Syn5	224	224	50,000	0.5097	**0.4777**	0.4851	0.5040	0.5091
Syn6	245	245	60,000	0.5258	**0.4775**	0.4899	0.5013	0.5034
Syn7	265	265	70,000	0.5133	**0.4753**	0.4841	0.5143	0.5153
Syn8	283	283	80,000	0.5236	**0.4783**	0.4925	0.5081	0.5075
Syn9	300	300	90,000	0.5231	**0.4850**	0.4976	0.5055	0.5071
Syn10	316	316	100,000	0.5285	**0.4854**	0.4968	0.5081	0.5129
Average	225	225	55,000	0.5214	**0.4731**	0.4873	0.5082	0.5046

4.7.3 Run-time Comparison Using UCI Data sets

In Subsection 4.7.2, runtime comparisons were presented using synthetic data sets. In this section runtime comparison are presented using the UCI data sets also used in

Subsection 4.7.4 below where the quality of the bandings produced are considered. Again the operation of the proposed 2D-BPM algorithm is compared with respect to the BC, MBA_{BFP} and MBA_{FP} algorithms. Table 4.4 shows the runtime results obtained (best results in bold font). For convenience the table also records: the number of records for each data sets and the number of attributes (after discretisation). From the table, it can be observed (as before) that there is a clear correlation between the number of records in the data sets and run time, as the number of records increases there is a corresponding increase in the processing time required. Whatever the case the table also clearly demonstrates that the proposed 2D-BPM algorithm requires less processing time to identify bandings than the three alternative banding algorithms considered. The worst recorded run time was obtained using the MBA_{BFP} algorithm. These results corroborate the results presented earlier in Subsection 4.7.2 above.

Table 4.4: Run-time (RT) Results (seconds) Using UCI data sets.

Data sets	# Rows	# Cols	runtime (secs)			
			2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Lympography	148	59	**0.01**	0.08	0.08	0.06
Hepatitis	155	56	**0.01**	0.08	0.06	0.06
Wine	178	68	**0.01**	0.09	0.09	0.06
Heart	303	52	**0.02**	0.08	0.12	0.11
HorseColic	368	85	**0.02**	0.09	0.20	0.12
Annealing	898	73	**0.05**	0.22	0.26	0.20
Mushroom	8124	90	**02.24**	08.47	09.07	08.14
Waveform	5000	101	**0.88**	02.28	03.05	02.41
PenDigits	10992	89	**02.81**	10.12	12.94	11.85
LetRecognition	20000	106	**10.28**	26.38	24.54	21.31
ChessKRvK	28056	58	**11.46**	23.27	27.90	27.81
Adult	48842	97	**76.74**	175.84	185.95	140.95
Average	10255	78	**08.71**	20.58	22.02	17.76

4.7.4 Banding Quality of 2D-BPM algorithm Using UCI Data sets

This section presents the results of the comparative analysis of the proposed 2D-BPM algorithm and the BC, MBA_{BFP} and MBA_{FP} algorithms with respect to the quality of the bandings produced using the UCI data sets. In Table 4.3 GBS comparisons regarding the quality of bandings produced with respect to a range of dot matrices of increasing size was presented. However, the comparison was conducted in terms of GBS values, the metric that the 2D-BPM algorithm seeks to minimise. It can be argued that using GBS favours the 2D-BPM algorithm, the BC, MBA_{BFP} and MBA_{FP} algorithms were not intended to operate using GBS. The BC algorithm seeks to maximize the MRM value (see Sub-section 2.3.1), while the MBA_{BFP} and MBA_{FP} and algorithms seek to maximise

Accuracy (see Sub-section 2.3.1.2). Thus comparisons were conducted using all three measures and the independent ABW measure presented in Section 2.6 of Chapter 2.

The results in terms of GBS, MRM, Acc and ABW are presented in Tables 4.5, 4.6, 4.7 and 4.8 respectively. In the tables best results, with respect to each data set, are presented in bold font. Note also that the data sets are listed according to number of rows. From Table 4.5, it can be seen that in terms of GBS, the proposed 2D-BPM algorithm produces the best results in all cases. As noted above, it can be argued that this is to be expected as the other algorithms are not directed at minimising GBS. In terms of *MRM* (Table 4.6), the BC algorithm produces best results in only 4 out of the 12 cases, the 2D-BPM algorithm produced the best result with respect to all the remaining cases. With respect to *Accuracy* (Table 4.7), the MBA_{BFP} algorithm performed well in only 5 out of the 12 cases, the 2D-BPM algorithm produced the best in the remaining seven case. It is interesting to note with respect to Tables 4.6 and 4.7 that the union of the data sets for which BC produced the best performance and MBA_{BFP} produced the best performance was the mushroom data set. It seems to be the case that BC algorithm works well with respect to a different subset of the data sets than the MBA_{BFP} algorithm.

The most interesting results are those produced using the independent *ABW* measure (Table 4.8), where the proposed 2D-BPM algorithm produces the best banding in every case. Note that the "before" banding results were worst in all cases indicating that the application of banding has made a difference. It should also be noted that the MBA_{FP} algorithm did not produce any best results.

TABLE 4.5: Quality of banding in terms of GBS using 2D UCI data set (best results presented in bold font).

Data sets	# Rows	# Cols	GBS				
			Bef. Band.	2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Lymphograph	148	59	0.4581	**0.2487**	0.4005	0.4540	0.4359
Hepatitis	155	56	0.4619	**0.2063**	0.3997	0.4279	0.4240
Wine	178	68	0.4564	**0.2785**	0.3965	0.4015	0.3970
Heart	303	52	0.4318	**0.1502**	0.4005	0.2833	0.3387
HorseColic	368	85	0.3857	**0.2367**	0.3702	0.3760	0.3801
Annealing	898	73	0.4133	**0.1218**	0.3448	0.3162	0.3300
Mushroom	8124	90	0.3473	**0.1774**	0.2977	0.3018	0.3284
Waveform	5000	101	0.3402	**0.2091**	0.2904	0.3215	0.2958
PenDigits	10992	89	0.3453	**0.2064**	0.2651	0.2874	0.2775
LetRecog.	20000	106	0.3325	**0.1682**	0.2561	0.2632	0.2751
ChessKRvK	28056	58	0.3473	**0.1791**	0.2629	0.2832	0.3699
Adult	48842	97	0.3662	**0.1294**	0.2738	0.2539	0.2869

TABLE 4.6: Quality of banding in terms of MRM using 2D UCI data set (best results presented in bold font).

Data sets	# Rows	# Cols	MRM				
			Bef. Band.	2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Lympho.	148	59	90.05	**94.36**	91.10	93.58	93.36
Hepatitis	155	56	86.55	**109.26**	94.84	101.44	102.45
Wine	178	68	101.59	109.27	**111.71**	105.77	108.08
Heart	303	52	205.05	**224.09**	215.29	215.35	214.30
HorseC.	368	85	201.34	231.38	**241.83**	219.26	213.30
Anneal.	898	73	450.64	**556.46**	540.44	541.46	540.71
Mushrm.	8124	90	4634.61	5098.44	**5191.25**	5004.24	4713.75
Wavefm.	5000	101	3074.02	**3189.91**	3173.97	3189.53	3189.76
PenDigit.	10992	89	6632.27	**6967.84**	6773.69	6634.89	6634.68
LetRecog.	20000	106	11730.43	**13598.88**	12445.07	13076.70	13162.99
ChessKR	28056	58	15626.24	18826.96	**18863.96**	18853.93	16345.88
Adult	48842	97	25507.77	**32869.56**	28156.61	32852.85	32842.56

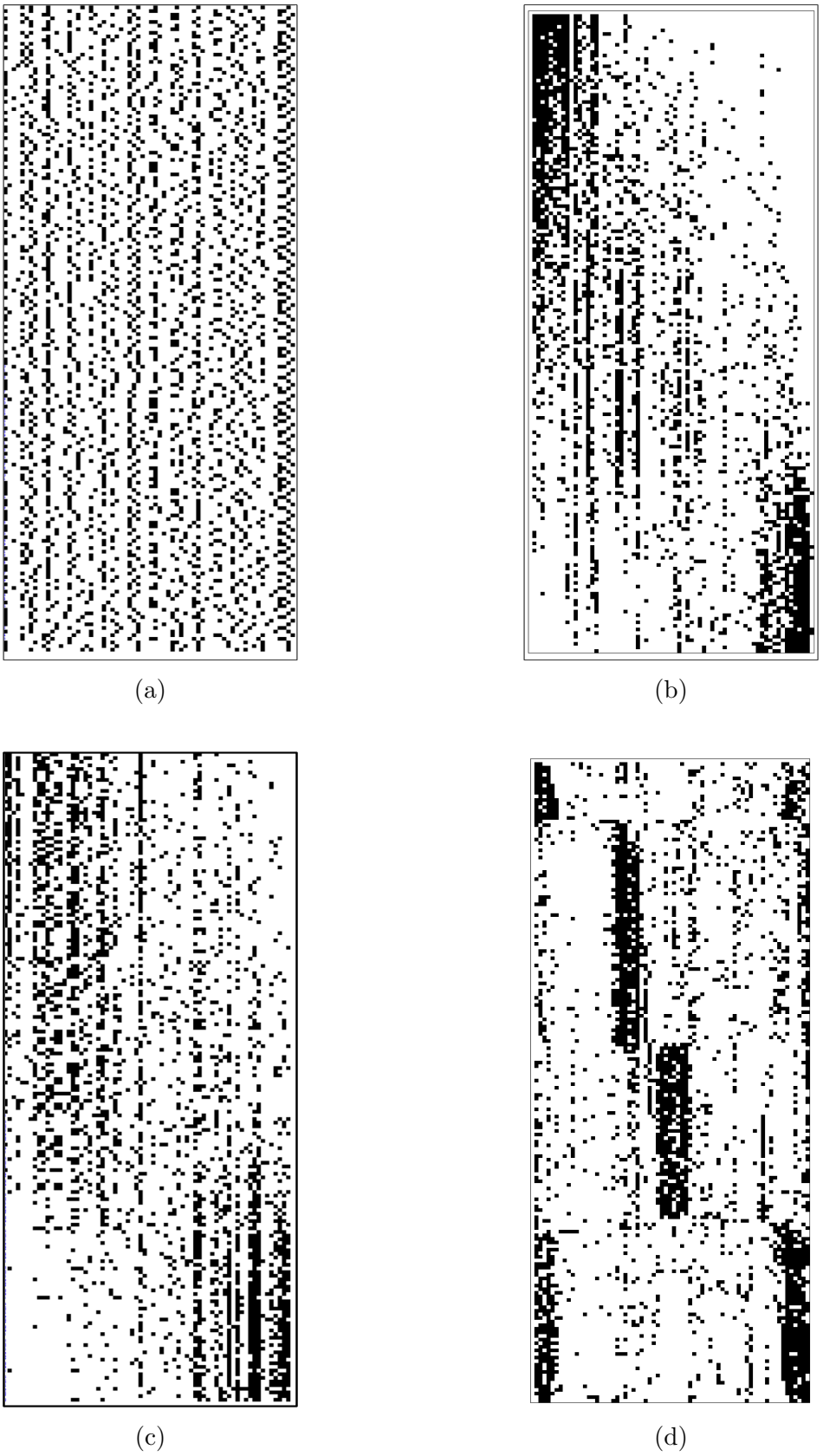
TABLE 4.7: Quality of banding in terms of Accuracy using 2D UCI data set (best results presented in bold font).

Data sets	# Rows.	# Cols.	Accuracy				
			Bef. Band.	2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Lymphograph	148	59	31.72	73.661	73.838	**73.887**	72.998
Hepatitis	155	56	30.02	78.701	74.631	**78.928**	79.677
Wine	178	68	49.33	**70.046**	68.700	68.565	69.070
Heart	303	52	46.86	**76.777**	71.480	74.759	75.389
HorseColic	368	85	46.01	**68.222**	66.933	66.997	67.076
Annealing	898	73	47.39	**80.772**	77.692	77.359	79.025
Mushroom	8124	90	40.28	68.211	69.123	**69.173**	63.119
Waveform	5000	101	49.76	**66.395**	59.875	66.270	61.588
PenDigits	10992	89	48.82	70.636	70.539	**71.802**	60.236
LetRecog.	20000	106	48.21	**73.313**	69.539	67.557	56.597
ChessKRvK	28056	58	46.96	**76.713**	72.121	67.275	65.722
Adult	48842	97	48.22	63.434	62.987	**64.099**	54.102

TABLE 4.8: Quality of banding in terms of ABW using 2D UCI data sets (best results presented in bold font).

Data sets	# Rows.	# Cols.	ABW				
			Bef. Band.	2D-BPM	BC	MBA_{BFP}	MBA_{FP}
Lympography	148	59	0.3356	**0.2804**	0.3324	0.2826	0.2887
Hepatitis	155	56	0.4438	**0.2957**	0.3438	0.2962	0.3032
Wine	178	68	0.4430	**0.2027**	0.3384	0.3061	0.3645
Heart	303	52	0.4346	**0.3016**	0.3423	0.3338	0.4142
HorseColic	368	85	0.4009	**0.3205**	0.3353	0.3881	0.4001
Annealing	898	73	0.4433	**0.3630**	0.3826	0.3779	0.4389
Mushroom	8124	90	0.4297	**0.2638**	0.3297	0.3845	0.3866
Waveform	5000	101	0.4372	**0.2414**	0.2833	0.2951	0.3774
PenDigits	10992	89	0.4276	**0.2197**	0.3276	0.2872	0.3318
LetRecog.	20000	106	0.4125	**0.2885**	0.3246	0.3152	0.3407
ChessKRvK	28056	58	0.4444	**0.2208**	0.3240	0.3246	0.3816
Adult	48842	97	0.4487	**0.3318**	0.3394	0.3617	0.4116

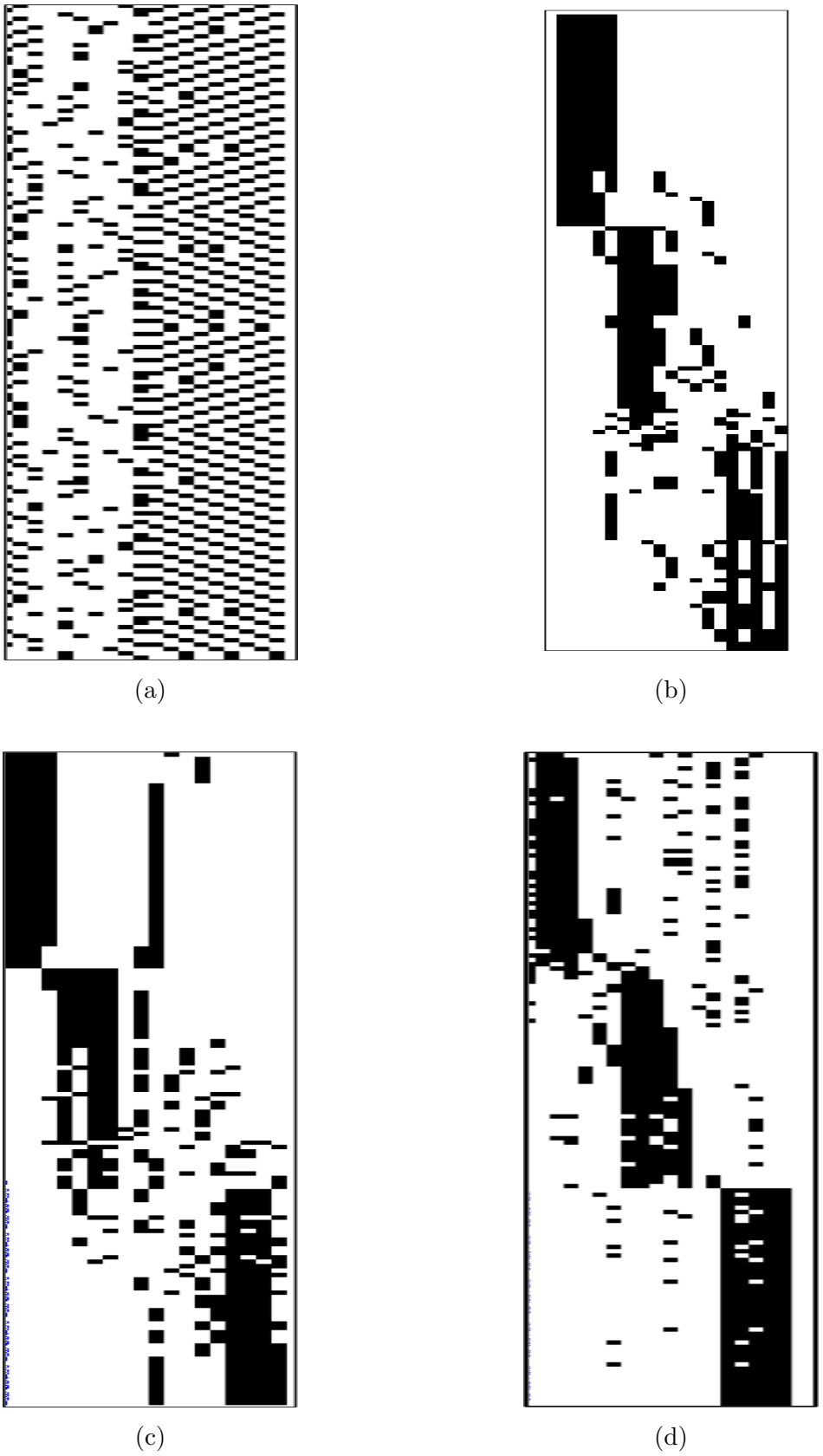
To enhance the appreciation of the results presented in Tables 4.5 to 4.8. Figures 4.8 to 4.10 show the dot matrices for the Wine, Iris and Glass UCI data sets before banding and after applying banding using the 2D-BPM, BC and MBA_{BFP} algorithms (dot matrices generated using the MBA_{FP} algorithm are not shown because this did not produce any "best" bandings). Inspection of these figures indicates that clear bandings can be identified in all cases. However, from further inspection of the figures it is suggested that the bandings produced using the proposed 2D-BPM algorithm are better. For example considering the bandings produced when the BC algorithm is applied to the Wine, Iris and Glass data sets (Figures 4.8(c), 4.9(c) and 4.10(c)) the banding is less dense than in the case of that produced using the 2D-BPM algorithm. Similarly, when the MBA_{BFP} algorithm is applied to the Wine, Iris and Glass data sets (Figures 4.8(d), 4.9(d) and 4.10(d)), the resulting banding includes dots (1s) in the top-right and bottom-left corners, while the 2D-BPM algorithm does not (it features a smaller bandwidth). It is therefore argued that the proposed GBS measure is a more effective measure for banding quality.



(a) (b)

(c) (d)

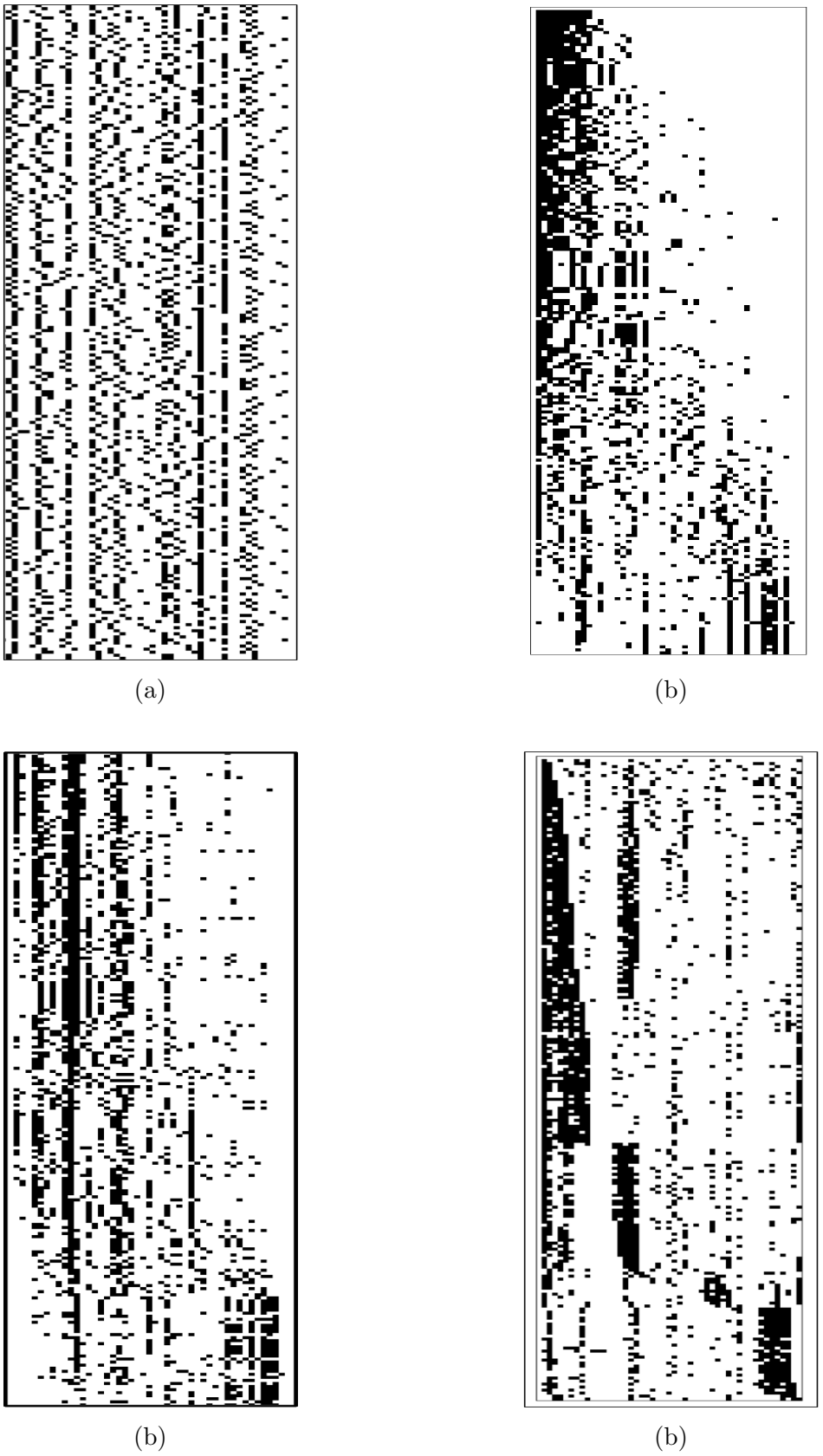
FIGURE 4.8: Wine raw data set: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm, (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA_{BFP} algorithm



(a) (b)

(c) (d)

FIGURE 4.9: Iris raw dataset: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm, (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA_{BFP} algorithm



(a) (b)

(b) (b)

FIGURE 4.10: Glass raw dataset: (a) Before banding, (b) Banding resulting from 2D-BPM algorithm (c) Banding resulting from BC algorithm and (d) Banding resulting from MBA_{BFP} algorithm

4.7.5 Effectiveness of Banding with respect to Frequent Item-set Mining (FIM)

As noted in Section 4.7 of this thesis, it is conjectured that banding has benefits in terms of enhancing the efficiency of some algorithms that use matrices or tabular zero-one data, in addition to being an indicator of some pattern that may exist in zero-one data. One example is Frequent Item-set Mining (FIM) [1, 2] where large binary valued data collections, stored in the form of sets of feature vectors (drawn from a vector space model of the data) are processed. Another example where banding may have benefits is with respect to algorithms that uses $n \times n$ affinity matrices, such as in the case of spectral clustering algorithms [126], to identify communities in networks (where n is the number of network nodes). This section presents the results from an experimental analysis conducted to determine the advantages that can be gained from banding in the context of FIM. The FIM process was described in Subsection 2.4.3 of Chapter 2.

For the experiments, the twelve data sets from the UCI machine learning data repository, considered previously, were again used. For the frequent item-set mining the Total From Partial (TFP) algorithm [30] was used, but any alternative FIM algorithm could equally well have been adopted. The TFP algorithm was applied to the data sets in banded and non-banded form (a support threshold σ of 2% was used). The results, respectively using 2D-BPM, MBA_{BFP}, MBA_{FP} and BC, are presented in Tables 4.9, 4.10, 4.11 and 4.12 (best results in bold font). In the Table the data sets are again listed according to number of rows. If we consider only Table 4.9, which shows the timings produced with respect to the proposed 2D-BPM algorithm, it can be seen that, if we do not include the time to conduct the banding, the FIM is much more efficient when using banded data than non-banded data. If the banding time is included, in 8 out of the 12 cases using 2D-BPM, the FIM is still more efficient. It is interesting to note from Table 4.9 that the 4 cases where when the banding is included, FIM is less efficient is where the number of rows is greater than 5,000.

Considering Tables 4.10, 4.11 and 4.12 the total banding and FIM time is shorter in 4 out of the 12 cases using MBA_{BFP} and MBA_{FP}, and 5 out of the 12 cases using BC. These results suggest that the bandings produced using 2D-BPM are somehow better. The four relevant data sets with respect to MBA_{BFP} and MBA_{FP} are the same (Lypography, Hepatitus, Annealing and Waveform).

TABLE 4.9: FIM runtime (seconds) with and without banding using 2D-BPM (σ = 2%)

Datasets	# Rows	# Cols	Banding Time (a)	FIM time with Banding (b)	Total ($a+b$)	FIM time without Banding
Lympography	148	59	0.010	**7.997**	**8.007**	12.658
Hepatitis	155	56	0.020	**0.055**	**0.075**	22.416
Wine	178	68	0.010	**0.155**	**0.165**	0.169
Heart	303	52	0.020	**0.294**	**0.314**	0.387
HorseColic	368	85	0.030	**0.899**	**0.929**	1.242
Annealing	898	73	0.080	**0.736**	**0.816**	2.889
Mushroom	8124	90	3.110	**874.104**	**877.214**	1232.740
WaveForm	5000	101	1.320	**119.220**	**120.540**	174.864
PenDigits	10992	89	3.730	**2.107**	5.837	**2.725**
LetRecognition	20000	106	12.460	**3.004**	15.464	**6.763**
ChessKRvK	28056	58	14.190	**0.082**	14.272	**0.171**
Adult	48842	97	83.960	**2.274**	86.234	**5.827**

TABLE 4.10: FIM runtime (seconds) with and without banding using MBA_{BFP} (σ = 2%)

Datasets	# Rows	# Cols	Banding Time (a)	FIM time with Banding (b)	Total ($a+b$)	FIM time without Banding
Lympography	148	59	0.077	**11.187**	**11.264**	12.658
Hepatitis	155	56	0.061	**19.104**	**10.165**	22.416
Wine	178	68	0.093	**0.211**	0.304	**0.169**
Heart	303	52	0.124	**0.461**	0.585	**0.387**
HorseColic	368	85	0.200	**2.134**	2.334	**1.242**
Annealing	898	73	0.260	**1.733**	**1.993**	2.889
Mushroom	8124	90	9.070	**1595.949**	1605.019	**1232.740**
WaveForm	5000	101	3.057	**125.624**	**128.781**	174.864
PenDigits	10992	89	12.940	**2.731**	15.671	**2.725**
LetRecognition	20000	106	24.538	**9.216**	30.759	**6.763**
ChessKRvK	28056	58	27.909	**0.075**	27.984	**0.171**
Adult	48842	97	185.955	**10.525**	196.480	**5.827**

TABLE 4.11: FIM runtime (seconds) with and without banding using MBA_{FP} (σ = 2%)

Datasets	# Rows	# Cols	Banding Time (a)	FIM time with Banding (b)	Total ($a+b$)	FIM time without Banding
Lympography	148	59	0.060	**11.331**	**11.391**	12.658
Hepatitis	155	56	0.059	**18.876**	**18.935**	22.416
Wine	178	68	0.060	**0.202**	0.262	**0.169**
Heart	303	52	0.109	**0.457**	0.566	**0.387**
HorseColic	368	85	0.122	**2.174**	2.296	**1.242**
Annealing	898	73	0.220	**1.985**	**2.205**	2.889
Mushroom	8124	90	8.140	**1695.349**	1703.489	**1232.740**
WaveForm	5000	101	2.416	**127.613**	**130.029**	174.864
PenDigits	10992	89	11.859	**2.741**	14.600	**2.725**
LetRecognition	20000	106	21.314	**9.216**	30.530	**6.763**
ChessKRvK	28056	58	27.815	**0.085**	27.900	**0.171**
Adult	48842	97	140.954	**11.225**	152.179	**5.827**

TABLE 4.12: FIM runtime (seconds) with and without banding using BC (σ = 2%)

Datasets	# Rows	# Cols	Banding Time (a)	FIM time with Banding (b)	Total ($a+b$)	FIM time without Banding
Lympography	148	59	0.080	**10.597**	**10.677**	12.658
Hepatitis	155	56	0.085	**19.007**	**19.092**	22.416
Wine	178	68	0.090	0.267	0.357	**0.169**
Heart	303	52	0.080	0.673	0.758	**0.387**
HorseColic	368	85	0.090	1.538	1.628	**1.242**
Annealing	898	73	0.200	**1.660**	**1.860**	2.889
Mushroom	8124	90	8.470	**941.725**	**950.195**	1232.740
LetRecognition	20000	106	26.380	8.214	34.504	**6.763**
WaveForm	5000	101	2.280	**129.173**	**131.452**	174.864
PenDigits	10992	89	10.120	3.528	21.158	**2.725**
ChessKRvK	28056	58	33.270	**0.081**	33.351	**0.171**
Adult	48842	97	175.840	**5.512**	181.352	**5.827**

4.8 Summary

This chapter has introduced the concepts of Banding Scores (BS) and Global Banding Scores (GBS). This chapter has also presented the 2D-BPM algorithm for identifying

bandings in 2D zero-one data and illustrated its operation using a worked example. The experimental analysis and evaluation of the 2D-BPM algorithm presented in this chapter, was conducted using the randomly generated synthetic and UCI data sets introduced in Chapter 3 and in comparison with the established BC, MBA$_{BFP}$ and MBA$_{FP}$ algorithms. The analysis was conducted in terms of; (i) Global Banding Score (GBS), (ii) runtime, (iii) the Average Band Width (ABW), (iv) Accuracy and (v) Mean Row Moment (MRM). Recall that ABW was designed to be an independent measure. The main findings from the reported evaluation of the proposed 2D-BPM algorithm indicated that:

1. The 2D-BPM algorithm produces better results than the other three banding algorithms considered in terms of both GBS and the independent ABW metric (The main finding from the reported evaluations indicated that the most effective and efficient algorithm was the proposed 2D-BPM algorithm (Section 4.7.3)).

2. In many cases the 2D-BPM algorithm also produced better results than the other three banding algorithms considered when using Accuracy (in 7 out of the 12 cases) and MRM (in 8 out of the 12 cases), despite the fact that 2D-BPM did not seek to maximise these metrics.

3. The 2D-BPM algorithm was consistently more efficient than the other three algorithms considered because it avoids the need to consider large number of permutations.

4. Banding improves the effectiveness of applications such as Frequent Itemset Mining (FIM).

Overall the 2D-BPM algorithm produce the best banding and consistently outperform the three algorithms considered.
In the next chapter the Approximate 3D Banded Pattern Mining (A3D-BPM) algorithm will be: presented, illustrated using a worked example and evaluated in the context of 3D data sets extracted from the CTS database.

Chapter 5

Approximate Banding Mechanism

5.1 Introduction

The previous chapter considered banding in 2D and presented the 2D-BPM algorithm for finding banded patterns in 2D zero-one data. This chapter, and the following two chapters, consider the banding problem in terms of higher dimensions commencing with 3D banding a special case of ND BPM algorithm. This chapter proposes the Approximate 3D Banded Pattern Mining (A3D-BPM) algorithm designed to find an "approximate" banding in 3D data. The algorithm is founded on the 2D-BPM algorithm presented in the foregoing chapter extended to address 3D data. As the name suggest, for reasons of efficiency, the algorithm features an approximation; the precise nature of this approximation will become clear later in this chapter. The conjecture was that despite producing an approximate banding the outcome would still be acceptable while at the same time being generated in a manner that would be more efficient than if an exact banding was generated. Whatever the case, to the best knowledge of the author, no work has been directed at the banding of 3D data other than the work presented in this and the following two chapters of this thesis.

Recall that a 2D binary valued data set is said to feature a banding if the dimension indexes can be ordered in such a way that the "dots" are arranged about the leading diagonal. The same applies in the case of 3D data (and ND data). Recall also that, given a reasonably complex data set, it is unlikely that a perfect banding can be achieved, however some "close to" best banding is always possible.

The rest of this chapter is organized as follows. We commence in Section 5.2 by considering the formalism associated with the 3D banding problem and the calculation of banding scores in the context of the A3D-BPM algorithm. Section 5.3 then goes on to consider the A3D-BPM algorithm in detail. A worked example illustrating how the A3D-BPM algorithm operates is presented in Section 5.4 and Section 5.5 then concludes the chapter with a brief summary. No evaluation is presented because at this point in the thesis there is no alternative algorithm with which the proposed A3D-BPM algorithm can be compared. This is done in Chapter 6 where alternative 3D BPM algorithms are considered.

5.2 3D Approximated Banding Formalism and Calculation of Banding Scores

In the context of the research presented in this thesis, a 3D data space can be conceptualised as comprising a $(k_1 \times k_2 \times k_3)$ grid where k_1 is the size of dimension one, k_2 is the size of dimension two and so on. The data space can be conceived of in terms of a x-y-z cartesian space; or, alternatively, as comprising column, rows and slices. The indexes associated with dimension x (columns) might be record numbers, the indexes associated with dimension y (rows) might be attribute value identifiers and the indexes associated with dimension z (slices) might be discrete time stamps. Note that a particular index p belonging to a dimension i will be indicated using the notation e_{i_p} and that the dimensions are not all necessarily of equal size. The set of dimensions is indicated using the notation $DIM = \{Dim_x, Dim_y, Dim_z\}$, where each dimension comprises a set of indexes (which we wish to order so as to reveal a best banding).

As before, each grid cube in the data space representing a "one" is conceptualised as containing a dot, whilst each grid cube representing a "zero" is conceptualised as being empty. Figures 5.1 and 5.2, present $3D$ configurations made up of three "columns" (Dim_x), three "rows" (Dim_y) and three "slices" (Dim_z). Figure 5.1 represents a perfect banding as defined in this thesis, whilst Figure 5.2 presents some alternative banding. Note that each dot can be defined by a coordinate tuple of the form $\langle x, y, z \rangle$ where $0 \leq x \leq k_1$, $0 \leq y \leq k_2$ and $0 \leq z \leq k_3$. Therefore, a $3D$ data set D can be considered to comprise a set of m dots, $D = \{d_1, d_2, ..., d_m\}$, such that each d_i is represented by a tuple of the form $\langle x_i, y_i, z_i \rangle$.

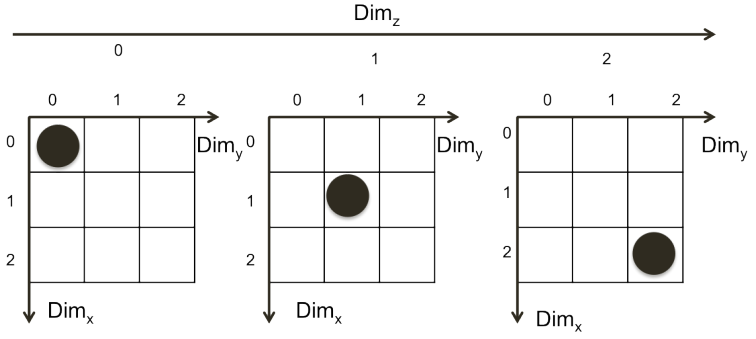


FIGURE 5.1: Example of a $3D$ dot Configuration featuring a perfect banding

As will become clearer later in this chapter the A3D-BPM algorithm operate by considering pairing of dimensions. Thus given two dimensions Dim_i and Dim_j, we calculate the banding scores for index p in Dim_i with respect to Dim_j indicated using the notation bs_{ij_p} as shown in Equation 5.1. The similarity between this and Equation 4.3 presented in the previous chapter should be noted. In Equation 5.1 the set W is the set of Dim_j indexes $\{w_1, w_2, \dots\}$, representing "dots" whose Dim_i coordinate equates to p.

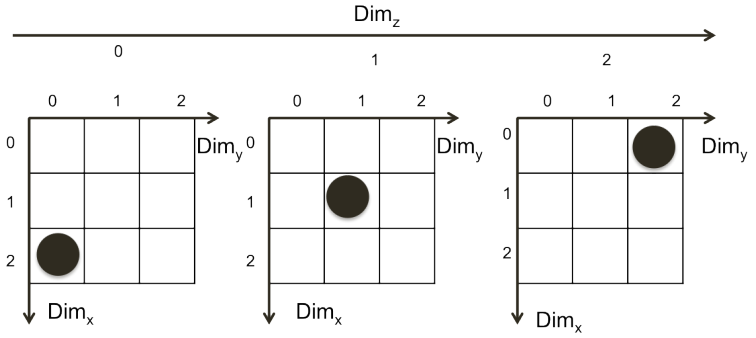


FIGURE 5.2: Example of a $3D$ dot Configuration featuring an alternative banding

$$bs_{ij_p} = \frac{\sum_{p=1}^{p=|W|} w_p}{\sum_{q=1}^{q=|W|}(|Dim_q| - k + 1)} \tag{5.1}$$

Note that using Equation 5.1 means that in the 3D case all three dimensions are not taken into consideration when calculating individual banding scores; only pairs of dimensions are considered. This is thus the approximation featured by the A3D-BPM algorithm. However, it was conjectured that this approximate approach would result in sufficiently accurate bandings without the need for the extra resource to calculate more complex (exact) banding scores. It should be noted that although the approach only considers pairs of dimensions this does not mean that the other dimensions are ignored (the 3rd dimensions in the 3D case), as all dimension pairing are considered during the process. The Global Banding Score for dimension Dim_i with respect to dimension Dim_j, given by GBS_{ij}, is calculated using Equation 5.2 where k_i is the size of dimension Dim_i.

$$GBS_{ij} = \frac{\sum_{p=1}^{p=k_i} bs_{ij_p}}{k_i} \tag{5.2}$$

The Global Banding Score for a dimension i, given by GBS_i, is then calculated using Equation 5.3 where I is the set of dimension identifiers excluding Dim_i.

$$GBS_i = \frac{\sum_{j=1}^{|I|} GBS_{ij}}{|I|} \tag{5.3}$$

Thus we have:

$$GBS_x = \frac{GBS_{xy} + GBS_{xz}}{2} \tag{5.4}$$

$$GBS_y = \frac{GBS_{yx} + GBS_{yz}}{2} \tag{5.5}$$

$$GBS_z = \frac{GBS_{zx} + GBS_{zy}}{2} \tag{5.6}$$

The overall GBS value is then calculated using Equation 5.7.

$$GBS = \frac{\sum_{i=1}^{i=|DIM|}}{|DIM|} = \frac{GBS_x + GBS_y + GBS_z}{3} \tag{5.7}$$

Usage of the above can be illustrated using the configurations featured in Figures 5.1 and 5.2. Starting with Figure 5.1 the set of banding scores for Dim_x, Dim_y and Dim_z, calculated using Equation 5.1, will be:

$$BS_{xy} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0 \qquad BS_{xz} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0$$

$$BS_{yx} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0 \qquad BS_{yz} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0$$

$$BS_{zx} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0 \qquad BS_{zy} = \frac{1}{3} + \frac{2}{3} + \frac{3}{3} = 2.0$$

Similarly, for the configuration shown in Figure 5.2, the set of banding scores for Dim_x, Dim_y and Dim_z will be:

$$BS_{xy} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33 \qquad BS_{xz} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33$$

$$BS_{yx} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33 \qquad BS_{yz} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33$$

$$BS_{zx} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33 \qquad BS_{zy} = \frac{1}{3} + \frac{2}{3} + \frac{1}{3} = 1.33$$

Referring back to the dot configuration shown in Figures 5.1 the GBS for: (i) dimension x with respect to dimension y, (ii) dimension x with respect to dimension z and (iii) dimension y with respect to dimension z, calculated using Equation 5.2, will be as follows:

$$GBS_{xy} = \frac{2}{3} = 0.67 \qquad GBS_{xz} = \frac{2}{3} = 0.67$$

$$GBS_{yx} = \frac{2}{3} = 0.67 \qquad GBS_{yz} = \frac{2}{3} = 0.67$$

$$GBS_{zx} = \frac{2}{3} = 0.67 \qquad GBS_{zy} = \frac{2}{3} = 0.67$$

and in the case of the configuration shown in Figure 5.2:

$$GBS_{xy} = \frac{1.33}{3} = 0.44 \qquad GBS_{xz} = \frac{1.33}{3} = 0.44$$

$$GBS_{yx} = \frac{1.33}{3} = 0.44 \qquad GBS_{yz} = \frac{1.33}{3} = 0.44$$

$$GBS_{zx} = \frac{1.33}{3} = 0.44 \qquad GBS_{zy} = \frac{1.33}{3} = 0.44$$

As noted above, to obtain the GBS for each dimension, we simply sum the indvidual banding scores and divide by the number of dimensions minus one using Equations 5.4, 5.5 and 5.6. Recall that we divide by 2 so as to normalise the dimension GBS values (because we have two dimensions pairings). The values GBS_x, GBS_y and GBS_z for the configuration in Figure 5.1 will thus be:

$$GBS_x = \frac{0.67 + 0.67}{2} = \frac{1.34}{2} = 0.67$$

$$GBS_y = \frac{0.67 + 0.67}{2} = \frac{1.34}{2} = 0.67$$

$$GBS_z = \frac{0.67 + 0.67}{2} = \frac{1.34}{2} = 0.67$$

and for the configuration shown in Figures 5.2:

$$GBS_x = \frac{0.44 + 0.44}{2} = \frac{0.88}{2} = 0.44$$

$$GBS_y = \frac{0.44 + 0.44}{2} = \frac{0.88}{2} = 0.44$$

$$GBS_z = \frac{0.44 + 0.44}{2} = \frac{0.88}{2} = 0.44$$

The GBS values are the same because the two configurations are symmetric about the diagonal.

Using Equation 5.7 the overall global banding Score (GBS) for the two configurations will be calculated as follows:

$$GBS = \frac{0.67 + 0.67 + 0.67}{3} = 0.67$$

$$GBS = \frac{0.44 + 0.44 + 0.44}{3} = 0.44$$

Note that this result serves to distinguish between the perfect banding shown in Figure 5.1 and the alternative banding shown in Figure 5.2.

5.3 Overview of Approximate 3D Banded Pattern Mining (A3D-BPM) Mechanism

The A3D-BPM algorithm operates in a similar manner to the 2D-BPM algorithm; we loop through the dimensions rearranging the dimension indexes according to the banding

score concept. The process continues until a best banding has been arrived at or we reach some maximum number of permitted iterations monitored by a counter variable.

For the A3D-BPM algorithm it should be noted that the banding scores are calculated in the same manner as the 2D-BPM algorithm presented in Chapter 4. However, this was a 2D mechanism, thus we have to consider all possible pairings. The maximum number of pairings can be calculated using Equation 5.8, where $|DIM|$ is the size of the set of dimensions DIM, in other words the number of dimensions. Where $|DIM| = 3$, as in the case of 3D data, the maximum number of parings will be $3 \times 2 = 6$.

$$Max \text{ pairings} = |DIM| \times (|DIM| - 1) \tag{5.8}$$

The pseudo code for the A3D-BPM algorithm is presented Algorithm 6. The inputs are (Lines 1 to 3): (i) a dot data set D, (ii) the set of dimensions $DIM = \{Dim_x, Dim_y, Dim_z\}$ and (iii) a maximum number of iterations *counter*. The output is a rearranged data space D that minimises the GBS value (Line 4). Because we are seeking to minimise the GBS score, on start up, the GBS value sofar is set to 1.0 (Line 5). The algorithm iteratively loops over the data space. On each iteration the algorithm rearranges the indexes in Dim_i according to the calculated banding scores. Recall that this is done by considering all possible 2D pairings. For each pairing Dim_{ij} the banding score bs_{ij_p} for each index p in Dim_i is calculated (Lines 10 to 18) with respect to Dim_j and used to rearrange dimensions Dim_i to give Dim'_i (Line 15). A GBS value for Dim_x, Dim_y and Dim_z is calculated and stored in a set G (Lines 19 to 23). Once all the pairings have been considered, a GBS_{new} value is calculated (Line 25). If GBS_{new} is worse than the current GBS_{sofar} value, or there has been no change, we exit with the current configuration D (Line 33). Otherwise we set D to D' and GBS_{new} to GBS_{sofar} (Line 29) and repeat. Note that although not shown in Algorithm 6, termination also occurs whenever no changes have taken place.

5.4 A Working Example Using the A3D-BPM Algorithm

This section presents a worked example to illustrate the operation of the proposed A3D-BPM algorithm using the 3D configuration shown in Figure 5.3. The same configuration is shown in Figures 5.4 and 5.5 but from different perspectives. Note that the data set is made up of: five "columns" (Dim_x), six "rows" (Dim_y) and two "slices" (Dim_z).

The A3D-BPM algorithm commence by calculating the BS_{ij_p} scores for Dim_x, Dim_y and Dim_z to obtain: $BS_{xy} = 1.0000$, $BS_{xz} = 0.6296$, $BS_{yx} = 0.8095$, $BS_{yz} = 0.6188$, $BS_{zx} = 0.2794$ and $BS_{zy} = 0.3333$. Using this set of scores the items in Dim_x, Dim_y and Dim_z are rearranged as shown in Figures 5.6, 5.7 and 5.8. The GBS for Dim_x, Dim_y and Dim_z are: $GBS_x = 0.8148$, $GBS_y = 0.7142$ and $GBS_z = 0.3064$; and the overall GBS value for the configuration is now:

$$GBS = \frac{0.8148 + 0.7142 + 0.3064}{3} = 0.6120$$

Algorithm 6: The A3D-BPM Algorithm

Input: DIM = a set of dimensions $\{Dim_x, Dim_y, Dim_z\}$ each comprised of a set of indexes
D = a binary valued data matrix subscribing to DIM
$counter$ = a maximum number of iterations
Output: D rearranged so as to minimise the GBS
$GBS_{sofar} = 1.0$
loop
  if $(counter == 0)$ **then**
    $break$
  end if
  for $i = 0$ **to** $i = |DIM|$ **do**
    for $j = i + 1$ **to** $j = |DIM|$ **do**
      for $p = 0$ **to** $p = k_i$ **do**
        bs_{ij_p} = Banding score for index p in Dim_i w.r.t. Dim_j calculated using Equation 5.1
      end for
      DIM'_i = Rearranged Dim_i according to banding scores for Dim_i w.r.t Dim_j
      $D' = D$ Rearranged according to Dim'_i
    end for
  end for
  for $i = 0$ **to** $i = |DIM|$ **do**
    for $j = 0$ **to** $j = |DIM|$ and $j \neq i$ **do**
      $G = GBS_{ij}$ calculated using Equation 5.2
    end for
  end for
  GBS_i calculated using Equation 5.3
  GBS_{new} = overall GBS calculated using G and Equation 5.7
  if $(GBS_{new} \geq GBS_{sofar})$ **then**
    break
  else
    $DIM = DIM'$, $D = D'$, $GBS_{sofar} = GBS_{new}$
  end if
  $counter = counter - 1$
end loop
Exit with D and GBS

On the second iterations, the process is repeated and the BS_{ij_p} scores for Dim_x, Dim_y and Dim_z are now: $BS_{xy} = 0.9524$, $BS_{xz} = 0.7593$, $BS_{yx} = 0.9286$, $BS_{yz} = 0.5805$ and $BS_{zx} = 0.2794$ and $BS_{zy} = 0.2963$. As a consequence the items in Dim_x, Dim_y and Dim_z are rearranged to produce the configuration shown in Figures 5.9, 5.10 and 5.11 .

The Global banding scores for Dim_x, Dim_y and Dim_z are now: $GBS_x = 0.8559$ (previously this was 0.8148), $GBS_y = 0.7546$ (was 0.6950) and $GBS_z = 0.2879$ (was 0.3064) and the overall GBS value is now:

$$GBS = \frac{0.8559 + 0.7546 + 0.2879}{3} = 0.6328$$

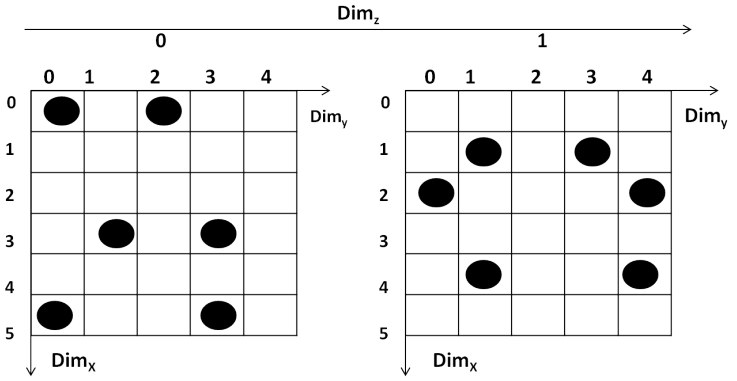


FIGURE 5.3: Input Data Perspective 1

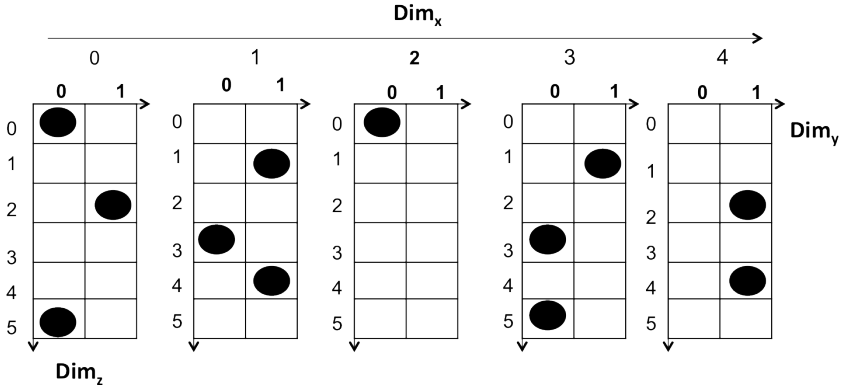


FIGURE 5.4: Input Data Perspective 2

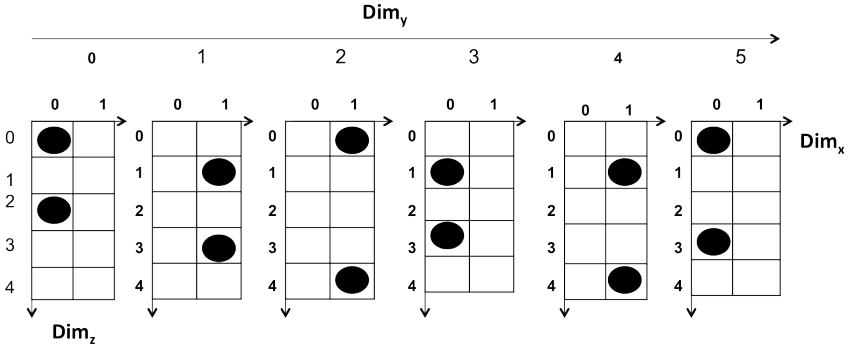


FIGURE 5.5: Input Data Perspective 3

On the previous iteration it was 0.6120, however no changes have been made on this second iteration so the process terminates.

5.5 Summary

This chapter has described the operation of the proposed A3D-BPM algorithm. The algorithm was presented in detail and its operation illustrated with a worked example.

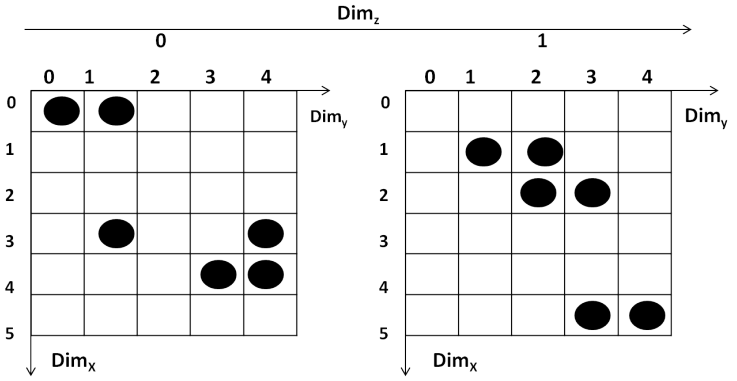


FIGURE 5.6: Input Data rearranged using A3D-BPM after the first iteration, perspective 1

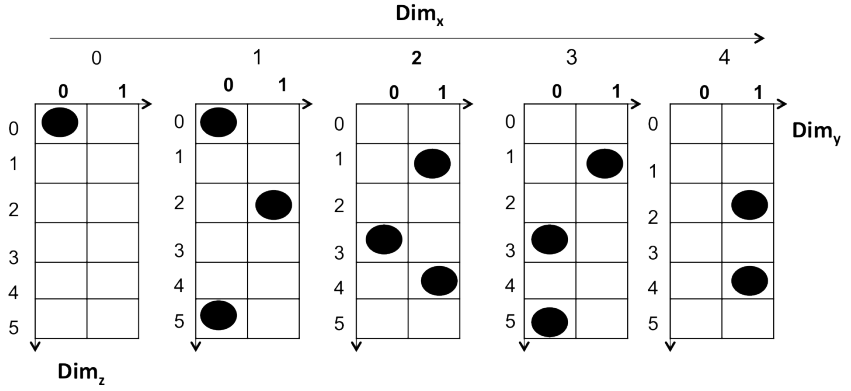


FIGURE 5.7: Input Data rearranged using A3D-BPM after the first iteration, perspective 2

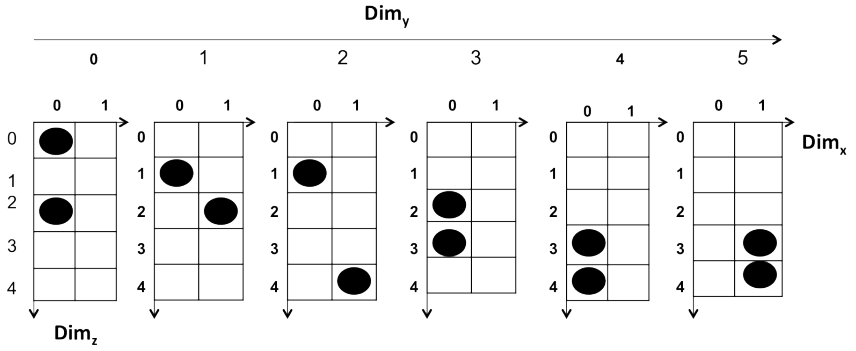


FIGURE 5.8: Input Data rearranged using A3D-BPM after the first iteration, perspective 3

The algorithm produces only an approximate banding in the sense that when calculating banding scores it only considers pairs of dimensions, all dimensions are not considered simultaneously when calculating banding scores. The significance of the A3D-BPM algorithm was that it was conjectured that sufficiently accurate bandings would be generated

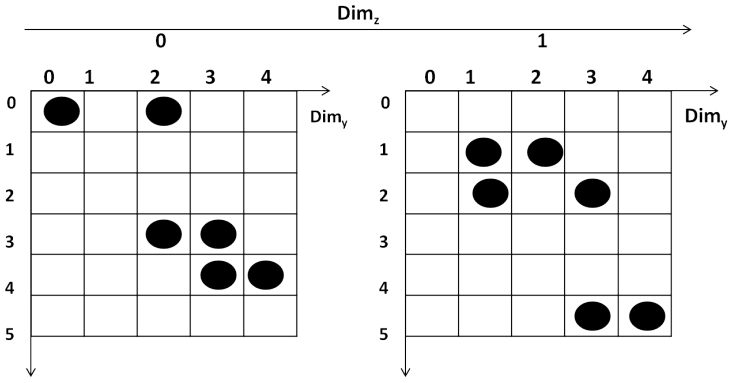


FIGURE 5.9: Input Data rearranged using A3D-BPM after the second iteration, perspective 1

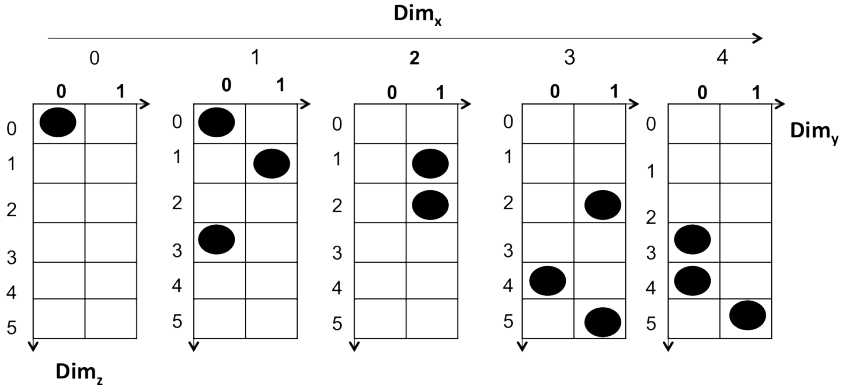


FIGURE 5.10: Input Data rearranged using A3D-BPM after the second iteration, perspective 2

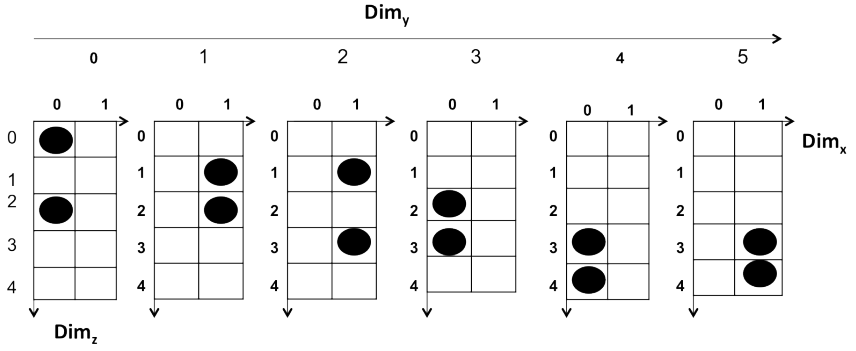


FIGURE 5.11: Input Data rearranged using A3D-BPM after the second iteration, perspective 3

without the complexity of considering all dimensions simultaneously, which it was anticipated would entail a computational overhead. To determine whether this conjecture was correct or not an exact 3D BPM algorithm was required so that comparisons could be made. The following chapter thus presents the Exact 3D Banded Pattern Mining (E3D-BPM) algorithm and also presents a comparison of the two algorithms.

Chapter 6

Exact Banding Mechanism

6.1 Introduction

In the previous chapter the Approximate 3D Banded Pattern Mining (A3D-BPM) algorithm was presented. The A3D-BPM algorithm was a natural progression from the 2D-BPM algorithm presented in Chapter 4, but it entailed a simplification in that banding scores were calculated in terms of 2D in a very similar manner to that used in the 2D-BPM algorithm. However, it was conjectured that the advantage offered might be that the finding of approximate banded patterns would be more efficient than finding exact patterns, and that the resulting patterns would be of sufficient quality. To establish whether this was indeed the case an exact 3D banding algorithm would be required. This is therefore discussed in this chapter. More specifically the Exact 3D Banded Pattern Mining (E3D-BPM) algorithm a special case of ND BPM algorithm is presented.

As will become clear later in this chapter, the proposed E3D-BPM algorithm is more complex than the A3D-BPM algorithm because it takes into consideration the entire data space. It does this by calculating banding scores using the concept of "distance from origin" of individual dots. These distances can be calculated in variety of ways; two obvious choices, and those considered in this chapter, are: (i) Euclidean and (ii) Manhattan distance calculation. To normalise the calculated banding score, again as will become clear later in this chapter, we use maximum "distance from origin". Because, given any reasonably sized data set, maximum distances would be frequently calculated this chapter also presents the idea of pre-calculating these distances and storing them in a maximum distance table (an M-Table).

The rest of this chapter is organised as follows. Section 6.2 presents some formal definitions to support the E3D-BPM algorithm. Section 6.3 considers the M-Table concept, while Section 6.4 considers the E3D-BPM algorithm it-self. Section 6.5 then presents an evaluation of the E3D-BPM algorithm in comparison with the A3D-BPM algorithm presented in the previous chapter. The evaluation was conducted in the context of the CTS data sets introduced in Chapter 3. A summary of the work considered in this chapter and some conclusions, are presented in Section 6.6.

6.2 Formalism and Banding Score Calculation

As in the case of the A3D-BPM algorithm, the data space of interest comprises a set of 3-Dimensions, $DIM = \{Dim_x, Dim_y, Dim_z\}$ such that each dimension comprises a set of index positions (which we wish to rearrange to achieve a banding). Note that the dimensions are not necessarily all of the same size. As before the dots within the space are referenced using x-y-z coordinate tuples of the form $\langle c_1, c_2, c_3 \rangle$. Consequently, again as in the case of the A3D-BPM algorithm, a dot data set $D = \{d_1, d_2, \dots\}$ comprises a set of coordinate tuples each representing a dot. The situation where more than one dot might occur at a location is excluded at the present, thus any given coordinate tuple can appear only once in D.

The Banding Score (BS) for a particular index j in dimension Dim_i, indicated in this chapter using the notation bs_{i_j}, is determined according to the location of the subset of dots $S = \{s_1, s_2, \dots\}$ in D whose c_i coordinate is equal to j (recall that each dot in D is define by a coordinate tuple of the form $\langle c_1, c_2, c_3 \rangle$). For each dot in S we calculate the distance to the origin in terms of a data sub-space that does not include the current dimension Dim_i. We exclude the current dimension because this is the dimension we want to rearrange. Thus the banding score bs_{i_j} is calculated as follow:

$$bs_{i_j} = \sum_{p=1}^{p=|S|} dist(s_p) \tag{6.1}$$

where $dist(s_p)$ is the distance from the "zero" origin of the data space to the point s_p. However, as before, we wish to normalise the Banding Scores (BS). To this end we need to divide by an equal number of maximum distances. Thus we need to devise a set $Max = \{m_1, m_2, \dots\}$ holding these maximum distances such that there is a one-to-one correspondence between the set Max and the set S. Recall that the assumption has been made that only one dot can be held at a given location. Thus the normalised banding score is calculated as follows:

$$bs_{i_j} = \frac{\sum_{p=1}^{p=|S|} dist(s_p)}{\sum_{p=1}^{p=|Max|} m_p} \tag{6.2}$$

As noted in the introduction, there are two obvious mechanisms for calculating the distance of a dot's location to the origin of the data space: (i) Euclidean (Equation 6.3) and (ii) Manhattan (Equation 6.4).

$$w = \sqrt{(c_1)^2 + (c_2)^2 + \cdots + (c_n)^2} \tag{6.3}$$

$$w = \sum_{k=1}^{k=n} c_k \tag{6.4}$$

Thus we have two variations of the E3D-BPM algorithm: (i) Euclidean E3D-BPM and (ii) Manhattan E3D-BPM.

Given the above, GBS values can be calculated as follows (regardless of whether Euclidean or Manhattan distance calculation is used). Recall that, a $3D$ data set D can be considered to comprise a set of m dots, $D = \{d_1, d_2, ..., d_m\}$ such that each d_i is represented by a tuple of the form $\langle c_1, c_2, c_3 \rangle$. Therefore, the GBS, for Dim_x, Dim_y and Dim_z may be obtained thus:

$$GBS_x = \frac{\sum_{j=0}^{j=k_1} bs_{i_j} \times (k_1 - j)}{k_1(k_1 + 1)/2} \tag{6.5}$$

$$GBS_y = \frac{\sum_{j=0}^{j=k_2} bs_{y_j} \times (k_2 - j)}{k_2(k_2 + 1)/2} \tag{6.6}$$

$$GBS_z = \frac{\sum_{j=0}^{j=k_3} bs_{z_j} \times (k_3 - j)}{k_3(k_3 + 1)/2} \tag{6.7}$$

and the total GBS for the given 3D configuration using Equation 6.8:

$$GBS = \frac{GBS_x + GBS_y + GBS_z}{3} \tag{6.8}$$

Usage of the above can be illustrated by considering the banding configurations used previously with respect to the A3D-BPM algorithm; these were given in Figures 5.1 and 5.2 in Chapter 5. Starting with the configuration given in Figure 5.1 and using the Euclidean variation of the E3D-BPM algorithm the banding scores for Dim_x, Dim_y and Dim_z (calculated using Equation 5.1) will be:

$$bs_{x_1} = \frac{0.0000}{2.8284} = 0.0000 \quad bs_{x_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{x_3} = \frac{2.8284}{2.8284} = 1.0000$$

$$bs_{y_1} = \frac{0.0000}{2.8284} = 0.0000 \quad bs_{y_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{y_3} = \frac{2.8284}{2.8284} = 1.0000$$

$$bs_{z_1} = \frac{0.0000}{2.8284} = 0.0000 \quad bs_{z_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{z_3} = \frac{2.8284}{2.8284} = 1.0000$$

The GBS values for Dim_x, Dim_y and Dim_z, GBS_x, GBS_y and GBS_z will then be:

$$GBS_x = \frac{0.0000 \times 2 + 0.5000 \times 1 + 1.0000 \times 0}{2(2 + 1)/2} = \frac{0.5000}{3} = 0.1667$$

$$GBS_y = \frac{0.0000 \times 2 + 0.5000 \times 1 + 1.0000 \times 0}{2(2 + 1)/2} = \frac{0.5000}{3} = 0.1667$$

$$GBS_z = \frac{0.0000 \times 2 + 0.5000 \times 1 + 1.0000 \times 0}{2(2+1)/2} = \frac{0.5000}{3} = 0.1667$$

Note that the above GBS_x, GBS_y and GBS_z values are the same because the configuration in Figure 5.1 is symmetrical. The final GBS is then as follows:

$$GBS = \frac{0.1667 + 0.1667 + 0.1667}{3} = 0.1667$$

Similarly when, using the Manhattan variation of the E3D-BPM algorithm the banding scores for Dim_x, Dim_y and Dim_z (for the configuration given in Figure 5.1) will be:

$$bs_{x_1} = \frac{0}{4} = 0.0000 \quad bs_{x_2} = \frac{2}{4} = 0.5000 \quad bs_{x_3} = \frac{4}{4} = 1.0000$$

$$bs_{y_1} = \frac{0}{4} = 0.0000 \quad bs_{x_2} = \frac{2}{4} = 0.5000 \quad bs_{x_3} = \frac{4}{4} = 1.0000$$

$$bs_{z_1} = \frac{0}{4} = 0.0000 \quad bs_{x_2} = \frac{2}{4} = 0.5000 \quad bs_{x_3} = \frac{4}{4} = 1.0000$$

The values for GBS_x, GBS_y and GBS_z will then be:

$$GBS_x = \frac{0.0 \times 2 + 0.5 \times 1 + 1.0 \times 0}{2(2+1)/2} = \frac{0.5}{3} = 0.1667$$

$$GBS_y = \frac{0.0 \times 2 + 0.5 \times 1 + 1.0 \times 0}{2(2+1)/2} = \frac{0.5}{3} = 0.1667$$

$$GBS_z = \frac{0.0 \times 2 + 0.5 \times 1 + 1.0 \times 0}{2(2+1)/2} = \frac{0.5}{3} = 0.1667$$

The final GBS value for the configuration will then be:

$$GBS = \frac{0.1667 + 0.1667 + 0.1667}{3} = 0.1667$$

If we now consider the configuration given in Figure 5.2 the set of banding scores for Dim_x, Dim_y and Dim_z using the Euclidean E3D-BPM algorithm will be:

$$bs_{x_1} = \frac{2.0000}{2.8284} = 0.7071 \quad bs_{x_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{x_3} = \frac{2.0000}{2.8284} = 0.7071$$

$$bs_{y_1} = \frac{2.0000}{2.8284} = 0.7071 \quad bs_{y_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{y_3} = \frac{2.0000}{2.8284} = 0.7071$$

$$bs_{z_1} = \frac{2.0000}{2.8284} = 0.7071 \quad bs_{z_2} = \frac{1.4142}{2.8284} = 0.5000 \quad bs_{z_3} = \frac{2.0000}{2.8284} = 0.7071$$

and the corresponding GBS_x, GBS_y and GBS_z values will be:

$$GBS_x = \frac{0.7071 \times 2 + 0.5000 \times 1 + 0.7071 \times 0}{2(2+1)/2} = \frac{1.9142}{3} = 0.6380$$

$$GBS_y = \frac{0.7071 \times 2 + 0.5000 \times 1 + 0.7071 \times 0}{2(2+1)/2} = \frac{1.9142}{3} = 0.6380$$

$$GBS_z = \frac{0.7071 \times 2 + 0.5000 \times 1 + 0.7071 \times 0}{2(2+1)/2} = \frac{1.9142}{3} = 0.6380$$

Consequently the final GBS value for the dot configuration given in Figure 5.2, using the Euclidean E3D-BPM algorithm will be:

$$GBS = \frac{0.6380 + 0.6380 + 0.6380}{3} = 0.6380$$

Using the Manhattan variation of the E3D-BPM algorithm applied to the dot configuration given in Figure 5.2 the set of banding scores for Dim_x, Dim_y and Dim_z will be:

$$bs_{x_1} = \frac{2}{4} = 0.5000 \qquad bs_{x_2} = \frac{2}{4} = 0.5000 \qquad bs_{x_3} = \frac{2}{4} = 0.5000$$

$$bs_{y_1} = \frac{2}{4} = 0.5000 \qquad bs_{y_2} = \frac{2}{4} = 0.5000 \qquad bs_{y_3} = \frac{2}{4} = 0.5000$$

$$bs_{z_1} = \frac{2}{4} = 0.5000 \qquad bs_{z_2} = \frac{2}{4} = 0.5000 \qquad bs_{z_3} = \frac{2}{4} = 0.5000$$

which will give rise to the following GBS values:

$$GBS_x = \frac{0.5 \times 2 + 0.5 \times 1 + 0.5 \times 0}{2(2+1)/2} = \frac{1.5}{3} = 0.5000$$

$$GBS_y = \frac{0.5 \times 2 + 0.5 \times 1 + 0.5 \times 0}{2(2+1)/2} = \frac{1.5}{3} = 0.5000$$

$$GBS_z = \frac{0.5 \times 2 + 0.5 \times 1 + 0.5 \times 0}{2(2+1)/2} = \frac{1.5}{3} = 0.5000$$

and a final GBS value of:

$$GBS = \frac{0.5 + 0.5 + 0.5}{3} = 0.5000$$

The above illustrations are summarised in Table 6.1 with respect to the final GBS values obtained. From the table it can be seen that, regardless of whether the Euclidean or Manhattan E3D-BPM variation is used, the resulting overall GBS value finally arrived at can be used to distinguish between the two configurations given in Figures 5.1 and 5.2.

TABLE 6.1: Summary of final GBS values obtained with respect to illustration given in Section 6.2

	Perfect banding (Figure 5.1)	Alternative banding (Figure 5.2)
Euclidean E3D-BPM	0.1667	0.6380
Manhattan E3D-BPM	0.1667	0.5000

6.3 Maximum Distance Tables

The mechanism for calculating banding scores presented in the previous section, Section 6.2 involves normalisation using maximum distances from (to) the origin. With reference to the illustrations using the exact banding process presented in Section 6.2, the maximum number of dots associated with each index in each dimension is one, we only need one maximum value, however, the maximum distance is calculated repeatedly (Equation 6.3 or 6.4 will be invoked again and again). The number of maximum distances required will be equivalent to the maximum number of dots held with respect to any one index in any dimension. Thus for most genuine data sets there will be many "maximum" distance calculations and it is likely that the same maximum distances will be calculated again and again. The idea presented in this section is that these values can be precalculated and stored in a table called an M-Table.

An example M-Table is given in Figure 6.1. In the figure the rows represent the dimensions and columns the maximum distances starting with the largest and then decreasing. The length of each row depends on the maximum number of dots with respect to any one index in the associated dimension. The value v_{ij} included in the table indicated the value for dimension i for the j^{th} dot. Note that with respect to the example given in Figure 6.1 the maximum number of dots per dimension is not equal. Figure 6.2(a) and (b) show the M-Tables (using Euclidean and Manhattan distance calculation respectively) for the "toy" configurations shown in Figures 5.1 and 5.2. In this case, because the maximum number of dots associated with each index in each dimension is one, we only have one maximum value per row. Also, because both configurations are symmetrical, the maximum distance is the same for each dimension.

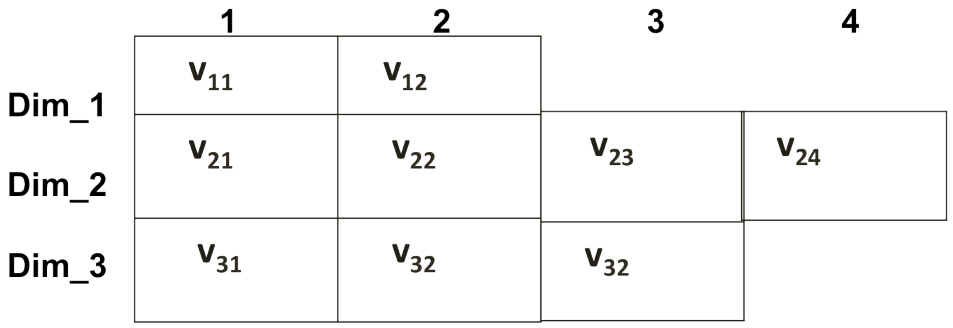


FIGURE 6.1: Example M-Table

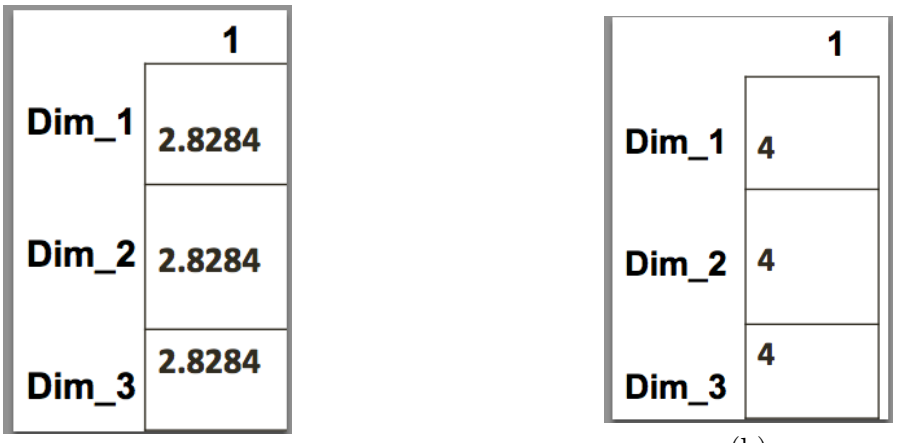


FIGURE 6.2: M-Tables for example configurations given in Figures 5.1 and 5.2: (a) Euclidean and (b) Manhattan

The remainder of this section is organised as follows. Subsection 6.3.1 considers the generation of M-Tables. It is proposed that this be done using an algorithm referred to as the Maximum Distance Calculation (MDC) Algorithm. An example of the construction of an M-Table, using the MDC algorithm, is then presented in Subsection 6.3.2.

6.3.1 M-Table Generation and the MDC Algorithm

From the foregoing, because maximum distances are calculated repeatedly, it is suggested that it might be expedient to calculate the potential maximum distances that may be required in advance and store these in a Maximum Distance Table (an M-Table). The number of dimension featured in such an M-Table will always be one less than $|DIM|$, the maximum number of dimensions. This is because, as noted above, when calculating banding scores we ignore the current dimension as this is the dimension we wish to rearrange. Note also that the entire data space does not need to be covered (this would require a significant computational overhead), we only need to consider the maximum number of dots that can occur with respect to each dimension.

The calculation of the longest possible distance from the origin to a dot within a ND space is straight forward as the maximum coordinates are known. The second most longest distance is harder, especially where the ND space under consideration is not symmentrical. Similarly with the third longest distance and so on. Other than for the maximum distance there will be a number of candidates locatons that will give the nth most longest distance. To generate an M-Table, given the foregoing, the Maximum Distance Calculation (MDC) algorithm is proposed.

The pseudo code for the proposed MDC algorithm is given in Algorithms 7 and Algorithm 8. Algorithm 7 is the "top-level" algorithm for calculating M-Tables while Algorithm 8 is used to calculate maximum values for a specified row in a desired M-Table. Returning to Algorithm 7, the inputs (Line 1) are: (i) the dimension sizes $\{k_1, k_2, k_3\}$ for the data space under consideration, (ii) the set of dimensions DIM defining the

data space and (iii) the dot data set under consideration D. The output (Line 2) is an M-Table as defined above. The algorithm commences (Lines 3 to 12) by determining the maximum number of dots for each dimensions $Dim_i \in DIM$ by considering each index j in dimension Dim_i in turn. The result is stored in the *MaxDots* array. The information is then used to define the size of the desired M-Tables (Line 13). We then loop through the dimensions (Lines 14 to 17). On each iteration we first (Line 15) collate the dimension sizes, excluding the current dimension Dim_i, and hold these in *DimSizes*. Then (Line 16) the function *calculateMtableRow* is called (Algorithm 8) to generate the required M-Tables row of maximum distances for the current dimension Dim_i. In this manner the M-Table is built up.

Algorithm 7: MDC Algorithm

```
1:  Input: K = {k1, k2, k3}, DIM = {Dimx, Dimy, Dimz} set of dimensions,D set of
    dots held in the data space under consideration
2:  output: M-Table
3:  MaxDots = {max1, max2, max3} = {0, 0, 0}
4:  loop
5:    for i = 1 to i = DIM do
6:      for j = 1 to j = Ki do
7:        count = number of dots with index equal to j for dimension i
8:        if (count > maxi) then
9:          maxi = count
10:       end if
11:     end for
12:   end for
13:   Define M-Table of size |K| by content of MaxDots
14:   for i = 1 to i = DIM do
15:     DimSizes = array of dimension sizes from K excluding dimension size for
        Dimi
16:     M-Tablei = calculateMtableRow(MaxDotsi,DimSizes)
17:   end for
18: end loop
```

The pseudo code for the *calculatedMtableRow* function is given in Algorithm 8. The inputs to the algorithm are: (i) the number of maximum values to be returned (thus the size of M-Table row under consideration) and (ii) the dimension sizes (excluding the current dimension). The output is a sequence of maximum distances, starting with the greatest distance, which become a row in a desired M-Table. On start up the location which will feature the maximum distance is identified and stored in the set *Locs* (Line 5). Recall that this location is a tuple of the form $\langle c_1, c_2, \ldots \rangle$ where each coordinate value corresponds to one of the dimension specified for the data space of interest excluding the current dimension (the dimension whose indexes we wish to rearrange). The associated maximum distance is then calculated and stored in the set *Dists* (Line 6). These row sets are updated as the algorithm progresses.

The algorithm then continues, in an iterative manner, according to the $numValues$ input parameter. On each iteration the longest distance $dist_j$ is extracted from the set $Dists$ (Line 8). The set $Locs$ is then pruned (Line 10) by removing the location loc_j associated with the maximum distance $dist_j$ identified in the previous line. The $Dists$ set is also pruned by removing $dist_j$ (Line 11). We then identify the two locations immediately above and to the left of loc_j (assuming the origin of the space under consideration is in the top-left hand corner) and store them in $NewLocs$ (Line 12). The associated set of distances are also calculated and stored in $NewDists$ (Line 13). In some cases, if we have reached either the top or left boundary of the data space, only one location will be generated. Given a data space populated entirely with dots the origin location will eventually be reached and no new locations will be generated. The sets $NewLocs$ and $NewDists$ are then merged with the existing (pruned) sets $Locs$ and $Dists$ such that no repetitions are included (Lines 14 and 15). The process repeats in this manner until the required maximum number of values is reached.

Algorithm 8: Calculate M-Table Row Algorithm

```
1:  Function: calculateMtableRow
2:  Input: numValues = the number of "maximum" values to be returned
3:  DimSizes = {k1, k2, ...} The dimension sizes excluding the current dimension
4:  output: Row = A list of maximum values of length numValues
5:  Locs = {loc1} = {⟨k1, k2, ...⟩}
6:  Dists = {dist1} = {distCalc(loc1)}
7:  for (i = 0 to i = numValues) do
8:      dist_j =getLongestdistIndex(Dists);
9:      Row[i] = dist_j
10:     Locs = (Locs − loc_j) Prune location loc_j from Locs
11:     Dists = (Dist − dist_j) Prune distance dist_j from Dists
12:     NewLocs = calculateNewLocations (loc_j)
13:     NewDists = calculateNewDistances (NewLocs)
14:     Locs = Locs ∪ NewLocs
15:     Dists = Dists ∪ NewDists
16: end for
```

6.3.2 M-Table Construction Example

This subsection presents an example to illustrate the operation of the MDC algorithm using a 3D configuration measuring $8 \times 5 \times 6$, in otherwords $K = \{8, 5, 6\}$ corresponding to Dim_1, Dim_2 and Dim_3 respectively. If we assume that the maximum number of dots associated with each dimension in this case is ten. Thus we need ten maximum values with respect to each dimension. It should be recalled that the maximum number of dots per dimension are not necessarily equal, however with respect to the example presented in this section an equal maximum number of dots was assumed. Recall from above that the number of dimensions to be considered when calculating M-Table values is always one less than $|DIM|$ (the maximum number of dimensions). Therefore in the case of the

example presented here, the MDC algorithm calculates the maximum values for each Dim_x, Dim_y and Dim_z in context of: (i) a 5×6 space, (ii) a 8×6 space and (iii) a 8×5 space. Figure 6.3 presents the order in which distance values were selected. Figures 6.3(a), (c) and (e) with respect to Euclidean distance calculation, and Figures 6.3(b), (d) and (f) with respect Manhattan distance calculation. In each case the first value, value 0 is at the bottom-right hand corner; after that the location of the following values varies.

Table 6.2 shows the associated calculations with respect to the orderings presented in Figures 6.3. In the tables the first column, loc_j, gives the location identifier; the second column the associated coordinates for the location; the third column the Euclidean or Manhattan distance calculation as appropriate; and the fourth (final) column the consequent distance. The associated M-Tables (Euclidean and Manhattan) are gven in Figures 6.4(a) and (b).

6.4 The Exact 3D Banded Pattern Mining (E3D-BPM) Algorithm

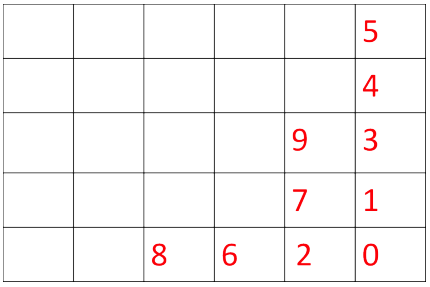
This section considers the proposed E3D-BPM algorithm in more detail. The section is divided into two subsections. The first, Subsection 6.4.1 considers the operation of the algorithm; pseudo code describing the algorithm is presented and discussed. Subsection 6.4.2 then gives a worked example of the algorithm's operation.

6.4.1 The E3D-BPM Algorithms

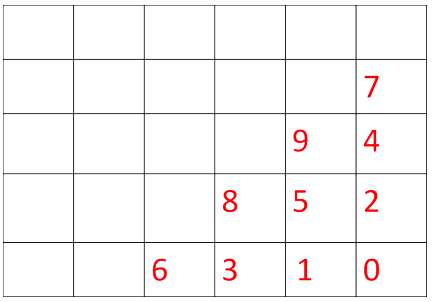
The pseudo code for the proposed E3D-BPM algorithm is presented in Algorithm 9. The inputs (Lines 1-3) are: (i) the set of dimensions $DIM = \{Dim_x, Dim_y, Dim_z\}$, (ii) a zero-one data set D and (iii) a maximum number of iterations *counter*. The output (Line 4) is a rearranged data space that serve to minimise the GBS value. As in the case of the 2D-BPM and A3D-BPM algorithms presented earlier in Chapters 4 and 5 respectively, the E3D-BPM algorithms proceeds in an iterative manner. On each iteration the indexes in the dimensions are rearranged according to the calculated banding scores (Line 12). Banding scores are calculated using Equation 6.2 and either Euclidean or Manhattan distance calculation with or without recourse to an M-Table (see above). This process continues until either: (i) the GBS is minimised or (ii) the number of iterations is reached. A worked example illustrating how the E3D-BPM algorithm operates is presented in the following subsection, Subsection 6.4.2 below.

6.4.2 A Working Example Using the E3D-BPM Algorithms

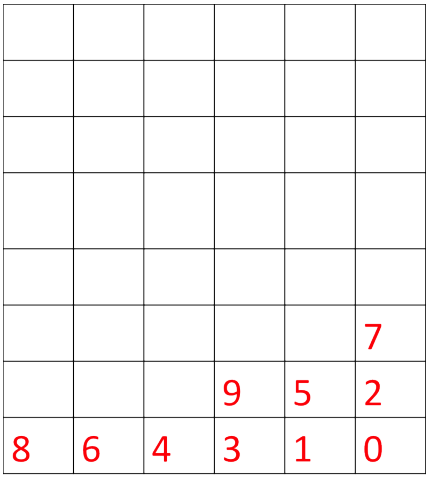
This subsection presents a working example to illustrate the operation of the E3D-BPM algorithm using the dot configuration used to describe the operation of the A3D-BPM algorithm in the previous chapter. More specifically the configuration shown, from



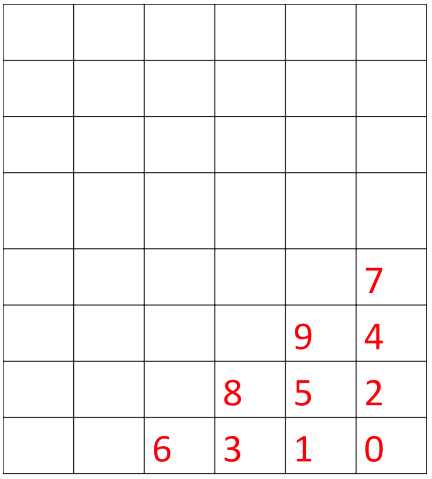
(a) $Dim_2 \times Dim_3$



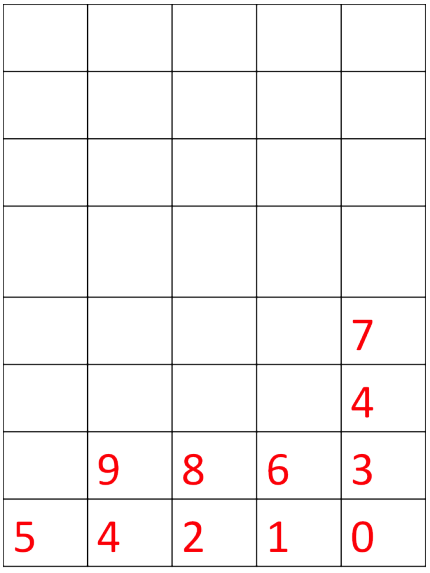
(b) $Dim_2 \times Dim_3$



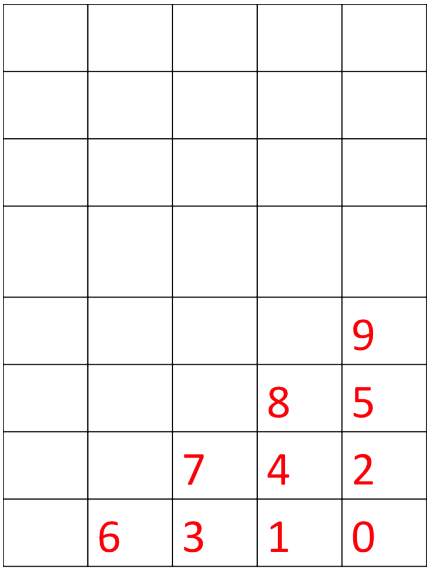
(c) $Dim_1 \times Dim_3$



(d) $Dim_1 \times Dim_3$



(e) $Dim_1 \times Dim_2$



(f) $Dim_1 \times Dim_2$

FIGURE 6.3: The order in which locations are selected when generating M-Tables using Euclidean and Manhattan distance calculation, for the spaces 5×6, 8×6 and 8×5.

three different perspectives, in Figures 5.3, 5.4 and 5.5 in Section 5.4 of Chapter 5. The operation of both the Euclidean and Manhattan variations of the E3D-BPM algorithm

TABLE 6.2: MDC using Euclidean and Manhattan distance calculations

loc_j	$index(i,j)$	Euclid.	Dist
loc_0	$(4,5)$	$\sqrt{4^2+5^2}$	6.4031
loc_1	$(3,5)$	$\sqrt{3^2+5^2}$	5.8309
loc_2	$(4,4)$	$\sqrt{4^2+4^2}$	5.6568
loc_3	$(2,5)$	$\sqrt{2^2+5^2}$	5.3851
loc_4	$(1,5)$	$\sqrt{1^2+5^2}$	5.0990
loc_5	$(0,5)$	$\sqrt{0^2+5^2}$	5.0000
loc_6	$(4,3)$	$\sqrt{4^2+3^2}$	5.0000
loc_7	$(3,4)$	$\sqrt{3^2+4^2}$	5.0000
loc_8	$(4,2)$	$\sqrt{4^2+2^2}$	4.4721
loc_9	$(2,4)$	$\sqrt{2^2+4^2}$	4.4721

(a)

loc_j	$index(i,j)$	Manhat.	Dist
loc_0	$(4,5)$	$(4+5)$	9
loc_1	$(4,4)$	$(4+4)$	8
loc_2	$(3,5)$	$(3+5)$	8
loc_3	$(4,3)$	$(4+3)$	7
loc_4	$(5,2)$	$(5+2)$	7
loc_5	$(4,3)$	$(3+4)$	7
loc_6	$(2,4)$	$(2+4)$	6
loc_7	$(1,5)$	$(1+5)$	6
loc_8	$(3,3)$	$(3+3)$	6
loc_9	$(2,4)$	$(2+4)$	6

(b)

loc_j	$index(i,j)$	Euclid.	Dist
loc_0	$(7,5)$	$\sqrt{7^2+5^2}$	8.6023
loc_1	$(7,4)$	$\sqrt{7^2+3^2}$	8.0622
loc_2	$(6,5)$	$\sqrt{6^2+5^2}$	7.8102
loc_3	$(7,3)$	$\sqrt{7^2+3^2}$	7.6157
loc_4	$(7,2)$	$\sqrt{7^2+2^2}$	7.2801
loc_5	$(6,4)$	$\sqrt{6^2+4^2}$	7.2111
loc_6	$(7,1)$	$\sqrt{7^2+1^2}$	7.0710
loc_7	$(5,4)$	$\sqrt{5^2+4^2}$	7.0710
loc_8	$(7,0)$	$\sqrt{7^2+0^2}$	7.0000
loc_9	$(6,3)$	$\sqrt{6^2+3^2}$	6.7082

(c)

loc_j	$index(i,j)$	Manhat.	Dist
loc_0	$(7,5)$	$(7+5)$	12
loc_1	$(7,4)$	$(7+4)$	11
loc_2	$(6,5)$	$(6+5)$	11
loc_3	$(7,3)$	$(7+3)$	10
loc_4	$(5,5)$	$(5+5)$	10
loc_5	$(6,4)$	$(6+4)$	10
loc_6	$(7,2)$	$(7+2)$	9
loc_7	$(4,5)$	$(4+5)$	9
loc_8	$(6,3)$	$(6+3)$	9
loc_9	$(5,4)$	$(5+4)$	9

(d)

loc_j	$index(i,j)$	Euclid.	Dist
loc_0	$(7,4)$	$\sqrt{7^2+4^2}$	8.0622
loc_1	$(7,3)$	$\sqrt{7^2+3^2}$	7.6157
loc_2	$(7,2)$	$\sqrt{7^2+2^2}$	7.2801
loc_3	$(6,4)$	$\sqrt{6^2+4^2}$	7.2111
loc_4	$(7,1)$	$\sqrt{7^2+1^2}$	7.0710
loc_5	$(7,0)$	$\sqrt{7^2+0^2}$	7.0000
loc_6	$(6,3)$	$\sqrt{6^2+3^2}$	6.7082
loc_7	$(5,4)$	$\sqrt{5^2+4^2}$	6.4031
loc_8	$(6,2)$	$\sqrt{6^2+2^2}$	6.3245
loc_9	$(6,1)$	$\sqrt{6^2+1^2}$	6.0827

(e)

loc_j	$index(i,j)$	Manhat.	Dist
loc_0	$(7,4)$	$(7+4)$	11
loc_1	$(7,3)$	$(7+3)$	10
loc_2	$(6,4)$	$(6+4)$	10
loc_3	$(7,2)$	$(7+2)$	9
loc_4	$(6,3)$	$(6+3)$	9
loc_5	$(5,4)$	$(5+4)$	9
loc_6	$(7,1)$	$(7+1)$	8
loc_7	$(6,2)$	$(6+2)$	8
loc_8	$(5,3)$	$(5+3)$	8
loc_9	$(4,4)$	$(4+4)$	8

(f)

will be considered so that the distinction between the operation of the variations can be made clear.

Considering the Euclidean E3D-BPM algorithm first; the set of banding scores for the Dim_x indexes locations are as follows: $bs_{x_0} = 0.4730$, $bs_{x_1} = 0.5581$, $bs_{x_2} = 0.0000$,

	1	2	3	4	5	6	7	8	9	10
Dim_1	6.4031	5.8309	5.6568	5.3851	5.0990	5.0000	5.0000	5.0000	4.4721	4.4721
Dim_2	8.6023	8.0622	7.8102	7.6157	7.2801	7.2111	7.0710	7.0710	7.0000	6.7082
Dim_3	8.0622	7.6157	7.2801	7.2111	7.0710	7.0000	6.7082	6.4031	6.3245	6.0827

(a)

	1	2	3	4	5	6	7	8	9	10
Dim_1	9	8	8	7	7	7	6	6	6	6
Dim_2	12	11	11	10	10	10	9	9	9	9
Dim_3	11	10	10	9	9	9	8	8	8	8

(b)

FIGURE 6.4: Example M-Tables (Euclidean and Manhattan) for the illustration of the operation of the MDC algorithm given in Section 6.3.2

$bs_{x_3} = 0.6154$, $bs_{x_4} = 0.4157$. The banding scores for Dim_y are: $bs_{y_0} = 0.1212$, $bs_{y_1} = 0.6546$, $bs_{y_2} = 0.6715$, $bs_{y_3} = 0.6063$, $bs_{y_4} = 0.7712$, $bs_{y_5} = 0.4851$. The banding scores for Dim_z are: $bs_{z_0} = 0.3126$ and $bs_{z_1} = 0.7657$. The indexes in Dim_x, Dim_y and Dim_z are thus rearranged accordingly to produce the configuration shown, from three different perspectives, in Figures 6.5, 6.6 and 6.7.

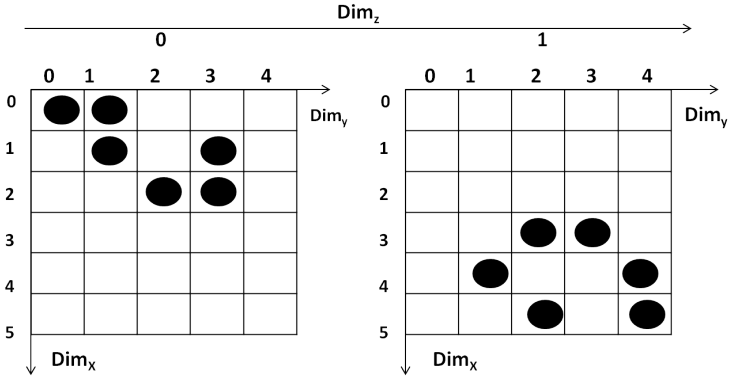


FIGURE 6.5: Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 1

The GBS values for Dim_x, Dim_y and Dim_z are then $GBS_x = 0.2788$, $GBS_y = 0.3439$ and $GBS_z = 0.1042$; and the total GBS value is:

$$GBS = \frac{0.2788 + 0.3439 + 0.1042}{3} = 0.2423$$

Algorithm 9: The E3D-BPM Algorithm

Input: DIM = a set of dimensions $\{Dim_x, Dim_y, Dim_z\}$ comprised of a set of indexes
D = binary valued data matrix subscribing to DIM
$counter$ = a maximum number of iterations
Output: A rearranged data space D that serves to minimise GBS
$GBS_{sofar} = 1.0$
loop
  if $(counter == 0)$ **then**
    $break$
  end if
  for $i = 0$ **to** $i = |DIM|$ **do**
    for $j = 0$ **to** $|Dim_i|$ **do**
      Calculate bs_{i_j} Banding score for current index j in Dim_i using Equation 6.2
    end for
    DIM' = Rearranged Dim_i according to banding scores for Dim_i
    D' = D Rearranged according to DIM'_i
  end for
  GBS_x = calculate GBS for Dim_x using Equation 6.5
  GBS_y = calculate GBS for Dim_y using Equation 6.6
  GBS_z = calculate GBS for Dim_z using Equation 6.7
  GBS_{new} = Global banding score for DIM' using Equation 6.8
  if $(GBS_{new} \geq GBS_{sofar})$ **then**
    break
  else
    $DIM = DIM'$, $D = D'$, $GBS_{sofar} = GBS_{new}$
  end if
  $counter = counter - 1$
end loop
Exit with D and GBS

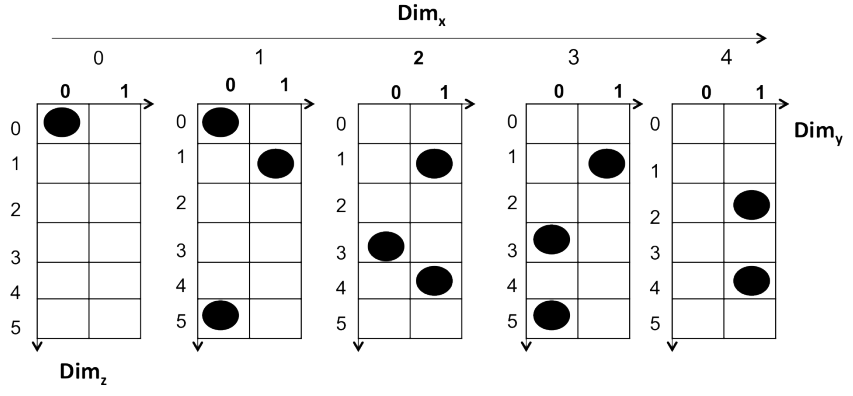


FIGURE 6.6: Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 2

On the second iterations, the set of banding scores for the Dim_x are now: $bs_{x_0} = 0.0000$, $bs_{x_1} = 0.3349$, $bs_{x_2} = 0.6708$, $bs_{x_3} = 0.4028$, $bs_{x_4} = 0.9043$. For Dim_y the banding scores are: $bs_{y_0} = 0.1212$, $bs_{y_1} = 0.3638$, $bs_{y_2} = 0.6063$, $bs_{y_3} = 0.6565$, $bs_{y_4} =$

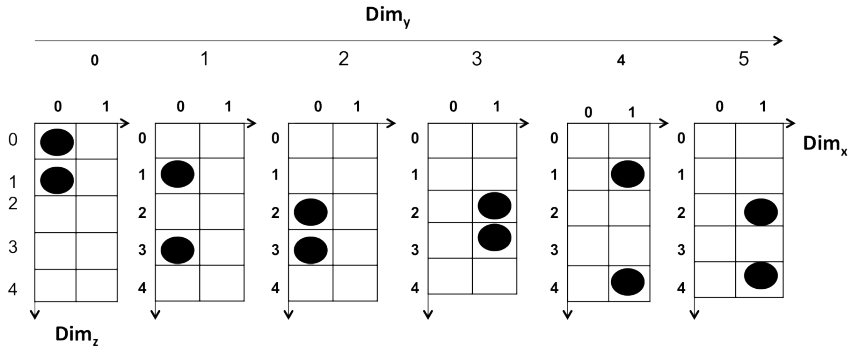


FIGURE 6.7: Input Data rearranged using Euclidean E3D-BPM after the first iteration, perspective 3

0.6715 and $bs_{y_5} = 0.8835$. And for Dim_z the banding scores are: $bs_{z_0} = 0.2546$ and $bs_{z_1} = 0.7864$. Using this set of scores the items in Dim_x, Dim_y and Dim_z are again rearranged as shown, again from three different perspectives, in Figures 6.8, 6.9 and 6.10.

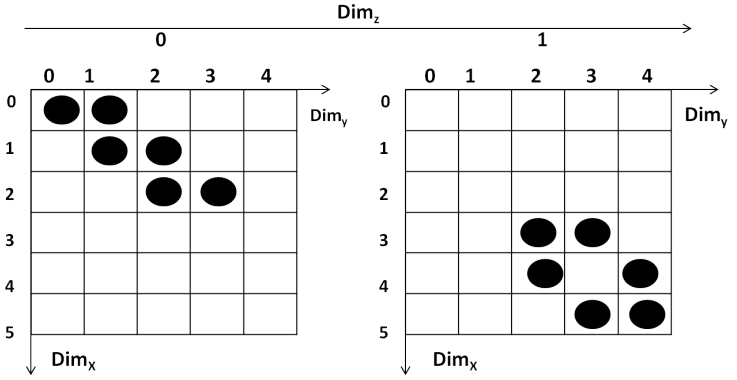


FIGURE 6.8: Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 1

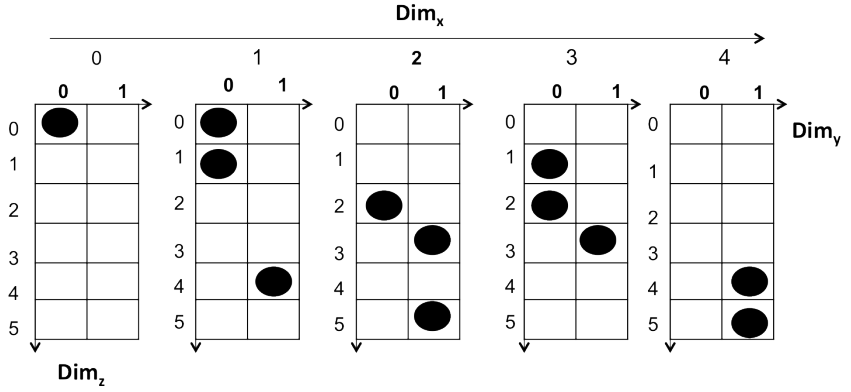


FIGURE 6.9: Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 2

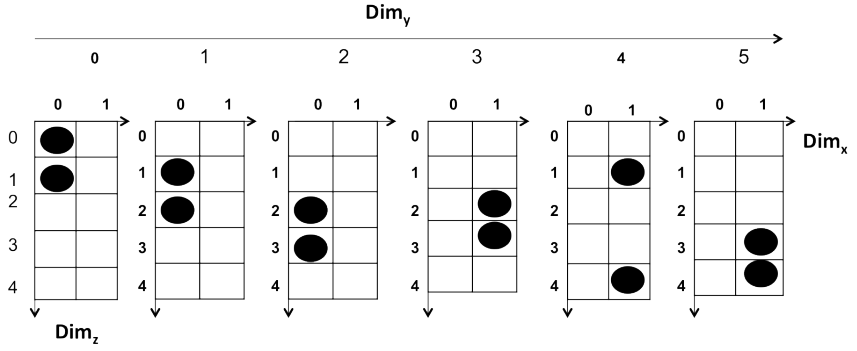


FIGURE 6.10: Input Data rearranged using Euclidean E3D-BPM after the second iteration, perspective 3

The GBS values for Dim_x, Dim_y and Dim_z are now: $GBS_x = 0.1833$, $GBS_y = 0.2793$ and $GBS_z = 0.0849$; and the overall total GBS value is now:

$$GBS = \frac{0.1833 + 0.2793 + 0.0849}{3} = 0.1825$$

However, no changes have been made on this second iteration so the process terminates.

We will now consider the Manhattan E3D-BPM variation with respect to the same input data as used for the above illustration of the Euclidean E3D-BPM variation. In this case the set of banding scores obtained for the Dim_x are: $bs_{x_0} = 0.4444$, $bs_{x_1} = 0.5556$, $bs_{x_2} = 0.0000$, $bs_{x_3} = 0.5556$ and $bs_{x_4} = 06667$. Similarly, the set of banding scores for Dim_y are: $bs_{y_0} = 0.2000$, $bs_{y_1} = 0.6000$, $bs_{y_2} = 0.6000$, $bs_{y_3} = 0.4000$, $bs_{y_4} = 0.7000$ and $bs_{y_5} = 0.3000$. The Dim_z banding scores are then: $bs_{z_0} = 0.4630$ and $bs_{z_1} = 0.5000$. As a consequence the indexes in the Dim_x, Dim_y and Dim_z are rearranged accordingly so as to produce the configuration shown (from three different perspectives) in Figures 6.11, 6.12 and 6.13.

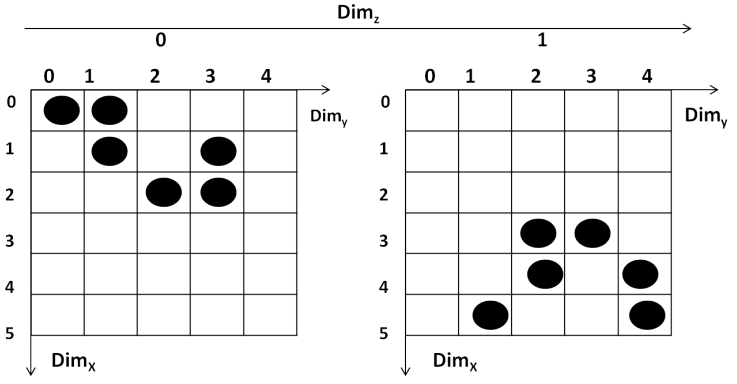


FIGURE 6.11: Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 1

The GBS values for Dim_x, Dim_y and Dim_z are now: $GBS_x = 0.2667$, $GBS_y = 0.3190$ and $GBS_z = 0.1543$; the total GBS value on completion of this first iteration is:

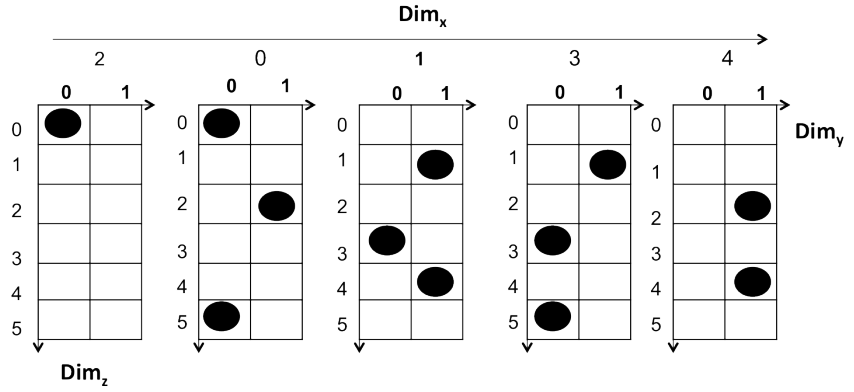


FIGURE 6.12: Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 2

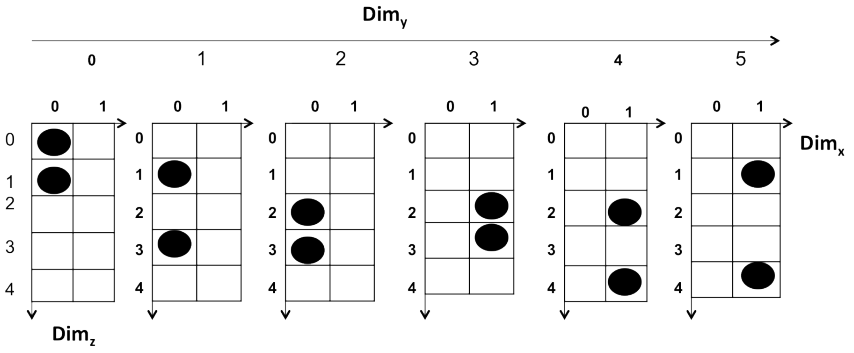


FIGURE 6.13: Input Data rearranged using Manhattan E3D-BPM after the first iteration, perspective 3

$$GBS = \frac{0.2667 + 0.3190 + 0.1543}{3} = 0.2467$$

Better than the 1.0 default start value, thus we proceed with a second itteration.

On the second iteration, the set of banding scores for Dim_x are: $bs_{x_0} = 0.0000$, $bs_{x_1} = 0.5000$, $bs_{x_2} = 0.7778$, $bs_{x_3} = 0.5556$ and $bs_{x_4} = 0.6667$. For Dim_y the banding scores are: $bs_{y_0} = 0.1000$, $bs_{y_1} = 0.7000$, $bs_{y_2} = 0.6000$, $bs_{y_3} = 0.5000$, $bs_{y_4} = 0.2000$, $bs_{y_5} = 0.9000$. And for Dim_z the banding scores are: $bs_{z_0} = 0.4630$ and $bs_{z_1} = 0.5926$. Using this set of scores the items in Dim_z, Dim_x and Dim_y are again rearranged as shown, with respect to the three perspectives presented used previously, in Figures 6.14, 6.15 and 6.16.

The GBS values for Dim_x, Dim_y and Dim_z are now: $GBS_x = 0.2482$, $GBS_y = 0.3000$ and $GBS_z = 0.1543$. The total GBS value at the end of the iteration two is thus:

$$GBS = \frac{0.2482 + 0.3000 + 0.1543}{3} = 0.2342$$

Better than the 0.2342 recorded previously, however, no changes have been made hence the process terminates. From the above it should be noted that the resulting final GBS values are different when using Euclidean and Manhattan E3D-BPM because they are

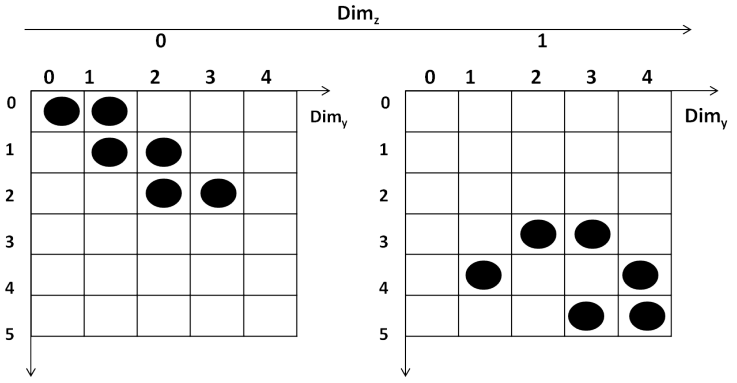


FIGURE 6.14: Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 1

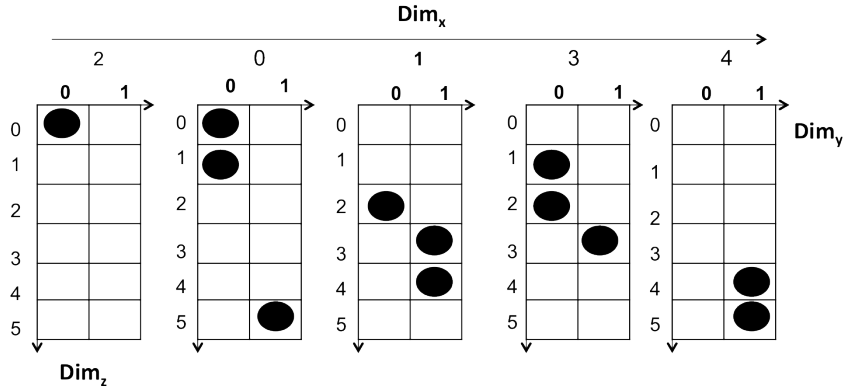


FIGURE 6.15: Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 2

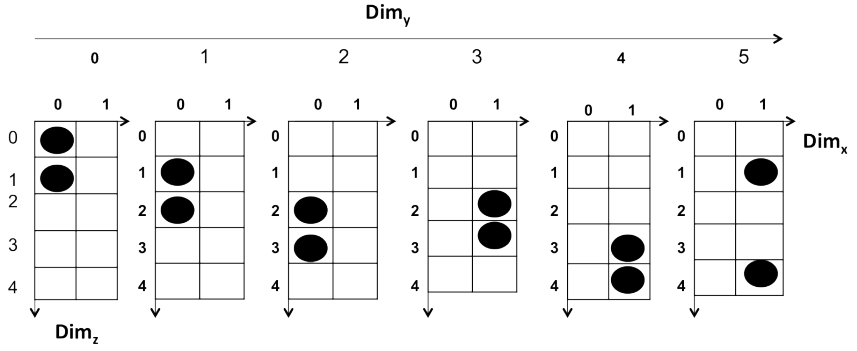


FIGURE 6.16: Input Data rearranged using Manhattan E3D-BPM after the second iteration, perspective 3

calculated differently; 0.1825 and 0.2423 respectively. Although the GBS associated with the Euclidean variation is better, in this simple example the same bandings are produced. As will be shown later in this chapter this not the case with respect to more complex examples.

6.5 Evaluation of E3D-BPM Mechanism

This section reports on the experimental analysis conducted to evaluate the performance of the proposed E3D-BPM algorithm presented in this chapter. The objectives of the evaluation were:

1. **Efficency:** To compare the operation of the A3D-BPM algorithm, presented previously in Chapter 5, and the E3D-BPM algorithm presented in this chapter, in terms of the runtime efficiency of the banding process in each case.
2. **M-Tables:** To determine whether, in relation to the E3D-BPM algorithm, it is better to use the concept of M-Tables or not.
3. **Effectiveness:** To compare the operation of the A3D-BPM algorithm, presented previously in Chapter 5, and the E3D-BPM algorithm presented in this chapter, in terms of the quality of the bandings produced in each case.

Recall that when using M-Tables maximum distance values are only calculated once, whislt when not using M-Tables the likelihood is that maximum distance values will be calculated many times.

The evaluation was conducted using the 3D CTS data sets introduced in Section 3.4 of Chapter 3. Recall that 48 3D data sets were extracted from the CTS database covering four selected counties and the years 2003, 2004, 2005 and 2006. The 3D CTS data sets was divided into three equal sized groups. For the first group, the Eastings data sets, the dimensions were: (i) Records, (ii) Attributes and (iii) Eastings. For the second group, the Northings data sets, the dimensions were: (i) Records, (ii) Attributes and (iii) Northings. And for the third group, the Temporal data sets, the dimensions were: (i) Records, (ii) Attributes and (iii) Time.

The outcomes from the experiments related to the first and second of the above objectives are presented in Subsection 6.5.1. It was anticipated that the quality of the bandings produced using the E3D-BPM algorithm would be better than those produced using the A3D-BPM algorithm; while the A3D-BPM algorithm would be more efficient than the E3D-BPM algorithm. The outcomes from the experiments related to the third of the above objectives are presented in Subsection 6.5.2. Note that in this case the E3D-BPM algorithm was run using both the Euclidean and Manhattan variations of the algorithm.

6.5.1 Comparison Between E3D-BPM And A3D-BPM Algorithms In Term of Run-times

This section considers the results from the comparative evaluation of the E3D-BPM and A3D-BPM algorithms in terms of the runtime. Runtime values using both Euclidean and Manhattan distance measurement and with and without the usage of M-Table, were obtained. The results are presented in Tables 6.3, 6.4 and 6.5. Table 6.3 shows the

results obtained using the Eastings data sets, Table 6.4 shows the results obtained using the Northings data sets and Table 6.5 shows the results obtained using the Temporal data sets. Each table includes a column indicating the number of records in each data set.

From Tables 6.3, 6.4 and 6.5, it can be seen, as might be expected, that there is a correlation between the number of records in the data sets and the run times, as the number of records increased the processing time increased correspondingly. As also anticipated, Manhattan distance calculation was more efficient than Euclidean distance calculation because it is simpler (both when using M-Tables and when not using M-Tables). More specifically the complexity of Manhattan distance calculation is given by $O(K_1(n-1))$ where n is the number of dimensions and K_1 is the complexity of an addition operation. In the case of Euclidean distance calculation, the complexity is given by $O(K_1(n-1)+K_2n+K_3)$, where K_2 is the complexity of a multiplication operation and K_3 is the complexity of a square root operation. Although the values of K_1, K_2 and K_3 vary according to how the associated operations are implemented the relationship $K_1 < K_2 < K_3$ usually holds. Returning to Tables 6.3, 6.4 and 6.5, it is interesting to note that there is a significant difference in run time between using M-Tables and not using M-Tables; using M-Tables is significantly more efficient. This was because, although the same maximum distances were used repeatedly, the calculation of the M-Tables is done only once, whilst when not using M-Tables the same maximum distances are calculate repeatedly, thereby introducing an additional computational overhead. There are mechanisms whereby the M-Table pre-calculation can further be improved and these are included as items for future work presented in Chapter 11. With respect to the comparison between the E3D-BPM and A3D-BPM algorithms, the approximate algorithm, as conjectured in the previous chapter, was the fastest. Although on each iteration of the A3D-BPM algorithm, more reordering is done because each dimension is considered with respect to each other dimension (thus six reorderings on each iteration compared to only three for E3D-BPM), there is much less calculation; although coupling the E3D-BPM algorithm with the usage of M-Tables does speed up its operation.

6.5.2 Comparison Between E3D-BPM And A3D-BPM Algorithms In Term of Global Banding Score (GBS) Values

This sub-section considers the results from the comparative evaluation conducted with respect to the E3D-BPM and A3D-BPM algorithms in terms of the bandings produced. This was measured in terms of GBS values. The results are presented in Tables 6.6 to Tables 6.8. Table 6.6 presents the results using the sixteen Eastings data sets, Table 6.7 presents the results using the Northings data sets, whilst Table 6.8 presents the results using the Temporal data sets. The naming convention used in the tables for the different variations of the E3D-BPM algorithm are: (i) "E3D-BPM$_M$" for Manhattan E3D-BPM, (ii) "E3D-BPM$_E$" for Euclidean E3D-BPM and (iii) "A3D-BPM" for the approximate 3D-BPM algorithm. Note that with respect to the tables the GBS results

TABLE 6.3: Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Eastings data sets.

Data Sets	# Recs.	runtime (sec)				
		E3D-BPM M-Tab.		E3D-BPM no M-Tab.		A3D-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-2003	178172	358.90	478.14	492.12	842.95	348.24
Abd-2004	173612	365.43	479.54	437.77	882.33	318.52
Abd-2005	157033	292.26	396.83	406.62	852.33	281.34
Abd-2006	236206	536.86	758.14	713.44	1248.21	374.40
Corn-2003	170245	440.89	648.22	499.65	831.94	277.99
Corn-2004	169053	333.35	483.91	442.25	845.95	228.80
Corn-2005	154589	299.45	433.47	412.25	791.20	249.03
Corn-2006	167281	341.27	438.91	414.39	905.88	316.35
Lanc-2003	167919	306.91	424.64	438.08	798.30	276.74
Lanc-2004	217566	559.22	741.41	670.54	1058.08	376.69
Lanc-2005	157142	253.89	387.12	402.26	844.84	201.93
Lanc-2006	196290	409.25	452.03	529.62	920.02	322.19
Nolf-2003	46977	42.83	58.90	47.09	94.69	30.99
Nolf-2004	46246	35.47	51.42	50.82	101.73	19.33
Nolf-2005	35914	20.91	49.61	37.98	105.12	12.75
Nolf-2006	45150	40.55	51.11	50.75	113.45	19.37
Average	144961	289.84	395.84	377.85	702.31	228.42

TABLE 6.4: Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Northings data sets

Data Sets	# Recs.	runtime (sec)				
		E3D-BPM M-Tab.		E3D-BPM no M-Tab.		A3D-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-2003	178172	443.70	661.73	471.11	942.95	376.23
Abd-2004	173612	426.87	582.33	453.52	862.85	377.60
Abd-2005	157033	364.24	508.62	466.01	742.85	273.93
Abd-2006	236206	505.28	1753.57	773.10	1121.4	455.99
Corn-2003	170243	336.52	447.06	446.47	822.80	317.46
Corn-2004	169053	355.01	497.07	458.72	861.86	322.62
Corn-2005	154589	326.65	448.06	412.57	865.66	284.15
Corn-2006	167281	390.26	467.53	470.10	828.12	313.20
Lanc-2003	167919	400.48	497.84	506.77	898.30	231.86
Lanc-2004	217566	461.83	592.33	542.88	1058.08	416.50
Lanc-2005	157142	322.43	463.37	435.09	844.84	278.40
Lanc-2006	196292	445.15	670.21	530.11	920.02	395.76
Nolf-2003	46977	40.89	60.58	49.93	93.72	29.86
Nolf-2004	46246	34.50	57.47	48.79	108.89	14.71
Nolf-2005	35914	27.40	54.80	46.58	108.37	14.15
Nolf-2006	45150	40.42	58.74	33.26	104.17	19.45
Average	144961	307.60	426.33	384.06	699.06	258.24

with and without the usage of M-Table are the same so only a single result is presented). Inspection of the results presented in the three tables confirms firstly, as expected, that the E3D-BPM algorithm (both variations) produced better bandings than the A3D-BPM algorithm. The difference between the operation of the Euclidean E3D-BPM and Manhattan E3D-BPM algorithms are because the first is better at differentiating between potential configurations. This can be illustrated by considering the 2D space given in Figure 6.17. Points A and B are both a Manhattan distance of 4 away from

TABLE 6.5: Comparative results in terms of Run time (seconds) using the E3D-BPM (with and without M-Tables) and A3D-BPM Algorithms applied to the Temporal data sets

Data Sets	# Recs.	runtime (sec)				
		E3D-BPM *M*-Tab.		E3D-BPM no *M*-Tab.		A3D-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-2003	178172	434.98	640.27	437.27	813.44	334.73
Abd-2004	173612	303.86	540.07	440.53	707.77	320.81
Abd-2005	157033	311.51	518.53	352.55	785.75	217.27
Abd-2006	236206	593.89	854.01	603.33	1116.56	420.03
Corn-2003	170243	430.64	539.71	441.66	839.71	319.59
Corn-2004	169053	330.95	437.11	418.99	853.78	317.33
Corn-2005	154589	296.25	319.16	395.11	837.36	215.63
Corn-2006	167281	343.16	404.13	483.42	834.71	316.57
Lanc-2003	167919	376.75	434.23	483.26	854.01	217.40
Lanc-2004	217566	601.14	624.96	678.70	1121.95	416.44
Lanc-2005	157142	289.51	327.36	351.30	876.91	216.66
Lanc-2006	196292	504.85	534.35	534.69	912.09	315.99
Nolf-2003	46977	48.05	59.25	51.27	89.35	26.56
Nolf-2004	46246	32.76	44.76	41.38	84.76	15.46
Nolf-2005	35914	26.46	35.67	28.17	80.33	13.49
Nolf-2006	45150	41.38	55.45	45.01	96.63	17.28
Average	144961	310.37	398.06	361.67	681.57	231.33

the origin. However, the Euclidean distance from the origin for point A is "3.1623" ($\sqrt{3^2+1^2}$) while that for point B is "2.8284" ($\sqrt{2^2+2^2}$).

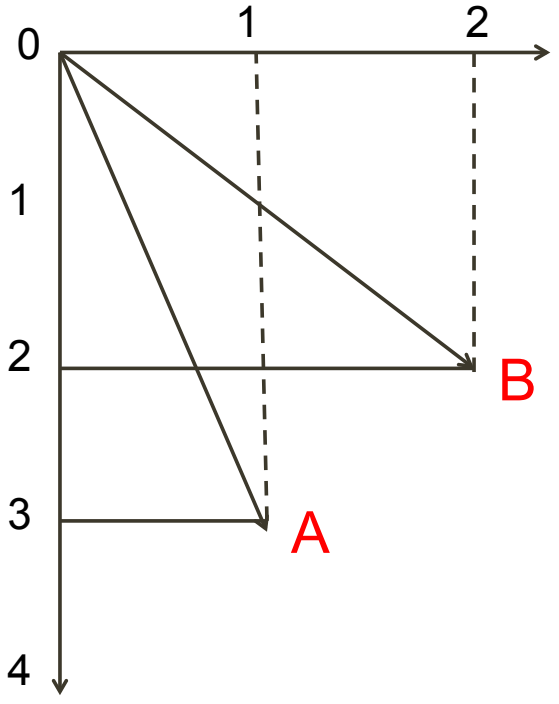


FIGURE 6.17: A 2D space illustrating the distinction between Euclidean and Manhattan distance calculation

TABLE 6.6: Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Eastings data sets (best results in bold font).

Counties	Year	# Recs	GBS		
			E3D-BPM$_M$	E3D-BPM$_E$	A3D-BPM
Aberdeenshire	2003	178172	**0.5533**	**0.5146**	0.6953
	2004	173612	0.6062	**0.5756**	0.6412
	2005	157033	0.6283	**0.5841**	0.7258
	2006	236206	0.6276	**0.6141**	0.6478
Cornwall	2003	170243	0.5007	**0.4763**	0.5360
	2004	169053	0.5118	**0.4924**	0.5936
	2005	154589	0.6032	**0.5347**	0.6135
	2006	167281	0.5553	**0.5106**	0.5933
Lancashire	2003	167919	0.5811	**0.5506**	0.6973
	2004	217566	0.5794	**0.5250**	0.6232
	2005	157142	0.5643	**0.4749**	0.6720
	2006	196292	0.6015	**0.4894**	0.6486
Norfolk	2003	46977	0.5932	**0.5876**	0.7282
	2004	46246	0.6603	**0.6592**	0.7161
	2005	35914	0.5922	**0.5783**	0.5932
	2006	45150	0.5449	**0.5392**	0.6865
Average		144961	0.5818	**0.5442**	0.6507

TABLE 6.7: Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Northing data sets (best results in bold font).

Counties	Year	# Recs	GBS		
			E3D-BPM$_M$	E3D-BPM$_E$	A3D-BPM
Aberdeenshire	2003	178172	0.5564	**0.5500**	0.7174
	2004	173612	0.5822	**0.5467**	0.6488
	2005	157033	0.5963	**0.5745**	0.7715
	2006	236206	0.6572	**0.6224**	0.7697
Cornwall	2003	170245	0.5193	**0.5092**	0.6466
	2004	169053	0.5737	**0.4908**	0.6069
	2005	154589	0.5662	**0.5285**	0.6579
	2006	167281	0.5451	**0.5048**	0.7466
Lancashire	2003	167919	0.5695	**0.5432**	0.5841
	2004	217566	0.5913	**0.4953**	0.6396
	2005	157142	0.5572	**0.5382**	0.5968
	2006	196290	0.5972	**0.5032**	0.6459
Norfolk	2003	46977	0.6407	**0.6097**	0.7747
	2004	46246	0.6952	**0.6622**	0.7940
	2005	35914	0.6287	**0.5940**	0.7323
	2006	45150	0.6309	**0.6124**	0.7498
Average		144961	0.5942	**0.5553**	0.6927

6.6 Summary

This chapter has presented the Exact Banded Pattern Mining (E3D-BPM) algorithm. The algorithm was considered in detail and its operation illustrated using a number of worked examples. Unlike the A3D-BPM algorithm presented in the previous chapter, that only considered dimension pairings when calculating banding scores and consequently produced approximate bandings, the E3D-BPM algorithm considered all dimensions simultaneously and consequently produced exact bandings. The algorithm operated by using distances of dots to the origin of the data space under consideration

TABLE 6.8: Comparative result in terms of GBS Using the E3D-BPM and A3D-BPM Algorithms applied to the Temporal data sets (best results in bold font).

Counties	Year	# Recs	GBS		
			E3D-BPM$_M$	E3D-BPM$_E$	A3D-BPM
Aberdeenshire	2003	178172	0.5721	**0.5163**	0.7996
	2004	173612	0.5531	**0.5314**	0.6352
	2005	157033	0.5638	**0.5625**	0.7110
	2006	236206	0.5457	**0.5255**	0.7517
Cornwall	2003	170245	0.5279	**0.5074**	0.8184
	2004	169053	0.5822	**0.5406**	0.7603
	2005	154589	0.5756	**0.5660**	0.7857
	2006	167281	0.5875	**0.5474**	0.7128
Lancashire	2003	167919	0.5995	**0.5424**	0.8220
	2004	217566	0.5297	**0.4671**	0.7135
	2005	157142	0.5425	**0.5295**	0.7013
	2006	196290	0.6586	**0.6378**	0.8485
Norfolk	2003	46977	0.6407	**0.6053**	0.8659
	2004	46246	0.6730	**0.6681**	0.7128
	2005	35914	0.6046	**0.5351**	0.7070
	2006	45150	0.6604	**0.6332**	0.7675
Average		144961	0.5886	**0.5572**	0.7571

to calculate banding scores. Two mechanisms were identified for calculating distances, Euclidean and Manhattan distance calculation. The banding scores were normalised with respect to maximum distances to the origin. To this end the observation was made that these maximum distances are calculated repeatedly and therefore it might be beneficial to pre-calculate these and store them in what was termed an M-Table. The evaluation of the E3D-BPM algorithm variations was conducted by comparing its operation with the A3D-BPM algorithm presented in the previous chapter, Chapter 5. From the reported evaluation the following overall observations can be made:

1. In terms of GBS, the best bandings were produced using the E3D-BPM algorithm with Euclidean distance calculation.

2. Using the A3D-BPM algorithm is more efficient than using E3D-BPM (regardless of whether Euclidean or Manhattan distance calculation is adopted or whether M-Tables are used or not).

3. Manhattan distance calculation is more efficient than Euclidean distance calculation (as expected).

4. The concept of M-Tables did offer efficiency advantages with respect to the E3D-BPM algorithm.

In the next chapter the work presented in this, and the previous Chapter 5, will be extended by considering bandings in ND data.

Chapter 7

ND Banded Pattern Mining Mechanisms

7.1 Introduction

In the previous chapters, Chapter 5 and Chapter 6, the A3D-BPM and the E3D-BPM algorithms. The reason for considering the 2D and 3D case, which are afterall special forms of the general ND case, was to facilitate reader understanding. This chapter extends the ideas presented in these previous two chapters by considering the adaptation of these algorithms in the context of ND data. More specifically two N-Dimensional Banded Pattern Mining (ND-BPM) algorithms are presented: the Approximate ND (AND) and the Exact ND (END) BPM algorithms. As before two mechanism for distances calculation, Euclidean and Manhattan, were considered. Note that the two ND-BPM algorithms are not significantly different from the 3D-BPM algorithms presented in Chapter 5 and Chapter 6 except that they are directed at the generation of ND bandings which adds an extra level of complexity. To this end a much more sophisticated mechanism for calculating M-Tables is required, this is also presented in this chapter.

The rest of the chapter is organised in a similar manner to the earlier chapters that described banded pattern mining algorithms. Section 7.2 presents some formal definitions to support the discussion of the ND-BPM algorithms, whilst Section 7.3 reviews the M-Table concept in terms of ND. Section 7.4 then consider the proposed AND-BPM and END-BPM algorithms in detail. Section 7.5 presents an evaluation of the AND-BPM and END-BPM algorithms in the context of the CTS data sets introduced in Chapter 3. Finally, Section 7.6 concludes the chapter with a brief summary of the main findings.

7.2 ND Banding Formalism and Calculation of Banding Score

In the context of the research presented in this thesis, an ND data space of interest is conceptualised in terms of a $(k_1 \times k_2 \times k_3 \times \cdots \times k_n)$. As with respect to earlier discussions, this space can be conceived of as being comprised of a $(k_1 \times k_2 \times k_3 \times \cdots \times k_n)$ "hyper-grid", where k_1 is the size of dimension one (Dim_1), k_2 is the size of dimension two (Dim_2) and so on. Note that the dimensions are not necessarily of equal size. Using this conceptualisation each location in this space representing a "one" contains a dot (a hyper-sphere to be more exact), a locaton representing a "zero" is empty. As before the challenge, given an ND data set, is to rearrange the indexes in each dimension so that the dots are arranged along the main diagonal (or as close to it as possible). For ND, the set of dimensions is represented using the notation $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$ where each subset Dim_i comprises a set of indexes. Thus each dot (hyper-sphere) in this ND space will be represented by a set of coordinates (indexes) $\langle c_1, c_2, \ldots, c_n \rangle$ (where n is the number of dimensions) such that $c_1 \in Dim_1$, $c_2 \in Dim_2$ and so on.

In the case of the AND-BPM algorithm, as before, individual banding scores are calculated by only considering dimension pairings. Previously it was conjectured that this approximate approach would result in sufficiently accurate bandings without the need for the extra resource required to calculate exact bandings using an exact BPM algorithm. In the case of 3D it was found that this was not necessarily the case; however, for completeness, the approximate approach is still consider in this chapter. Recall that given two dimensions Dim_i and Dim_j, the banding score for index p in Dim_i with respect to Dim_j, bs_{ij_p}, is calculated as follows:

$$bs_{ij_p} = \frac{\sum_{p=1}^{p=|W|}(w_p)}{\sum_{q=1}^{q=|W|}(|Dim_q| - k + 1)} \tag{7.1}$$

where the set W is the set of Dim_j indexes representing "dots" whose Dim_i coordinate equates to p ($W = \{w_1, w_2, \ldots\}$). Note that Equation 7.1 is identical to Equation 5.1 given in Chapter 5.

The normalised GBS for each dimension i with respect to dimension j (GBS_{ij}) is then calculated as follows:

$$GBS_{ij} = \frac{\sum_{p=1}^{p=k_i} bs_{ij_p}}{k_i} \tag{7.2}$$

Again note that Equation 7.2 is identical to Equation 5.2 given in Chapter 5.

The GBS for a dimension i is then given by the sum of the GBS for the individual pairings divided by the number of pairings (which will be the number of dimensions minus one):

$$GBS_i = \frac{\sum_{j=1}^{j=|DIM|, j \neq i} GBS_{ij}}{|DIM| - 1} \tag{7.3}$$

The normalised overall GBS for the entire configuration is then calculated thus:

$$GBS = \frac{\sum_{i=1}^{i=|DIM|} GBS_i}{|DIM|} \tag{7.4}$$

Putting equations 7.3 and 7.4 together we get:

$$GBS = \frac{\sum_{i=1}^{i=|DIM|} \sum_{j=1}^{j=|DIM|, j \neq i} GBS_{ij}}{|DIM| \times |DIM| - i} \tag{7.5}$$

In the case of the END-BPM, the normalised banding score bs_{i_p} for index p in dimension i is calculated by dividing the sum of the distances that each relevant dot is from the origin by the sum of the maximum distances that the dots can be from the origin:

$$bs_{i_j} = \frac{\sum_{p=1}^{p=|W|} dist(w_p)}{\sum_{q=1}^{q=|M|} m_q} \tag{7.6}$$

where: (i) W is set of dots whose Dim_i index equates to p and (ii) M is a set of maximum distances ($|W| = |M|$). More specifically $W = \{w_1, w_2, \dots\}$, where each element is a tuple describing the coordinates $\langle c_1, c_2, \dots \rangle$ of a dot in terms of the set DIM but excluding the current dimension Dim_i. Note that distances can be calculated in terms of Euclidean or Manhattan distance according to which variation of the END-BPM algorithm is being used.

The normalised GBS for a dimension Dim_i is obtained by adding up all the individual bs_{i_p} scores and dividing by the size of the dimension:

$$GBS_i = \frac{\sum_{p=1}^{p=k_i} bs_{i_p}}{k_i} \tag{7.7}$$

The overall normalised GBS is obtained by adding up all the individual GBS_i and dividing by the total number of dimensions:

$$GBS = \frac{\sum_{i=1}^{i=|DIM|} GBS_i}{|DIM|} \tag{7.8}$$

7.3 M-Tables in ND Space

In Section 6.3 of the previous chapter, Chapter 6, the concept of M-Tables was introduced including the Maximum Distance Calculation (MDC) algorithm. This section considers M-Tables in the context of ND data spaces. As before maximum distances from the origin (zero location) of the ND data space under consideration are calculated in the context of all the dimension in the ND data space under consideration but excluding the current dimension (the dimension we wish to reorder). For ND space, M-Tables are calculated in almost the same manner as for 3D space. In other words using the MDC algorithm given in Algorithms 7 and Algorithm 8. Recall that Algorithm 7 dimensions the desired

M-Table while Algorithm 8 calculates the maximum distance values for a given row in an M-Table. The only distinction between the 3D M-Table algorithm and that required for ND is that in Algorithm 8, Line 12, next locations are calculated by identifying the neighbouring $|DIM| - 1$ locations with respect to the appropriate dimensions (not the current dimension whose indexes are being reordered). In the previous version of the algorithm only two dimensions required consideration. Thus for ND there may be many more "next locations" than in the case of 3D.

7.4 ND Banded Pattern Mining (ND-BPM) Algorithms

This section provides more detail concerning the two variations of the proposed ND-BPM algorithms:

1. The Approximate ND Banded Pattern Mining (AND-BPM) algorithm.
2. The Exact ND Banded Pattern Mining (END-BPM) algorithm.

The section is divided into two subsections. The AND-BPM algorithm is discussed in further detail in Subsection 7.4.1 whilst the END-BPM algorithm is discussed in further detail in the following subsection, Subsection 7.4.2. The section is concluded with Subsection 7.4.3 which considers the theoretical complexity of the proposed ND-BPM algorithms.

7.4.1 Approximate ND Banded Pattern Mining (AND-BPM) Algorithm

The AND-BPM algorithm operates in a similar manner as the A3D-BPM algorithm presented in Chapter 5, the only distinction is with respect to the number of dimensions to be considered. For the AND-BPM algorithm, as noted above, the banding is conducted by considering all possible pairings. The maximum number of pairings can be calculated using Equation 5.8 given in Chapter 5 reproduced and, for convenience, in Equation 7.9. Thus if we have $|DIM| = 5$, as in the case of ND data sets used for evaluation purposes with respect to the work presented in this chapter, the number of pairings will be $5 \times 4 = 20$.

$$Max \text{ pairings} = |DIM| \times (|DIM| - 1) \tag{7.9}$$

The pseudo code for the AND-BPM algorithm is presented in Algorithm 10. The input are (Lines 1 to 3): (i) the set of dimensions $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$ for the data space under consideration, (ii) a dot data set D, comprising a set of tuples of the form $\langle c_1, c_2, \ldots \rangle$, describing the location of each dot in the data space, and (iii) a maximum iteration counter. The output is a rearranged dot data set D that minimises the GBS value (Line 4). The algorithm iteratively loops over the data space. On each iteration the algorithm attempts to rearrange the indexes in the set of dimensions DIM.

It does this by considering all possible dimension pairings pq. For each pairing the bs_{ij_p} value for each index p in dimension Dim_i is calculated with respect to Dim_j (Line 13). The calculated BS values are then used to rearrange the dimension Dim_i (Line 15) and consequently the data space D (Line 16). Once all pairings for dimension Dim_i have been calculated a GBS value for the dimension is calculated (Line 18). Once all dimensions have been considered the final GBS for this iteration is obtained, GBS_{new} (Line 21). If GBS_{new} is worse (higher) than the current GBS value (GBS_{sofar}), or there has been no change (not shown in Algorithm 10), the algorithm exits with the current configuration D (Line 29). Otherwise, D is set to D', and GBS_{new} is set to GBS_{sofar} (Line 25), and the process repeats.

Algorithm 10: The AND-BPM Algorithm

1: **Input:** DIM = a set of dimensions $\{Dim_1, Dim_2, \ldots, Dim_n\}$
2: D =binary valued data matrix subscribing to DIM
3: $counter$ = a maximum number of iterations
4: **Output:** D Rearranged data space that serves to minimise GBS
5: $GBS_{sofar} = 1.0$
6: **loop**
7: **if** $(counter == 0)$ **then**
8: *break*
9: **end if**
10: **for** $i = 1$ **to** $i = |DIM| - 1$ **do**
11: **for** $j = i + 1$ **to** $j = |DIM|$ and $j \neq i$ **do**
12: **for** $p = 1$ **to** $p = |K_i|$ **do**
13: Calculate bs_{ij_p} for index p in Dim_i w.r.t. Dim_i using Equation 7.1 as appropriate
14: **end for**
15: DIM' = Rearranged Dim_i according to bs_{ij_p} for Dim_i
16: D' = D Rearranged according to DIM'_i
17: **end for**
18: GBS_{ij} = Global banding score for each Dim_i with respect to Dim_j using Equation 7.2
19: **end for**
20: GBS_i = Calculated GBS value for Dim_i using Equation 7.3
21: GBS_{new} = Global banding score for DIM' using Equation 7.5
22: **if** $(GBS_{new} \geq GBS_{sofar})$ **then**
23: break
24: **else**
25: $DIM = DIM'$, $D = D'$, $GBS_{sofar} = GBS_{new}$
26: **end if**
27: $counter = counter - 1$
28: **end loop**
29: Exit with D and GBS

7.4.2 Exact ND Banded Pattern Mining (END-BPM) Algorithm

As in the case of the AND-BPM algorithm the END-BPM algorithm operates in a similar manner to the E3D-BPM algorithm. The pseudo code for the END-BPM algorithm (using either Manhattan or Euclidean distance weighting calculation) is presented in Algorithm 11. As before the inputs are (Lines 1 to 3): (i) the set of dimensions $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$ for the dot data space under consideration, (ii) a dot data set D (comprising tuples of the form $\langle c_1, c_2, \ldots \rangle$) and (iii) a maximum number of iterations counter. The output (Line 4) is a rearranged data space D that serves to minimise the GBS value. As in the case of the previously proposed BPM algorithms the END-BPM algorithm iteratively loops over the data space calculating banding scores for each index p in each dimension Dim_i. For each dimension, the bs_{i_p} values are used to rearrange the indexes in the dimension (Line 14). Once all dimensions have been calculated a GBS_{new} value is calculated (Line 18). If GBS_{new} is worse (higher) than the current GBS value (GBS_{sofar}) the algorithm exits with the current configuration D (Line 24). Otherwise D is set to D' and GBS_{sofar} to GBS_{new} (Line 20), and the process is repeated.

Algorithm 11: The END-BPM Algorithm

```
1:  Input DIM, a set of dimensions {Dim_1, Dim_2, ..., Dim_n}
2:  D, binary valued data matrix subscribing to DIM
3:  counter = a maximum number of iterations
4:  Output: D Rearranged data space that serves to minimise GBS
5:  GBS_sofar = 1.0
6:  loop
7:    if (counter == 0) then
8:      break
9:    end if
10:   for i = 1 to i = |DIM| do
11:     for p = 1 to p = k_i do
12:       Calculate bs_{i_p} for current index p Dim_i using Equation 7.6
13:     end for
14:     DIM' = Rearranged Dim_i according to bs_{i_j} for Dim_i
15:     D' = D Rearranged according to for DIM'_i
16:   end for
17:   GBS_i = Calculated GBS value for Dim_i using Equation 7.7
18:   GBS_new = Global banding score for DIM' using Equation 7.8
19:   if (GBS_new ≥ GBS_sofar) then
20:     DIM = DIM', D = D', GBS_sofar = GBS_new
21:   end if
22:   counter = counter − 1
23: end loop
24: Exit with D and GBS
```

7.4.3 Theoretical Complexity of the ND Banded Pattern Mining (ND-BPM) Algorithm

The theoretical complexity of the ND-BPM algorithms is largely founded on the number of times that the indexes in each dimension are rearranged on a single iteration of the algorithm. Considering a single dimension and the AND-BPM algorithm, the complexity of the banding identification can be said to be $O(n-1)$ (where n is the number of dimensions) because dimension pairings are considered. In the case of the END-BPM algorithm, and considering only one dimension the complexity of the banding score calculation is then given by $O(1)$ because banding score are calculated with respect to all other dimensions. Taking into account the overall number of dimension rearrangements that take place on each iteration; the complexity of the AND-BPM algorithm, per iteration, is given by $O(n(n-1))$; while for the END-BPM algorithm it given by $O(n)$. Note that in 2D the complexity is the same for both algorithms; but as n is increased the complexity increases in a linear manner with respect to the END-BPM algorithm, and in an exponential manner with respect to the AND-BPM algorithm (Figure 7.1). However, as will be demonstrated later in the evaluation section included in this chapter, Section 7.5, there are some further subtleties in the calculation of banding scores that makes the AND-BPM algorithm faster than the END-BPM algorithm.

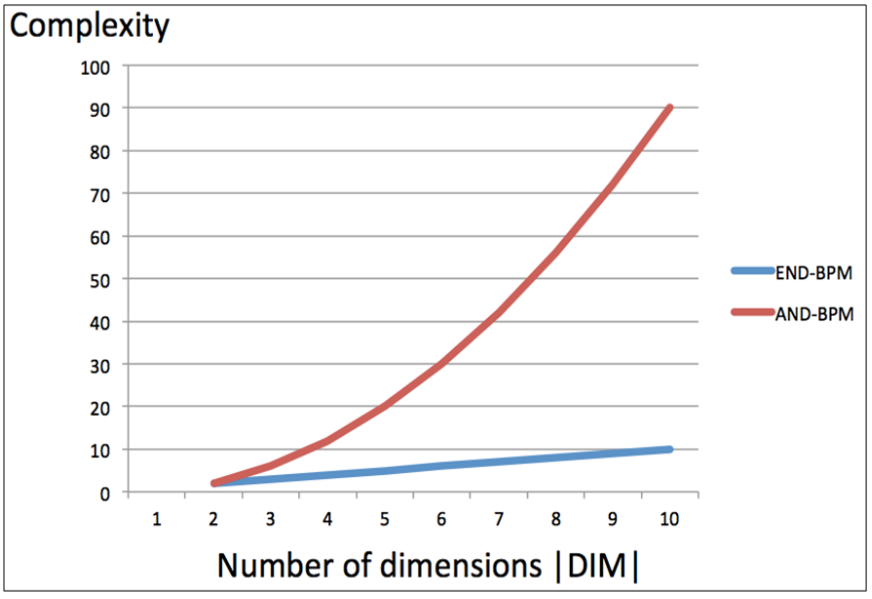


FIGURE 7.1: Compartive Complexity of END-BPM and AND-BPM algorithms

7.5 Evaluations of the ND Banded Pattern Mining (ND-BPM) Mechanism

This section reports on the experimental analysis conducted to evaluate the operation of the proposed ND-BPM algorithms. The evaluation consists of two components (objectives):

1. To compare the efficiency of the performance of the AND-BPM and END-BPM algorithms in terms of runtime (seconds).

2. To compare the effectiveness of the AND-BPM and END-BPM algorithms in terms of the bandings produced measured using GBS values.

For the evaluation the sixty-four 5D data sets extracted from the CTS database and introduced in Section 3.4 of Chapter 3 were used. Recall that these comprised data for four counties (Aberdeenshire, Cornwall, Lancashire and Norfolk) for each quarter of the years 2003, 2004, 2005 and 2006. Note also that with respect to the END-BPM algorithm both variations are considered, Euclidean and Manhattan distance measurement. With respect to the runtime comparison we also consider the use (or not) of M-Tables. The first of the above objectives is considered in Subsection 7.5.1, while the second is considered in Subsection 7.5.2.

7.5.1 Comparison Between END-BPM And AND-BPM Algorithms In Term of Run-times

The section considers the results from the comparative evaluation of the END-BPM and AND-BPM algorithms in terms of runtime. In the case of the END-BPM algorithm runtimes using both Euclidean and Manhattan distance measurement, and with and without M-Tables, were recorded. The results are presented in Tables 7.1, 7.2, 7.3 and 7.4 for the years 2003, 2004, 2005 and 2006 respectively. The 5 dimensions comprised: (i) Records (movement of a number of animals of the same breed and gender, on the same day, specific sender location to specific receiver location), (ii) Attributes, (iii) Eastings (x-cordinate holding area), (iv) Northings (y-cordinate holding area) and (v) Time (in months). From the tables it can firstly be noted that the runtime results vary with respect to different counties, quadrants and years. Closer inspection of the table indicates that this is to be expected because there is a correlation between the number of records in the data sets and the run-times; as the number of records increased the processing time also increased. More specifically, from the tables, the following can be noted:

1. Using the AND-BPM algorithm is more efficient than using END-BPM (regardless of whether Euclidean or Manhattan distance calculation is adopted or the usage of M-Table or not).

2. Using the M-Table requires less runtime than when not using such tables (a result that corroborates the results obtained with respect to earlier reported experiments regarding the E3D-BPM algorithm).

3. Using Manhattan distance calculation is faster than Euclidean distance calculation, again corroborating results obtained earlier.

TABLE 7.1: Runtime results (seconds) for 2003 5D CTS data sets using: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM

Month id	# Recs.	runtime (sec)				
		END-BPM *M* Tab.		END-BPM and no *M* Tab.		AND-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-Q1	42962	15.66	48.68	61.13	69.13	10.95
Abd-Q2	46187	19.82	50.95	60.01	79.04	16.95
Abd-Q3	41181	30.29	43.86	58.32	61.89	08.83
Abd-Q4	47842	28.01	45.16	32.22	51.59	16.44
Corn-Q1	40501	28.73	43.82	45.35	53.42	07.83
Corn-Q2	39626	20.32	39.33	51.60	75.91	06.92
Corn-Q3	40226	33.13	54.41	67.87	71.86	07.58
Corn-Q4	49890	48.92	54.68	61.35	80.43	18.88
Lanc-Q1	34325	27.66	46.68	51.02	67.29	05.13
Lanc-Q2	40926	36.50	50.95	63.11	72.74	09.91
Lanc-Q3	45765	25.74	52.03	59.85	64.87	13.86
Lanc-Q4	47392	36.29	55.52	80.52	88.89	15.99
Nolf-Q1	11280	05.32	26.70	19.65	36.21	01.58
Nolf-Q2	14557	17.04	25.85	47.40	56.82	02.29
Nolf-Q3	9460	10.48	22.23	45.17	56.20	01.27
Nolf-Q4	11680	13.34	25.84	46.38	53.15	02.23
Average	35238	24.82	42.92	54.39	64.97	09.17

TABLE 7.2: Runtime results (seconds) for 2004 5D CTS data sets using: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM

Month id	# Recs.	runtime (sec)				
		END-BPM *M* Tab.		END-BPM and no *M* Tab.		AND-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-Q1	43900	27.37	45.01	51.47	69.13	12.02
Abd-Q2	43221	35.44	53.12	60.44	72.82	17.72
Abd-Q3	38496	30.22	36.43	48.93	62.15	06.52
Abd-Q4	47995	34.40	46.77	41.22	60.41	18.65
Corn-Q1	40126	22.14	43.71	55.32	65.71	07.32
Corn-Q2	38226	25.63	44.67	54.42	78.78	16.12
Corn-Q3	38751	20.22	33.57	50.80	83.57	15.23
Corn-Q4	51950	33.75	50.83	64.82	94.86	20.88
Lanc-Q1	53976	40.71	62.49	66.99	96.02	22.35
Lanc-Q2	54326	60.36	74.91	75.58	113.69	30.83
Lanc-Q3	53926	65.43	70.63	72.95	103.53	33.60
Lanc-Q4	65694	73.97	85.04	90.52	125.94	40.73
Nolf-Q1	11701	05.49	29.06	13.57	45.84	01.91
Nolf-Q2	12993	08.44	31.71	15.31	47.16	02.78
Nolf-Q3	9290	07.49	26.43	28.01	38.20	01.32
Nolf-Q4	12262	06.07	21.12	16.08	28.68	02.48
Average	138552	31.08	47.21	50.40	74.16	15.65

The distinction between the operation of the AND-BPM and END-BPM algorithms merits some further discussion. Earlier, in Subsection 7.4.3, it was noted that the complexity of the END-BPM algorithms is less than that for the AND-BPM algorithm when considering the number of reorderings that take place on each iteration. However, this calculation did not take into account the complexity of the banding score calculation which, for the AND-BPM algorithm is much simpler than for the END-BPM algorithm. Consequently, as indicated by the results presented in Tables 7.1, 7.2, 7.3 and 7.4, the AND-BPM algorithm is more efficient than the END-BPM algorithm (despite the use

TABLE 7.3: Runtime results (seconds) for 2005 5D CTS data sets using:: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM

Month id	# Recs.	runtime (sec)				
		END-BPM *M* Tab.		END-BPM and no *M* Tab.		AND-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-Q1	41086	20.23	55.52	64.08	74.46	12.17
Abd-Q2	41317	20.06	43.50	54.06	103.50	15.99
Abd-Q3	30635	23.86	38.45	43.86	78.55	11.13
Abd-Q4	43995	36.02	56.19	56.12	106.91	26.52
Corn-Q1	40226	27.89	50.10	45.35	63.42	16.03
Corn-Q2	38076	25.34	44.94	59.14	61.76	14.16
Corn-Q3	31301	23.15	34.76	67.05	74.94	13.86
Corn-Q4	44986	32.77	52.06	78.27	82.26	20.52
Lanc-Q1	45526	39.76	55.97	59.46	84.34	23.96
Lanc-Q2	38676	28.29	45.80	58.34	75.83	15.38
Lanc-Q3	30351	26.74	40.62	56.70	69.22	10.79
Lanc-Q4	42591	39.87	46.83	56.82	96.23	20.93
Nolf-Q1	8557	02.71	21.13	12.71	20.13	01.69
Nolf-Q2	10549	03.17	27.48	23.17	44.48	02.35
Nolf-Q3	7066	02.23	20.15	22.23	33.15	01.04
Nolf-Q4	9742	02.87	23.55	22.87	35.55	01.81
Average	31543	22.19	41.07	48.76	69.05	13.02

TABLE 7.4: Runtime results (seconds) for 2006 5D CTS data sets using:: (i) Manhattan END-BPM and Euclidean END-BPM and M-Tables (ii) Manhattan END-BPM and Euclidean END-BPM and no M-Tables and (iii) AND-BPM

Month id	# Recs.	runtime (sec)				
		END-BPM *M* Tab.		END-BPM and no *M* Tab.		AND-BPM
		Manhat.	Euclid.	Manhat.	Euclid.	
Abd-Q1	54196	40.45	87.51	88.22	107.51	14.12
Abd-Q2	56876	52.58	86.94	78.18	106.94	15.03
Abd-Q3	56026	53.59	82.15	71.26	102.17	14.69
Abd-Q4	69108	56.55	84.19	76.52	184.19	43.50
Corn-Q1	38276	22.52	46.01	53.79	96.01	10.56
Corn-Q2	41099	33.79	55.56	66.39	85.64	22.52
Corn-Q3	40601	33.03	76.89	61.03	86.89	13.49
Corn-Q4	47305	44.60	51.61	64.26	91.16	28.45
Lanc-Q1	41176	30.55	55.18	60.51	70.26	22.94
Lanc-Q2	48601	37.56	55.97	67.52	75.79	31.53
Lanc-Q3	51151	34.60	40.60	60.26	76.67	24.62
Lanc-Q4	55362	37.90	52.04	72.04	108.24	27.98
Nolf-Q1	9659	03.35	24.52	13.65	34.20	01.61
Nolf-Q2	13707	07.11	22.54	33.24	42.67	02.67
Nolf-Q3	8945	04.10	21.78	28.12	31.88	01.69
Nolf-Q4	12839	06.55	28.50	42.38	48.60	02.14
Average	39490	31.18	54.50	58.59	84.30	17.35

of M-Tables).

7.5.2 Comparison Between END-BPM And AND-BPM Algorithms In Term of Global Banding Score (GBS) Values

This section considers the result from the comparative evaluation conducted with respect to the END-BPM and AND-BPM algorithms in terms of the final GBS values produced. The results are presented in Tables 7.5, 7.6, 7.7 and 7.8 for the years 2003, 2004, 2005 and 2006 respectively. In this case the naming conventions used are: (i) "END-BPM$_M$"

to indicate the Manhattan variation of END-BPM algorithm, (ii) "END-BPM$_E$" to indicate the Euclidean variation of the END-BPM algorithm and (iii) "AND-BPM" to indicate the AND-BPM algorithm. The results presented in the tables confirm that the END-BPM algorithm (both variations) produced better banding results than the AND-BPM algorithm (although the latter was more efficient). In addition, as also noted with respect to the reported evaluation of the END-BPM algorithm, the Euclidean distance measurement was found to out-perform Manhattan distance measurement.

It should be recalled from Subsection 6.5.2 of Chapter 6, that the difference between the operation of the Euclidean E3D-BPM and Manhattan E3D-BPM algorithms was that the former is better at differentiating between potential configurations than the latter. This can be better illustrated by considering the 3D data space presented in Figure 7.2. The data space features points A, B and C. These have a Manhattan distance to the origin of "6", "6" and "9". However the Euclidean distance from the origin of point A is "3.7416" ($\sqrt{3^2+1^2+2^2}$), point B is "3.4641" ($\sqrt{2^2+2^2+2^2}$) and point C is "5.3651" ($\sqrt{4^2+3^2+2^2}$). Thus it can be observed that Manhattan distance does not serve to differentiate between the two points A and B, whilst the Euclidean distance does.

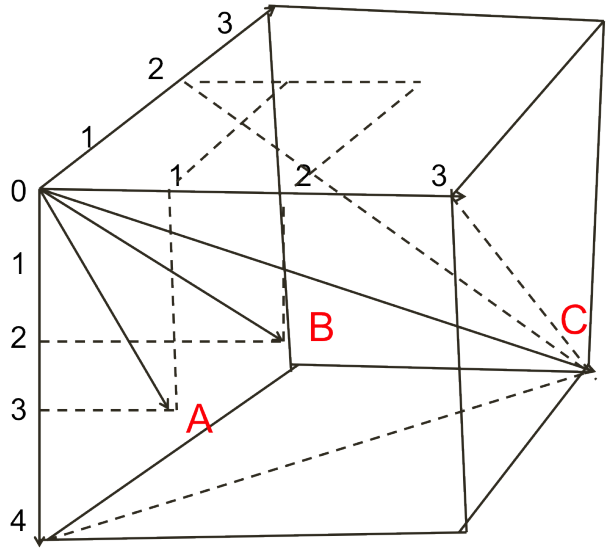


FIGURE 7.2: A 3D space illustrating the distinction between Euclidean and Manhattan distance calculation

Closer inspection of the bandings generated using the 5D CTS data sets indicated various phenomena. For example it could be observed that:

1. Male cattle breeds from Aberdeenshire county were moved more often in the east of the county for the year 2003 than in the west of the county.

2. In 2005 male cattle of age = 1 in the Cornwall county were more frequently moved in the north of the county than in the south.

TABLE 7.5: GBS results for 2003 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM

Month id	# Recs	GBS		
		END-BPM$_M$	END-BPM$_E$	AND-BPM
Abd-Q1	42962	0.5272	**0.4831**	0.8795
Abd-Q2	46187	0.5281	**0.4754**	0.8776
Abd-Q3	41181	0.5322	**0.4824**	0.8824
Abd-Q4	47842	0.5324	**0.4734**	0.8957
Corn-Q1	40501	0.5070	**0.4658**	0.9420
Corn-Q2	39626	0.5222	**0.4634**	0.9837
Corn-Q3	40226	0.5276	**0.5144**	0.8663
Corn-Q4	49890	0.5286	**0.5097**	0.9983
Lanc-Q1	34325	0.5351	**0.5235**	0.9126
Lanc-Q2	40926	0.5338	**0.5317**	0.8816
Lanc-Q3	45765	0.5360	**0.5213**	0.9961
Lanc-Q4	47392	0.5254	**0.5090**	0.9859
Norf-Q1	11526	0.5152	**0.4658**	0.8159
Norf-Q2	14311	0.5198	**0.4702**	0.8376
Norf-Q3	9460	0.5315	**0.5202**	0.8820
Norf-Q4	11680	0.5243	**0.4642**	0.8227
Average	35238	0.5267	**0.4921**	0.9037

TABLE 7.6: GBS results for 2004 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM

Month id	# Recs	GBS		
		END-BPM$_M$	END-BPM$_E$	AND-BPM
Abd-Q1	43900	0.5146	**0.5009**	0.9318
Abd-Q2	43221	0.5182	**0.5077**	0.8852
Abd-Q3	38429	0.5312	**0.5030**	0.8836
Abd-Q4	47995	0.5205	**0.5066**	0.8085
Corn-Q1	40126	0.5405	**0.5163**	0.8671
Corn-Q2	38226	0.5352	**0.5213**	0.8095
Corn-Q3	38751	0.5335	**0.5186**	0.9164
Corn-Q4	51950	0.5327	**0.5252**	0.9246
Lanc-Q1	53976	0.5313	**0.5152**	0.9746
Lanc-Q2	54326	0.5338	**0.5336**	0.9014
Lanc-Q3	53926	0.5340	**0.5285**	0.8458
Lanc-Q4	65694	0.5234	**0.4697**	0.8938
Norf-Q1	11701	0.5235	**0.5208**	0.9174
Norf-Q2	12993	0.5308	**0.5169**	0.9149
Norf-Q3	9290	0.5224	**0.5128**	0.9334
Norf-Q4	12262	0.5350	**0.5069**	0.9485
Average	38552	0.5285	**0.5128**	0.8973

3. With respect to the county of Norfolk, for the years 2003 and 2006, fewer cattle were moved in the east and north of the county than the south and west.

4. With respect to the county of Aberdeenshire, for all the data sets considered, more cattle were moved in the east and north of the county than in the south and west.

The above examples give an illustration of the kind of information that can be extracted from data as a result of the application of banding.

TABLE 7.7: GBS results for 2005 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM

Month id	# Recs	GBS		
		END-BPM$_M$	END-BPM$_E$	AND-BPM
Abd-Q1	41086	0.5285	**0.4157**	0.8219
Abd-Q2	41317	0.5299	**0.4136**	0.8716
Abd-Q3	30635	0.5433	**0.5174**	0.9141
Abd-Q4	43995	0.5288	**0.5021**	0.8484
Corn-Q1	40226	0.4231	**0.3768**	0.9357
Corn-Q2	38076	0.4582	**0.4056**	0.8960
Corn-Q3	31301	0.5293	**0.5209**	0.9460
Corn-Q4	44986	0.5312	**0.4746**	0.9520
Lanc-Q1	45526	0.5266	**0.5230**	0.9101
Lanc-Q2	38676	0.5835	**0.4027**	0.8441
Lanc-Q3	30351	0.5342	**0.5124**	0.8995
Lanc-Q4	42591	0.5262	**0.5229**	0.9498
Norf-Q1	8557	0.5225	**0.4676**	0.8375
Norf-Q2	10549	0.5240	**0.4689**	0.8230
Norf-Q3	7066	0.5221	**0.5472**	0.9232
Norf-Q4	9742	0.5240	**0.4688**	0.8357
Average	31543	0.5207	**0.4713**	0.8880

TABLE 7.8: GBS results for 2006 data sets using: (i) Manhattan END-BPM, (ii) Euclidean END-BPM and (iii) AND-BPM

Month id	# Recs	GBS		
		END-BPM$_M$	END-BPM$_E$	AND-BPM
Abd-Q1	54196	0.5059	**0.4959**	0.9061
Abd-Q2	56878	0.5226	**0.4874**	0.8890
Abd-Q3	56026	0.5032	**0.4786**	0.9366
Abd-Q4	69108	0.5967	**0.4788**	0.9375
Corn-Q1	38276	0.5265	**0.5109**	0.9239
Corn-Q2	41099	0.5358	**0.5358**	0.9918
Corn-Q3	40601	0.5333	**0.5196**	0.9189
Corn-Q4	47305	0.5285	**0.5248**	0.8680
Lanc-Q1	41176	0.5249	**0.5429**	0.9058
Lanc-Q2	48601	0.5275	**0.5244**	0.9863
Lanc-Q3	51151	0.5390	**0.5187**	0.8025
Lanc-Q4	55362	0.5289	**0.5271**	0.9103
Norf-Q1	9659	0.5248	**0.4945**	0.8811
Norf-Q2	13707	0.5343	**0.4912**	0.9220
Norf-Q3	8945	0.5399	**0.4789**	0.9544
Norf-Q4	12839	0.5371	**0.5091**	0.8896
Average	39490	0.5308	**0.5096**	0.9140

7.6 Summary

This chapter has presented N-Dimensional Banded Pattern Mining (ND-BPM). Two algorithms were presented, the AND-BPM and END-BPM algorithms. These were based respectively on the A3D-BPM and E3D-BPM algorithms presented in the previous chapter. As before the exact algorithm featured both Euclidean and Manhattan variations and the option to use M-Tables or not. The evaluation of the algorithms was conducted

by comparing their operation in terms of runtime (seconds) and the final GBS values arrived at. From the reported evaluation the following main findings can be noted:

1. The AND-BPM algorithm was more efficient than the END-BPM algorithm (regardless of the variation used) because the banding score calculation mechanism was much simpler than in the case of the mechanism used with respect to the END-BPM algorithm.

2. In terms of the recorded GBS values, although the approximate algorithm is faster and produced an approximate, the best (most accurate) bandings were produced using the END-BPM algorithm with Euclidean distance calculation because Euclidean distance calculation is better able to differentiate between potential configurations.

3. The concept of M-Tables, in the context of the END-BPM algorithm, offered efficiency advantages.

Note that to use either exact or approximate banding, given a particular application depends on whether, the user wishes to maximise accuracy or efficiency. In most case, it is desirable to maximise accuracy.

In the next chapter the ideas presented in this chapter will be further developed to address the situation where locations within the data space can hold more than one dot. The significance, as will become clear later in this thesis, is in the context of the sampling and segmentation mechanisms proposed in Chapter 9 to allow much larger data sets (than those considered so far) to be banded.

Chapter 8

Multiple Dot Mechanism

8.1 Introduction

In the previous chapter, Chapter 7 two ND-BPM algorithms were presented, AND-BPM and END-BPM. Both algorithms were developed from earlier work on 2D and 3D banding presented earlier in this thesis. All the banding algorithms considered in this thesis so far have assumed that the maximum number of dots held at a location is one. This makes sense if we are considering data sets comprised of records and attributes of some kind, as in the case of the evaluation data sets identified in Chapter 3. We can envisage situations where this might not be the case. One such situation is where we are determining a banding in the context of a subset of the available dimensions. Why we might want to do this is explored in the following Chapter. In preparation for the work presented in the following chapter this chapter presents the multiple dots Banded Pattern Mining (MD-BPM) algorithm; more specifically, following on from the work in the previous chapter, two variations of the MD-BPM algorithm are presented, approximate and exact (MD-ABPM and MD-EBPM). Note that although the MD-BPM algorithms are designed for the "multiple dots" situation, they will work equally well where we have one dot per location although some unnecessary processing will be conducted.

The reminder of this chapter is arranged as follows. Section 8.2 presents some formal definitions to support the MD-BPM algorithm discussion, while Section 8.3 considers the MD-ABPM and MD-EBPM algorithms in detail. Section 8.4 presents an example illustrating the operation of the proposed MD-BPM algorithms. Finally in Section 8.5 the chapter is concluded with a brief summary. Note that some further evaluation using the proposed MD-BPM algorithms was conducted although not expressely reported on in this chapter. More specifically, experiments were conducted to determine the number of iterations that would be required for the MD-BPM algorithms to find a best banding. However, the results were very similar to those reported for the 2D-BPM algorithm in Section 4.7.1 of Chapter 4. Hence these have been included in Appendix D.

8.2 Multiple Dot Banding Formalism and Calculation of Banding Score

The data space of interest, as before comprise, a set of dimensions DIM, where $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$. As before the dimensions are not necessarily of equal size, and each dimension Dim_i comprises a sequence of k index values $\{e_{i1}, e_{i2}, \ldots, e_{ik}\}$. However, in this case each location may contain zero, one or more dots. The precise distribution of the dots depends on the nature of the application domain. As before each dot (hyper-sphere in ND space) will be represented by a set of coordinates: $\langle c_1, c_2, \ldots, c_n \rangle$. The challenge is then to rearrange the indexes in the dimensions so that the dots are arranged along the leading diagonal (or as close to it as possible) taking into consideration that individual locations may hold multiple dots. Note that having multiple dots at a cell location is not the same as considering integer valued data sets. The two are very different the latter would require an entire rethink of the banding score concept. The potential for applying the work presented in this thesis to alternative data format is considered as a potential avenue for future work (see Chapter 11). As will become clear later in the chapter, the idea is to consider co-located dots as a "meta dots".

In the context of the MD-ABPM algorithm the banding score bs_{ij_p} for an index p in dimension Dim_i with respect to dimension Dim_j is calculated as follows:

$$bs_{ij_p} = \frac{\sum_{u=1}^{u=|W_{ij_p}|} w_u \times q_u}{\sum_{v=1}^{v=|W_{ij_p}|} (k_i - q + 1) \times q'_v} \tag{8.1}$$

where:

- W_{ij_p} The set of Dim_j indexes for the locations that feature index p in Dim_i and hold one or more dots, $W = \{w_1, w_2, \ldots\}$.
- q_u The number of dots at location w_u, $q_u \in Q_{ij_p}$, where Q_{ij_p} is the set of the number of dots at index p in Dim_i with respect to dimension Dim_j $Q_{ij_p} = \{q_1, q_2, \ldots\}$, $|Q_{ij_p}| = |W_{ij_p}|$.
- q'_v The vth element in the set Q'_{ij_p}, the set of location quantities Q_{ij_p} but in descending order of size so that elements with the largest number of dots are associated with the maximum distance from the origin.

The GBS for Dim_i with respect to Dim_j (GBS_{ij}) is calculated in the same way as before:

$$GBS_{ij} = \frac{\sum_{p=1}^{p=k_i} bs_{ij_p}}{k_i} \tag{8.2}$$

The GBS for a dimension Dim_i is then given by (as before):

$$GBS_i = \frac{\sum_{j=1}^{j=|DIM|, j \neq i} GBS_{ij}}{|DIM| - 1} \tag{8.3}$$

The normalised overall GBS for the entire configuration is then calculated thus (as before):

$$GBS = \frac{\sum_{i=1}^{i=|DIM|} GBS_i}{|DIM|} \tag{8.4}$$

In the case of the MD-EBPM algorithm, the banding score bs_{i_p} for an index p in Dim_i is calculated as follows:

$$bs_{i_p} = \frac{\sum_{u=1}^{u=|W_{i_p}|} w_u * q_u}{\sum_{v=1}^{v=|M_{i_p}|} m_v * q'_v} \tag{8.5}$$

W_{i_p} The set of distances from the origin for the locations that feature index p in Dim_i and hold at least one dot $W_{i_p} = \{w_1, w_2, \dots\}$. Note that the distances can be calculated using either Euclidean or Manhattan distance calculation.

q_u The number of dots at location w_u, $q_u \in Q_{i_p}$, $Q_{i_p} = \{q_1, q_2, \dots\}$, $|Q_{i_p}| = |W_{i_p}|$.

q'_v The vth element in the set Q'_{i_p}, the set of location quantities Q_{i_p}, but in descending order of size so that elements with the largest number of dots are associated with the maximum distance from the origin.

The GBS for a dimension Dim_i is calculated in the same way as before:

$$GBS_i = \frac{\sum_{p=1}^{p=k_i} bs_{i_p}}{k_i} \tag{8.6}$$

The overall normalised GBS is then:

$$GBS = \frac{\sum_{i=1}^{i=|DIM|} GBS_i}{|DIM|} \tag{8.7}$$

8.3 MD Banded Pattern Mining (MD-BPM) Algorithms

The MD-ABPM and MD-EBPM algorithms operate in the same manner as the AND-BPM and END-BPM algorithms presented in the previous chapter other than including provision for multiple dots as described above. We iterate through the dimensions and use the individual banding scores to reorder the dimension indexes. In the case of the MD-EBPM algorithm the use of M-Tables might again be expedient. Because of the similarity with the previous algorithms the pseudo code for the MD-ABPM and MD-EBPM algorithms are not presented here. The mechanism for constructing M-Tables is identical to that presented in the previous chapter except that the concept of meta-dots is used where dots are co-located, so also not detailed in this chapter.

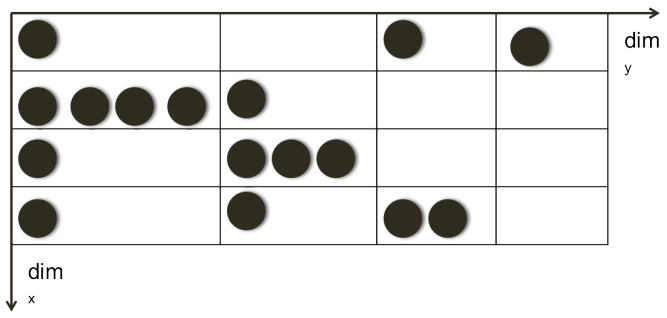


FIGURE 8.1: Input "Dot matrix" for worked example

8.4 A Worked Example Using the MD-BPM Algorithms

This section presents a working example illustrating the operation of the MD-EBPM algorithm; the MD-ABPM algorithm will operate in a similar manner other than in how the banding scores are calculated and thus a worked example using the MD-ABPM algorthm is not included here. For this illustration the 2D 4×4 configuration given in Figure 8.1 will be used. The configuration features $DIM = \{x, y\}$, $Dim_x = \{0, 1, 2, 3\}$ and $Dim_y = \{0, 1, 2, 3\}$ with multiple dots in some cells. The input D to the MD-EBPM algorithms is thus:

$$D = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 1,1\rangle,$$
$$\langle 1,2\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \langle 2,3\rangle, \langle 2,3\rangle, \langle 3,0\rangle\}.$$

The MD-EBPM algorithm starts by considering dimension x first, the banding scores are calculated (taking into account the number of dots per location) using Equation 8.5. This produces the banding scores $\{0.60, 0.83, 0.75, 0.00\}$, calculated as shown in Table 8.1. Consequently we rearrange the indexes (elements) in Dim_x in ascending order of their banding scores to produce the result shown in Figure 8.2.

$$D = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle,$$
$$\langle 2,0\rangle, \langle 2,3\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,2\rangle, \langle 3,2\rangle, \langle 3,3\rangle\}.$$

TABLE 8.1: Calculation of banding scores for dimension x (iteration 1)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*1)+(1*4)$ $+(2*1)+(3*1)$ $=9.0$	$(0*1)+(1*1)$ $+(2*1)+(3*4)$ $=15.0$	0.60
1	$(1*1)+(2*3)$ $+(3*1)=10.0$	$(1*1)+(2*1)$ $+(3*3)=12.0$	0.83
2	$(0*1)+(3*2)$ $=6.0$	$(2*1)+(3*2)$ $=8.0$	0.75
3	$(0*1)=0.0$	$((3*1)=3.0$	0.00
		Total	2.18

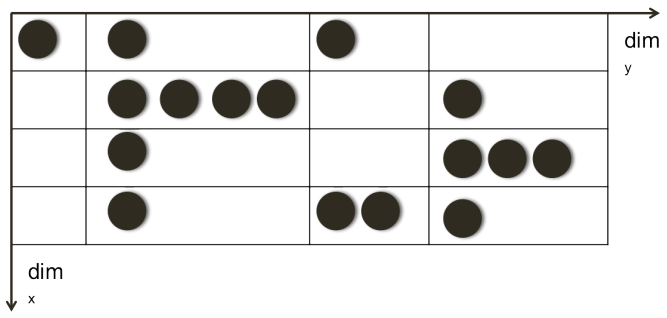


FIGURE 8.2: Dot matrix after rearrangement of Dim_x (iteration 1)

TABLE 8.2: Calculation of banding scores for dimension y (iteration 1)

Index	Dist from origin	Max. dist. from origin	bs
0	$(0*1)+(1*1)$ $+(2*1)=3.0$	$(1*1)+(2*1)$ $+(3*1)=6.0$	0.50
1	$(1*4)+(3*1)$ $=7.0$	$(2*1)+(3*4)$ $=14.0$	0.50
2	$(1*1)+(3*3)$ $=10.0$	$(2*1)+(3*3)$ $=11.0$	0.91
3	$(1*1)+(2*2)$ $+(3*1)=8.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.89
		Total	2.80

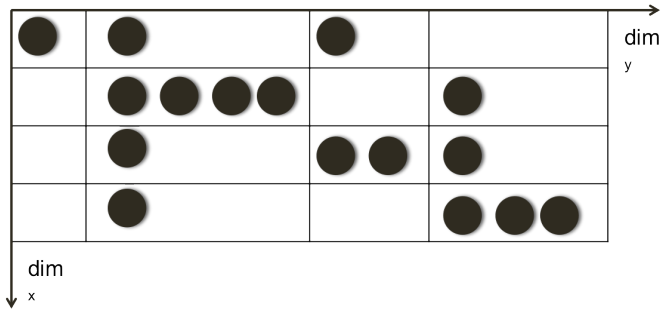


FIGURE 8.3: Dot matrix after rearrangement of Dim_y (iteration 1)

Considering dimension y next, we calculate the banding scores as shown in Table 8.2. This produces the banding scores $\{0.50, 0.50, 0.91, 0.89\}$. The indexes (elements) in y are more or less already in ascending order of bs_y; we only need to swap the last two elements (the effect is that the index with the greater number of dots is moved to be nearer the centre of the data space). The result is as shown in Figure 8.3. We now have:

$$D' = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle,$$
$$\langle 2,0\rangle, \langle 2,2\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

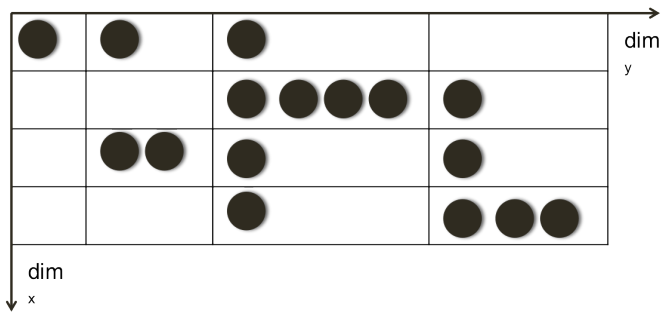


FIGURE 8.4: Dot matrix after rearrangement of Dim_x (iteration 2)

The final GBS for this configuration is then calculated using Equation 8.7 (the sum of the individual banding scores divided by the total number of indexes in the configuration):

$$GBS = \frac{0.0}{3.0} + \frac{9.0}{15.0} + \frac{6.0}{8.0} + \frac{10.0}{12.0} + \frac{3.0}{6.0} + \frac{7.0}{14.0} + \frac{8.0}{9.0} + \frac{10.0}{11.0} = 0.6122$$

Testing the new GBS value against the stored value we find that $GBS_{new} < GBS_{sofar}$ (recall that GBS_{sofar} was set to 1.0 on start up), thus $GBS_{sofar} = GBS_{new}$ and $D = D'$ and the process is repeated. Note that the maximum number of iterations has not yet been reached.

On the next iteration new banding score values are first calculated for Dim_x. The new banding scores produced for Dim_x are $\{0.00, 0.60, 0.50, 1.00\}$ calculated as shown in Table 8.3. The indexes in Dim_x are arranged accordingly; the result is as shown in Figure 8.4. Similarly, new banding scores are produced for Dim_y, $\{0.50, 0.79, 0.78, 1.00\}$, calculated as shown in Table 8.4. As a result the indexes in Dim_y are also rearranged accordingly. The result is as shown in Figure 8.5 (we only needed to swap the second and third indexes). We now have:

$$D' = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 2,0\rangle, \langle 2,1\rangle, \langle 2,1\rangle, \langle 2,1\rangle,$$
$$\langle 2,1\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

TABLE 8.3: Calculation of banding scores for dimension x (iteration 2)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*1) = 0.0$	$(3*1) = 3.0$	0.00
1	$(0*1) + (1*4)$ $+(2*1) + (3*1)$ $= 9.0$	$(0*1) + (1*1)$ $+(2*1) + (3*4)$ $= 15.0$	0.60
2	$(0*1) + (2*2)$ $= 4.0$	$(1*2) + (3*2)$ $= 8.0$	0.50
3	$(1*1) + (2*1)$ $(3*3) = 12.0$	$(1*1) + (2*1)$ $(3*3) = 12.0$	1.00
		Total	2.10

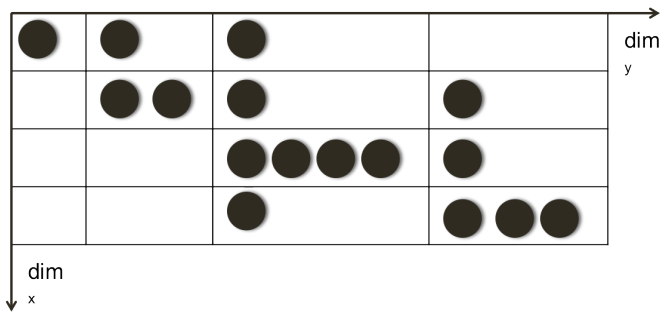


FIGURE 8.5: Dot matrix after rearrangement of Dim_y (iteration 2)

The new GBS' value is calculated as follows:

$$GBS = \frac{0.0}{3.0} + \frac{4.0}{8.0} + \frac{9.0}{15.0} + \frac{12.0}{12.0} + \frac{3.0}{6.0} + \frac{7.0}{9.0} + \frac{11.0}{14.0} + \frac{11.0}{11.0} = 0.6392$$

This newly calculated GBS value of 0.6392 is greater (worse) than the previously calculated value of $GBS = 0.6122$, so the algorithm exits with D from the previous iteration. Thus:

$$D = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 2,1\rangle, \langle 3,1\rangle,$$
$$\langle 0,2\rangle, \langle 2,2\rangle, \langle 2,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

TABLE 8.4: Calculation of banding scores for dimension y (iteration 2)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*1)+(1*1)$ $+(2*1)=3.0$	$(1*1)+(2*1)$ $+(3*1)=6.0$	0.50
1	$(2*4)+(3*1)$ $=11.0$	$(2*1)+(3*4)$ $=14.0$	0.79
2	$(1*2)+(2*1)$ $+(3*1)=7.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.78
3	$(2*1)+(3*3)$ $=11.0$	$(2*1)+(3*3)$ $=11.0$	1.00
		Total	3.07

Further worked examples using the MD-BPM algorithms are presented in Appendix A

8.5 Summary

This short chapter has presented two Multiple Dot Banded Pattern Mining (MD-BPM) algorithms, the MD-ABPM and MD-EBPM algorithms. The first was based on the AND-BPM algorithm, and the second on the END-BPM algorithm (both presented in previous chapters). Of note were the mechanisms used to calculate banding scores such that the possibility of multiple dots existing at locations was taken into account. The

significance of MD-BPM is that it can be used to find bandings with respect to subsets of the dimensions that might feature in a given dot data set. We can imagine situations where we might wish to do this, but this is also important where we are considering very large data sets that cannot be held in primary storage. How bandings can be discovered in such very large data sets is considered in the following chapter. The central idea is that we rearrange a subset of the data and then apply the discovered banding to the entire data set. To do this the MD-BPM approach presented in this chapter must be adopted. In the following chapter, the discovery of bandings in very large data sets is presented in more detail.

Chapter 9

Discovering Bandings in Big Data Using Sampling and Segmentation

9.1 Introduction

In the previous chapter Multiple Dots Banded Pattern Mining (MD-BPM) was considered. At the end of the chapter it was noted that MD-BPM has application with respect to the identification of banding in the context of big data. For the purpose of this chapter we define big data as data that cannot be easily stored and processed in primary storage. Other definitions of big data exist [3, 64, 6, 117]. In this chapter two techniques are proposed for discovering bandings in big data; sampling and segmentation. In the first, banding is conducted with respect to a subset, a "sample", of the data; and the identified banding is then applied to the remainder of the data. In the second the data is divided into "segments" and the banding conducted with respect to each segment. The identified individual banding configurations are then combined to identify a global banding.

The challenge with respect to the sampling technique is how best to identify a sample that is representative of the entire data set. The challenge with respect to the segmentation technique is how best to combine the identified banding configurations (it is likely that the configuration for each segment will not be identical to the rest). In both cases the sampling or segmentation needs to be conducted with respect to one of the dimensions, for the evaluation presented later in this chapter the dimension representing records was used. Whatever dimension is selected this dimension should not be considered when banded pattern mining is applied to the data, consequently locations defined within the context of the remaining dimensions may hold multiple dots, and hence MD-BPM will be required. This can be illustrated as follows. Given $DIM = \{Dim_x, Dim_y, Dim_z\}$ and a data configuration:

$$D =$$
$$\{\langle 0,0,0\rangle, \langle 0,0,1\rangle, \langle 0,0,3\rangle, \langle 0,1,1\rangle, \langle 0,1,2\rangle, \langle 1,1,2\rangle, \langle 1,1,3\rangle, \langle 1,1,4\rangle, \langle 1,1,5\rangle, \langle 1,2,2\rangle$$
$$\langle 2,2,3\rangle, \langle 2,2,4\rangle, \langle 2,2,5\rangle, \langle 2,3,0\rangle, \langle 2,3,4\rangle, \langle 3,2,4\rangle, \langle 3,3,3\rangle, \langle 3,3,4\rangle\}.$$

if we remove dimension Dim_x, we are left with a 2D data configuration that features multiple dots as follows:

$$D' = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,3\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle, \langle 1,5\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 2,5\rangle$$
$$\langle 3,0\rangle, \langle 3,4\rangle, \langle 2,4\rangle, \langle 3,3\rangle, \langle 3,4\rangle\}.$$

The proposed sampling and segmentation techniques operate using either the MD-ABPM or MD-EBPM algorithm (as described in the foregoing chapters). In the case of the MD-EBPM algorithm it was noted previously that this can operate using either Euclidean or Manhattan distance calculation with or without M-Tables. Both the Euclidean and Manhattan variations are considered in this chapter coupled with the use of M-Tables. The reason behind considering only variations with M-Tables, and not without, was beacause the previously reported experiments in Subsection 6.5.1 in Chapter 6 and subsection 7.5.1 in Chapter 7 had demonstrated that this was significantly more efficient.

The rest of this chapter is organised as follows. Section 9.2 presents an overview of the BPM sampling technique, while Section 9.3 consider the BPM segmentation technique. Section 9.4 then presents a comparative evaluation of the two techniques. The chapter is concluded in Section 9.5 with a summary and some conclusions.

9.2 BPM Sampling Technique

This section presents the BPM sampling technique for identifying bandings in large ND data sets. As already noted, processing very large data sets (data sets that will not fit into primary storage) remains a challenge, especially when the data sets under consideration is comprised of many dimensions. The suggested proposed solution presented in this section was to adopt a sampling technique whereby a best banding was identified using a subset S of the original dot data sets D and then applied to the entire data set. As noted in the introduction to this chapter the challenge when using sampling is to select an appropriately representative subset of the original data set. From the literature, a number of data sampling techniques have been proposed (see for example [24, 66, 37, 79, 121]). Recall also from Subsection 2.5.1 that these were itemised as follows: (i) Simple Random Sampling, (ii) Systematic Selection, (iii) Stratified Random Sampling and (iv) Cluster Sampling. However with respect to the work presented in this thesis, the stratified sampling technique was adopted, where the data set is divided into subgroups (strata) and records randomly selected from each subgroup. The reason for adopting stratified sampling, as opposed to the other sampling techniques that might have been selected, was that it was considered to provide a highly representative sample of the whole data set.

Another important consideration with respect to the sampling approach is deciding on sample size. Clearly this has to be small enough to allow it to be processed but large enough for it to be representative. In [75] the following formula was proposed to determine a sample size:

$$S \geq \left(\frac{z * \sigma}{MOE}\right)^2 \tag{9.1}$$

where: (i) z corresponds to the value of confidence level taken from the standard normal distribution to estimate the sample mean, (ii) σ is the standard deviation of the sample data and (iii) MOE is the Margin Of Error used to determine the reliability of the sample size. The formula presented in [75] was adopted with respect to the work presented in this thesis determine a minimum sample size:

$$S \geq \left(\frac{1.96 * 1414}{20}\right)^2 = 19202$$

where the value 1.96 is the z score equivalent to 95% confidence level, the value 1414 is the standard deviation of the sample size and 20 is the a margin of error. Because the above was considered to be a minimum sample size, for the evaluation presented later in this a chapter, a sample size of 24,000 was actually used.

As noted in the introduction to this chapter the stratification was conducted with respect to a particular dimension. The obvious dimension to use was the record dimension (assuming such a dimension exists with respect to the data set under consideration). In the context of the evaluation presented later in this chapter, where 5D data sets were considered, the record dimension was used. Which dimension is selected depends very much on the application domain. Whatever case the sampling as proposed here, and the segmentation proposed in Section 9.3, will both feature the possibility of multiple dots at a location.

The derivation of bandings using the proposed sampling technique is described by the pseudo code presented in Algorithm 12. The algorithm incorporates calls to either the MD-ABPM or the MD-EBPM algorithms presented in the previous chapter. The pseudo code assumes that these algorithms have been adapted so that they return a set of reordered dimensions DIM' (the algorithms as presented in the previous chapter returned a rearranged data space configuration D). Returning to the pseudo code presented in Algorithm 12 the input, as in the case of the BPM algorithms previously presented in this thesis, are: (i) a dot data set D comprised of a set of tuples of the form $\langle c_1, c_2, \dots \rangle$, (ii) a set of dimensions $DIM = \{Dim_1, Dim_2, \dots\}$ and (iii) a maximum number of iterations counter. The output (Line 6) is a reconfigured data set D that features a "best banding". The algorithm proceeds by identifying a data sample S using the stratified sampling technique detailed above. This sample is then used for banding discovery and a reconfigured set of dimensions returned, DIM' (Line 8). This banding configuration is then applied to the entire input data set D to give a reconfigured data set that features a "best" banding D' (Line 9). The final GBS value is then calculated

(Line 10) using Equation 8.4 or Equation 8.7 as appropriate from Chapter 8 depending on whether approximate or exact banding is adopted.

Algorithm 12: BPM with Sampling Algorithm

1: **Input:**
2: D = Binary valued input data set
3: DIM = the set of indexes per dimension
4: *counter* = The maximum number of iterations
5: **Output:**
6: D = The original data set D re-arranged so as to display as near a banding as possible
7: S = A subset of records from D
8: $DIM' = nd\_bpmAlgorithm(S, DIM, counter)$ (either Algorithm 10 or 11)
9: D' = Data set D rearranged according to DIM'
10: GBS_{new} = Final GBS calculated using either Equation 8.4 or 8.7

9.3 BPM Segmentation Technique

Following on from the previous section on sampling, this section presents the BPM segmentation technique. As already noted the basic idea was to conduct banding sequentially using a sequence of data segments R taken from a single large ND data set D and then to combine the different bandings on completion. Again the segmentation has to be conducted with respect to a particular dimension. In the context of the evaluation presented later in this chapter the record dimension was again used. The size of R will depend on the size of the data subset that can be processed at any one time. With respect to the evaluation presented later in this chapter $|R| = 6$ was used.

There are two ways that the bandings that result from the proposed segmentation technique can be combined: (i) best GBS and (ii) most frequent. The first is done by conducting bandings on the segmented data sets and then selecting the banding with the best GBS and applying this to the whole data set. The second involves selecting the most frequently occurring banding from all the potential segment bandings and then applying this to the entire data sets to achieve an overall banding. In the unexpected situation where two or more most frequent bandings are found one will have to be selected in an arbitrary manner. In the evaluation reported on below both mechanisms are considered in further detail.

The psuedo code for the BPM segmentation technique is presented in Algorithm 13. The input to the algorithm (Lines 1 to 3) comprises: (i) a dot data set D, (ii) the set of dimensions $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$ for the data space under consideration and (iii) a maximum iteration counter. The output is the dot data set D rearranged so as to feature a best banding (Line 4). As in the case of the sampling technique the algorithm incorporates calls to either the MD-ABPM or the MD-EBPM algorithms presented in the previous chapter. The pseudo code assumes that these algorithms have been adapted so that they return a tuple of the form $\langle b_i, g_i \rangle$, where: (i) b' is a

dimension configuration reordered so as to feature a best banding, and (ii) g' is the associated GBS value (the algorithms as presented in the previous chapter return a rearranged data space configuration D). The segmentation algorithm commence (Line 5) by segmenting the data set D to give a collection of segments R. As the algorithm proceeds each configuration b_i and associated GBS values g_i are stored in a set of sets B and and a set GBS_{SET} respectively. These are defined in (Lines 6 and 7) in the algorithm. The algorithm then iteratively loops over the set of segments R. On each iteration a configuration for the current segment R is determined (Line 9). The resulting configuration is then stored in the set B (Line 10) and the associated GBS value in the set GBS_{SET} (Line 11). We then (Line 13) select the "best" configuration b' using either a most frequent or best GBS strategy. This selected configuration is then applied to the entire data set D to give a reconfigured data set D' (Line 14). The process is completed with the calculation of the final GBS value (Line 15).

Algorithm 13: BPM with Segmentation Algorithm

1: **Input:** D = Binary valued input data set
2: $DIM = \{Dim_1, Dim_2, \ldots, Dim_n\}$ the set of indexes per dimension
3: $counter$ = The maximum number of iterations
4: **Output:** D' = The original data set D rearranged so as to display a "best" banding and the associated GBS' value
5: R = A sequence of data segments from D
6: $B = \{b_1, b_2, \ldots, b_{|R|}\}$ A set of sets in which to hold bandings
7: $GBS_{SET} = \{g_1, g_2, \ldots, g_{|R|}\}$ A set to hold GBS values associated with each banding
8: **for** $i = 1$ **to** $i = |R|$ **do**
9: $\quad \langle b_i, g_i \rangle = nd\_bpmAlgorithm(R, DIM, counter)$ (either Algorithm 10 or 11)
10: $\quad B = B \cup b_i$
11: $\quad GBS_{SET} = GBS_{SET} \cup g_i$
12: **end for**
13: b_i = selected banding from B
14: D' = Data set D rearranged according to contents of b'
15: GBS_{new} = Final global banding score for D' calculated using either Equation 8.4 or 8.7

9.4 Evaluation

This section reports on the experimental analysis conducted to evaluate the performance of the BPM sampling and segmentation techniques advocated in this chapter. The evaluation was conducted using the 3D, 4D and 5D data sets extracted from the CTS database introduced in Section 3.4.4 of Chapter 3. For the evaluation a total of 48 data sets were used: 16 3D data sets, 16 4D data sets and 16 5D data sets. In each case the data sets covered the four identified counties Aberdeenshire, Cornwall, Lancashire and Norfolk. Note also that the data sets spanned the years 2003, 2004, 2005 and 2006. In the case of the 3D data sets the three dimensions were: (i) Records, (ii)

Attributes and (iii) Time (in months). For the 4D data sets the four dimensions were: (i) Records, (ii) Attributes, (iii) Eastings and (iv) Northings. For the 5D data sets the dimensions were: (i) Records, (ii) Attributes, (iii) Eastings, (iv) Northings and (v) Time (in months). The Eastings and Northings were discretised into 10 sub-ranges. In all cases the sampling/segmentation was done with respect to the record dimension. Note that there is no gold standard or benchmark data sets. The Average Band Width (ABW), measure was defined as an independent mechanism for measuring bandings. In the context of the utility of the bandings produced, some examples are given in Section 7.5.2 of Chapter 7.1 in the context of the CTS data sets.

TABLE 9.1: Statistical summary of number of records per segmentation

Year	**Segmentation Data sets**					
	1	2	3	4	5	6
Aberdeenshire						
2003	28158	34033	26958	25036	37115	26872
2004	27285	34648	25188	23732	33088	29253
2005	24532	33458	24413	21295	26082	27253
2006	40427	40778	39952	37889	40072	37083
Cornwall						
2003	21949	28860	26327	25879	37083	30145
2004	20454	29659	27551	28409	33713	29267
2005	17503	28886	25059	23263	31611	28267
2006	22636	27369	29370	25182	33452	29272
Lancashire						
2003	21183	27714	24506	27667	37599	29250
2004	33032	40274	38730	37129	40500	38257
2005	20076	25448	25326	25991	33321	26980
2006	26373	27716	32164	34081	40095	35861
Norfolk						
2003	7012	9701	9124	5753	7747	7640
2004	7123	8894	8677	5645	7989	7918
2005	4882	7769	6455	4630	5911	6267
2006	5608	8905	8853	5830	7740	8214

For the sampling each data set was divided into subsets where each subset represented a month in a particular year, thus we have 12 subsets per data set. 2000 records were selected from each subset to give a total sample size of 24000 records. In the case of the segmentation, the data set was divided into a sequence of six data segments, six because this corresponded to the maximum amount of data could easily be processed on a single machine. A statistical summary concerning the number of records per segment in the segmentation data sets is presented in Table 9.1.

The objectives of the evaluation reported on in this section were as follows:

1. To determine the relative efficiency of the two techniques; sampling and segmentation in terms of run time (seconds).

2. To determine the relative effectiveness of the two techniques in terms of the quality of the bandings produced measured using the final GBS value arrived at.

3. To determine the effectiveness of the two alternative selection mechanisms used with respect to segmentation by comparing the quality of the bandings produced in terms of the final GBS value arrived at.

The first of the above objectives is considered in Sub-section 9.4.1 below, while the second and third are considered in Sub-section 9.4.2.

9.4.1 Efficiency Comparison Using Sampling and Segmentation

This subsection considers the results from the comparative evaluation of the sampling and segmentation techniques in terms of run time (in seconds). The evaluation was conducted by considering the Euclidean MD-EBPM$_E$, Manhattan MD-EBPM$_M$ and MD-ABPM algorithms. Note also that with respect to the reported run times this is the time taken to identify the final global best banding.

The results are presented in Tables 9.2, 9.3, 9.4 and 9.5 (best results highlighted in bold font). Table 9.2 and 9.3 show the run time results obtained using the 3D and 4D data sets. Table 9.2 gives the results for the counties of Aberdeenshire and Cornwall, while Table and 9.3 gives the results for the counties of Lancashire and Norfolk. For each data set recall that the data sets are split across years. In the tables the last four columns give the recorded run times in each case. The runtime results presented in the tables are the average of ten runs. For each data set, and with reference to the table, run times are given for: (i) the run time used to determine a banding for the selected sample S, (ii) the run time used to determine a banding with respect to the collection of segments, (iii) the run time to determine the final banding using the sampling technique, (iv) the run time to produce the final global banding using the segmentation technique and selecting the configuration according to the best GBS value and (v) the run time to produce the final global banding using segmentation technique and selecting the configuration according to the most frequent combination found. For the 4D data sets run times obtained using the MD-EBPM (both variations) and the MD-ABPM algorithms are given. For the 3D data only the run time results obtained using MD-EBPM are presented because the MD-ABPM algorithm when applied to 2D data (3D minus one dimension) operates in the same manner as the MD-EBPM algorithm. Tables 9.4 and 9.5 presents the sampling and segmentation runtime results obtained with respect to the 5D data sets; the tables are organised in the same manner as the previous two tables.

From the tables, as before, it can be seen that there is a correlation between the number of records in the data sets and the run-time. As data sets size increases there is a corresponding increase in processing time. With respect to the 4D data sets the results presented shows that using sampling and segmentation, and using the MD-ABPM algorithm, requires less run time than when using sampling and segmentation and the

TABLE 9.2: Sampling and Segmentation Runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Aberdeenshire and Cornwall)

	Year	**runtime (sec)**			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	02.68	02.26	**01.47**	01.58
Banding of Segments		16.02	15.19	**13.21**	11.98
Final band. using Sampling		18.38	16.08	**14.83**	16.13
Final band. Seg. (Best GBS)		22.37	16.38	**15.56**	14.96
Final band. Seg. (most freq.)		22.26	16.23	**15.01**	14.17
Banding of Sample	2004	01.76	01.56	**01.39**	01.92
Banding of Segments		16.82	15.61	**12.70**	11.97
Final band. using Sampling		15.84	14.36	**10.36**	13.13
Final band. Seg. (Best GBS)		17.13	13.56	**12.46**	13.97
Final band. Seg. (most freq.)		17.26	13.20	**12.45**	13.57
Banding of Sample	2005	02.95	02.45	**01.61**	01.85
Banding of Segments		17.95	15.05	**13.62**	12.72
Final band. using Sampling		18.24	14.74	**13.24**	12.98
Final band. Seg. (Best GBS)		30.84	17.77	**13.09**	12.95
Final band. Seg. (most freq.)		30.73	17.18	**13.90**	12.06
Banding of Sample	2006	01.79	01.69	**01.42**	01.48
Banding of Segments		20.04	16.04	**15.72**	13.83
Final band. using Sampling		20.74	15.87	**10.88**	14.47
Final band. Seg. (Best GBS)		21.31	16.61	**13.94**	14.66
Final band. Seg. (most freq.)		21.13	16.59	**13.41**	14.06
Cornwall					
Banding of Sample	2003	01.71	01.61	**01.38**	01.51
Banding of Segments		15.25	15.05	**12.18**	11.82
Final band. using Sampling		20.87	18.30	**16.30**	16.13
Final band. Seg. (Best GBS)		20.39	18.74	**14.50**	16.64
Final band. Seg. (most freq.)		20.32	18.40	**14.25**	16.60
Banding of Sample	2004	03.41	02.41	**01.74**	02.55
Banding of Segments		18.44	16.26	**14.09**	12.64
Final band. using Sampling		16.44	14.11	**13.05**	18.28
Final band. Seg. (Best GBS)		15.67	13.47	**12.44**	14.20
Final band. Seg. (most freq.)		15.99	13.30	**12.25**	14.10
Banding of Sample	2005	02.90	02.60	**02.09**	01.53
Banding of Segments		18.22	15.37	**13.26**	12.23
Final band. using Sampling		14.61	14.26	**12.16**	14.32
Final band. Seg. (Best GBS)		17.18	15.31	**14.19**	14.27
Final band. Seg. (most freq.)		17.08	15.26	**14.10**	14.17
Banding of Sample	2006	06.01	05.31	**02.40**	02.27
Banding of Segments		18.54	16.80	**13.95**	12.75
Final band. using Sampling		18.60	17.88	**15.69**	14.50
Final band. Seg. (Best GBS)		18.61	17.34	**14.08**	13.30
Final band. Seg. (most freq.)		18.34	17.16	**14.61**	13.20

MD-EBPM algorithm (both variations). The main points to note from the results presented in the tables are:

1. When using either sampling or segmentation, using either the MD-EBPM or the MD-ABPM algorithm, bandings can be successfully identified in large ND data sets within reasonable computation time.

TABLE 9.3: Sampling and Segmentation Runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Lancashire and Norfolk)

	Year	**runtime (sec)**			
		Euclid.	**Manhat.**	**Approx.**	3D
Lancashire					
Banding of Sample	2003	02.97	02.10	**01.70**	01.59
Banding of Segments		18.60	16.69	**14.59**	12.18
Final band. using Sampling		18.93	16.59	**14.39**	13.94
Final band. Seg. (Best GBS)		19.81	16.38	**14.69**	20.34
Final band. Seg. (most freq.)		19.18	16.38	**14.69**	20.32
Banding of Sample	2004	02.91	02.10	**01.15**	01.48
Banding of Segments		19.96	17.86	**09.35**	09.90
Final band. using Sampling		19.99	18.24	**14.04**	15.25
Final band. Seg. (Best GBS)		20.17	19.53	**17.73**	18.95
Final band. Seg. (most freq.)		20.70	19.53	**17.73**	18.50
Banding of Sample	2005	02.62	01.97	**01.87**	01.57
Banding of Segments		18.63	15.75	**13.88**	12.17
Final band. using Sampling		18.87	16.65	**15.97**	16.08
Final band. Seg. (Best GBS)		17.67	16.39	**15.35**	16.15
Final band. Seg. (most freq.)		17.15	16.85	**14.50**	16.05
Banding of Sample	2006	02.43	02.37	**01.93**	01.57
Banding of Segments		16.98	15.59	**13.46**	12.44
Final band. using Sampling		18.83	18.64	**12.44**	14.72
Final band. Seg. (Best GBS)		19.57	17.84	**14.70**	15.95
Final band. Seg. (most freq.)		19.96	17.17	**14.37**	15.59
Norfolk					
Banding of Sample	2003	05.09	02.59	**02.24**	01.51
Banding of Segments		18.20	15.49	**13.49**	12.46
Final band. using Sampling		12.09	11.03	**09.75**	11.77
Final band. Seg. (Best GBS)		20.34	17.96	**15.43**	14.51
Final band. Seg. (most freq.)		20.43	17.60	**15.34**	14.45
Banding of Sample	2004	07.24	05.94	**03.41**	02.99
Banding of Segments		20.45	17.40	**14.40**	13.51
Final band. using Sampling		27.05	17.14	**12.77**	16.42
Final band. Seg. (Best GBS)		15.81	13.66	**12.14**	10.81
Final band. Seg. (most freq.)		15.99	13.30	**12.25**	10.31
Banding of Sample	2005	02.16	02.10	**01.74**	01.15
Banding of Segments		16.05	15.74	**13.60**	11.43
Final band. using Sampling		15.02	11.23	**10.46**	10.85
Final band. Seg. (Best GBS)		17.30	14.90	**12.42**	14.10
Final band. Seg. (most freq.)		17.05	14.49	**12.24**	14.14
Banding of Sample	2006	01.83	01.53	**01.29**	01.52
Banding of Segments		15.48	14.01	**13.83**	12.54
Final band. using Sampling		14.74	13.03	**11.30**	11.10
Final band. Seg. (Best GBS)		14.11	13.06	**12.94**	12.30
Final band. Seg. (most freq.)		14.01	13.51	**12.40**	12.90

2. Regardless of whether sampling or segmentation is adopted, the MD-EBPM algorithm was always slower than when using the MD-ABPM algorithm.

With respect to the last point it should be recalled, from the results presented in the previous chapter, that although the MD-ABPM algorithm was found to be faster (because it entails less calculations) it was not as effective in terms of the final GBS values produced (this will be demonstrated further in the following sub-section).

TABLE 9.4: Sampling and Segmentation Runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Aberdeenshire and Cornwall)

	Year	**runtime (sec)**		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	24.69	19.22	**14.72**
Banding of Segments		39.14	37.98	**29.97**
Final band. using Sampling		36.94	31.89	**29.09**
Final band. Seg. (Best GBS)		36.51	31.40	**29.13**
Final band. Seg. (most freq.)		36.45	31.42	**29.33**
Banding of Sample	2004	26.93	21.78	**12.41**
Banding of Segments		39.90	33.82	**27.90**
Final band. using Sampling		34.40	32.24	**30.10**
Final band. Seg. (Best GBS)		34.98	32.66	**31.30**
Final band. Seg. (most freq.)		34.87	32.04	**31.50**
Banding of Sample	2005	15.66	14.56	**09.19**
Banding of Segments		31.92	29.41	**27.54**
Final band. using Sampling		31.35	29.67	**26.60**
Final band. Seg. (Best GBS)		31.58	29.49	**26.26**
Final band. Seg. (most freq.)		31.50	29.25	**29.30**
Banding of Sample	2006	18.41	17.58	**07.41**
Banding of Segments		38.28	32.34	**25.06**
Final band. using Sampling		35.41	30.18	**28.13**
Final band. Seg. (Best GBS)		36.44	31.24	**28.23**
Final band. Seg. (most freq.)		36.24	31.24	**28.36**
Cornwall				
Banding of Sample	2003	24.09	17.59	**09.61**
Banding of Segments		39.91	37.34	**35.20**
Final band. using Sampling		37.65	33.06	**28.15**
Final band. Seg. (Best GBS)		38.85	33.35	**28.60**
Final band. Seg. (most freq.)		38.80	33.28	**28.68**
Banding of Sample	2004	22.62	14.15	**10.61**
Banding of Segments		55.02	47.58	**39.10**
Final band. using Sampling		36.28	31.81	**29.80**
Final band. Seg. (Best GBS)		36.34	32.23	**29.08**
Final band. Seg. (most freq.)		36.29	32.50	**29.42**
Banding of Sample	2005	27.43	23.52	**14.92**
Banding of Segments		46.82	39.85	**31.11**
Final band. using Sampling		27.02	24.62	**21.23**
Final band. Seg. (Best GBS)		28.37	25.45	**20.32**
Final band. Seg. (most freq.)		28.25	25.07	**20.45**
Banding of Sample	2006	29.89	20.89	**15.41**
Banding of Segments		49.91	46.86	**43.07**
Final band. using Sampling		33.64	30.24	**28.05**
Final band. Seg. (Best GBS)		34.81	30.32	**28.24**
Final band. Seg. (most freq.)		34.28	30.62	**28.14**

To enhance the appreciation of the results obtained, Figure 9.1 and 9.2 presents the results from Tables 9.2, 9.3, 9.4 and 9.5 in bar graph form. In the figures the numbering along the x-axis should be interpreted as follows: (i) Aberdeenshire 2003,

Table 9.5: Sampling and Segmentation Runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM Algorithms with M-Tables (Lancashire and Norfolk)

	Year	**runtime (sec)**		
		Euclid.	**Manhat.**	**Approx.**
Lancashire				
Banding of Sample	2003	24.19	20.10	**10.02**
Banding of Segments		48.29	43.14	**32.12**
Final band. using Sampling		34.30	32.22	**22.02**
Final band. Seg. (Best GBS)		32.36	30.25	**25.03**
Final band. Seg. (most freq.)		32.12	30.08	**25.04**
Banding of Sample	2004	24.29	20.45	**14.09**
Banding of Segments		49.98	45.85	**31.26**
Final band. using Sampling		38.24	35.20	**30.23**
Final band. Seg. (Best GBS)		39.33	35.23	**30.12**
Final band. Seg. (most freq.)		39.15	35.14	**30.15**
Banding of Sample	2005	19.25	15.55	**09.84**
Banding of Segments		35.35	30.66	**27.35**
Final band. using Sampling		31.46	28.31	**26.25**
Final band. Seg. (Best GBS)		32.72	29.51	**27.32**
Final band. Seg. (most freq.)		32.50	29.41	**27.30**
Banding of Sample	2006	20.31	15.30	**11.49**
Banding of Segments		38.21	34.22	**29.71**
Final band. using Sampling		32.35	30.27	**28.02**
Final band. Seg. (Best GBS)		32.63	30.15	**28.32**
Final band. Seg. (most freq.)		32.25	30.34	**28.34**
Norfolk				
Banding of Sample	2003	20.84	18.84	**10.72**
Banding of Segments		32.77	28.06	**15.30**
Final band. using Sampling		35.09	25.35	**18.30**
Final band. Seg. (Best GBS)		36.74	26.43	**18.41**
Final band. Seg. (most freq.)		36.40	26.35	**18.96**
Banding of Sample	2004	29.29	25.60	**21.99**
Banding of Segments		35.13	30.94	**28.00**
Final band. using Sampling		35.90	32.36	**24.16**
Final band. Seg. (Best GBS)		36.81	32.42	**24.79**
Final band. Seg. (most freq.)		36.18	32.24	**24.90**
Banding of Sample	2005	25.25	20.55	**17.84**
Banding of Segments		35.59	28.10	**25.07**
Final band. using Sampling		18.69	16.79	**14.79**
Final band. Seg. (Best GBS)		18.79	16.50	**15.31**
Final band. Seg. (most freq.)		18.15	16.70	**15.10**
Banding of Sample	2006	36.01	25.31	**22.40**
Banding of Segments		44.33	32.48	**29.68**
Final band. using Sampling		19.77	15.70	**13.41**
Final band. Seg. (Best GBS)		17.46	14.27	**12.93**
Final band. Seg. (most freq.)		17.62	14.20	**12.27**

(ii) Aberdeenshire 2004, (iii) Aberdeenshire 2005, (iv) Aberdeenshire 2006, (v) Cornwall 2003, (vi) Cornwall 2004, (vii) Cornwall 2005, (viii) Cornwall 2006, (ix) Lancashire 2003, (x) Lancashire 2004, (xi) Lancashire 2005, (xii) Lancashire 2006, (xiii) Norfolk

2003, (xvi) Norfolk 2004, (xv) Norfolk 2005, (xiv) Norfolk 2006. From comparison of the figures, it can be seen that there is a significant difference in the run time between using sampling and segmentation and the MD-ABPM and MD-EBPM algorithms. From the figures it can also be confirmed that, regardless of whether sampling or segmentation is used, the Manhattan variation of the MD-EBPM algorithm performs better than the Euclidean variation (fewer calculations are required).

With respect to the results presented in this subsection using the sampling technique these were obtained using a sample size of 24,000 for reasons noted in Section 9.2 above. Some further experiments using sample sizes of 12,000 and 36,000 are reported on in Appendix B. The results from these experimentss coroborate the results presented in this chapter so were not reported on in the body of this thesis.

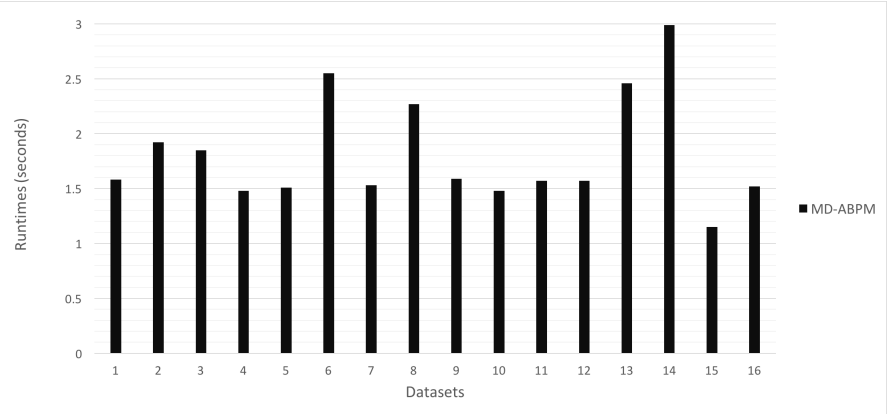
9.4.2 Effectiveness of MD-EBPM and MD-ABPM Algorithms Using Sampling and Segmentation

This section considers the evaluation of the proposed sampling and segmentation techniques with respect to the effectiveness of the techniques, using the MD-EBPM (both variations) and MD-ABPM algorithms, in the context of ND data. To determine the effectiveness of the techniques the final GBS values produced were considered. Experiments were conducted, as in the case of the foregoing sub-section, with respect to 3D, 4D and 5D data.

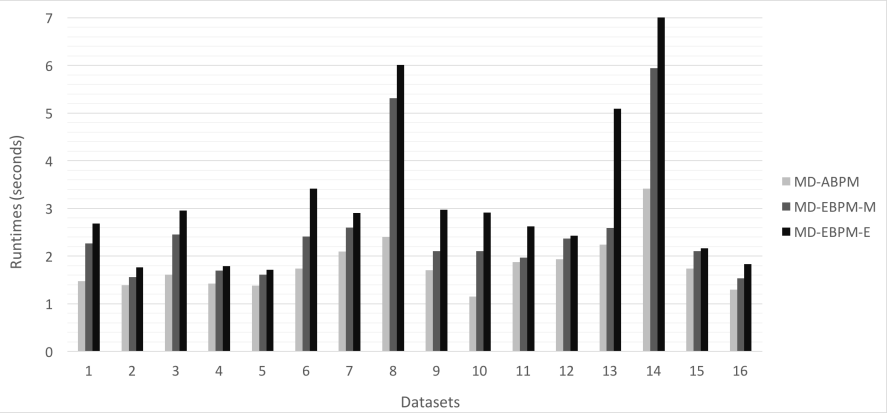
The results in the context of the 3D and 4D data are presented in Tables 9.6 and 9.7, while Tables 9.8 and 9.9 presents the results in the context of the 5D data. In the tables, the columns reference the three MD-BPM (Euclidean MD-EBPM, Manhattan MD-EBPM and Approximate MD-ABPM) algorithms. The rows indicate: (i) the final GBS value obtained with respect to the the banding identified in the selected sample S, (ii) the final GBS value with respect to the best configuration obtained when uisng segmentation, (iii) the final overall GBS values obtained when the sample banding configuration is applied to the entire data set, (iv) the final overall GBS value obtained when the best configuration from the segmentation is selected using best GBS and applied to the entire data set, (v) the final overall GBS value obtained when the best configuration from the segmentation is selected using the most frequently occurring configuration and applied to the entire data set and (vi) the GBS value obtained when no banding is conducted.

As noted in the previous subsection in 3D the sampling and segmentation technique using either the MD-EBPM or MD-ABPM algorithm operate in the same manner, and consequently produce the same GBS values; therefore only one of the results is presented.

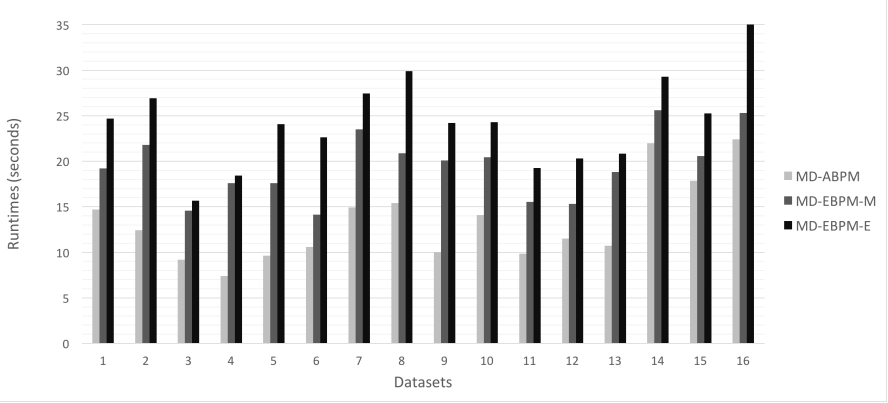
For ease of understanding the results from Tables 9.6, 9.7, 9.8 and 9.9 are also presented in bar graph form in Figures 9.3 and 9.4. Figure 9.3 shows the GBS results in terms of the 3D data sets with respect to both sampling and segmentation for 2003, while Figure 9.4 shows the GBS results in terms of the 4D and 5D data sets for 2003 with respect to sampling and segmentation. Note that the graph results for 2004, 2005



(a) 3D Data



(b) 4D Data

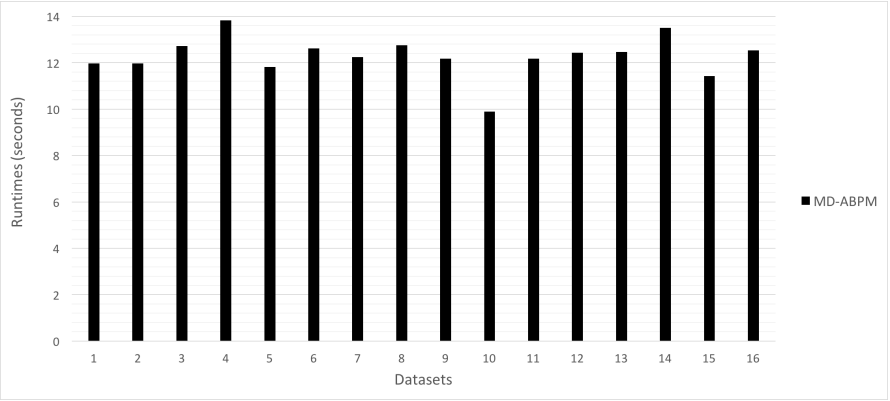


(c) 5D Data

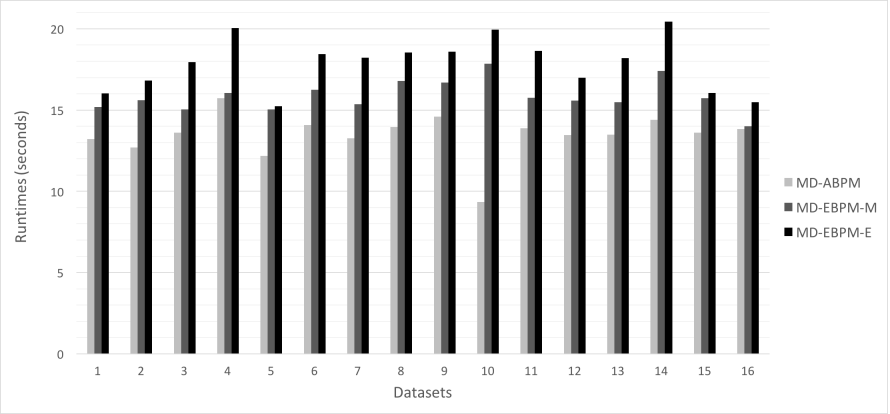
FIGURE 9.1: Sampling Run time results for 2003-2006 using the MD-EBPM$_E$, MD-EBPM$_M$ and MD-ABPM algorithms in the context of: (a) 3D, (b) 4D and (c) 5D data sets

and 2006 were very similar to those obtain for 2003, so were not included in the body of the thesis.

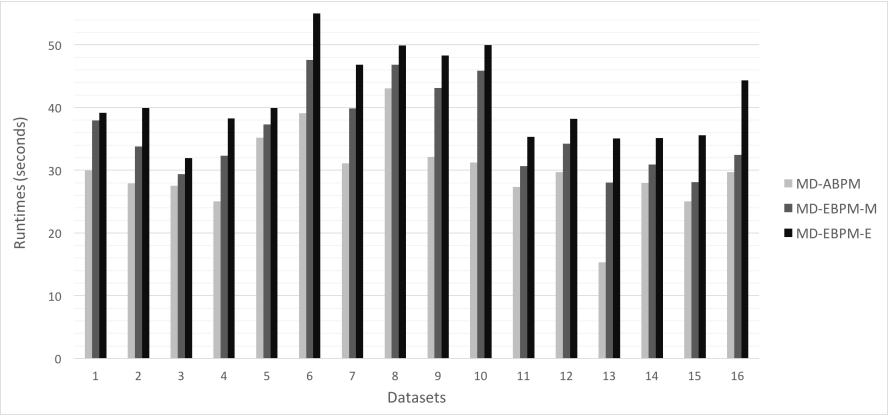
Inspection of the tables indicates that when using sampling and segmentation, and the Euclidean variation of the MD-EBPM algorithm, best GBS values were produced. From the tables, it can also be seen that by imposing the banding identified in a sample



(a) 3D Data



(b) 4D Data



(c) 5D Data

FIGURE 9.2: Segmentation Run time results for 2003-2006 with respect to entire data set using the MD-EBPM$_E$, MD-EBPM$_M$ and MD-ABPM algorithms in the context of: (a) 3D, (b) 4D and (c) 5D data sets

on the entire data set, the GBS values for the entire data set was improved (this is to be expected). Similarly, when using segmentation (regardless of whether best GBS or most frequent selection was adopted) the final GBS was improved.

More specifically, from the tables, it can be observed that:

1. The application of both the sampling and segmentation techniques served to produce an effective banding, in terms of the final GBS values obtained; better than if no banding was applied.

2. Segmentation tended to produce a better overall banding than sampling because the banding produced using segmentation was the best of a number of dimension index re-orderings (unlike in the case of sampling).

3. Out of the two segmentation banding combination techniques considered, best GBS and most frequent, in most cases the most frequent combination technique produced a better overall GBS.

4. The most effective MD-BPM algorithm, in terms of GBS, and in the context of both sampling and segmentation, was the Euclidean MD-EBPM.

The results presented in Tables 9.6, 9.7, 9.8 and 9.9 below, with respect to the segmentation technique only list GBS values for the segment with the best (lowest) GBS value. For completeness the GBS values with respect to all the segments are given in Appendix C.

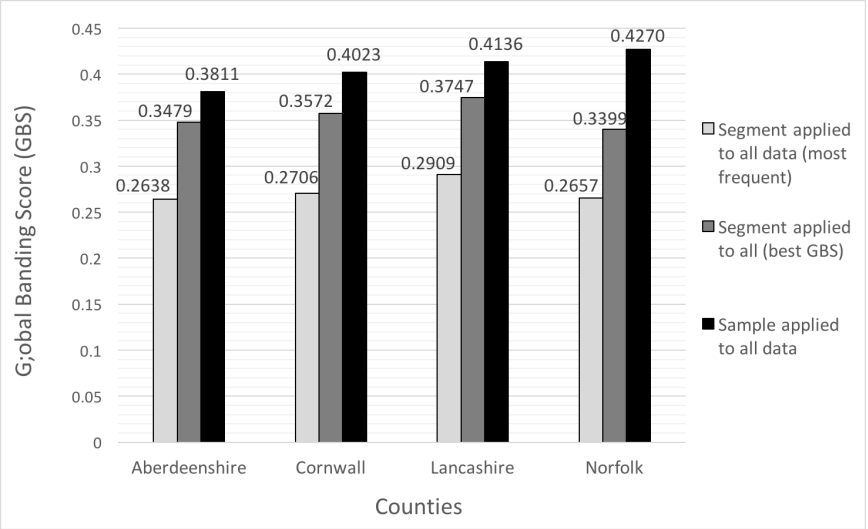


FIGURE 9.3: GBS values for 2003 comparison of 3D data sets using the Euclidean MD-EBPM Algorithm

TABLE 9.6: Sampling and Segmentation GBS Result for 2003 to 2006 3D and 4D CTS data set (Aberdeenshire and Cornwall)

	Year	GBS			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	0.2929	0.2936	0.4006	0.3770
Banding of Segments		0.2391	0.2398	0.3398	0.3049
Final band. using Sampling		0.3061	0.3133	0.4158	0.3811
Final band. Seg. (Best GBS)		0.2780	0.2794	0.4037	0.3479
Final band. Seg. (most freq.)		**0.2247**	**0.2350**	**0.2807**	**0.2638**
no banding		0.3176	0.3261	0.4557	0.3937
Banding of Sample	2004	0.2409	0.2420	0.3526	0.3686
Banding of Segments		0.2396	0.2442	0.3411	0.3023
Final band. using Sampling		0.2497	0.2586	0.3620	0.3710
Final band. Seg. (Best GBS)		0.2400	0.2490	0.3401	0.3403
Final band. Seg. (most freq.)		**0.1724**	**0.1937**	**0.2799**	**0.2514**
no banding		0.2383	0.2964	0.4242	0.3947
Banding of Sample	2005	0.3113	0.3180	0.4155	0.3869
Banding of Segments		0.2138	0.2366	0.3087	0.2890
Final band. using Sampling		0.3206	0.3276	0.4294	0.3977
Final band. Seg. (Best GBS)		0.2650	0.2677	0.3680	0.3278
Final band. Seg. (most freq.)		**0.2233**	**0.2280**	**0.3097**	**0.2551**
no banding		0.3290	0.3305	0.4595	0.3987
Banding of Sample	2006	**0.2397**	**0.2483**	**0.3449**	0.3670
Banding of Segments		0.2231	0.2305	0.3213	0.3106
Final band. using Sampling		**0.2397**	**0.2483**	**0.3449**	0.3709
Final band. Seg. (Best GBS)		0.2266	0.2284	0.3285	0.3322
Final band. Seg. (most freq.)		**0.1819**	**0.1902**	**0.2694**	**0.2421**
no banding		0.2743	0.2756	0.3936	0.3733
Cornwall					
Banding of Sample	2003	0.2911	0.2923	0.4180	0.4039
Banding of Segments		0.2503	0.2556	0.3606	0.3263
Final band. using Sampling		0.2996	0.3083	0.4306	0.4048
Final band. Seg. (Best GBS)		0.2835	0.2871	0.4155	0.3572
Final band. Seg. (most freq.)		**0.2190**	**0.2256**	**0.3340**	**0.2706**
no banding		0.3139	0.3152	0.4557	0.4370
Banding of Sample	2004	0.3181	0.3194	0.3743	0.3944
Banding of Segments		0.2717	0.2750	0.3855	0.3140
Final band. using Sampling		0.3243	0.2723	0.3901	0.4023
Final band. Seg. (Best GBS)		0.2639	0.2793	0.3372	0.3524
Final band. Seg. (most freq.)		**0.2020**	**0.2051**	**0.2867**	**0.2756**
no banding		0.3213	0.3281	0.4570	0.4043
Banding of Sample	2005	0.2730	0.2758	0.3943	0.3696
Banding of Segments		0.2365	0.2412	0.3427	0.3073
Final band. using Sampling		0.2822	0.2881	0.4097	0.3786
Final band. Seg. (Best GBS)		0.2779	0.2865	0.3331	0.3580
Final band. Seg. (most freq.)		**0.2281**	**0.2314**	**0.3284**	**0.2785**
no banding		0.2281	0.2314	0.4633	0.4142
Banding of Sample	2006	0.3065	0.3081	0.4336	0.3886
Banding of Segments		0.2812	0.2858	0.3971	0.3160
Final band. using Sampling		0.3092	0.3177	0.4455	0.3901
Final band. Seg. (Best GBS)		0.2844	0.2923	0.4049	0.3449
Final band. Seg. (most freq.)		**0.1767**	**0.1999**	**0.2979**	**0.2676**
no banding		0.3220	0.3234	0.4542	0.4060

Table 9.7: Sampling and Segmentation GBS Result for 2003 to 2006 3D and 4D CTS data set (Lancashire and Norfolk)

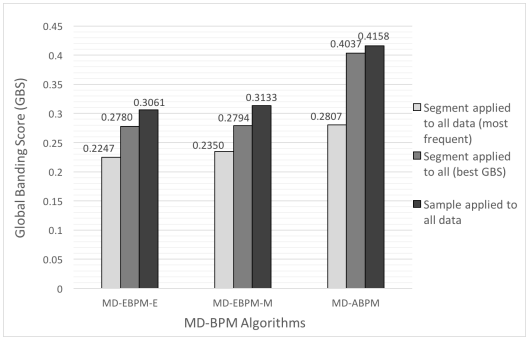
	Year	GBS			
		Euclid.	**Manhat.**	**Approx.**	3D
Lancashire					
Banding of Sample	2003	0.2947	0.3024	0.4199	0.4105
Banding of Segments		0.2807	0.2830	0.3966	0.3394
Final band. using Sampling		0.3076	0.3146	0.4393	0.4136
Final band. Seg. (Best GBS)		0.2922	0.3041	0.4176	0.3747
Final band. Seg. (most freq.)		**0.2092**	**0.2104**	**0.3083**	**0.2909**
no banding		0.3139	0.3236	0.4906	0.4206
Banding of Sample	2004	0.2817	0.2855	0.3959	0.3593
Banding of Segments		0.2317	0.2343	0.3190	0.2417
Final band. using Sampling		0.3076	0.3146	0.4393	0.3769
Final band. Seg. (Best GBS)		0.2421	0.2445	0.3449	0.3925
Final band. Seg. (most freq.)		**0.1996**	**0.2083**	**0.2770**	0.2655
no banding		0.3237	0.3248	0.4430	0.3908
Banding of Sample	2005	0.2940	0.2965	0.4142	0.3974
Banding of Segments		0.2678	0.2741	0.3830	0.3354
Final band. using Sampling		0.2948	0.2978	0.4241	0.4007
Final band. Seg. (Best GBS)		0.2787	0.2893	0.3589	0.3797
Final band. Seg. (most freq.)		**0.2010**	**0.2022**	**0.2797**	**0.2782**
no banding		0.3061	0.3100	0.4392	0.4059
Banding of Sample	2006	0.2847	0.2874	0.4071	0.3970
Banding of Segments		0.2731	0.2765	0.3875	0.3346
Final band. using Sampling		0.2856	0.2880	0.4113	0.3999
Final band. Seg. (Best GBS)		0.2899	0.2962	0.3252	0.3729
Final band. Seg. (most freq.)		**0.2029**	**0.2163**	**0.2863**	**0.3040**
no banding		0.3085	0.3122	0.4386	0.4011
Norfolk					
Banding of Sample	2003	0.3079	0.3216	0.4575	0.4255
Banding of Segments		0.2191	0.2300	0.3171	0.3192
Final band. using Sampling		0.3103	0.3226	0.4578	0.4270
Final band. Seg. (Best GBS)		0.2742	0.2788	0.3930	0.3399
Final band. Seg. (most freq.)		**0.2172**	**0.2265**	**0.3100**	**0.2657**
no banding		0.3319	0.3333	0.4653	0.4370
Banding of Sample	2004	0.2810	0.2910	0.4005	0.3607
Banding of Segments		0.2678	0.2688	0.3715	0.2894
Final band. using Sampling		0.2917	0.3017	0.4166	0.3653
Final band. Seg. (Best GBS)		0.2689	0.2696	0.3840	0.3278
Final band. Seg. (most freq.)		**0.2051**	**0.2062**	**0.2915**	**0.2453**
no banding		0.3076	0.3093	0.4507	0.3698
Banding of Sample	2005	0.3215	0.3279	0.4422	0.4058
Banding of Segments		0.2174	0.2226	0.3137	0.2910
Final band. using Sampling		0.3222	0.3287	0.4469	0.4075
Final band. Seg. (Best GBS)		0.2713	0.2782	0.3832	0.2837
Final band. Seg. (most freq.)		**0.2093**	**0.2253**	**0.3390**	**0.2511**
no banding		0.3291	0.3302	0.4548	0.4148
Banding of Sample	2006	0.2566	0.2629	0.3635	0.3802
Banding of Segments		0.1978	0.2036	0.2835	0.2653
Final band. using Sampling		0.2568	0.2654	0.3645	0.3817
Final band. Seg. (Best GBS)		0.2161	0.2251	0.3068	0.3154
Final band. Seg. (most freq.)		**0.1752**	**0.1807**	**0.2461**	**0.2537**
no banding		0.2600	0.2668	0.4172	0.3986

TABLE 9.8: Sampling and Segmentation GBS Result for 2003 to 2006 5D CTS data set (Aberdeenshire and Cornwall)

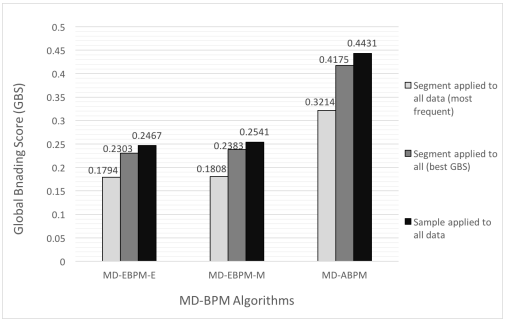
	Year	GBS		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	0.2352	0.2370	0.4226
Banding of Segments		0.1735	0.1761	0.3208
Final band. using Sampling		0.2467	0.2541	0.4431
Final band. Seg. (Best GBS)		0.2303	0.2383	0.4175
Final band. Seg. (most freq.)		**0.1794**	**0.1808**	**0.3214**
no banding		0.2580	0.2653	0.4781
Banding of Sample	2004	0.2020	0.2027	0.3814
Banding of Segments		0.1744	0.1794	0.3180
Final band. using Sampling		0.2085	0.2099	0.3945
Final band. Seg. (Best GBS)		0.2013	0.2104	0.3429
Final band. Seg. (most freq.)		**0.1542**	**0.1553**	**0.2667**
no banding		0.2383	0.2462	0.4415
Banding of Sample	2005	0.2496	0.2582	0.3620
Banding of Segments		0.1592	0.1712	0.3056
Final band. using Sampling		0.2584	0.2655	0.4581
Final band. Seg. (Best GBS)		0.2219	0.2299	0.4018
Final band. Seg. (most freq.)		**0.1908**	**0.1927**	**0.3547**
no banding		0.2671	0.2699	0.4837
Banding of Sample	2006	0.2002	0.2088	**0.3780**
Banding of Segments		0.1679	0.1713	0.3219
Final band. using Sampling		0.2015	0.2099	**0.3789**
Final band. Seg. (Best GBS)		0.1922	0.1943	0.3591
Final band. Seg. (most freq.)		**0.1469**	**0.1568**	**0.2770**
no banding		0.2238	0.2299	0.4092
Cornwall				
Banding of Sample	2003	0.2402	0.2428	0.4426
Banding of Segments		0.1822	0.1842	0.3421
Final band. using Sampling		0.2488	0.2508	0.4543
Final band. Seg. (Best GBS)		0.2357	0.2385	0.4398
Final band. Seg. (most freq.)		**0.1822**	**0.1945**	**0.3425**
no banding		0.2589	0.2613	0.4670
Banding of Sample	2004	0.2178	0.2260	0.4050
Banding of Segments		0.2011	0.2052	0.3650
Final band. using Sampling		0.2277	0.2289	0.4156
Final band. Seg. (Best GBS)		0.2321	0.2353	0.4224
Final band. Seg. (most freq.)		**0.1643**	**0.1675**	**0.3021**
no banding		0.2577	0.2617	0.4831
Banding of Sample	2005	0.2271	0.2290	0.4175
Banding of Segments		0.1705	0.1753	0.3196
Final band. using Sampling		0.2342	0.2368	0.4289
Final band. Seg. (Best GBS)		0.2321	0.2351	0.4219
Final band. Seg. (most freq.)		**0.1886**	**0.2029**	**0.3483**
no banding		0.2657	0.2687	0.4821
Banding of Sample	2006	0.2510	0.2520	0.4554
Banding of Segments		0.2123	0.2138	0.3876
Final band. using Sampling		0.2542	0.2633	0.4589
Final band. Seg. (Best GBS)		0.2352	0.2385	0.4292
Final band. Seg. (most freq.)		**0.1666**	**0.1742**	**0.3096**
no banding		0.2626	0.2699	0.4754

Table 9.9: Sampling and Segmentation GBS Result for 2003 to 2006 5D CTS data set (Lancashire and Norfolk)

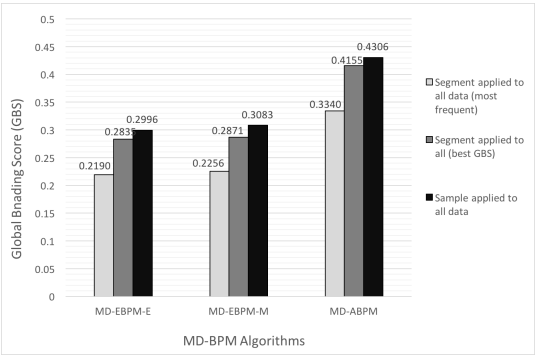
	Year	GBS		
		Euclid.	**Manhat.**	**Approx.**
Lancashire				
Banding of Sample	2003	0.2434	0.2488	0.4429
Banding of Segments		0.2070	0.2106	0.3777
Final band. using Sampling		0.2544	0.2631	0.4673
Final band. Seg. (Best GBS)		0.2436	0.2465	0.4482
Final band. Seg. (most freq.)		**0.1770**	**0.1775**	**0.3041**
no banding		0.2609	0.2647	0.4830
Banding of Sample	2004	0.2384	0.2451	0.4179
Banding of Segments		0.1733	0.1777	0.3180
Final band. using Sampling		0.2483	0.2550	0.4344
Final band. Seg. (Best GBS)		0.2070	0.2163	0.3886
Final band. Seg. (most freq.)		**0.1742**	**0.1764**	**0.3266**
no banding		0.2650	0.2676	0.4749
Banding of Sample	2005	0.2456	0.2541	0.4437
Banding of Segments		0.1999	0.2017	0.3703
Final band. using Sampling		0.2470	0.2546	0.4475
Final band. Seg. (Best GBS)		0.2319	0.2342	0.4255
Final band. Seg. (most freq.)		**0.1620**	**0.1658**	**0.3021**
no banding		0.2518	0.2559	0.4660
Banding of Sample	2006	0.2382	0.2466	0.4369
Banding of Segments		0.2026	0.2065	0.3770
Final band. using Sampling		0.2419	0.2455	0.4515
Final band. Seg. (Best GBS)		0.2370	0.2395	0.4385
Final band. Seg. (most freq.)		**0.1620**	**0.1829**	**0.3043**
no banding		0.2535	0.2576	0.4635
Norfolk				
Banding of Sample	2003	0.2663	0.2695	0.4663
Banding of Segments		0.1629	0.2632	0.3004
Final band. using Sampling		0.2695	0.2698	0.4573
Final band. Seg. (Best GBS)		0.2316	0.2366	0.4206
Final band. Seg. (most freq.)		**0.1644**	**0.1881**	**0.3588**
no banding		0.2795	0.2817	0.4959
Banding of Sample	2004	0.2350	0.2449	0.4266
Banding of Segments		0.1972	0.1999	0.3612
Final band. using Sampling		0.2418	0.2517	0.4432
Final band. Seg. (Best GBS)		0.2237	0.2317	0.4200
Final band. Seg. (most freq.)		**0.1603**	**0.1703**	**0.3253**
no banding		0.2512	0.2544	0.4859
Banding of Sample	2005	0.2572	0.2647	0.4540
Banding of Segments		0.1615	0.1628	0.3016
Final band. using Sampling		0.2580	0.2657	0.4551
Final band. Seg. (Best GBS)		0.2273	0.2347	0.4154
Final band. Seg. (most freq.)		**0.1760**	**0.1788**	**0.3341**
no banding		0.2640	0.2664	0.4781
Banding of Sample	2006	0.2134	0.2225	0.3948
Banding of Segments		0.1482	0.1497	0.2824
Final band. using Sampling		0.2139	0.2172	0.4004
Final band. Seg. (Best GBS)		0.1857	0.1869	0.3460
Final band. Seg. (most freq.)		**0.1327**	**0.1436**	**0.2708**
no banding		0.2153	0.2242	0.4985



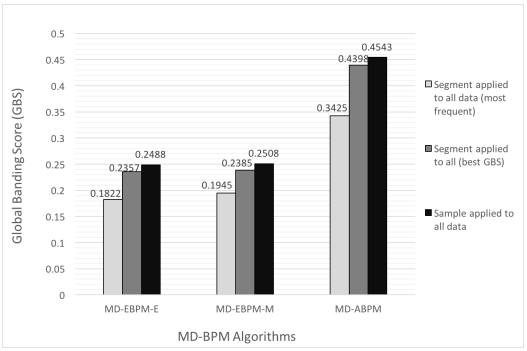
(a) Aberdeenshire 4D



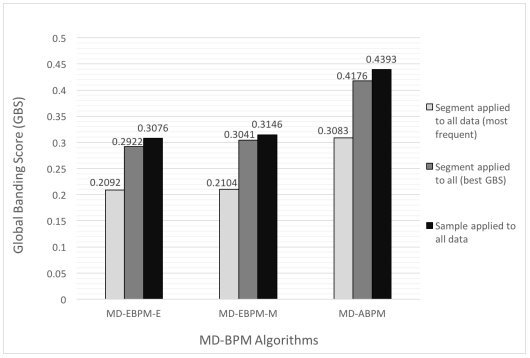
(b) Aberdeenshire 5D



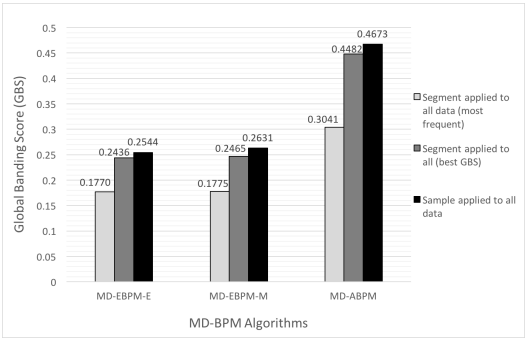
(c) Cornwall 4D



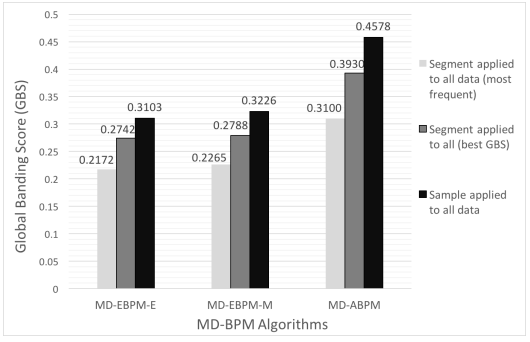
(d) Cornwall 5D



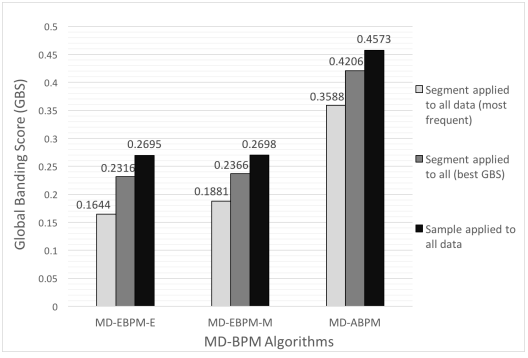
(e) Lancashire 4D



(f) Lancashire 5D



(g) Norfolk 4D



(h) Norfolk 5D

FIGURE 9.4: GBS values comparison for 4D and 5D data set in the context of: (i) Segmentation banding applied to all data (most frequent), (ii) Segmentation banding applied to all data (best GBS) and (iii) Sample banding applied to all data using the Euclidean MD-EBPM, Manhattan MD-EBPM and MD-ABPM Algorithms

9.5 Summary

This chapter has presented the BPM sampling and segmentation techniques for identifying bandings in very large data sets, data sets that can not be held in primary storage. The techniques were combined with the MD-ABPM and MD-EBPM algorithms presented in the previous chapter. Both techniques were fully described and evaluated using a series of experiments to illustrate the operation of the proposed techniques, and the banding concept in general, in the context of: (i) 3D, (ii) 4D and (iii) 5D data sets. From the reported evaluations, the following overall observations can be made:

1. Both the sampling and segmentation techniques were found to be effective with respect to identifying bandings in large data sets better than if no banding was applied.

2. In the context of both sampling and segmentation, segmentation produced a better banding than the sampling technique; the reason being that the banding produced using segmentation was conducted by considering the entire dataset (best bandings were obtained from a number of segment bandings) while in the case of sampling only a subset of the data set was used.

3. In the context of segmentation the eventual global banding produced using most frequent configuration selection (selection of the most frequently occuring bandings from all possible segment bandings) tended to produce better (more accurate) bandings than in the case of best GBS value configuration selection (selection of the banding from segment bandings with the best GBS values).

4. The most efficient MD-BPM algorithm, in terms of runtime, in the context of both sampling and segmentation was the MD-ABPM algorithm.

5. The most effective MD-BPM algorithm in terms of the final overall GBS obtained, in the context of both sampling and segmentation, was the MD-EBPM algorithm (regardless of the variation used).

6. The MD-ABPM algorithm in the context of sampling and segmentation produce the worst GBS values because of the general disadvantages of the MD-ABPM algorithms noted previously in Sub-section 9.4.2.

The following chapter considers mechanisms for determining the statistical significance of the banded pattern mining concept in the context of randomly generated synthetic data sets using normal (Gaussian) distribution curves.

Chapter 10

Statistical Significance Testing Using Gaussian Distributions

10.1 Introduction

In the foregoing chapters a variety of BPM algorithms have been presented. The reported evaluations indicated that in all cases a better GBS value was produced after banding than existed prior to banding. The question remained as to whether the detected bandings were in fact statistically significant or not. This short chapter reports on an exploration of a mechanism that can be adopted to determine whether a banding is statistically significant or not. The basic idea was that if we had n randomly generated dot data sets, all featuring the same dimensions and approximately the same density, each of these data sets would have a GBS value associated with it. It was assumed that these values would be distributed following the normal (Gaussian) distribution. Given a GBS value generated after banding had been applied the expectation was that this would be located away from the median of this distribution by a distance of at least one standard deviation. The normal (Gaussian) distribution was selected because in the absence of any information to the contrary it was assumed that the data sets to which banding was to be applied were likely to follow this distribution; the Gaussian distribution is the most common continuous probability distribution. Further reasons were that the Gaussian distribution is easy to work with and many statistical test can be derived from it. This chapter explores this idea and demonstrates that normal distributions can be usefully employed to establish the statistical significance of banding.

The rest of the chapter is organised as follows; Section 10.2 presents an overview of statistical significance testing in the context of the banding concept investigated in this thesis. Section 10.3 then reports on the process for generating normal distributions with respect to a set of example data set configurations without banding. This is followed in Section 10.4 with examples of how the normal distributions generated in the previous section can be used for the purpose of testing the statistical significance of generated banded patterns. Finally, Section 10.5 concludes the chapter.

10.2 Overview of Statistical Significance Testing

This section presents a more detailed discussion, than that presented in the previous section, of statistical significance testing. Given a randomly generated synthetic data set, some form of banding will exist as indicated by the associated GBS value. However, as noted above, the question is whether the identified banding is statistically significant or not. From the literature, there are a number of statistical techniques used to perform statistical significance comparison. With respect to the work presented in this thesis, the normal (Gaussian) distribution was used. The normal distribution is concerned with the operation of a continuous probability distribution [44, 57, 73, 99, 100, 34, 43] that represents a real-valued random variable. The normal distribution is described by the probability density function $\phi(x)$ given in Equation 10.1, where x is an observation of some kind. Note that the factor $\sqrt{2\pi}$ ensures the total area under curve $\phi(x)$ is one [44, 46, 47, 57, 118] and that the distribution has a unit variance (unit standard deviation).

$$\phi(x) = \frac{e^{1/2}x^2}{\sqrt{2\pi}} \tag{10.1}$$

Though, authors differ on which normal distribution should be called the "standard" one, Gauss [63] defined standard normal distribution as having variance $\sigma^2 = 1/2$ and a probability density function of:

$$\phi(x) = \frac{e^{x^2}}{\sqrt{\pi}} \tag{10.2}$$

while Stigler [119, 120] define standard normal distribution as having a variance $\sigma^2 = 1/(2\pi)$ and a probability density function of:

$$\phi(x) = e^{x^{2\pi}} \tag{10.3}$$

Using the probability density function $\phi(x)$ given in Equations 10.1, 10.2 and 10.3, for a range of values for x describes a "bell curve" [107] with a mean μ, a standard deviation σ and a variance σ^2. Figure 10.1, taken from [69] presents a number of examples of bell curves associated with the normal (or Gaussian) distribution. In the figure the X-axis indicates a range of values for the variable x from -5 to 5 and the Y-axis represents the frequency or probability of the occurence count. The red curve in the figure is the standard normal curve with ($\mu = 0$, $\sigma = 1$), The blue and green curves represents the normal curves with ($\mu = 0$, $\sigma = 0.2$) and ($\mu = -2$, $\sigma = 0.5$), whilst the purple curve is a normal curve with ($\mu = 0$, $\sigma = 5.0$). Thus the normal distribution is symmetric about its mean μ. As such it may not be a suitable model for variables that are inherently positive or strongly skewed. The normal distribution value tends to zero when the value x lies more than a few standard deviations away from the mean.

Standard normal distribution values are often presented in tabular form; Figure 10.2 gives an example taken from [88]. Note that in the table the variable z is used instead

of x as used in the above discussion and in Figure 10.1. Note also that in the table the "0.1"s run along the Y-axis and the "0.01"s along the X-axis (in this way we avoid a very large table).

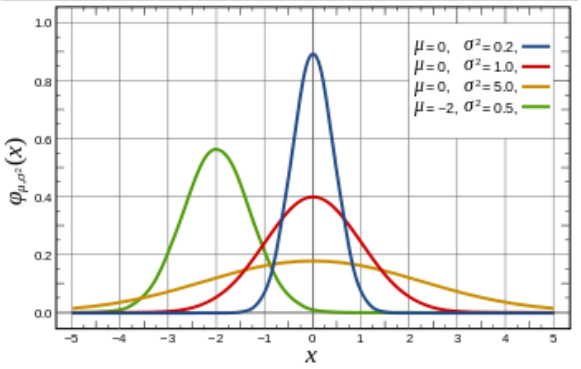


FIGURE 10.1: Gaussian or Normal Probability Curve [69]

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.00	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.10	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.20	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.30	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.40	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.50	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.60	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.70	0.7580	0.7611	0.7642	0.7673	O.7704	0.7734	O.7764	O.7794	0.7823	0.7852
0.80	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.90	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.00	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.10	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.20	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.30	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.40	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.50	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.60	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.70	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.80	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.90	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.00	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.10	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.20	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.30	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.40	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.50	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.60	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.70	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.80	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.90	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.00	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.10	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9931
3.20	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.30	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.40	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998

FIGURE 10.2: Standard Normal Distribution Table [88]

In the Normal distribution, the three-sigma rule is used to show the percentage of values that lie within a band around the mean width of "one", "two" and "three" standard deviations; this means that; 68.27%, 95.45% and 99.73% of the values lie within one, two and three standard deviations from the mean. In other words, for the normal distribution, values of less than one standard deviation away from the mean account for 68.27% of the values, two standard deviation from the mean account for 95.45% of the values and three standard deviation account for 99.73% of the values. Figure 10.3 taken from [105], illustrates the three-sigma rule for the normal distribution (see also [36, 129]).

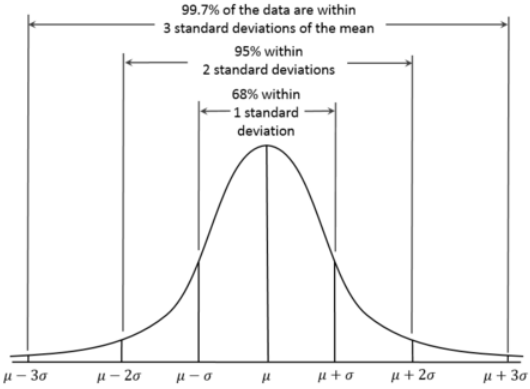


FIGURE 10.3: three-sigma rule for the normal distribution [105]

10.3 Normal Distribution Curve Generation

The theoretical foundation for testing the significance of a banding (expressed in terms of a GBS value) was presented in the foregoing section. This section reports on the adopted process whereby this theory was applied by considering the generation of collections of banding normal distributions. The idea was to create a bank of normal distribution curves, from randomly generated data sets to which banding had not been applied, which could then be used to establish whether a generated banding was significant or not in terms of distance from the mean. Of course there will be different distributions associated with different data as defined by the parameters for these data sets (size and dot density).

To demonstrate the application of the above theoretical approach to testing, two sets of experiments were conducted, each involving a collection of 1000 data sets grouped into batches of 100 according to row/column size. More specifically the row and column dimensions used were: (i) 100×100, (ii) 141×141, (iii) 173×173, (iv) 200×200, (v) 224×224, (vi) 245×245, (vii) 265×265 and (viii) 285×285, (ix) 300×300 and (x) 316×316. The effect was to have data sets ranging from 10,000 to 100,000 locations in steps of 1,000. The distinction between the two sets of experiments was the dot density used:

- **Static Dot Density value**: Experiments using a collection of data sets, using a static dot density of 10%.
- **Range of Dot Density Values**: Experiments using dot density values ranged from 10% to 50% increasing in steps of 10% (each data set size featured five different dot densities distributed evenly).

The rational for the second set of experiments was to determine the more general applicability of the approach. The data sets were generated using the LUCS-KDD generator used with respect to previously reported experiments [29]. The results were then used to define ten normal distributions, one for each data set configuration. The normal distributions associated with the first set of experiments is discussed in further

detail in Subsection 10.3.1 while that associated with the second set is discussed in Subsection 10.3.2.

10.3.1 Static Dot Density

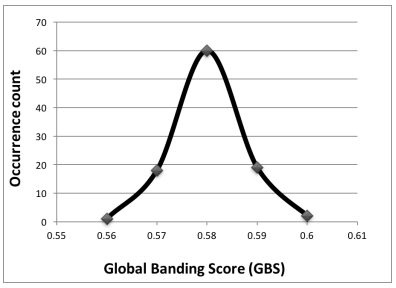
In this subsection, the experimental result using a static dot density of 10% is presented. Table 10.1 lists the natural GBS occurrence counts for each data set configuration (without banding), whilst Table 10.2 lists the accompanying μ, σ and one and two standard deviation limits. Figure 10.4 shows the normal distribution curves associated with the distributions (and the information in Tables 10.1 and 10.2). Inspection of the figure (and tables) indicates that similar distribution curves result regardless of data set size. The significance of these distribution curves is that they can now be used to compare GBS values obtained from similar data sets (same size and density) after banding has taken place. This is illustrated in the following section.

Table 10.1: List of GBS Occurrence Counts per data set configuration (static dot density)

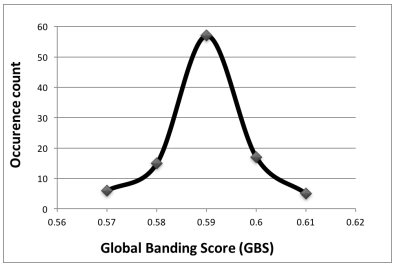
	Data sets									
GBS	100 × 100	141 × 141	173 × 173	200 × 200	224 × 224	245 × 245	265 × 265	283 × 283	300 × 300	316 × 316
0.56	1	-	-	-	-	-	-	-	-	-
0.57	18	6	1	1	-	-	-	-	-	-
0.58	60	15	5	10	-	-	-	-	-	-
0.59	19	57	26	78	1	1	-	1	-	-
0.60	2	17	46	10	20	15	2	14	5	3
0.61	-	5	21	1	58	65	18	35	20	18
0.62	-	-	1	-	20	18	61	34	53	59
0.63	-	-	-	-	1	1	17	15	18	18
0.64	-	-	-	-	-	-	2	1	4	2
Total	100	100	100	100	100	100	100	100	100	100

Table 10.2: Mean and Standard Deviation values extracted from data presented in Table 10.1 (static dot density)

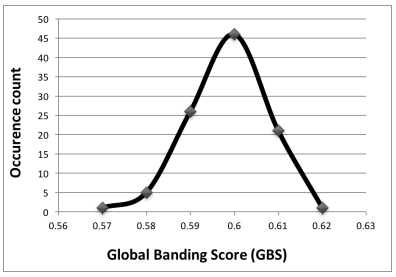
		Data sets									
		100 × 100	141 × 141	173 × 173	200 × 200	224 × 224	245 × 245	265 × 265	283 × 283	300 × 300	316 × 316
	μ	0.58	0.59	0.60	0.61	0.61	0.61	0.62	0.615	0.62	0.62
	σ	0.01	0.01	0.02	0.01	0.02	0.01	0.01	0.02	0.01	0.01
1SD	$\mu - \sigma$	0.57	0.58	0.58	0.60	0.59	0.60	0.61	0.595	0.61	0.61
	$\mu + \sigma$	0.59	0.60	0.62	0.62	0.63	0.62	0.63	0.635	0.63	0.63
2SD	$\mu - 2\sigma$	0.56	0.57	-	0.59	-	0.59	0.60	-	0.60	0.60
	$\mu + 2\sigma$	0.60	0.61	-	0.63	-	0.63	0.64	-	0.64	0.64



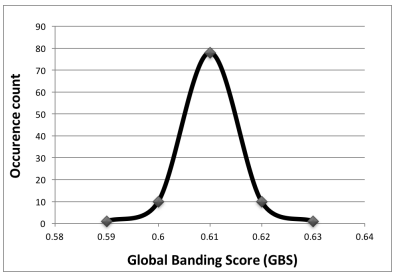
(a) 100×100



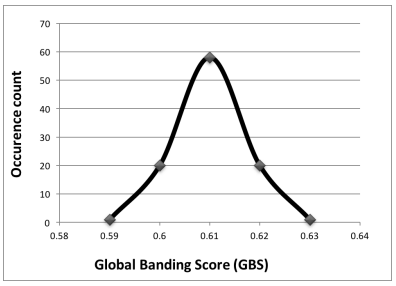
(b) 141×141



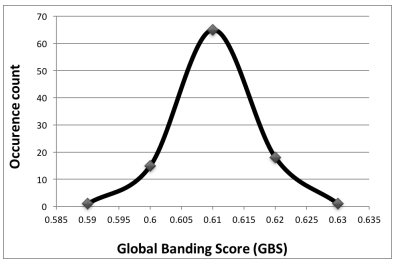
(c) 173×173



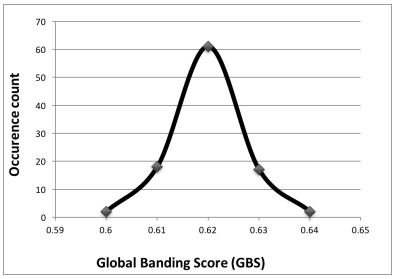
(d) 200×200



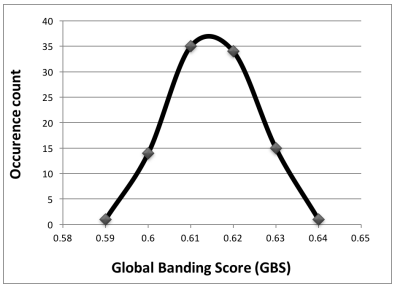
(e) 224×224



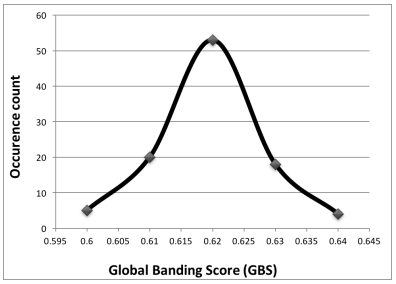
(f) 245×245



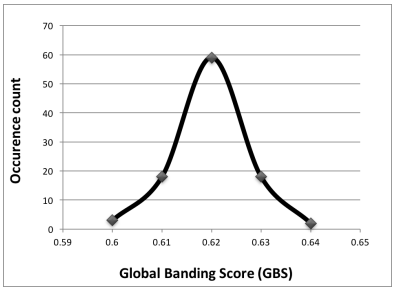
(g) 265×265



(h) 283×283



(i) 300×300



(j) 316×316

FIGURE 10.4: Standard distribution curves for data presented in Table 10.1 (static dot density)

10.3.2 Range of Dot Density Values

This subsection considers the normal distributions that result with respect to data set generated using a range of dot density values instead of a static value. More specifically dot density values ranging from 10% to 50% increasing in steps of 10%. In the same manner as in the previous subsection. Table 10.3 lists the natural GBS occurrence counts for each data set configuration (without banding), whilst Table 10.4 lists the accompanying μ, σ and one and two standard deviation limits. The associated normal distribution curves are given in Figure 10.5. Inspection of the figure indicates that similar distributions are produced; however, comparison with the distribution curves presented previously in Figure 10.4 indicates a marked difference in shape indicating that it is not a "one size fits all" situation. The significance of the distribution curves, as already noted was that they can be used to compare the GBS values obtained from data sets (same size but different dot densities) after banding has taken place to determine if the resulting banding is statistically significant or not.

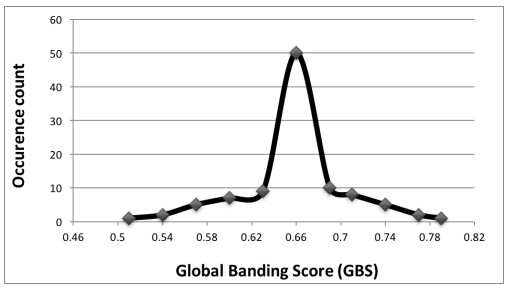
Table 10.3: List of GBS Occurrence Counts per data set configuration (ranged dot density)

	Data sets									
GBS	100 × 100	141 × 141	173 × 173	200 × 200	224 × 224	245 × 245	265 × 265	283 × 283	300 × 300	316 × 316
0.51	1	1	-	-	-	-	-	-	-	-
0.52	-	1	1	1	1	2	-	1	-	-
0.53	-	-	-	-	-	3	1	-	1	1
0.54	2	-	5	3	3	-	1	3	-	-
0.55	-	-	-	-	-	5	-	-	3	5
0.56	-	-	6	-	5	-	2	-	-	-
0.57	5	-	-	-	-	7	-	-	7	7
0.58	-	3	8	4	7	-	-	5	-	-
0.59	-	-	-	-	-	-	-	-	9	-
0.60	7	-	14	5	10	9	4	7	-	10
0.61	-	-	-	-	-	-	-	-	-	-
0.62	-	4	-	9	12	12	10	-	-	-
0.63	9	-	-	-	-	-	-	10	14	12
0.64	-	5	-	-	25	27	15	-	-	-
0.65	-	9	27	-	-	-	-	-	-	-
0.66	50	-	-	15	11	-	-	-	-	32

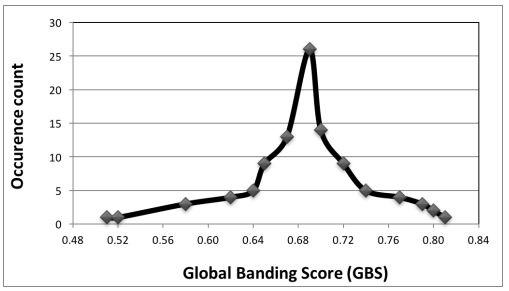
0.67	-	13	-	-	-	-	-	12	31	-
0.68	-	-	-	23	10	10	35	-	-	-
0.69	10	26	-	-	-	-	-	26	-	-
0.70	-	14	15	15	-	-	-	-	-	-
0.71	8	-	-	-	-	-	-	-	-	-
0.72	-	9	9	10	7	9	14	11	15	11
0.73	-	-	-	-	-	-	-	-	-	-
0.74	5	5	8	6	5	7	9	10	9	9
0.75	-	-	-	-	-	-	-	-	-	-
0.76	-	-	-	5	-	-	5	-	-	-
0.77	2	4	-	-	-	-	-	-	-	-
0.78	-	-	6	3	3	5	2	7	7	7
0.79	1	3	-	1	-	-	-	5	-	-
0.80	-	2	1	-	1	3	1	2	3	5
0.81	-	1	-	-	-	1	1	1	1	1
Total	100	100	100	100	100	100	100	100	100	100

TABLE 10.4: Mean and Standard Deviation values extracted from data presented in Table 10.3 (ranged dot density)

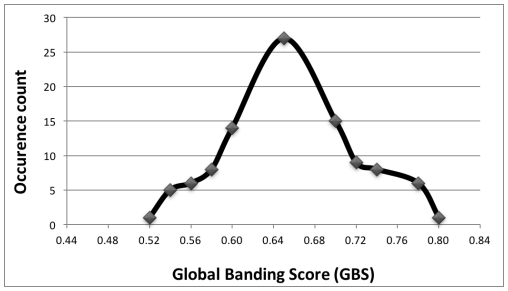
		Data sets									
		100 × 100	141 × 141	173 × 173	200 × 200	224 × 224	245 × 245	265 × 265	283 × 283	300 × 300	316 × 316
	μ	0.66	0.68	0.65	0.68	0.64	0.64	0.68	0.68	0.67	0.66
	σ	0.06	0.07	0.06	0.09	0.07	0.09	0.07	0.09	0.07	0.09
1SD	$\mu - \sigma$	0.60	0.61	0.57	0.59	0.57	0.55	0.61	0.59	0.60	0.57
	$\mu + \sigma$	0.72	0.75	0.71	0.77	0.71	0.73	0.75	0.77	0.74	0.75
2SD	$\mu - 2\sigma$	0.54	-	0.53	-	0.50	-	0.54	-	0.53	-
	$\mu + 2\sigma$	0.78	-	0.77	-	0.78	-	0.82	-	0.81	-



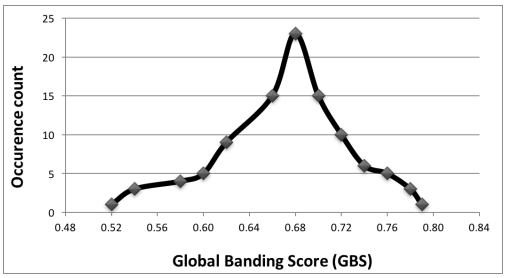
(a) 100×100



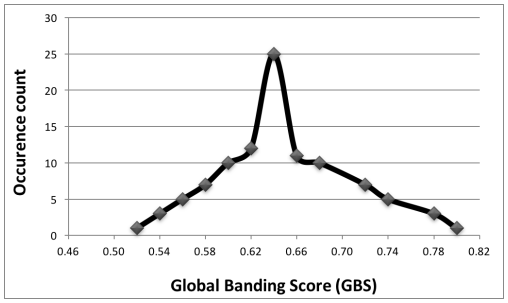
(b) 141×141



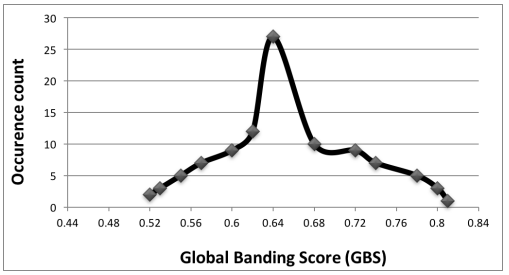
(c) 173×173



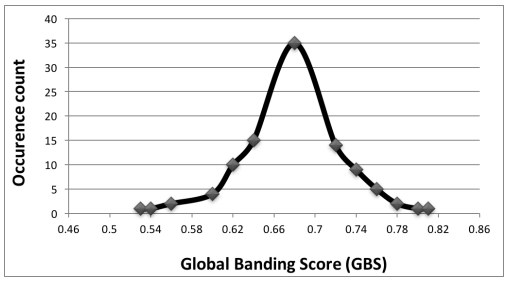
(d) 200×200



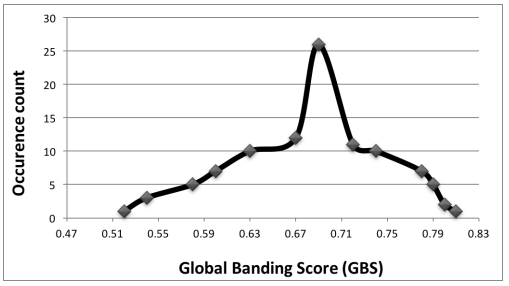
(e) 224×224



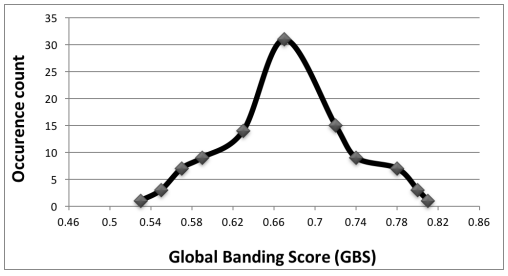
(f) 245×245



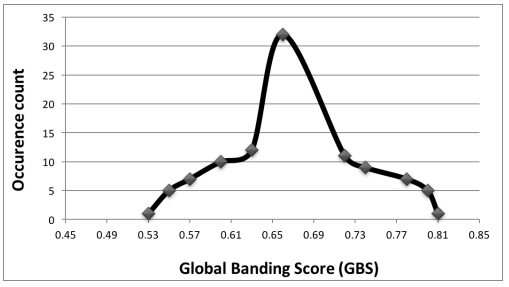
(g) 265×265



(h) 283×283



(i) 300×300



(j) 316×316

FIGURE 10.5: Standard distribution curves for data presented in Table 10.3 (ranged dot density)

10.4 Banded Pattern Mining Significance Testing

To evaluate the proposed approach to significance testing of generated banded patterns two set of experiments were conducted using: (i) a static dot density value of 10% and (ii) a range of dot density values (the same range as used to generate the distribution curves described above). In each case a number of additional synthetics data sets were generated, 10 for each of the dot data set configuration used above to generate distribution curves. The resulting GBS values produced as a result of applying banding were then compared with the normal distributions. Note that for this purpose the 2D-BPM banding algorithm (reported on in Section 4.5 of Chapter 4) was used. The results are presented in Tables 10.5 and 10.6. In the tables, for each data set configuration, the columns indicate: (i) the average GBS value obtained after banding, (ii) the distance of the average GBS value from the corresponding (μ) value shown in Tables 10.2 and 10.4 as appropriate, (iii) whether the results were significant or not (yes/no) with respect to one standard deviation (1SD) and (iv) whether the results were significant or not (yes/no) with respect to two standard deviation (2SD). From the tables it can be seen that the generated average GBS values after banding had been applied in every case was found to be located at least one or two standard deviations away from the median. It is therefore argued that these bandings are statistically significant. The results also show that the proposed mechanism of determining the statistical significant of bandings is a viable approach; the normal (Gaussian) distribution can be effectively used to determine the statistical significance of bandings.

Table 10.5: GBS results with Normal Distribution (static dot density)

Data Set	**Mean GBS**	**Distance from μ**	**Significant w.r.t 1SD (yes/no)**	**Significant w.r.t 2SD (yes/no)**
100 × 100	0.41	0.02	no	yes
141 × 141	0.42	0.01	yes	no
173 × 173	0.43	0.01	yes	no
200 × 200	0.44	0.01	yes	no
224 × 224	0.46	0.02	no	yes
245 × 245	0.45	0.02	no	yes
265 × 265	0.45	0.03	no	yes
283 × 283	0.46	0.02	yes	no
300 × 300	0.46	0.03	no	yes
316 × 316	0.46	0.03	no	yes

10.5 Summary

This chapter has presented some ideas on how to determine whether generated bandings are statistically significant or not. Two set of experiments were conducted using: (i) a static dot density value and (ii) a range of dot density values. The idea was that any dot data set irrespective of the dot density used and size, will feature some form of banding

TABLE 10.6: GBS results with Normal Distribution (ranged dot density)

Data Set	**Mean GBS**	**Distance from μ**	**Significant w.r.t 1SD (yes/no)**	**Significant w.r.t 2SD (yes/no)**
100 × 100	0.57	0.10	no	yes
141 × 141	0.57	0.09	no	yes
173 × 173	0.58	0.09	no	yes
200 × 200	0.59	0.09	yes	no
224 × 224	0.59	0.09	no	yes
245 × 245	0.59	0.09	yes	no
265 × 265	0.59	0.09	no	yes
283 × 283	0.59	0.09	yes	no
300 × 300	0.60	0.09	no	yes
316 × 316	0.60	0.09	yes	no

defined by a GBS value and these values will form a normal distribution. Whether, after indexes have been reordered using the banding score concept, the resulting banding is significant or not can then be determined by how far the new GBS value is away from the mean of the associated normal distribution (μ). To analyse this approach twenty normal distributions were derived using ten 2D data set configurations. The usage of these distributions were then evaluated by using them to determine the significance of a number of further bandings. The evaluation results presented indicated the significance of bandings with respect to either 1SD or 2SD. A criticism of the approach is that the normal distribution for a data set under consideration has to be derived in each case, however, the results show that it is possible to generate generic normal distribution curves using ranges of density values (but a fixed size). The experiments clearly indicated a useful mechanism for determining whether a banding is statistically significant or not. In the following chapter, the thesis is concluded with a summary of the work presented, along with the main findings in the context of the research objectives presented in Chapter 1, and some suggestions for future work.

Chapter 11

Conclusion and Future Research Works

11.1 Introduction

This concluding chapter presents an overall summary of work described in this thesis with the main findings and contributions. The chapter also provide some suggestions for future work. The chapter is arranged as follows. In Section 11.2, a summary of the thesis is presented. The main finding and contributions are then reported in Section 11.3. Finally some suggested ideas for future work are presented in Section 11.4 in the context of further potential areas of research based on the work described in this thesis.

11.2 Summary

This section presents the summary of the work presented in this thesis. The thesis commenced in Chapter 2 with a review of previous work and then went on in Chapter 3 to consider the data sets used with respect to the evaluations reported on later in the thesis. Three categories of data set were considered: (i) synthetic data sets generated using the random data generator proposed in [29], (ii) selected data sets from the UCI machine learning data repository and (iii) data sets extracted from the GB Cattle Tracing System (CTS) database. The second and third categories were the main focuses of the evaluations presented in this thesis, while the first was intended to illustrate the wider applicability of the BPM idea. For the second and third categories, the raw data sets were translated into a zero-one format so that BPM, as envisaged in this thesis, could be applicable.

The following five chapters, presented a sequence of BPM algorithms of increasing sophistication directed at larger and larger dot data sets, from 2D to "big data" commencing with 2D data in Chapter 4. Each of these chapters were structured in a similar manner comprising formalism, algorithm and evaluation sections. Chapter 4 presented the required formalism for 2D banding and presented the 2D-BPM algorithm. The operation of this algorithm was compared with three previously proposed banding

algorithms: (i) the Barycenter (BC), (ii) the Minimum Banded Augmentation "Bidirectional Fixed Permutation" (MBA_{BFP}) and (iii) the Minimum Banded Augmentation "Fixed Permutation" (MBA_{FP}) algorithms. The reported evaluation indicated that the proposed 2D-BPM algorithm produced better results than the other three banding algorithms considered, both in terms of GBS and the independent ABW metric. The proposed 2D-BPM algorithm was also consistently more efficient than the other three algorithms considered, because it did not require the generation and testing of many permutations.

Chapters 5 and 6 then considered two alternative 3D approaches to BPM, approximate and exact. The chapters provided essential "stepping stone" material for the work on ND-BPM presented in the following chapter. In Chapters 5 and 6 the A3D-BPM and E3D-BPM algorithms were proposed. The first was a variation of the 2D-BPM algorithm presented in the previous chapter but applied to 3D in that dimension pairings were considered. An important element of the proposed E3D-BPM algorithm was the number of maximum distance calculations required for banding purposes. The idea presented was to precalculate the maximum distances and store these in an M-Table. It was also noted that, in the case of the E3D-BPM algorithm, there were a number of ways of calculating the distances of dots from the origin of the data space. Thus, two variations of the E3D-BPM algorithm were proposed, Euclidean and Manhattan. The reported evaluation comparing the usage of the A3D-BPM and the E3D-BPM algorithms found that: (i) the GBS values produced using exact BPM were better than those produced using the A3D-BPM algorithm, (ii) the Euclidean variation of the E3D-BPM algorithm was more effective than the Manhattan variation, (iii) in the case of the E3D-BPM algorithm the use of M-Tables produced efficiency advantages, (iv) A3D-BPM was more efficient than E3D-BPM (regardless of whether Euclidean or Manhattan distance calculation was adopted or M-Tables were used or not) and (v) the Manhattan variation of the E3D-BPM algorithm was more efficient than the Euclidean variation.

Chapter 7 proposed two N-Dimensional BPM algorithms founded on the 3D algorithms presented in the foregoing chapters, the AND-BPM and END-BPM algorithms. The main issue here was how best to scale up the 3D BPM algorithm to operate in ND. The evaluation was conducted by comparing the usage of the AND-BPM and the END-BPM in terms of GBS and runtime (using M-Tables and without M-Tables, and using Euclidean and Manhattan distance calculation in the case of END-BPM). The recorded evaluation confirmed the results obtained with respect to the evaluation for 3D-BPM algorithms. Namely that the AND-BPM algorithm is more efficient than the END-BPM algorithm (regardless of whether Euclidean or Manhattan distance calculation was adopted or the usage of M-Tables or not), while the best banding with respect to GBS was produced using the END-BPM algorithm with Euclidean distance calculation.

Chapter 8 proposed the MD-BPM algorithm in the context of the approximate and exact BPM approaches presented earlier. Two multiple dot BPM algorithms were proposed, the MD-ABPM and MD-EBPM algorithms. The main issue here was how to

address a situation where a location holding multiple dots can be banded. The significance of the proposed multiple dot BPM algorithms was with respect to the banding of very large (big data) dot data sets. This was considered in Chapter 9 where two techniques for banding very large dot data sets were proposed, sampling and segmentation. Sampling involved applying a banding to a data set sample and applying this to the entire data set. The issue here was how best to identify an appropriate sample. Segmentation involved dividing the data set into chunks, called "segments", banding each segment and then combining the banding definitions (index arrangements). The issue here was how best to combine the configurations, two mechanisms were proposed: best GBS and most frequent. The recorded evaluation indicated that the MD-EBPM algorithm was less efficient than the MD-ABPM algorithm. However, the best banding with respect to the GBS was produced using the MD-EBPM algorithm with Euclidean distance calculation. Of the big data banding techniques it was found that the segmentation technique combined with most frequent banding segment selection was the best. It was also noted that when using either sampling or segmentation the overall GBS for the entire data set improved compared to the GBS when no banding was applied.

Chapter 10 then considered the statistical significance of the bandings that might be produced using the proposed BPM algorithms. The idea presented was that the GBS values associated with a particular data set size and density will have a normal distribution associated with it and that this distribution could be used to determine the significance of bandings in terms of how far a GBS value resulting from a banding exercise was from the mean of the associated distribution. This in turn could be expressed in terms of standard deviations.

11.3 Main Findings and Contribution

This section revisits the overriding research question presented in Chapter 1 (Section 1.5), and the associated subsidiary research questions. The section addresses these in terms of the "main findings" of the research presented in this thesis. The section is organised by considering each of the identified subsidiary research questions first and then returning to the overriding research question.

1. **Mechanisms and Techniques**: *"What mechanisms and techniques can best be employed to identify a best banding? What are the most suitable techniques for obtaining a best banding?"*. The challenge of which mechansim and techniques can best be employed to identify a best banding was resolved initially by proposing the "banding score" concept. However, there were a variety of ways in which banding scores could be calculated depending on the number of dimensions and size of the dot data sets under consideration. The idea of banding scores was incorporated into a sequence of BPM algorithms which operated by iterating over dimensions and reordering each dimension in turn; in case of the approximate algorithms the same dimension might be reordered several times in a single iteration. According

to the conducted evaluation, presented in Chapter 4, the usage of the banding score mechanism incorporated into a BPM algorithm was found to be significantly more effective than the other banding algorithms considered. The reason for this was that the proposed 2D-BPM mechanism identified bandings such that the indexes in each dimension were allocated a banding score which could be used to rearrange the indexes, thus avoiding the computational expense of considering large numbers of permutations (as in the case of some of the other algorithms considered). The conducted experimental analysis established that the best mechanism/technique for identifying bandings was to use the banding score concept proposed by the author (because it avoided the consideration of large numbers of permutations and because it produced better bandings).

2. **"Best" Banding**: *"What is a banding? How is a best banding determined? How is the goodness of a banding measured?"*. Banding was defined, with respect to the proposed BPM algorithms, in terms of a final Global Banding Score (GBS) arrived at the end of the proposed process, a number between "0" and "1", where a GBS of 0 indicates a best (perfect) banding and "1" the worst (most imperfect) banding. The GBS value was the value that the proposed BPM algorithms thus wished to minimise. It was noted that the competitor algorithms used alternative mechanisms for measuring best banding, and thus it would be unfair to measure their performance using GBS. Hence an independent measure, the ABW metric, was used for comparison purposes (with good results with respect to the proposed BPM algorithms). Overall it was found that the proposed GBS measure was an effective measure for measuring banding quality. Note that with respect to the competitor algorithms BC seeks to maximise the MRM and MBA seeks to maximise accuracy, while the proposed BPM algorithm seeks to minimise GBS. Hence, for a fair comparison, an independent mechanism, the ABW mechanism, was proposed that measures the quality of banding in an independent manner.

3. **ND Banded Data**: *"What are the mechanisms that can best be employed to ensure that any proposed banding algorithm will scale up to operate in ND?"*. The challenge of determining whether the proposed BPM algorithm would scale up was initially addressed by considering the development of a number of ND algorithms and variations. The expedient of the use of M-Tables was also considered. The END-BPM and AND-BPM algorithms were proposed for identifying bandings in ND data. However, whatever algorithm is used, there will always be a point where a dot data set is too large to be processed in primary storage. To this end two techniques were proposed whereby very large data sets (big data sets) could be processed, sampling and segmentation. The reported evaluation conducted to evaluate these two techniques indicated that these techniques could be successfully applied to find bandings in large (big) dot data sets.

4. **Multiple "Dots"**: *"How is the issue of more than one "1" value (dot) being located at a location in a ND data matrix best addressed?"*. The sampling and segmentation techniques proposed to address the banding of very large data sets entailed the selection of a reference dimension with respect to which the sampling/segmentation would be conducted. This dimension would therefore need to be excluded from the banding exercise. This in turn resulted in locations in the remaining data matrix possibly holding more than one dot. Initially it was assumed that locations could only hold one dot, the proposed sampling and segmentation techniques thus necessitated that this assumption could no longer hold. The challenge of having more than one dot at individual locations in ND zero-one (dot) data was resolved by proposing the Multiple Dots mechanism in the context of both the approximate and exact BPM algorithms proposed earlier leading to the MD-ABPM and MD-EBPM algorithms. These algorithms were then employed in the context of the proposed sampling and segmentation techniques to band very large dot data sets.

5. **Statistically Significant**: *"What is the most appropriate mechanism for determining whether a best banding, when identified, is statistically significant or not? "*. The challenge of identifying the most appropriate mechansim for determining whether a derived banding was statistically significant or not was addressed towards the end of the thesis. The idea was to use the anticipated standard distribution for a given dot data set configuration. Evaluation of the proposed approach indicated that this was a good mechanism for establishing the statistical significance of generated bandings using the proposed BPM algorithms.

Returning the main research question:

What are the most appropriate mechanisms and techniques required to identify banded patterns in ND zero-one data spaces in a manner that is both effective and efficient?

From the foregoing, a number of BPM algorithms were considered founded on the concept of the "Banding Score" mechanism. Of note were the following algorithms: (i) 2D-BPM, (ii) A3D-BPM, (iii) E3D-BPM, (iv) AND-BPM (v) and END-BPM, (vi) MD-ABPM and (vii) MD-EBPM. From the evaluation conducted, each of the BPM algorithms provided different advantages. However, the best banding was produced using the exact BPM algorithms in the context of 3D and ND data sets, whilst the most efficient was the approximate BPM algorithm (also in the context of both 3D and ND data). The Multiple Dot (MD-BPM) mechanism was proposed to address the possibility of some cells holding more than one dot which in turn was utilised in the context of sampling and segmentation for banding very large data sets. A mechanism for determining the statistical significance of the bandings produced was also formulated. Overall the reported evaluations indicated that by using the BPM algorithms effective bandings can be achieved.

The main contributions of the research presented in this thesis were presented in Chapter 1. These are restated here, for completeness, as follows:

1. The concept of a **banding score** that supports the identification of bandings in zero-one data without considering large numbers of permutations (the reason being that the proposed BPM algorithms presented in this thesis use the banding score concept that avoids the need to generate and test large numbers of permutations by assigning to each individual index in each individual dimension banding scores and then reordering accordingly) (Chapters 4, 5, 6, 7, 8 and 9). This is arguably the most significant contribution of the work.
2. The 2D-BPM algorithm for discovering bandings in 2D data sets (Chapter 4).
3. The Approximate 3D (A3D) and Exact 3D (E3D) BPM algorithms, including the Euclidean and Manhattan variations of the E3D-BPM algorithm (Chapters 5 and 6).
4. The Approximate ND (AND) and Exact ND (END) BPM algorithms (Chapter 7).
5. A mechanism for addressing the situation where a location holds multiple dots (Chapter 8) in the context of both approximate and exact BPM (the MD-ABPM and MD-EBPM algorithms).
6. A mechanism for applying bandings to very large data sets using a sampling technique integrated into the banded pattern mining process (Chapter 9).
7. A mechanism for applying bandings to very large data sets using a segmentation technique integrated into the banded pattern mining process (Chapter 9).
8. An independent mechanism, the Average Band Width (ABW) mechanism, for measuring the quality of a banding to support comparison of BPM algorithms (ABW calculates the average distance of dots from the main diagonal and is measured according to the length of the normal from each dot to the leading diagonal, while the GBS mechanism calculates the normalised sum of the individual banding scores)(Chapter 4).
9. A mechanism for considering the statistical significance of an identified banding (Chapter 10).
10. Some insights into the CTS database (Chapter 3).

11.4 Future Work

The work presented in this thesis has demonstrated that in the context of ND zero-one data, banded pattern mining can be effectively achieved using the proposed BPM

algorithms. Despite the results produced, improvements and enhancements can be envisioned. This concluding section suggests some potential areas for future work as follows.

1. **Utilising alternative high performance computing approaches to Banded Pattern Mining (BPM)**: One limiting factor of the proposed BPM algorithms, as discussed in the foregoing chapters, was the computing resources required to identify bandings in very large data set. Although, two techniques were considered, sampling and segmentation, another potential solution that merits further investigation is the adoption of some form of multi-core or parallel computing solution to bandings that will improve the efficiency and effectiveness of the BPM algorithms, allowing them to be applied to very large ND data sets. Investigating of appropriate parallel approaches is thus considered to be a fruitful avenue for future work.

2. **Alternative Evaluation**: To date the proposed BPM algorithms have only been applied to: (i) synthetically generated data sets, (ii) data sets from the UCI machine learning repository and (iii) data sets extracted from the Great Britain (GB) Cattle Tracing System (CTS). A much wider evaluation seems desirable. Even in the context of the CTS application, the data sets used with respect to the work presented in this thesis, were limited to only four specific counties (Aberdeenshire, Cornwall, Lancashire and Norfolk); in the context of the CTS application it would be beneficial to consider a greater number of counties.

3. **Using alternative data sets especially non-binary data**: Only binary valued data sets was considered with respect to the work presented in this thesis, this was because of the wide usage of such data sets in many application domains. Where necessary, for evaluation purposes, data sets were translated into this format. It is suggested that it would be worth investigating bandings in a non-binary data set contexts. The idea here, is to relax the requirement for bandings from zero-one data to either positive integers or real valued numbers, where the structure in the banded patterns can be described as a variation from large to small values. The assumption is that in the case of a real valued dataset, the banding score for each index in each dimension can be calculated by taking into account the values in each cell (location).

4. **Visualisation of ND Banded Patterns**: The ability to generate a visualisation for a banded pattern and display the banding result graphically can be difficult to comprehend, especially in the context of ND, therefore an effective visualisation tool is desirable. The availabilty of such a tool would be of great help to users in that it would: (i) provide valuable insights into the data sets and (ii) provides different views of the data sets. Recall, referring back to chapter 2, that one of the motivations for banding was data visualisation.

5. **Investigation of other ways of assessing the statistical significance of Banded Patterns**: In chapter 10 a mechanism for assessing the statistical significance of bandings was proposed. This was shown to operate well, however it necessitated the generation of normal distributions for each data set configuration (defined in terms of number of row and columns and the density). Better mechanisms for determining the statistical significance of generated bandings would therefore be a fruitful avenue for further research.

6. **Further improvement on M-Table generation**: In chapters 6 and 7 the idea of M-Tables was presented in the context 3D and ND exact BPM. The advantage offered by the use of M-Tables was that it increased the efficiency of the proposed exact BPM algorithms. Using the proposed algorithms one global M-Table was generated given a particular banding problem. However, it is suggested that another way of doing this might be to store the maximum distance values for each dimension separately using individual M-Tables. The assumption here is that this might further improve the efficiency of the proposed Exact BPM algorithms and therefore provide another suggested area for further research.

Overall the work presented in this thesis has produced a significant improvement over alternative approaches to identifying bandings in 2D data; an approach that scales up to higher dimensions.

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N.Swami. Mining association rules between sets of items in large databases. *SIGMOD'93*, pages 207–216, 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings 20th International Conference on Very Large Data Bases, VLDB*, pages 487–499, 1994.

[3] Elmustapha Syed Ali Ahmad and Rashid A. Saeed. A survey of big data cloud computing security. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 3:78–85, 2014.

[4] Shakil Ahmed, Frans Coenen, and Paul Leng. Data structures for association rule mining:t-trees and p-trees. *IEEE Transactions on Data and Knowledge Engineering*, 16:774–778, 2006.

[5] Shakil Ahmed, Frans Coenen, and Paul Leng. Tree- based partition of data for association rule mining. *Knowledge and Information Sytems*, 10:315–331, 2006.

[6] R. V. Akhila and R. G. Rakesh. Study and analysis of big data in cloud computing. *International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS)*, 3:416–422, 2015.

[7] Farid Alizadeh, Richard M. Karp, Lee A. Newberg, and Deborah K. Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. 13:52–76, 1995.

[8] E. L. Allgower. Exact inverse of certain band matrices. *Numerical Mathematics*, 21:279–284, 1973.

[9] P Arabie and L.J Hubert. An overview of combinatorial data analysis. in p arabie, lj hubert, g soete (eds)clustering and classification. *World Scientific, River Edge, NJ*, 27:5–63, 1996.

[10] Jonathan E. Atkins, Erik G. Boman, and Bruce Hendrickson. Spectral algorithm for seriation and the consecutive ones problem. *Journal of Computing SIAM*, 28:297–310, 1998.

[11] Cevdet Aykanat, Ali Pinar, and Umit V. Catalyurek. Permuting sparse rectangular matrices into block-diagonal form. *Journal on Scientific Computing (SIAM)*, 25:1860–1879, 2004.

[12] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.

[13] Arindam Banerjee, Chase Krumpelman, Joydeep Ghosh, Sugato Basu, and Rahmand J. Mooney. Model-based overlapping clustering. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, pages 532–537, 2005.

[14] Stephen T. Barnard, Alex Pothen, and Horst D. Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications*, 2:317–334, 1995.

[15] Jacques Bertin. *Graphics and graphic information processing, chapter 1D, 2D, 3D, pages 6265.* Morgan Kaufmann Publishers Incorporation, San Francisco, CA, USA, 1981.

[16] Jacques Bertin. Graphics and graphic information processing. *In Stuart K. Card, Jock Mackinlay and Ben Shneiderman editors, Readings in information visualisation: using vision to think, Morgan Kaufmann, San Francisco, USA*, pages 62–65, 1999.

[17] Jacques Bertin. Matrix theory of graphics. *Information Design Journal*, 10:5–19, 2001.

[18] C. I. Blake and C. J. Merz. Uci repository of machine learning databases. `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.htm`, 1998.

[19] Z. Bohte. Bounds for rounding errors in the gaussian elimination for band systems. *Journal of Applied Mathematics*, 16:133–142, 1975.

[20] Kellogg S. Booth and George S. Lueker. Testing of the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Science*, 13:335–379, 1976.

[21] Christian Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementation ACM Press*, pages 1–5, 2005.

[22] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic item-set counting and implication rules for market basket data. In *Proceedings ACM SIGMOD Conference*, pages 255–256, 1997.

[23] Ivan Bruha. From machine learning to knowledge discovery: Survey of preprocessing and post-processing. *Journal of Intelligent Data Analysis*, 4:363–374, 2000.

[24] Audrey M. Burnam and Paul Koegel. Methodology for obtaining a representative sample of homeless persons:the los angeles skid row study. *Evaluation Review*, 12:117–52, 1988.

[25] K. Y. Cheng. Minimising the bandwidth of sparse symmetric matrices. *Computing Springer-Verlag*, 11:103–110, 1973.

[26] K. Y. Cheng. Note on minimising the bandwidth of sparse symmetric matrices. *Journal of Computing Springer-Verlag*, 11:27–30, 1973.

[27] Yizong Cheng and George M. Church. Biclustering of expression data. In *Proceedings of Intelligent Systems Molecular Biology*, pages 93–103, 2000.

[28] P. Chinn, J. Chvatalova, A. K Dewdney, and N. E. Gibbs. The bandwidth problem for graphs and matrices: a survey. *Journal of Graphs Theory*, 6:223–254, 1992.

[29] Frans Coenen. Lucs-kdd data generator software. `http://www.csc.liv.ac.uk/$\sim$frans/KDD/Software/LUCS_KDD_DataGen_Generator.html`, 2003.

[30] Frans Coenen, Graham Goulbourne, and Paul Leng. Computing association rules using partial totals. In *de Raedt, L. and Siebes, A. (Eds) Principles of Data Mining and Knowledge Discovery,Proc PKDD 2001,Springer-Verlag LNAI 2168*, pages 54–66, 2001.

[31] Frans Coenen, Graham Goulbourne., and Paul Leng. Tree structure for association rule mining. *Journal of Data Mining and Knowledge Discovery*, 8:25–51, 2004.

[32] Frans Coenen and Paul Leng. Finding association rules with some very frequent attributes. In *Procedngs of PKDD 2002 Conference,Helsinki, August 2002: Lecture Notes in AI 2431, Springer-Verlag*, pages 99–111, 2002.

[33] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithm*. MIT Press, 2001.

[34] M. Thomas Cover and A. Joy. *Elements of Information Theory*. John Wiley and Sons, 2006.

[35] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of ACM National Conference Association for Computing Machinery New York*, pages 157–172, 1969.

[36] Veronica Czitrom and Patrick D. Spagon . Statistical case studies for industrial process improvement. *SIAM*, pages 342–345, 1997.

[37] Horvitz G. Daniel. Sampling and field procedures of the pittsburgh morbidity survey. *Public Health Representation*, 67:1003–1012, 1952.

[38] G. M DelCorso and G. Manzini. Finding exact solutions to the bandwidth minimisation problem. *Journal of Computing*, 62:183–203, 1999.

[39] Stephen B. Deutsch and John J. Martin. An ordering algorithm for analysis of data arrays. *Operation Research*, 19:1350–1362, 1971.

[40] G. H Dueck and J. Jeffs. A heuristic bandwidth reduction algorithm. *Journal of Combinatorial Mathematics and Computation*, 18:97–108, 1995.

[41] IAIN S. Duff. A survey of sparse matrix research. *Proceedings of IEEE*, 65:500–535, 1977.

[42] Peter Eades and Nicholas C. Wormald . Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11:379–403, 1994.

[43] Wesstein W. Eric. Normal distribution. `http://mathworld.wolfram.com/NormalDistribution.html`, 2015.

[44] Lukac Eugene and King Edgar. A property of normal distribution. *The Annals of Mathematics*, 11, 2004.

[45] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[46] W. Feller. *Introduction to Probability Theory and Its Applications.* Wiley New York volume 1, 3rd ed, 1968.

[47] W. Feller. *Introduction to Probability Theory and Its Applications.* Wiley New York volume 2, 3rd ed, page 45, 1971.

[48] Golub H. Fene, Van Loan, and F. Charles. *Matrix Computation (3rd ed.).* John Wiley and Sons ISBN 978-0-471-50023-0, 1989.

[49] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.

[50] Fajwel Fogel, Alexandre d'Aspremont, and Milan Vojnovic. Spectral ranking using seriation. *Journal of Machine Learning Research*, 17:1–45, 2016.

[51] Paul Leng Frans Coenen and Shakil Ahmed. Data structure for association rule mining: T.trees and p-trees. *IEEE Transactions on Data and Knowledge Engineering*, 16:774–778, 2004.

[52] Petrie FWM. Sequences in prehistoric remains. *Journal of the Anthropological Institute*, 29:295301, 1996.

[53] M. R Garey, R. L Graham, D. S Johnson, and A. D. Knuth. Complexity for bandwidth minimisation. *SIAM Journal of Applied Mathematics*, 34:477–495, 1978.

[54] M. R Garey and D. S Johnson. *Computers and Intractability: A Guide to the Thoery of NP-Completeness.* W.H Freeman, 1979.

[55] Gemma C. Garriga, Esa Junttila, and Heikki Mannila. Banded structures in binary matrices. *Proceedings Knowledge Discovery in Data Mining (KDD08)*, pages 292–300, 2008.

[56] Gemma C. Garriga, Esa Junttila, and Heikki Mannila. Banded structures in binary matrices. *Knowledge Discovery and Information System*, 28:197–226, 2011.

[57] Marsaglia George. Evaluating the normal distribution. *Journal of Statistics Software*, 11, 2004.

[58] Norman E Gibbs, Jr William G. Poole, and Paul K. Stockmeyer. An algorithm for reducing the bandwidth and profile of sparse matrix. *SIAM Journal of Numerical Analysis*, 13:236–250, 1976.

[59] Bart Goethals. Survey on frequent pattern mining. technical report. *Helsinki Institute for Information Technology HIIT*, 2003.

[60] Gene H. Golub and Charles F.VanLoan. *Matrix Computations*. Johns Hopkins University Press, 1989.

[61] Sulekha Goyat. The basis of market segmentation: A critical review of literature. *European Journal Of Business and Management*, 3:45–54, 2011.

[62] D. M Green and R. R. Kao. Data quality of the cattle tracing system in great britain. *Veterinary Record*, 161:439–443, 2007.

[63] Carl Friedrich Guass. *Theory of Motion of the Heavenly Bodies Moving about the Sun in Conic Sections*. Little Brown and Company, 1857.

[64] Venkatesh H, Shrivatsa D Perur, and Nivedita Jalihal. A study on use of big data in cloud computing environment. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 6:2076–2078, 2015.

[65] Mohammad Taghi Hajiaghayi and Yashar Ganjali. A note on the consecutive ones submatrix problem. *Journal of Information Processing Letter*, 83:163–166, 2002.

[66] F. M. Hemphill. A sample survey of home injuries. *Public Health Representation*, 67:1026–1034, 1952.

[67] Roger A. Horn and Charles R. Johnson. *matrix Analysis (2nd ed.)*. Cambridge University Press ISBN 978-0-521-54823-6, 2013.

[68] Wen-Lian Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 43:1–16, 2002.

[69] Patel Jagadish, K Read, and B. l. Campbel. *Handbook on Normal Distribution (2nd ed)*. CRC Press, 1996.

[70] Ruoming Jin, Yang Xiang, David Fuhry, and Feodor F. Dragan. Overlapping matrix pattern visualization: a hypergraph approach. In *Proceedings of the 8th IEEE International Con- ference on Data Mining (ICDM08)*, pages 313–322, 2008.

[71] Roberto J. Bayardo Jr, Rakesh Agrawal, and Dimotrios Gunopolos. Constraint-based rule mining in large, dense databases. In *Proceedings of 15th International Conference on Data Engineering*, 1999.

[72] Micheal Junger and Petra Mutzel. 2-layer straight line crossing minimisation: performance of exact and heuristic algorithms.. *Journal Graph Algorithms and Applications*, 1:125, 1997.

[73] Krishnamoorty Kalimuthu. *Handbook of Statistical Distribution with Applications*. Chapmall and Hall / CRC Press: ISBN 1-58488-635-8, 2006.

[74] Atkinson A. Kendall. *Introduction to Numerical Analysis (2nd ed.)*. John Wiley and Sons ISBN 978-0-471-50023-0, 1989.

[75] Adrienne Kirby, Val Gebski, and Anthony C. Keech. Determining the sample size in a clinical trial. *Medical Journal*, 177:256–257, 2002.

[76] M. Koebe and J. Knochel. On the block alignment problem. *Journal of Information Processing Cybernet*, 26:377–378, 1990.

[77] Y. Koren. Drawing graphs by eigenvectors: theory and practice. *Computers and Mathematics with Applications*, 49:1867–1888, 2005.

[78] Sugiyama Kozo, Tagawa Shojiro, and Toda Mitsuhiko. Methods for visual understanding of hierarchical system structures. *IEEE Transaction on Systems, Man and Cybernetics*, 11:109–125, 1981.

[79] Paul S. Levy and Stanley Lemeshow. *Sampling of populations: Methods and applications*. 2008.

[80] Zhi-Chao Li, Pi-Lian He, and Ming Lei. A high efficient aprioritid algorithm for mining association rule. *Machine Learning and Cybernetics*, 3:1812–1815, 2005.

[81] Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining*, 3(2):7091, 2010.

[82] Innar Liiv, Rain Opik, Jaan Ub1, and John Stasko. Visual matrix explorer for collaborative seriation. *Journal of Machine Learning Research*, 4:8587, 2012.

[83] Andrew Lim, Brian Rodrigues, and Fei Xiao. Discrete optimisation: Heuristics for matrix bandwidth reduction. *European Journal of Operational Research*, pages 69–91, 2006.

[84] Yong Ma, Shihong Lao, Erina Takikawa, and Masato Kawade. Discriminant analysis in correlation similartity measure space. In *Proceedings of the 24th International Conference on Machine Learning ICML'07 New York, NY USA*, pages 577–584, 2007.

[85] Oded Maimon and Lior Rokach. Data mining and knowledge discovery handbook, chapter introduction to knowledge discovery and data mining. *Springer*, pages 1–18, 2010.

[86] Hekki Mannila and Evimaria Terzi. Nestedness and segmented nestedness. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining New York ACM*, pages 480–489, 2007.

[87] Lipson Marc and Lipschutz Seymour. *Schaum's outline of theory and problems of Linear Algebra.* McGraw-Hill New York pp69-80 ISBN 978-0-07-136200-9, 2001.

[88] Robert Markland. *Topics in Management Science 3rd ed.* John Wiley and Sons New York, 1989.

[89] R. S Martin and J. H. Wilkinson. Solution of symmetric and unsymmetric band equations and the calculation of eigenvalues of band matrices. *Numerische Mathematik*, 9:279–301, 1967.

[90] Rafael Marti, Manuel Laguna, Fred Glover, and Vicente Campos. Reducing the bandwidth of a sparse matrix with tabu search. *European Journal of Operational Research*, 135:211–220, 2001.

[91] Erkki Makinen and Harri Siirtola. Reordering the reorderable matrix as an algorithmic problem. In *Proceedings of the 1st International Conference on the Theory and Application of Diagrams (Diagrams00) Springer-Verlag*, pages 453–2467, 2000.

[92] Erkki Makinen and Harri Siirtola. The barycenter heuristic and the reorderable matrix. *Informatica*, 29:357–363, 2005.

[93] Chris Mueller. Sparse matrix reordering algorithms for cluster identification. *Machine Learning in Bioinformatics*, 2004.

[94] Samuel Myllykangas, J. Himberg, T. Bohling, B. Nagy, J. Hollman, and S. Knuutila. Dna copy number amplification profiling of human neoplasms. *Oncogene*, 25:7324–7332, 2006.

[95] M. E Newman and M. Girvan. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences USA (PNAS) 99(12)*, pages 7821–7826, 2003.

[96] Andrew Y. Ng, Micheal I. Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. in t. dietterich, s. becker, and z. ghahramani (eds.). *Advances in Neural Information Processing Systems MIT Press*, 14:849–856, 2002.

[97] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of International MultiConference of Engineers and Computer Scientist IMECS*, pages 380–384, 2013.

[98] Puteri N. E. Nohuddin, Rob Christley, Frans Coenen, and Christian Setzkorn. Trend mining in social networks: A study using a large cattle movement database. *Advances in Data mining, Applications and Theoretical Aspects LNAI 6171*, pages 464–475, 2010.

[99] Johnson L. Norman, Kotz Samuel, and Balakrishnan Narayanaswamy. *Continuos Univariate Distribution Volume 1.* Wiley ISBN 0-471-58495-9, 1994.

[100] Johnson L. Norman, Kotz Samuel, and Balakrishnan Narayanaswamy. *Continuos Univariate Distribution Volume 2.* Wiley ISBN 0-471-58495-9, 1995.

[101] Marcus Oswald and Gerhard Reinelt. The weighted consecutive ones problem for a fixed number of rows or columns. *Operations Research Letters*, 31:350–356, 2003.

[102] Ch. H Papadimitriou. The np-completeness of the bandwidth minimisation problem. *Computing*, 16:263–270, 1976.

[103] Eatefenia Pinana, Isaac Plana, Vicente Campos, and Rafeal Marti. Grasp and path relinking for the matrix bandwidth minimization. *In print European Journal of Operational Research. http://www.uv.es/marti/*, 2001.

[104] Alex Pothen, Horst D. Simon, and Kang-Pu Paul Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal of Matrix Analysis Application*, 11:430–452, 1990.

[105] F. Pukelsheim. The three sigma rule. *American Statistician*, 48:88–91, 1994.

[106] Kai Puolamki, Mikeal Fortelius, and Heikki Mannila. Seriation in paleontological data using markov chain monte monte carlo methods. *PLoS Computational Biology*, 2, 2006.

[107] Herrnstein J. Richard and Murray Charles. *The Bell Curve: Intelligence and Class Structure in American Life.* Free Press ISBN 0-02-914673-9, 1994.

[108] CJ. Van. Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, 1979.

[109] Susan E. Robinson and Rob M. Christley. Identifying temporal variation in reported birth, death and movements of cattle in britain. In *BMC Veterinary Research*, pages 2–11, 2006.

[110] W. S Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16:293–301, 1951.

[111] Antonio Robles-Kelly and Edwin .R . Hancock. Graph edit distance from spectral seriation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:365–378, 2005.

[112] Brian Rodrigues, Fei Xiao, and Andrew Lim. A new node centroid algorithm for bandwidth minimisation. *18th International Joint Conference on Artificial Intelligence (IJCAI) Mexico. Research Collection Lee Kong China School Business*, 2003.

[113] Richard Rosen. Matrix bandwidth minimisation. In *Proceedings of ACM National conference*, pages 585–595, 1968.

[114] Dibb Sally and Simkin Lyndon. The market segmentation workbook: Target marketing for marketing managers. *Routledge London*, 1996.

[115] Jianbo Shi and Jitendra Malik. Normalised cuts and image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[116] Harri Siirtola and Eekki Makinen. Constructing and reconstructing of reorderable matrix. *Information Visualisation*, 4:32–48, 2005.

[117] Chris Snijders, Uwe Matzat, and Ulf-Dietrich Reips. Big data: Big gaps of knowledge in the field of internet science. *International Journal of Internet Science*, 7:1–5, 2012.

[118] M. R. Spiegel. *Theory and Problems of Probability and Statistics.* New York: McGraw-Hill, pages 109-111, 1992.

[119] Stigler M. Stephen. Mathematical statistics in early states. *The annals of the Statistics*, 6:239–265, 1978.

[120] Stigler M. Stephen. *Statistics on Table.* Harvard University Press, 1999.

[121] Karthik Suresh, Sanjeev V.Thomas, and Geetha Suresh. Design, data analysis and sampling technique for clinical research. *Ann Indian Academy Neurology*, 14:287–290, 2011.

[122] Jinsong Tan and Louxin Zhang. The consecutive ones submatrix problem for sparse matrices. *Algorithmica*, 48:287–299, 2007.

[123] Kohonen Teuvo. *Self-Organising.* Berlin, Springer-Verlag, 2001.

[124] Alan Tucker. A structure theorem for the consecutive 1s property. *Journal of Combinatorial Theory Series B*, 12:153–162, 1972.

[125] Marinus Veldhorst. Approximation of the consecutive ones matrix augmentation problem. *Journal on Computing SIAM.*, 14:709–729, 1985.

[126] Ulrike von Luxburg. A tutorial on spectral clustering. *AI Magazine in Statistics and Computing*, 17:395–416, 2007.

[127] Srikant Q. Vu and Rakesh Agrawal. Mining association rules with item constraints. In *Proceedings of KDD'97*, 1997.

[128] Niko Vuokko. Consecutive ones property and spectral ordering. In *Proceedings of the 10th SIAM International Conference on Data Mining (SDM10)*, pages 350–360, 2010.

[129] D. J Wheeter and D. S. Chambers. Understanding statistical process control. *SPC Press*, 1992.

Appendix A

Additional 2D-BPM Worked Examples

A.1 Introduction

In this appendix some additional worked examples to those given in Chapters 4 and 8, in the context of 2D (single and multiple dots per location), are presented. These were not included in the body of the thesis because of space limitations and are thus presented here. The appendix is organised as follows. Sub-appendix A.2, presents additional worked examples illustrating the operation of a 2D-BPM algorithm, whilst Sub-appendix A.3 presents additional worked examples illustrating the operation of a MD-BPM algorithm.

A.2 A Worked Example Using 2D-BPM Algorithm

This sub-appendix presents additional working examples illustrating the operation of the 2D-BPM algorithm. Two worked examples are considered, one using a 5×5 matrix, and another using a 5×4 matrix. The first 2D example is presented in Sub-appendix A.2.1 and the second in Sub-appendix A.2.2.

A.2.1 A Worked Example 1

Considering a 2D matrix measuring 5×5 is shown in Figure A.1. Thus $k_1 = 5$ and $k_2 = 5$ and:

$$D = \{\langle 1,1\rangle, \langle 1,3\rangle, \langle 1,5\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \\ \langle 3,5\rangle, \langle 4,3\rangle, \langle 4,4\rangle, \langle 4,5\rangle, \langle 5,1\rangle, \langle 5,2\rangle, \langle 5,4\rangle\}.$$

The 2D-BPM algorithm commences by considering the x-dimension first, the calculated banding scores are shown in Table A.3; the sequence of banding scores is $\{0.67, 0.75, 0.67, 1.00, 0.58\}$. We thus rearrange the indexes in Dim_x in ascending order of banding score. The result is as shown in Figure A.2. We now have:

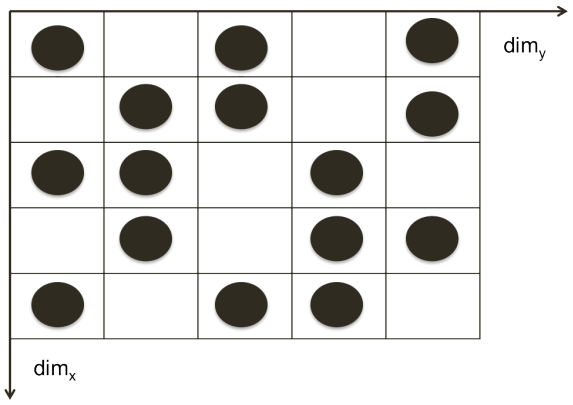


FIGURE A.1: Input matrix

TABLE A.1: Example 1 Calculation of banding scores for dimension x (iteration 1)

Index	Dist from origin	Max. dist. from origin	bs
1	$1+3+5$ $=8.0$	$3+4+5$ $=12.0$	0.67
2	$2+3+4$ $=9.0$	$3+4+5$ $=12.0$	0.75
3	$1+2+5$ $=8.0$	$3+4+5$ $=12.0$	0.67
4	$3+4+5$ $=12.0$	$3+4+5$ $=12.0$	1.00
5	$1+2+4$ $=7.0$	$3+4+5$ $=12.0$	0.58

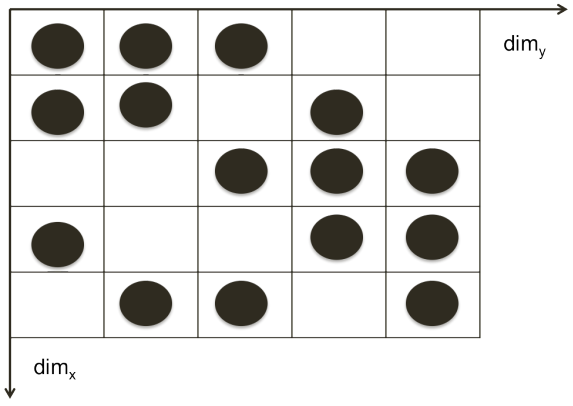


FIGURE A.2: Input matrix after rearrangement of Dim_x (iteration 1)

$$D' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,4\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 2,5\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,5\rangle, \langle 4,2\rangle, \langle 4,3\rangle, \langle 4,4\rangle, \langle 5,3\rangle, \langle 5,4\rangle, \langle 5,5\rangle\}.$$

Considering dimension y next, the banding scores are calculated as shown in Table A.2. This produced a set of banding scores $\{0.50, 0.58, 0.91, 0.83, 0.83\}$. Thus in this case the indexes in Dim_y are more or less already arranged in ascending order of banding score,

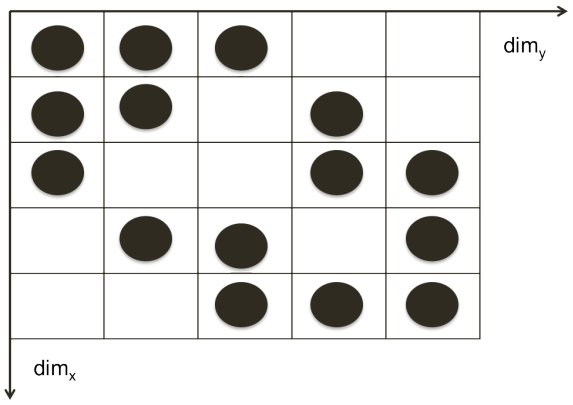


FIGURE A.3: Input matrix after rearrangement of Dim_y (iteration 1

we simply need to move the third row to the last place. The result is as shown in Figure A.3. We now have:

TABLE A.2: Example 1 Calculation of banding scores for dimension y (iteration 1)

Index	Dist from origin	Max. dist. from origin	bs
1	$1+2+3$ $= 6.0$	$3+4+5$ $= 12.0$	0.50
2	$1+2+4$ $= 7.0$	$3+4+5$ $= 12.0$	0.58
3	$3+4+5$ $= 11.0$	$3+4+5$ $= 12.0$	1.00
4	$1+4+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83
5	$2+3+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83

$$D'' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,1\rangle, \langle 2,4\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,4\rangle, \langle 3,5\rangle, \langle 4,2\rangle, \langle 4,3\rangle, \langle 4,5\rangle, \langle 5,3\rangle, \langle 5,4\rangle, \langle 5,5\rangle\}.$$

The GBS_x and GBS_y values are then calculated as follows (note that the individual GBS values for the columns have changed because of the reorganisation of the rows):

$$GBS_x = \frac{(0.67 \times 5) + (0.75 \times 4) + (0.67 \times 3) + (1.0 \times 2) + (0.58 \times 1)}{1+2+3+4+5} = \frac{10.94}{15} = 0.7293$$

$$GBS_y = \frac{(0.50 \times 5) + (0.58 \times 4) + (1.0 \times 3) + (0.83 \times 2) + (0.83 \times 1)}{1+2+3+4+5} = \frac{10.31}{15} = 0.6873$$

The overall global banding score (GBS) value is then calculated as:

$$GBS = \frac{0.7293 + 0.6873}{2} = 0.7083 \tag{A.1}$$

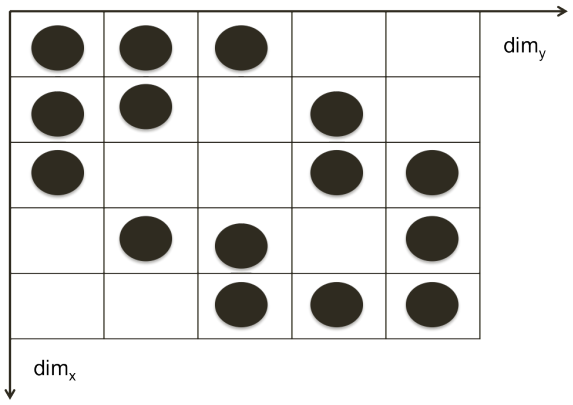


FIGURE A.4: Input matrix after rearrangement of Dim_x (iteration 2

The process is then repeated because we have reduced the GBS value and because the maximum number of iterations has not yet been reached. In this second iteration the banding scores, $\{0.50, 0.58, 0.83, 0.83, 1.00\}$, are produced for dimension x calculated as shown in Table A.3, and thus no changes to the index (element) ordering in dimension x is undertaken; the result remains as shown in Figure A.4. Similarly, the banding scores, $\{0.50, 0.58, 0.83, 0.83, 1.00\}$, are produced for dimension y calculated as shown in Table A.4, as a result no changes were made with respect to dimension y either and hence the process terminates. Note that the GBS values for the columns and rows are the same because the configuration is now symmetrical. The resulting configuration remains as shown in Figure A.5 and:

$$D'' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,1\rangle, \langle 2,4\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,5\rangle,$$
$$\langle 4,1\rangle, \langle 4,2\rangle, \langle 4,4\rangle, \langle 4,5\rangle, \langle 5,3\rangle, \langle 5,4\rangle, \langle 5,5\rangle\}.$$

TABLE A.3: Example 1 Calculation of banding scores for dimension x (iteration 2)

Index	Dist from origin	Max. dist. from origin	*bs*
1	$1+2+3$ $= 6.0$	$3+4+5$ $= 12.0$	0.50
2	$1+2+4$ $= 7.0$	$3+4+5$ $= 12.0$	0.58
3	$1+4+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83
4	$2+3+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83
5	$3+4+5$ $= 12.0$	$3+4+5$ $= 12.0$	1.00

$$GBS_x = \frac{(0.50 \times 5) + (0.58 \times 4) + (0.83 \times 3) + (0.83 \times 2) + (1.0 \times 1)}{1+2+3+4+5} = \frac{9.97}{15} = 0.6647$$

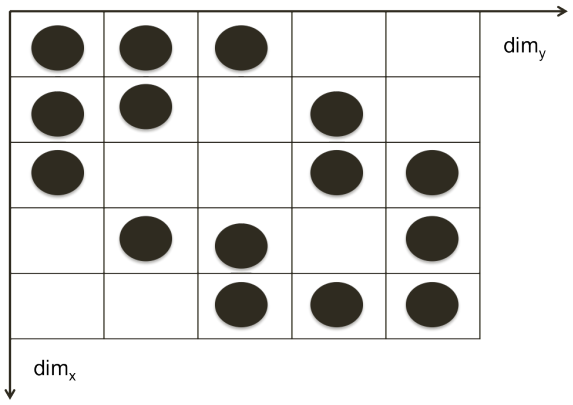


FIGURE A.5: Input matrix after rearrangement of Dim_y (iteration 2

TABLE A.4: Example 1 Calculation of banding scores for dimension y (iteration 2)

Index	Dist from origin	Max. dist. from origin	*bs*
1	$1+2+3$ $= 6.0$	$3+4+5$ $= 12.0$	0.50
2	$1+2+4$ $= 7.0$	$3+4+5$ $= 12.0$	0.58
3	$1+4+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83
4	$2+3+5$ $= 10.0$	$3+4+5$ $= 12.0$	0.83
5	$3+4+5$ $= 12.0$	$3+4+5$ $= 12.0$	1.00

$$GBS_y = \frac{(0.50 \times 5) + (0.58 \times 4) + (0.83 \times 3) + (0.83 \times 2) + (1.0 \times 1)}{1+2+3+4+5} = \frac{9.97}{15} = 0.6647$$

The overall GBS value is then calculated as:

$$GBS = \frac{0.6647 + 0.6647}{2} = 0.6647 \tag{A.2}$$

A.2.2 A Worked Example 2

For the second worked example a 2D matrix measuring 5×4 was used as shown in Figure A.16. Thus $k_1 = 5$ and $k_2 = 4$ and:

$$D = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,4\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 2,5\rangle, \langle 3,1\rangle, \langle 3,5\rangle, \langle 4,2\rangle, \langle 4,3\rangle, \}.$$

Considering the x-dimension first, the banding scores (calculated as shown in Table A.5) are $\{0.58, 1.00, 0.67, 0.56\}$. We thus rearrange the indexes in Dim_x in ascending order of banding score. The result is as shown in Figure A.7 and:

$$D' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,4\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,5\rangle, \langle 4,3\rangle, \langle 4,4\rangle, \langle 4,5\rangle, \}.$$

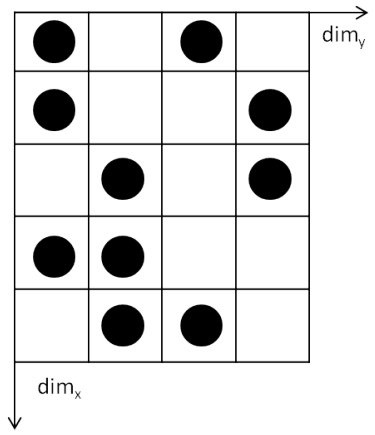


FIGURE A.6: Example matrix

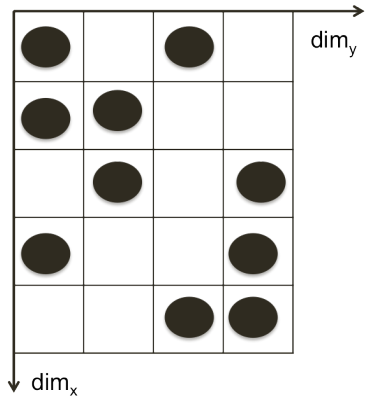


FIGURE A.7: Example matrix after rearrangement of Dim_x (iteration 1)

TABLE A.5: Example 2 Calculation of banding scores for dimension x (iteration 1)

Index	Dist from origin	Max. dist. from origin	*bs*
1	$1+2+4$ $=7.0$	$3+4+5$ $=12.0$	0.58
2	$3+4+5$ $=12.0$	$3+4+5$ $=12.0$	1.00
3	$1+5=6.0$	$4+5=9.0$	0.67
4	$2+3=5.0$	$4+5=9.0$	0.56

Considering the y-dimension next the banding score (calculated as shown in Table A.6) are $\{0.57, 0.43, 0.86, 0.71, 1.00\}$. We thus rearrange the indexes in Dim_y in ascending order of banding score to give the result shown in Figure A.8.

The banding scores for the x and y dimensions are then calculated as follows:

$$GBS_x = \frac{(0.58 \times 4) + (1.0 \times 3) + (0.67 \times 2) + (0.56 \times 1)}{1+2+3+4} = \frac{7.220}{10} = 0.7220$$

TABLE A.6: Example 2 Calculation of banding scores for dimension y (iteration 1)

Index	Dist from origin	Max. dist. from origin	bs
1	$1+3=4.0$	$3+4=7.0$	0.57
2	$1+2=3.0$	$3+4=7.0$	0.43
3	$2+4=6.0$	$3+4=7.0$	0.86
4	$1+4=5.0$	$3+4=7.0$	0.71
5	$3+4=7.0$	$3+4=7.0$	1.00

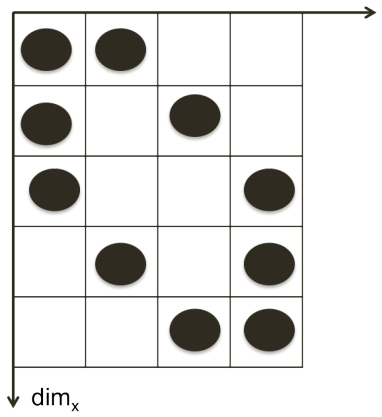


FIGURE A.8: Example matrix after rearrangement of Dim_y (iteration 1)

$$GBS_y = \frac{(0.57 \times 5) + (0.43 \times 4) + (0.86 \times 3) + (0.71 \times 2) + (1.0 \times 1)}{1+2+3+4+5} = \frac{9.48}{15} = 0.6320$$

The overall GBS value is then calculated as:

$$GBS = \frac{0.7220 + 0.6320}{2} = 0.6770 \tag{A.3}$$

The overall GBS value has been reduced, and we have not reached the maximum number of iterations, thus the process is repeated. New banding scores of $\{0.50, 0.56, 0.67, 1.00\}$ are produced for dimension x calculated as shown in Table A.7, and we thus rearrange the indexes (elements) in x accordingly; the result is as shown in Figure A.9. Similarly, new banding scores of $\{0.43, 0.58, 0.71, 0.86, 1.00\}$ are produced for dimension y calculated as shown in Table A.8, as a result no changes were made. The result is as shown in Figure A.10 and:

$$D'' = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,1\rangle, \langle 2,4\rangle, \langle 3,2\rangle, \langle 3,5\rangle, \langle 4,3\rangle, \langle 4,4\rangle, \langle 4,5\rangle, \}.$$

$$GBS_x = \frac{(0.50 \times 4) + (0.56 \times 3) + (0.67 \times 2) + (1.0 \times 1)}{1+2+3+4} = \frac{6.02}{10} = 0.6020$$

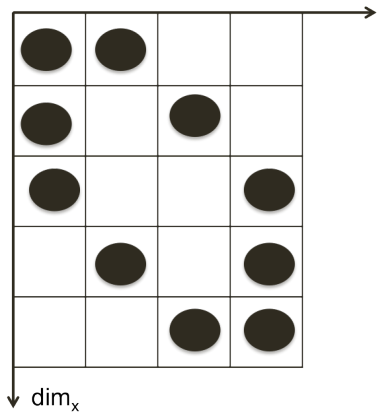


FIGURE A.9: Example matrix after rearrangement of Dim_x (iteration 2)

TABLE A.7: Example 2 Calculation of banding scores for dimension x (iteration 2)

Index	Dist from origin	Max. dist. from origin	bs
1	$1+2+3 = 6.0$	$3+4+5 = 12.0$	0.50
2	$1+4 = 5.0$	$4+5 = 9.0$	0.56
3	$2+5 = 6.0$	$4+5 = 9.0$	0.67
4	$3+4+5 = 12.0$	$3+4+5 = 12.0$	1.00

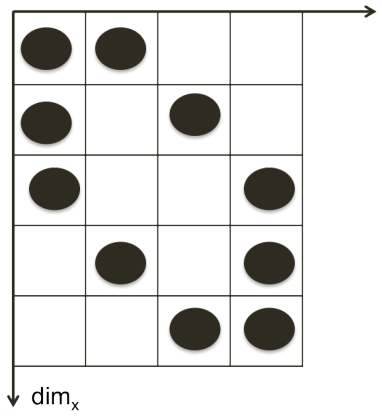


FIGURE A.10: Example matrix after rearrangement of Dim_y (iteration 2)

$$GBS_y = \frac{(0.43 \times 5) + (0.57 \times 4) + (0.71 \times 3) + (0.86 \times 2) + (1.0 \times 1)}{1+2+3+4+5} = \frac{9.28}{15} = 0.6187$$

The overall Global Banding Score (GBS) value is then calculated as:

$$GBS = \frac{0.6020 + 0.6187}{2} = 0.6104 \tag{A.4}$$

TABLE A.8: Example 2 Calculation of banding scores for dimension y (iteration 2)

Index	Dist from origin	Max. dist. from origin	bs
1	$1+2=3.0$	$3+4=7.0$	0.43
2	$1+3=4.0$	$3+4=7.0$	0.57
3	$1+4=5.0$	$3+4=7.0$	0.71
4	$2+4=6.0$	$3+4=7.0$	0.86
5	$3+4=7.0$	$3+4=7.0$	1.00

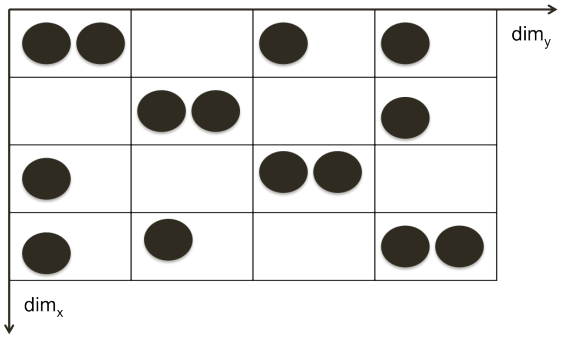


FIGURE A.11: Dot matrix

A.3 A Worked Example Using MD-BPM Algorithm

This sub-appendix presents two working examples illustrating the operation of the MD-EBPM algorithm. Both examples uses a 2D configuration measuring 4×4 with multiple dots in some cells. The first example is presented in Sub-appendix A.3.1, while the second example is presented in Sub-appendix A.3.2.

A.3.1 A Multiple Dots Example 1

Given the 2D 4×4 configuration given in Figure A.11. The configuration features: $DIM = \{x, y\}$, $Dim_x = \{0, 1, 2, 3\}$ and $Dim_y = \{0, 1, 2, 3\}$ with multiple dots in some cells. Note that multiple dots are arranged along the leading diagonal and that the data configuration is symmetric about the leading diagonal. The input D to the MD-EBPM algorithm is thus:

$$D = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,3\rangle, \langle 2,0\rangle,$$
$$\langle 2,2\rangle, \langle 2,2\rangle, \langle 3,0\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

The MD-EBPM algorithm starts by considering dimension x first, the banding scores are calculated, taking into account the number of dots per location, as shown in Table A.9. This produces the banding scores $\{0.56, 0.63, 0.50, 0.78\}$. Thus, we rearrange the indexes (elements) in Dim_x in ascending order of their banding score to produce the result shown in Figure A.12 and:

$$D' = \{\langle 0,0\rangle, \langle 0,2\rangle, \langle 0,2\rangle, \langle 1,0\rangle, \langle 1,0\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,1\rangle,$$
$$\langle 2,1\rangle, \langle 2,3\rangle, \langle 3,0\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

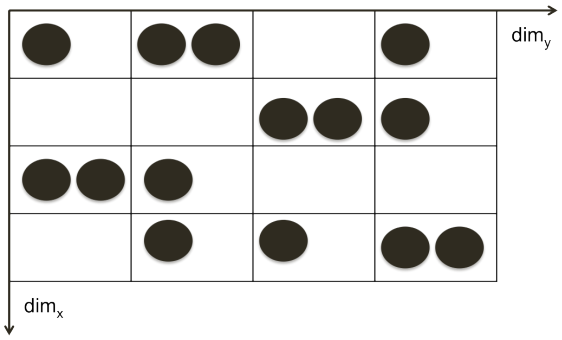


FIGURE A.12: Dot matrix after rearrangement of Dim_x (iteration 1)

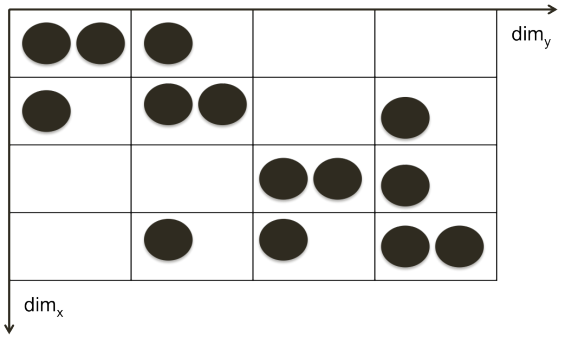


FIGURE A.13: Dot matrix after rearrangement of Dim_y (iteration 1)

TABLE A.9: Example 3 Calculation of banding scores for dimension x (iteration 1)

Index	Dist from origin	Max. dist. from origin	bs
0	$(0*2)+(2*1)$ $+(3*1)=5.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.56
1	$(1*2)+(3*1)$ $=5.0$	$(2*1)$ $+(3*2)=8.0$	0.63
2	$(0*1)+(2*2)$ $=4.0$	$(2*1)+(3*2)$ $=8.0$	0.50
3	$(0*1)+(1*1)$ $+(3*2)=7.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.78

Considering dimension y next, the banding scores in this case are calculated as shown in Table A.10 (taking into account the number of dots per location). This produces the banding scores $\{0.56, 0.88, 0.13, 1.00\}$. Thus, we rearrange the elements in Dim_y in ascending order of their banding score to produce the configuration shown in Figure A.13 and:

$$D'' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,3\rangle, \langle 2,2\rangle, \\ \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

The GBS for this configuration is then calculated using Equation 8.7 given in Chapter 8 (the sum of the individual banding scores divided by the total number of indexes in the configuration):

TABLE A.10: Example 3 Calculation of banding scores for dimension y (iteration 1)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*1)+(1*2)$ $(3*1)=5.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.56
1	$(2*2)+(3*1)$ $=7.0$	$(2*1)+(3*2)$ $=8.0$	0.88
2	$(0*2)+(1*1)$ $=1.0$	$(2*1)+(3*2)$ $=8.0$	0.13
3	$(1*1)+(2*1)$ $+(3*2)=9.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	1.00

$$GBS = \frac{5.0}{8.0} + \frac{5.0}{9.0} + \frac{4.0}{8.0} + \frac{7.0}{9.0} + \frac{1.0}{8.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{9.0}{9.0} = 0.6324$$

The process is then repeated but the same banding scores; $\{0.13, 0.56, 0.88, 1.00\}$ as before are produced for dimension x (calculated as shown in Table A.11); thus no changes to the elements in x dimension results. The result is as shown in Figure A.14. Similarly, the same banding scores $\{0.13, 0.56, 0.88, 1.00\}$ are also produced for dimension y (calculated as shown in Table A.12), as a result no changes to the ordering of the elements in the y index are undertaken either. The result is as shown in Figure A.15. As before:

$$D'' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 1,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,3\rangle, \langle 2,2\rangle,$$
$$\langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

TABLE A.11: Example 3 Calculation of banding scores for dimension x (iteration 2)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*2)+(1*1)$ $=1.0$	$(2*1)+(3*2)$ $=8.0$	0.13
1	$(0*1)+(2*2)$ $(3*1)=5.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.56
2	$(2*2)+(3*1)$ $=7.0$	$(2*1)+(3*2)$ $=8.0$	0.88
3	$(1*1)+(2*1)$ $+(3*2)=9.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	1.00

And

$$D'' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,3\rangle, \langle 2,2\rangle,$$
$$\langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

The GBS for this configuration is then calculated as follows (using Equation 8.7 from Chapter 8):

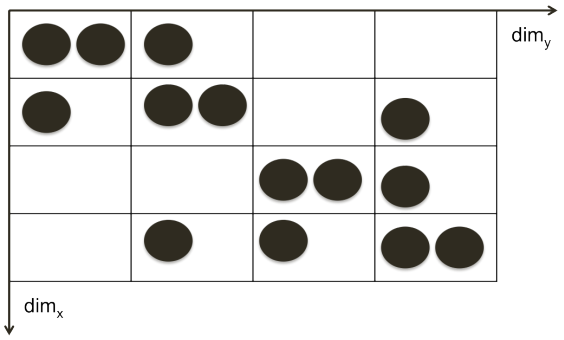


FIGURE A.14: Dot matrix after rearrangement of Dim_x (iteration 2)

TABLE A.12: Example 3 Calculation of banding scores for dimension y (iteration 2)

Index	Dist from origin	Max. dist. from origin	bs
0	$(0*2)+(1*1)$ $=1.0$	$(2*1)+(3*2)$ $=8.0$	0.13
1	$(0*1)+(1*2)$ $+(3*1)=5.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.56
2	$(2*2)+(3*1)$ $=7.0$	$(2*1)+(3*2)$ $=8.0$	0.88
3	$(1*1)+(2*1)$ $+(3*2)=9.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	1.00

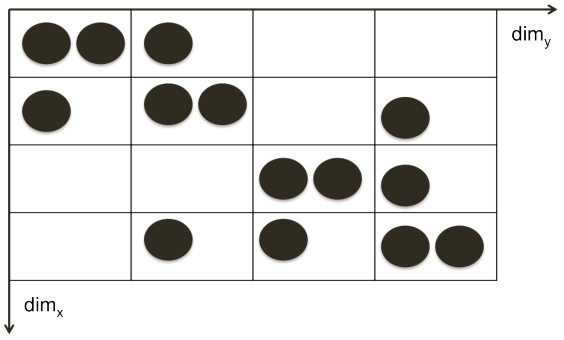


FIGURE A.15: Dot matrix after rearrangement of Dim_y (iteration 2)

$$GBS = \frac{1.0}{8.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{9.0}{9.0} + \frac{1.0}{8.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{9.0}{9.0} = 0.6471$$

Because there have been no changes after iteration 2, the algorithm exits with D'.

A.3.2 A Multiple Dots Example 2

For the second example the 2D multiple dot configuration, measuring 4×4, presented in Figure A.16 was used. Again the configuration is symmetrical about the leading diagonal. The configuration features $DIM = \{x, y\}$, $Dim_x = \{0, 1, 2, 3\}$ and $Dim_y = \{0, 1, 2, 3\}$ with multiple dots in some cells. The input D to the MD-BPM algorithm is thus:

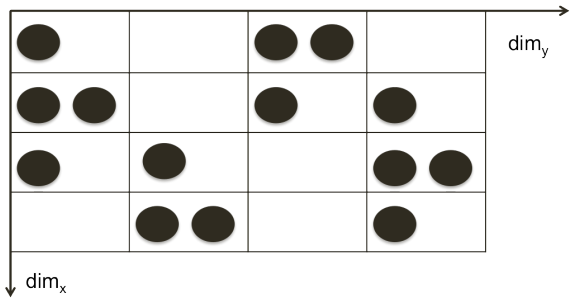


FIGURE A.16: Input dot matrix

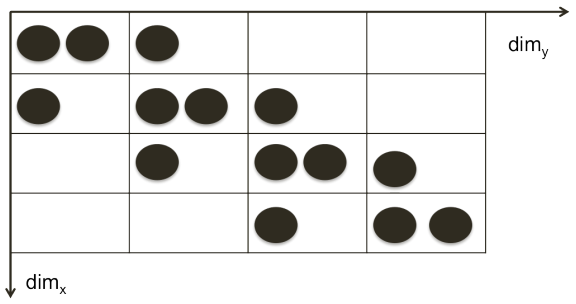


FIGURE A.17: Input dot matrix after rearrangement of Dim_x (iteration 1)

$$D = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \\ \langle 2,0\rangle, \langle 2,1\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,2\rangle, \langle 3,3\rangle\}.$$

The MD-BPM algorithm starts by considering dimension x first. The banding score is calculated as shown in Table A.13. This produces banding scores of $\{0.44, 1.00, 0.13, 0.89\}$. Thus, we rearrange the indexes (elements) in Dim_x in ascending order of their banding score to produce the result shown in Figure A.17 and:

$$D' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \\ \langle 2,2\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

TABLE A.13: Example 4 Calculation of banding scores for dimension x (iteration 1)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*1)+(1*2)$ $+(2*1)=4.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.44
1	$(2*1)+(3*2)$ $=8.0$	$(2*1)+(3*2)$ $=8.0$	1.00
2	$(0*2)+(1*1)$ $=1.0$	$(2*1)+(3*2)$ $=8.0$	0.13
3	$(1*1)+(2*2)$ $+(3*1)=8.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.89

Considering dimension y next, the banding score is calculated, as shown in Table A.14, to produce $\{0.13, 0.44, 0.88, 1.00\}$. Thus, we rearrange the elements in Dim_y in ascending order of their banding score to produce the result shown in Figure A.18 and:

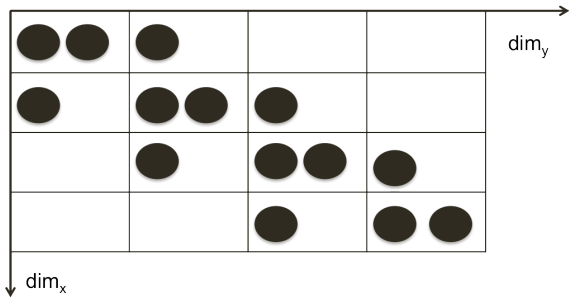


FIGURE A.18: Input dot matrix after rearrangement of Dim_y (iteration 1)

$$D'' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \\ \langle 2,2\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

TABLE A.14: Example 4 Calculation of banding scores for dimension y (iteration 1)

Index	Dist from origin	Max. dist. from origin	*bs*
0	$(0*2)+(1*1)$ $=1.0$	$(2*1)+(3*2)$ $=8.0$	0.13
1	$(0*1)+(1*2)$ $+(2*1)=4.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.44
2	$(1*1)+(2*2)$ $+(3*1)=8.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.88
3	$(2*1)+(3*2)$ $=8.0$	$(2*1)+(3*2)$ $=8.0$	1.00

The GBS for this configuration is then:

$$GBS = \frac{1.0}{8.0} + \frac{4.0}{9.0} + \frac{8.0}{9.0} + \frac{8.0}{8.0} + \frac{1.0}{8.0} + \frac{4.0}{9.0} + \frac{8.0}{9.0} + \frac{8.0}{8.0} = 0.6176$$

The process is repeated on the next iteration. However in this case the same overall GBS value is produced (indicating that a best banding has already been arrived at). The rearranged dot matrix is as follows (Figure A.18) and:

$$D'' = \{\langle 0,0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \\ \langle 2,2\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

Appendix B

Additional Sampling Experimental Result

B.1 Introduction

In this appendix some additional experimental results to those given in Chapter 9 in the context of run time (in seconds) and GBS values are presented. More specifically results obtained using sample sizes of 12,000 and 36,000 records are presented. Recall that in Chapter 9 only results using a sample size of 24,000 records (2,000 per month) were considered. This was because of space restrictions within the main body of the thesis. For completeness these additional results are thus presented here. The objective was to determine what effect sample size had on the process. Note that M-Tables were used throughout.

The appendix is organised in the same manner as in the case of Chapter 9 by dividing it into four sub-appendixes; run time is considered in Sub-appendix B.2 to B.4 with respect to sample sizes of 12,000 and 36,000 respectively, whilst GBS values are considered in Sub-appendix B.3 to B.5. The results corroborated the results presented earlier, namely:

1. The most efficient MD-BPM algorithm, in terms of runtime, in the context of both sample sizes of 12,000 and 36,000 was the MD-ABPM.

2. The most effective MD-BPM algorithm, in terms of GBS value, in the context of both sample sizes of 12,000 and 36,000 was the Euclidean MD-EBPM.

3. The MD-ABPM algorithm in the context of both sample sizes of 12,000 and 36,000 produce the worst GBS values.

4. Both sample sizes of 12,000 and 36,000 were also found to be effective with respect to idenfying a banding in large data sets.

B.2 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Technique in Terms of Run time (RT) for 12,000 Records (1,000 per month)

Tables B.1 and B.2 presents the runtime comparative evaluation for the MD-EBPM and MD-ABPM algorithms in terms of 3D, 4D and 5D data, and the sampling technique, using a sample size of 12,000 records (1,000 records per month).

Table B.1: Sampling runtime results (seconds) for 3D and 4D CTS data sets using MD-EBPM and MD-ABPM algorithms with M-Tables

	Year	runtime (sec)			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	02.38	02.02	**01.27**	01.28
Final band. using Sampling		15.09	14.55	**12.83**	16.15
Banding of Sample	2004	01.53	01.39	**01.09**	01.55
Final band. using Sampling		16.66	13.48	**10.36**	12.48
Banding of Sample	2005	02.37	02.30	**01.23**	01.48
Final band. using Sampling		17.18	15.15	**14.74**	11.05
Banding of Sample	2006	01.53	01.37	**01.12**	01.30
Final band. using Sampling		16.46	15.78	**10.88**	14.52
Cornwall					
Banding of Sample	2003	01.50	01.30	**01.18**	01.35
Final band. using Sampling		17.04	16.40	**15.30**	17.93
Banding of Sample	2004	02.75	02.23	**01.54**	01.94
Final band. using Sampling		25.13	17.08	**15.05**	14.19
Banding of Sample	2005	02.23	02.15	**01.19**	01.41
Final band. using Sampling		24.00	18.17	**14.26**	12.03
Banding of Sample	2006	04.21	03.82	**01.35**	02.17
Final band. using Sampling		25.70	17.13	**15.69**	13.36
Lancashire					
Banding of Sample	2003	01.83	01.72	**01.07**	01.13
Final band. using Sampling		24.74	17.04	**15.93**	13.85
Banding of Sample	2004	02.76	02.26	**01.05**	01.13
Final band. using Sampling		22.25	19.07	**14.04**	12.83
Banding of Sample	2005	02.36	01.28	**01.17**	01.06
Final band. using Sampling		24.23	19.17	**15.97**	12.82
Banding of Sample	2006	02.35	02.20	**01.27**	01.42
Final band. using Sampling		27.93	18.64	**12.44**	21.45
Norfolk					
Banding of Sample	2003	02.85	01.97	**01.24**	01.39
Final band. using Sampling		15.65	11.58	**09.75**	17.27
Banding of Sample	2004	05.12	04.92	**02.40**	01.75
Final band. using Sampling		21.79	19.89	**12.77**	11.70
Banding of Sample	2005	02.08	02.00	**01.60**	01.04
Final band. using Sampling		19.31	13.45	**11.76**	12.06
Banding of Sample	2006	01.60	01.47	**01.19**	01.44
Final band. using Sampling		20.64	13.03	**11.23**	14.75

Table B.2: Sampling runtime results (seconds) for 5D CTS data sets using MD-EBPM and MD-ABPM algorithms with M-Tables

	Year	**runtime (sec)**		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	19.50	15.66	**10.97**
Final band. using Sampling		43.47	39.02	**21.09**
Banding of Sample	2004	19.87	17.07	**16.40**
Final band. using Sampling		36.33	31.25	**20.94**
Banding of Sample	2005	15.66	14.56	**09.19**
Final band. using Sampling		47.88	45.67	**42.56**
Banding of Sample	2006	20.30	18.67	**16.21**
Final band. using Sampling		32.75	30.32	**25.10**
Cornwall				
Banding of Sample	2003	24.09	17.59	**09.61**
Final band. using Sampling		50.87	51.66	**48.27**
Banding of Sample	2004	25.62	17.15	**14.61**
Final band. using Sampling		77.18	61.28	**41.28**
Banding of Sample	2005	20.43	13.52	**11.92**
Final band. using Sampling		43.53	41.53	**40.11**
Banding of Sample	2006	20.89	17.89	**15.41**
Final band. using Sampling		94.80	84.52	**70.25**
Lancashire				
Banding of Sample	2003	15.19	12.19	**10.02**
Final band. using Sampling		96.30	92.22	**82.02**
Banding of Sample	2004	19.29	15.45	**10.09**
Final band. using Sampling		85.90	83.20	**63.02**
Banding of Sample	2005	15.25	12.55	**10.84**
Final band. using Sampling		49.82	47.13	**34.24**
Banding of Sample	2006	16.31	13.30	**11.49**
Final band. using Sampling		52.79	42.76	**39.62**
Norfolk				
Banding of Sample	2003	15.84	12.84	**10.72**
Final band. using Sampling		20.90	20.70	**18.30**
Banding of Sample	2004	19.29	15.60	**10.99**
Final band. using Sampling		35.90	34.36	**24.16**
Banding of Sample	2005	15.25	13.55	**10.84**
Final band. using Sampling		18.69	16.79	**14.79**
Banding of Sample	2006	16.01	13.31	**10.40**
Final band. using Sampling		21.77	21.70	**20.41**

B.3 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Techniques in Terms of Global Banding Score (GBS) for 12,000 records (1,000 per month)

Tables B.3 and B.4 presents the GBS comparative evaluation of the MD-EBPM and MD-ABPM Algorithms in terms of 3D, 4D and 5D data, and the sampling technique using a sample size of 12,000 records (1,000 records per month).

Table B.3: Sampling GBS result for 2003 to 2006 3D and 4D CTS data set

	Year	GBS			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	**0.2674**	**0.2688**	**0.3732**	**0.3494**
Final band. using Sampling		0.2853	0.2873	0.4039	0.3632
Banding of Sample	2004	**0.2192**	**0.2200**	**0.3180**	**0.3425**
Final band. using Sampling		0.2270	0.2279	0.3283	0.3464
Banding of Sample	2005	**0.2876**	**0.2901**	**0.3987**	**0.3684**
Final band. using Sampling		0.3028	0.3057	0.4198	0.3811
Banding of Sample	2006	**0.2256**	**0.2272**	**0.3256**	**0.3516**
Final band. using Sampling		0.2337	0.2353	0.3401	0.3591
Cornwall					
Banding of Sample	2003	**0.2748**	**0.2751**	**0.4000**	**0.3882**
Final band. using Sampling		0.2843	0.2851	0.4150	0.3891
Banding of Sample	2004	**0.2496**	**0.2518**	**0.3565**	**0.3349**
Final band. using Sampling		0.2710	0.2734	0.3901	0.3589
Banding of Sample	2005	**0.2680**	**0.2698**	**0.3872**	**0.3615**
Final band. using Sampling		0.2775	0.2796	0.3964	0.3691
Banding of Sample	2006	**0.2954**	**0.2966**	**0.4209**	**0.3718**
Final band. using Sampling		0.3003	0.3011	0.4455	0.3766
Lancashire					
Banding of Sample	2003	**0.2796**	**0.2813**	**0.4005**	**0.3918**
Final band. using Sampling		0.2928	0.2950	0.4171	0.3943
Banding of Sample	2004	**0.2695**	**0.2726**	**0.3959**	**0.3544**
Final band. using Sampling		0.2812	0.2849	0.4088	0.3598
Banding of Sample	2005	**0.2809**	**0.2818**	**0.4045**	**0.3861**
Final band. using Sampling		0.2817	0.2978	0.4096	0.3910
Banding of Sample	2006	**0.2739**	**0.2818**	**0.3912**	**0.3814**
Final band. using Sampling		0.2829	0.2856	0.4098	0.3960
Norfolk					
Banding of Sample	2003	**0.3058**	**0.3179**	**0.4460**	**0.4124**
Final band. using Sampling		0.3081	0.3208	0.4502	0.4142
Banding of Sample	2004	**0.2583**	**0.2598**	**0.3667**	**0.3285**
Final band. using Sampling		0.2698	0.2713	0.3806	0.3447
Banding of Sample	2005	**0.3124**	**0.3142**	**0.4253**	**0.3977**
Final band. using Sampling		0.3134	0.3171	0.4268	0.4006
Banding of Sample	2006	**0.2330**	**0.2353**	**0.3395**	**0.3625**
Final band. using Sampling		0.2461	0.2486	0.3518	0.3724

Table B.4: Sampling GBS result for 2003 to 2006 5D CTS data set

	Year	GBS		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	**0.2171**	**0.2192**	**0.3882**
Final band. using Sampling		0.2327	0.2332	0.4161
Banding of Sample	2004	**0.1871**	**0.1883**	**0.3485**
Final band. using Sampling		0.1932	0.1945	0.3611
Banding of Sample	2005	**0.2334**	**0.2346**	**0.4226**
Final band. using Sampling		0.2456	0.2471	0.4382
Banding of Sample	2006	**0.1899**	**0.1912**	**0.3527**
Final band. using Sampling		0.1964	0.1977	0.3662
Cornwall				
Banding of Sample	2003	**0.2276**	**0.2319**	**0.4208**
Final band. using Sampling		0.2357	0.2399	0.4298
Banding of Sample	2004	**0.2178**	**0.2260**	**0.4050**
Final band. using Sampling		0.2263	0.2275	0.4156
Banding of Sample	2005	**0.2214**	**0.2245**	**0.4175**
Final band. using Sampling		0.2283	0.2319	0.4289
Banding of Sample	2006	**0.2425**	**0.2428**	**0.4407**
Final band. using Sampling		0.2465	0.2471	0.4589
Lancashire				
Banding of Sample	2003	**0.2339**	**0.2350**	**0.4297**
Final band. using Sampling		0.2443	0.2459	0.4522
Banding of Sample	2004	**0.2235**	**0.2335**	**0.4179**
Final band. using Sampling		0.2323	0.2429	0.4344
Banding of Sample	2005	**0.2360**	**0.2362**	**0.4329**
Final band. using Sampling		0.2372	0.2546	0.4400
Banding of Sample	2006	**0.2325**	**0.2388**	**0.4220**
Final band. using Sampling		0.2406	0.2413	0.4380
Norfolk				
Banding of Sample	2003	**0.2613**	**0.2632**	**0.4663**
Final band. using Sampling		0.2638	0.2653	0.4673
Banding of Sample	2004	**0.2177**	**0.2181**	**0.3899**
Final band. using Sampling		0.2259	0.2264	0.4175
Banding of Sample	2005	**0.2487**	**0.2513**	**0.4540**
Final band. using Sampling		0.2524	0.2552	0.4551
Banding of Sample	2006	**0.1977**	**0.1980**	**0.3710**
Final band. using Sampling		0.2074	0.2075	0.3863

B.4 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Technique in Terms of Run time (RT) for 36,000 Records (3,000 per month)

Tables B.5 and B.6 presents the runtime comparative evaluation of the MD-EBPM and MD-ABPM algorithms in terms of 3D, 4D and 5D data, sampling technique using a sample size of 36,000 records (3,000 per month).

TABLE B.5: Sampling runtime results (seconds) for 3D and 4D CTS data sets using the MD-EBPM and MD-ABPM algorithms with M-Tables

	Year	**runtime (sec)**			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	08.05	05.02	**02.74**	02.08
Final band. using Sampling		15.09	14.55	**12.83**	16.15
Banding of Sample	2004	11.39	09.28	**05.22**	04.12
Final band. using Sampling		16.66	13.48	**10.36**	12.48
Banding of Sample	2005	18.17	16.20	**08.06**	05.07
Final band. using Sampling		17.18	15.15	**14.74**	11.05
Banding of Sample	2006	15.09	12.09	**09.02**	06.14
Final band. using Sampling		16.46	15.78	**10.88**	14.52
Cornwall					
Banding of Sample	2003	16.02	14.10	**09.18**	07.87
Final band. using Sampling		17.04	17.04	**16.30**	17.93
Banding of Sample	2004	17.99	15.16	**12.17**	07.02
Final band. using Sampling		25.13	17.08	**15.05**	14.19
Banding of Sample	2005	19.11	17.27	**13.19**	05.11
Final band. using Sampling		24.00	14.17	**14.26**	12.03
Banding of Segments		18.54	16.80	**13.95**	12.75
Final band. using Sampling		25.70	17.13	**15.69**	13.36
Lancashire					
Banding of Sample	2003	16.38	15.27	**11.70**	04.31
Final band. using Sampling		24.75	17.04	**18.93**	13.85
Banding of Sample	2004	16.76	14.46	**08.15**	02.13
Final band. using Sampling		22.25	19.07	**14.04**	12.83
Banding of Sample	2005	14.30	12.34	**11.23**	07.54
Final band. using Sampling		24.23	14.17	**15.97**	12.82
Banding of Sample	2006	12.65	10.08	**09.23**	04.24
Final band. using Sampling		27.93	18.64	**12.44**	21.45
Norfolk					
Banding of Sample	2003	12.43	10.23	**08.40**	03.63
Final band. using Sampling		15.65	11.58	**09.75**	17.27
Banding of Sample	2004	13.21	11.23	**09.23**	04.54
Final band. using Sampling		21.79	19.89	**12.77**	11.70
Banding of Sample	2005	12.23	10.04	**09.46**	05.43
Final band. using Sampling		19.31	13.45	**11.76**	12.06
Banding of Sample	2006	14.07	12.25	**09.09**	03.42
Final band. using Sampling		20.64	13.03	**11.23**	14.75

Table B.6: Sampling runtime results (seconds) for 5D CTS data sets using the MD-EBPM and MD-ABPM algorithms with M-Tables

	Year	runtime (sec)		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	34.69	28.22	**20.14**
Final band. using Sampling		47.85	41.89	**31.09**
Banding of Sample	2004	30.39	27.78	**22.41**
Final band. using Sampling		45.80	43.94	**40.94**
Banding of Sample	2005	28.60	24.56	**19.19**
Final band. using Sampling		47.88	45.67	**42.56**
Banding of Sample	2006	24.22	20.85	**15.14**
Final band. using Sampling		56.01	50.11	**45.10**
Cornwall				
Banding of Sample	2003	27.90	22.59	**17.09**
Final band. using Sampling		50.87	51.66	**48.27**
Banding of Sample	2004	30.25	17.15	**12.11**
Final band. using Sampling		77.18	61.28	**41.28**
Banding of Sample	2005	30.31	26.25	**14.20**
Final band. using Sampling		43.53	41.53	**41.11**
Banding of Sample	2006	30.81	25.12	**20.14**
Final band. using Sampling		94.80	84.52	**70.25**
Lancashire				
Banding of Sample	2003	30.19	25.91	**18.02**
Final band. using Sampling		96.30	92.22	**82.02**
Banding of Sample	2004	39.29	24.45	**19.09**
Final band. using Sampling		85.90	83.20	**63.02**
Banding of Sample	2005	25.52	22.50	**19.48**
Final band. using Sampling		49.82	47.13	**34.24**
Banding of Sample	2006	28.31	23.03	**17.42**
Final band. using Sampling		52.79	42.76	**39.62**
Norfolk				
Banding of Sample	2003	25.48	20.14	**12.27**
Final band. using Sampling		20.90	20.70	**18.30**
Banding of Sample	2004	39.19	29.05	**27.34**
Final band. using Sampling		35.90	34.36	**24.16**
Banding of Sample	2005	29.02	22.25	**19.14**
Final band. using Sampling		18.69	16.79	**14.79**
Banding of Sample	2006	29.11	25.11	**17.01**
Final band. using Sampling		21.77	21.70	**20.41**

B.5 Comparison of MD-EBPM and MD-ABPM Algorithms Using Sampling Techniques in Terms of Global Banding Score (GBS) for 36,000 Records (3,000 per month)

Tables B.7 and B.8 presents the GBS comparative evaluation of the MD-EBPM and MD-ABPM algorithms in terms of 3D, 4D and 5D data, sampling technique using a sample size of 36,000 records (3,000 per month).

TABLE B.7: Sampling GBS result for 2003 to 2006 3D and 4D CTS data set

	Year	GBS			
		Euclid.	**Manhat.**	**Approx.**	3D
Aberdeenshire					
Banding of Sample	2003	**0.3034**	0.3246	**0.3732**	0.3934
Final band. using Sampling		0.3117	**0.3142**	0.4039	**0.3887**
Banding of Sample	2004	**0.2525**	**0.2545**	0.4172	**0.3766**
Final band. using Sampling		0.2539	0.2550	**0.3007**	0.3839
Banding of Sample	2005	**0.3137**	**0.3164**	**0.4184**	**0.3924**
Final band. using Sampling		0.3149	0.3175	0.4260	0.3944
Banding of Sample	2006	**0.2256**	**0.2272**	**0.3256**	**0.3516**
Final band. using Sampling		0.2502	0.2523	0.3078	0.3827
Cornwall					
Banding of Sample	2003	**0.2986**	**0.3002**	**0.3308**	**0.4144**
Final band. using Sampling		0.2994	0.3013	0.3368	0.4188
Banding of Sample	2004	**0.2878**	0.3013	0.4085	0.3944
Final band. using Sampling		0.2916	**0.2939**	**0.4083**	**0.3892**
no banding		0.3213	0.3281	0.4570	0.4043
Banding of Sample	2005	**0.2789**	**0.2816**	**0.4078**	**0.3809**
Final band. using Sampling		0.2871	0.2896	0.4190	0.3885
Banding of Sample	2006	0.3096	0.3121	**0.4296**	0.3964
Final band. using Sampling		**0.3003**	**0.3011**	0.4513	**0.3766**
Lancashire					
Banding of Sample	2003	**0.3085**	**0.3119**	**0.4413**	**0.4166**
Final band. using Sampling		0.3093	0.3127	0.4491	0.4192
Banding of Sample	2004	**0.2790**	**0.2806**	**0.3991**	**0.3639**
Final band. using Sampling		0.2902	0.2924	0.4184	0.3817
Banding of Sample	2005	**0.2937**	**0.2968**	0.4313	**0.4036**
Final band. using Sampling		0.2951	0.2972	**0.4096**	0.4068
Banding of Sample	2006	0.2990	0.3016	**0.4309**	**0.4202**
Final band. using Sampling		**0.2856**	**0.2999**	0.4323	0.4228
Norfolk					
Banding of Sample	2003	**0.3095**	**0.3246**	**0.4358**	0.4265
Final band. using Sampling		0.3181	0.3253	0.4436	**0.4250**
Banding of Sample	2004	**0.3003**	0.3095	**0.4205**	0.3932
Final band. using Sampling		0.3008	**0.2713**	0.4207	**0.3927**
Banding of Sample	2005	**0.3182**	0.4010	**0.4421**	0.4058
Final band. using Sampling		0.3189	**0.3221**	0.4434	**0.4042**
Banding of Sample	2006	**0.2572**	**0.2597**	**0.3724**	**0.3867**
Final band. using Sampling		0.2600	0.2603	0.4014	0.3880

TABLE B.8: Sampling GBS result for 2003 to 2006 5D CTS data set

	Year	GBS		
		Euclid.	**Manhat.**	**Approx.**
Aberdeenshire				
Banding of Sample	2003	0.3034	0.3246	0.4359
Final band. using Sampling		**0.2502**	**0.2527**	**0.3489**
Banding of Sample	2004	**0.2095**	**0.2113**	**0.3886**
Final band. using Sampling		0.2101	0.2116	0.3962
Banding of Sample	2005	**0.2514**	**0.2528**	**0.4568**
Final band. using Sampling		0.2529	0.2545	0.4610
Banding of Sample	2006	**0.2073**	**0.2088**	0.3898
Final band. using Sampling		0.2083	0.2099	**0.3880**
Cornwall				
Banding of Sample	2003	**0.2451**	**0.2466**	**0.4517**
Final band. using Sampling		0.2994	0.3013	0.4595
Banding of Sample	2004	**0.2352**	0.2489	**0.4420**
Final band. using Sampling		0.2384	**0.2407**	0.4440
Banding of Sample	2005	**0.2300**	**0.2340**	**0.4316**
Final band. using Sampling		0.2357	0.2397	0.4466
Banding of Sample	2006	0.2538	**0.2548**	**0.4580**
Final band. using Sampling		**0.2465**	0.2561	0.4625
Lancashire				
Banding of Sample	2003	**0.2537**	**0.2559**	**0.4620**
Final band. using Sampling		0.2565	0.2575	0.4664
Banding of Sample	2004	**0.2346**	**0.2377**	**0.4240**
Final band. using Sampling		0.2441	0.2550	0.4431
Banding of Sample	2005	**0.2469**	0.2481	0.4627
Final band. using Sampling		0.2470	**0.2546**	**0.4475**
Banding of Sample	2006	0.2481	**0.2488**	**0.4627**
Final band. using Sampling		**0.2413**	0.2522	0.4635
Norfolk				
Banding of Sample	2003	**0.2687**	**0.2695**	**0.4750**
Final band. using Sampling		0.2690	0.2698	0.5005
Banding of Sample	2004	0.2513	**0.2518**	**0.4606**
Final band. using Sampling		**0.2264**	0.2524	0.4665
Banding of Sample	2005	**0.2572**	**0.2647**	0.4733
Final band. using Sampling		0.2580	0.2657	**0.4581**
Banding of Sample	2006	**0.2134**	0.2225	**0.4001**
Final band. using Sampling		0.2139	**0.2172**	0.4065

Appendix C

Additional Experimental Result on Segmentation

C.1 Introduction

In this appendix, the full experimental results with respect to the evaluation of the segmentation techniques are presented. Recall from Chapter 9 that, in the context of the segmentation technique, the data set D was divided into a sequence of six equal sized segments, to give a set of segments R. However, due to space restriction within the main body of the thesis results for only the segment featuring the best GBS value were presented; results for the remaining segments were not included. For completeness these additional results are thus presented here.

The appendix is organised as follows. Sub-appendix C.2, presents the quality of banding result in terms of GBS using the Euclidean MD-EBPM Algorithm, Sub-appendix C.3 presents the banding result in terms of GBS using the Manhattan MD-EBPM Algorithm, and Sub-appendix C.4 presents the results using the MD-ABPM Algorithm. In each case, for each county and year combination, the tables give the individual GBS results per segment and the GBS result had that segment been selected and the associated banding applied to the entire data set. The final best GBS score in each case is highlighted in bold font.

From the tables it can be seen that if we use best GBS value as the criterion for selecting a segmentation, in many cases, the best segment is not selected. This was born out by the results presented in the body of the thesis where the most frequent approach for selecting a banding was found to produce a best banding.

C.2 Effectiveness Results in Terms of GBS Using the Euclidean MD-EBPM Algorithm

The GBS results presented in this sub-appendix are those obtained using the Euclidean MD-EBPM algorithm. Tables C.1, C.2, C.3 and C.4 presents the GBS values in terms

of the counties considered: Aberdeenshire, Cornwall, Lancashire and Norfolk.

TABLE C.1: GBS results using the Euclidean MD-EBPM algorithm for Aberdeenshire

Data segment	**Year**	**MD-EBPM$_E$**		
id	**id**	**3D**	**4D**	**5D**
Aberdeenshire				
Banding of Segment 1	**2003**	0.3242	0.2384	0.1768
Banding of Segment 2		0.3332	0.2492	0.1836
Banding of Segment 3		0.3244	0.2398	0.1756
Banding of Segment 4		**0.3049**	**0.2371**	**0.1726**
Banding of Segment 5		0.3623	0.3259	0.2364
Banding of Segment 6		0.3083	0.2451	0.1768
Final band. using Segment 1		0.3607	0.2877	0.2382
Final band. using Segment 2		0.3677	0.2928	0.2407
Final band. using Segment 3		0.3528	0.2954	0.2431
Final band. using Segment 4		0.3643	0.2871	0.2372
Final band. using Segment 5		0.3863	0.3214	0.2640
Final band. using Segment 6		**0.3479**	**0.2780**	**0.2303**
Banding of Segment 1	**2004**	0.3744	0.2665	0.2002
Banding of Segment 2		0.3539	0.2578	0.1914
Banding of Segment 3		0.3318	0.2426	0.1798
Banding of Segment 4		0.3299	0.2489	0.1838
Banding of Segment 5		0.3165	0.2432	0.1786
Banding of Segment 6		**0.3023**	**0.2385**	**0.1736**
Final band. using Segment 1		0.3842	0.2792	0.2284
Final band. using Segment 2		0.3785	0.2665	0.2191
Final band. using Segment 3		0.3507	0.2508	0.2101
Final band. using Segment 4		0.3707	0.2553	0.2122
Final band. using Segment 5		0.3570	0.2509	0.2103
Final band. using Segment 6		**0.3403**	**0.2400**	**0.2013**
Banding of Segment 1	**2005**	0.2899	**0.2131**	**0.1587**
Banding of Segment 2		0.3206	0.2487	0.1841
Banding of Segment 3		0.3237	0.2513	0.1843
Banding of Segment 4		0.2986	0.2361	0.1729
Banding of Segment 5		0.3506	0.3099	0.2271
Banding of Segment 6		**0.2890**	0.2360	0.1702
Final band. using Segment 1		**0.3278**	0.2708	0.2253
Final band. using Segment 2		0.3468	0.2717	0.2259
Final band. using Segment 3		0.3603	0.2848	0.2353
Final band. using Segment 4		0.3440	**0.2650**	**0.2219**
Final band. using Segment 5		0.3923	0.3157	0.2565
Final band. using Segment 6		0.3573	0.2785	0.2319
Banding of Segment 1	**2006**	**0.3106**	**0.2218**	**0.1673**
Banding of Segment 2		0.3116	0.2292	0.1712
Banding of Segment 3		0.3427	0.2487	0.1829
Banding of Segment 4		0.3182	0.2325	0.1705
Banding of Segment 5		0.3413	0.2520	0.1729
Banding of Segment 6		0.3213	0.2463	0.1803
Final band. using Segment 1		**0.3322**	**0.2266**	**0.1922**
Final band. using Segment 2		0.3359	0.2307	0.1949
Final band. using Segment 3		0.3588	0.2532	0.2093
Final band. using Segment 4		0.3533	0.2442	0.2034
Final band. using Segment 5		0.3677	0.2572	0.1940
Final band. using Segment 6		0.3653	0.2482	0.2083

TABLE C.2: GBS results using the Euclidean MD-EBPM algorithm for Cornwall

Data segment id	**Year id**	**MD-EBPM$_E$**		
		3D	**4D**	**5D**
Cornwall				
Banding of Segment 1	**2003**	0.3442	0.2584	0.1888
Banding of Segment 2		0.3341	**0.2497**	**0.1817**
Banding of Segment 3		0.3483	0.2659	0.1914
Banding of Segment 4		0.3297	0.2555	0.1836
Banding of Segment 5		0.3676	0.2999	0.2266
Banding of Segment 6		**0.3263**	0.2591	0.1840
Final band. using Segment 1		0.3735	0.2874	0.2381
Final band. using Segment 2		0.3717	0.2866	0.2383
Final band. using Segment 3		0.3715	0.2944	0.2429
Final band. using Segment 4		**0.3572**	**0.2835**	**0.2357**
Final band. using Segment 5		0.3996	0.3025	0.2493
Final band. using Segment 6		0.3628	0.2852	0.2365
Banding of Segment 1	**2004**	0.3324	**0.2697**	0.2036
Banding of Segment 2		0.3528	0.2866	0.2149
Banding of Segment 3		0.3335	0.2812	0.2092
Banding of Segment 4		0.3358	0.2805	0.2079
Banding of Segment 5		0.3307	0.2833	0.2082
Banding of Segment 6		**0.3140**	0.2731	**0.1993**
Final band. using Segment 1		0.3566	0.2853	0.2350
Final band. using Segment 2		0.3779	0.2945	0.2415
Final band. using Segment 3		0.3710	0.2922	0.2395
Final band. using Segment 4		0.3762	**0.2639**	0.2385
Final band. using Segment 5		0.3661	0.2931	0.2404
Final band. using Segment 6		**0.3524**	0.2826	**0.2321**
Banding of Segment 1	**2005**	0.3276	0.2423	0.1769
Banding of Segment 2		0.3202	0.2409	0.1747
Banding of Segment 3		0.3206	0.2420	0.1746
Banding of Segment 4		**0.3073**	**0.2362**	**0.1699**
Banding of Segment 5		0.3864	0.3148	0.2350
Banding of Segment 6		0.3144	0.2496	0.1776
Final band. using Segment 1		0.3657	0.2821	0.2339
Final band. using Segment 2		0.3594	0.2822	0.2328
Final band. using Segment 3		0.3679	0.2835	0.2355
Final band. using Segment 4		0.3635	**0.2779**	**0.2321**
Final band. using Segment 5		0.3880	0.3147	0.2579
Final band. using Segment 6		**0.3580**	0.2783	0.2316
Banding of Segment 1	**2006**	**0.3186**	**0.790**	**0.2107**
Banding of Segment 2		0.3246	0.2843	0.2128
Banding of Segment 3		0.3269	0.2925	0.2174
Banding of Segment 4		0.3269	0.2856	0.2119
Banding of Segment 5		0.3550	0.3133	0.2309
Banding of Segment 6		0.3322	0.3011	0.2207
Final band. using Segment 1		**0.3449**	**0.2844**	**0.2352**
Final band. using Segment 2		0.3520	0.2896	0.2385
Final band. using Segment 3		0.3524	0.2948	0.2423
Final band. using Segment 4		0.3514	0.2926	0.2411
Final band. using Segment 5		0.3837	0.3177	0.2583
Final band. using Segment 6		0.3697	0.3042	0.2491

TABLE C.3: GBS results using the Euclidean MD-EBPM algorithm for Lancashire

Data segment id	**Year id**	**MD-EBPM$_E$**		
		3D	**4D**	**5D**
Lancashire				
Banding of Segment 1	**2003**	0.3556	**0.2793**	0.2096
Banding of Segment 2		0.3755	0.2996	0.2244
Banding of Segment 3		0.3604	0.2991	0.2255
Banding of Segment 4		0.3779	0.3065	0.2027
Banding of Segment 5		0.3673	0.2969	0.2182
Banding of Segment 6		**0.3394**	0.2796	**0.2047**
Final band. using Segment 1		**0.3747**	0.2965	**0.2436**
Final band. using Segment 2		0.3949	0.3086	0.2521
Final band. using Segment 3		0.3834	0.3110	0.2529
Final band. using Segment 4		0.4025	**0.2922**	0.2601
Final band. using Segment 5		0.4002	0.3123	0.2549
Final band. using Segment 6		0.3830	0.2991	0.2466
Banding of Segment 1	**2004**	0.2465	0.2330	0.1766
Banding of Segment 2		0.2568	0.2335	0.1776
Banding of Segment 3		0.3345	0.2859	0.2135
Banding of Segment 4		0.3538	0.2966	0.2210
Banding of Segment 5		**0.2417**	**0.2308**	**0.1722**
Banding of Segment 6		0.3101	0.2470	0.1768
Final band. using Segment 1		0.3197	0.2598	0.2192
Final band. using Segment 2		**0.3025**	**0.2421**	**0.2070**
Final band. using Segment 3		0.3585	0.3001	0.2477
Final band. using Segment 4		0.3767	0.3005	0.2491
Final band. using Segment 5		0.3105	0.2451	0.2092
Final band. using Segment 6		0.3696	0.2967	0.2512
Banding of Segment 1	**2005**	0.3586	0.2726	0.2037
Banding of Segment 2		0.3475	**0.2678**	0.2009
Banding of Segment 3		0.3631	0.2823	0.2115
Banding of Segment 4		0.3526	0.2780	0.2060
Banding of Segment 5		0.3608	0.2846	0.2099
Banding of Segment 6		**0.3354**	0.2728	**0.1988**
Final band. using Segment 1		0.3843	0.3041	0.2381
Final band. using Segment 2		**0.3797**	0.2901	**0.2319**
Final band. using Segment 3		0.3866	**0.2787**	0.2387
Final band. using Segment 4		0.3834	0.2893	0.2386
Final band. using Segment 5		0.3953	0.2926	0.2414
Final band. using Segment 6		0.3862	0.2870	0.2359
Banding of Segment 1	**2006**	0.3465	0.2768	0.2088
Banding of Segment 2		0.3569	0.2739	0.2047
Banding of Segment 3		0.3613	0.2821	0.2101
Banding of Segment 4		**0.3451**	**0.2721**	**0.2011**
Banding of Segment 5		0.3619	0.2886	0.2156
Banding of Segment 6		0.3586	0.2858	0.2084
Final band. using Segment 1		**0.3729**	0.2927	0.2417
Final band. using Segment 2		0.3816	0.2937	0.2427
Final band. using Segment 3		0.3916	0.3008	0.2473
Final band. using Segment 4		0.3854	**0.2899**	**0.2393**
Final band. using Segment 5		0.3901	0.2975	0.2459
Final band. using Segment 6		0.3951	0.3042	0.2498

TABLE C.4: GBS results using the Euclidean MD-EBPM algorithm for Norfolk

Data segment id	**Year id**	**MD-EBPM$_E$**		
		3D	**4D**	**5D**
Norfolk				
Banding of Segment 1	**2003**	0.3192	0.2290	0.1712
Banding of Segment 2		0.3277	0.2395	0.1774
Banding of Segment 3		0.3239	0.2407	0.1764
Banding of Segment 4		**0.2889**	**0.2188**	**0.1605**
Banding of Segment 5		0.3509	0.2936	0.2306
Banding of Segment 6		0.3207	0.2493	0.1806
Final band. using Segment 1		0.3576	0.2873	0.2365
Final band. using Segment 2		0.3717	0.2995	0.2475
Final band. using Segment 3		0.3600	0.2956	0.2439
Final band. using Segment 4		**0.3399**	**0.2742**	**0.2316**
Final band. using Segment 5		0.3881	0.2920	0.2570
Final band. using Segment 6		0.3591	0.2936	0.2442
Banding of Segment 1	**2004**	0.3201	0.2550	0.2091
Banding of Segment 2		0.3085	0.2713	0.2017
Banding of Segment 3		0.3065	0.2760	0.2036
Banding of Segment 4		0.3092	0.2788	0.2065
Banding of Segment 5		0.2980	0.2671	0.1968
Banding of Segment 6		**0.2894**	**0.2665**	**0.1951**
Final band. using Segment 1		0.3427	0.2810	0.2365
Final band. using Segment 2		0.3336	0.2801	0.2311
Final band. using Segment 3		0.3568	0.2993	0.2445
Final band. using Segment 4		0.3508	0.2900	0.2393
Final band. using Segment 5		0.3476	0.2811	0.2325
Final band. using Segment 6		**0.3278**	**0.2689**	**0.2237**
Banding of Segment 1	**2005**	0.2985	**0.2149**	0.1618
Banding of Segment 2		0.3219	0.2203	0.1643
Banding of Segment 3		0.2911	0.2387	0.1750
Banding of Segment 4		**0.2910**	0.2194	**0.1611**
Banding of Segment 5		0.3381	0.3001	0.2206
Banding of Segment 6		0.2933	0.2259	0.1609
Final band. using Segment 1		0.3387	**0.2713**	**0.2273**
Final band. using Segment 2		0.3465	0.2735	0.2304
Final band. using Segment 3		**0.2837**	0.2811	0.2351
Final band. using Segment 4		0.3447	0.2751	0.2304
Final band. using Segment 5		0.3805	0.3061	0.2501
Final band. using Segment 6		0.3397	0.2740	0.2286
Banding of Segment 1	**2006**	0.2718	**0.1947**	**0.1477**
Banding of Segment 2		0.3860	0.2216	0.1658
Banding of Segment 3		0.2888	0.2110	0.1576
Banding of Segment 4		0.2726	0.2109	0.1568
Banding of Segment 5		0.2944	0.2204	0.1622
Banding of Segment 6		**0.2653**	0.2023	0.1491
Final band. using Segment 1		**0.3154**	0.2170	**0.1857**
Final band. using Segment 2		0.3405	0.2347	0.1979
Final band. using Segment 3		0.3296	0.2257	0.1908
Final band. using Segment 4		0.3211	0.2214	0.1893
Final band. using Segment 5		0.3402	0.2338	0.1978
Final band. using Segment 6		0.3165	**0.2161**	0.1861

C.3 Effectiveness Results in Terms of GBS Using the Manhattan MD-EBPM Algorithm

This sub-appendix gives the results using the Manhattan variation of the MD-EBPM Algorithm. The results obtained, with respect to each county, are listed in Tables C.5, C.6, C.7 and C.8.

TABLE C.5: GBS results using the Manhattan MD-EBPM algorithm for Aberdeenshire

Data segment id	**Year id**	**MD-EBPM$_M$**		
		3D	**4D**	**5D**
Aberdeenshire				
Banding of Segment 1	**2003**	0.3242	0.2398	0.1776
Banding of Segment 2		0.3332	0.2503	0.1840
Banding of Segment 3		0.3244	0.2408	0.1761
Banding of Segment 4		**0.3049**	**0.2391**	**0.1735**
Banding of Segment 5		0.3623	0.3281	0.2381
Banding of Segment 6		0.3083	0.2461	0.1773
Final band. using Segment 1		0.3607	0.2946	0.2399
Final band. using Segment 2		0.3677	0.2968	0.2482
Final band. using Segment 3		0.3528	0.2977	0.2499
Final band. using Segment 4		0.3643	0.2897	0.2385
Final band. using Segment 5		0.3863	0.3274	0.2662
Final band. using Segment 6		**0.3479**	**0.2794**	**0.2383**
Banding of Segment 1	**2004**	0.3744	0.2690	0.2009
Banding of Segment 2		0.3539	0.2589	0.1921
Banding of Segment 3		0.3318	0.2434	0.1811
Banding of Segment 4		0.3299	0.2507	0.1850
Banding of Segment 5		0.3165	0.2442	0.1794
Banding of Segment 6		**0.3023**	**0.2396**	**0.1744**
Final band. using Segment 1		0.3842	0.2875	0.2294
Final band. using Segment 2		0.3785	0.2693	0.2281
Final band. using Segment 3		0.3507	00.2597	0.2190
Final band. using Segment 4		0.3707	0.2593	0.2213
Final band. using Segment 5		0.3570	0.2539	0.2193
Final band. using Segment 6		**0.3403**	**0.2490**	0.2104
Banding of Segment1	**2005**	0.2899	**0.2138**	**0.1592**
Banding of Segment 2		0.3206	0.2499	0.1850
Banding of Segment 3		0.3237	0.2521	0.1849
Banding of Segment 4		0.2986	0.2366	0.1733
Banding of Segment 5		0.3506	0.3129	0.2297
Banding of Segment 6		**0.2890**	0.2368	0.1712
Final band. using Segment 1		**0.3278**	0.2786	0.2331
Final band. using Segment 2		0.3468	0.2798	0.2339
Final band. using Segment 3		0.3603	0.2884	0.2435
Final band. using Segment 4		0.3440	**0.2677**	**0.2299**
Final band. using Segment 5		0.3923	0.3172	0.2577
Final band. using Segment 6		0.3573	0.2794	0.2400
Banding of Segment 1	**2006**	**0.3106**	**0.2231**	**0.1679**
Banding of Segment 2		0.3116	0.2305	0.1721
Banding of Segment 3		0.3427	0.2503	0.1839
Banding of Segment 4		0.3182	0.2342	0.1713
Banding of Segment 5		0.3413	0.2544	0.1735
Banding of Segment 6		0.3213	0.2481	0.1809
Final band. using Segment 1		**0.3322**	**0.2284**	**0.1943**
Final band. using Segment 2		0.3359	0.2394	0.1962
Final band. using Segment 3		0.3588	0.2577	0.2175
Final band. using Segment 4		0.3533	0.2465	0.2117
Final band. using Segment 5		0.3677	0.2589	0.1968
Final band. using Segment 6		0.3653	0.2496	0.2177

Table C.6: GBS results using the Manhattan MD-EBPM algorithm For Cornwall

Data segment id	**Year id**	**MD-EBPM$_M$**		
		3D	**4D**	**5D**
Cornwall				
Banding of Segment 1	**2003**	0.3442	0.2590	0.1896
Banding of Segment 2		0.3341	**0.2503**	**0.1822**
Banding of Segment 3		0.3483	0.2661	0.1923
Banding of Segment 4		0.3297	0.2556	0.1842
Banding of Segment 5		0.3676	0.3036	0.2296
Banding of Segment 6		**0.3263**	0.2593	0.1845
Final band. using Segment 1		0.3735	0.2971	0.2446
Final band. using Segment 2		0.3717	0.3062	0.2449
Final band. using Segment 3		0.3715	0.2991	0.2487
Final band. using Segment 4		0.3572	0.2880	**0.2385**
Final band. using Segment 5		0.3996	0.3087	0.2499
Final band. using Segment 6		**0.3628**	0.2871	0.2497
Banding of Segment 1	**2004**	0.3324	**0.2717**	0.2052
Banding of Segment 2		0.3528	0.2883	0.2159
Banding of Segment 3		0.3335	0.2830	0.2105
Banding of Segment 4		0.3358	0.2825	0.2095
Banding of Segment 5		0.3307	0.2843	0.2093
Banding of Segment 6		**0.3140**	0.2750	**0.2011**
Final band. using Segment 1		0.3566	0.2932	0.2350
Final band. using Segment 2		0.3779	0.3026	0.2482
Final band. using Segment 3		0.3710	0.3007	0.2485
Final band. using Segment 4		0.3762	**0.2393**	0.2398
Final band. using Segment 5		0.3661	0.2979	0.2404
Final band. using Segment 6		**0.3524**	0.2886	**0.2353**
Banding of Segment 1	**2005**	0.3276	0.2426	0.1775
Banding of Segment 2		0.3202	0.2412	0.1754
Banding of Segment 3		0.3206	0.2425	0.1753
Banding of Segment 4		**0.3073**	**0.2365**	**0.1705**
Banding of Segment 5		0.3864	0.3302	0.2396
Banding of Segment 6		0.3144	0.2499	0.1778
Final band. using Segment 1		0.3657	0.2901	**0.2351**
Final band. using Segment 2		0.3594	**0.2865**	0.2391
Final band. using Segment 3		0.3679	0.2906	0.2417
Final band. using Segment 4		0.3635	0.2791	0.2399
Final band. using Segment 5		0.3880	0.3191	0.2590
Final band. using Segment 6		**0.3580**	0.2866	0.2388
Banding of Segment 1	**2006**	**0.3186**	**0.2812**	**0.2123**
Banding of Segment 2		0.3246	0.2858	0.2144
Banding of Segment 3		0.3269	0.2933	0.2186
Banding of Segment 4		0.3269	0.2873	0.2138
Banding of Segment 5		0.3550	0.3167	0.2335
Banding of Segment 6		0.3322	0.3041	0.2235
Final band. using Segment 1		**0.3449**	**0.2923**	**0.2385**
Final band. using Segment 2		0.3520	0.2979	0.2470
Final band. using Segment 3		0.3524	0.2977	0.2471
Final band. using Segment 4		0.3514	0.2969	0.2495
Final band. using Segment 5		0.3837	0.3183	0.2583
Final band. using Segment 6		0.3697	0.3114	0.2562

Table C.7: GBS results using the Manhattan MD-EBPM algorithm for Lancashire

Data segment id	**Year id**	**MD-EBPM$_M$**		
		3D	**4D**	**5D**
Lancashire				
Banding of Segment 1	**2003**	0.3556	**0.2807**	0.2106
Banding of Segment 2		0.3755	0.3014	0.2264
Banding of Segment 3		0.3604	0.3020	0.2229
Banding of Segment 4		0.3779	0.3080	0.2272
Banding of Segment 5		0.3673	0.3000	0.2199
Banding of Segment 6		**0.3394**	0.2830	**0.2070**
Final band. using Segment 1		**0.3747**	**0.3041**	**0.2465**
Final band. using Segment 2		0.3949	0.3062	0.2541
Final band. using Segment 3		0.3834	0.3190	0.2551
Final band. using Segment 4		0.4025	0.3109	0.2603
Final band. using Segment 5		0.4002	0.3133	0.2562
Final band. using Segment 6		0.3830	0.3059	0.2483
Banding of Segment 1	**2004**	0.2465	0.2343	0.1779
Banding of Segment 2		0.2568	0.2342	0.1789
Banding of Segment 3		0.3345	0.2888	0.2157
Banding of Segment 4		0.3538	0.2971	0.2226
Banding of Segment 5		**0.2417**	**0.2317**	**0.1733**
Banding of Segment 6		0.3101	0.2474	0.1777
Final band. using Segment 1		0.3197	0.2682	0.2275
Final band. using Segment 2		**0.3025**	**0.2445**	**0.2163**
Final band. using Segment 3		0.3585	0.3169	0.2494
Final band. using Segment 4		0.3767	0.3186	0.2581
Final band. using Segment 5		0.3105	0.2483	0.2183
Final band. using Segment 6		0.3696	0.3036	0.2586
Banding of Segment 1	**2005**	0.3586	0.2746	0.2053
Banding of Segment 2		0.3475	**0.2678**	0.2017
Banding of Segment 3		0.3631	0.2847	0.2137
Banding of Segment 4		0.3526	0.2789	0.2076
Banding of Segment 5		0.3608	0.2875	0.2118
Banding of Segment 6		**0.3354**	0.2741	**0.1999**
Final band. using Segment 1		0.3843	0.3072	0.2456
Final band. using Segment 2		**0.3797**	0.2969	**0.2342**
Final band. using Segment 3		0.3866	**0.2787**	0.2396
Final band. using Segment 4		0.3834	0.2893	0.2387
Final band. using Segment 5		0.3953	0.2951	0.2494
Final band. using Segment 6		0.3862	0.2942	0.2431
Banding of Segment 1	**2006**	0.3465	0.2768	0.2099
Banding of Segment 2		0.3569	0.2765	0.2065
Banding of Segment 3		0.3613	0.2834	0.2122
Banding of Segment 4		**0.3451**	**0.2731**	**0.2026**
Banding of Segment 5		0.3619	0.2928	0.2181
Banding of Segment 6		0.3586	0.2876	0.2105
Final band. using Segment 1		**0.3729**	**0.2962**	0.2396
Final band. using Segment 2		0.3816	0.2965	0.2403
Final band. using Segment 3		0.3916	0.3027	0.2443
Final band. using Segment 4		0.3854	0.2967	**0.2370**
Final band. using Segment 5		0.3901	0.2988	0.2435
Final band. using Segment 6		0.3951	0.3054	0.2475

Table C.8: GBS results using the Manhattan MD-EBPM algorithm for Norfolk

Data segment id	**Year id**	**MD-EBPM$_M$**		
		3D	**4D**	**5D**
Norfolk				
Banding of Segment 1	**2003**	0.3192	0.2300	0.1718
Banding of Segment 2		0.3277	0.2403	0.1780
Banding of Segment 3		0.3239	0.2416	0.1788
Banding of Segment 4		**0.2889**	**0.2191**	**0.1629**
Banding of Segment 5		0.3509	0.2936	0.2326
Banding of Segment 6		0.3207	0.2498	0.1810
Final band. using Segment 1		0.3576	0.2938	**0.2366**
Final band. using Segment 2		0.3717	0.3056	0.2482
Final band. using Segment 3		0.3600	0.2979	0.2458
Final band. using Segment 4		**0.3399**	**0.2788**	0.2392
Final band. using Segment 5		0.3881	0.2966	0.2583
Final band. using Segment 6		0.3591	0.2988	0.2483
Banding of Segment 1	**2004**	0.3201	0.2750	0.2106
Banding of Segment 2		0.3085	0.2722	0.2029
Banding of Segment 3		0.3065	0.2780	0.2054
Banding of Segment 4		0.3092	0.2791	0.2080
Banding of Segment 5		**0.2980**	0.2688	0.1999
Banding of Segment 6		0.2894	**0.2678**	**0.1972**
Final band. using Segment 1		0.3427	0.2887	0.2375
Final band. using Segment 2		0.3336	0.2894	0.2327
Final band. using Segment 3		0.3568	0.2999	0.2454
Final band. using Segment 4		0.3508	0.2989	0.2397
Final band. using Segment 5		0.3476	0.2897	0.2346
Final band. using Segment 6		**0.3278**	**0.2696**	**0.2317**
Banding of Segment 1	**2005**	0.2985	**0.2174**	0.1628
Banding of Segment 2		0.3219	0.2226	0.1649
Banding of Segment 3		0.2911	0.2396	0.1753
Banding of Segment 4		**0.2910**	0.2208	**0.1615**
Banding of Segment 5		0.3381	0.3027	0.2231
Banding of Segment 6		0.2933	0.2275	0.1629
Final band. using Segment 1		0.3387	**0.2782**	**0.2347**
Final band. using Segment 2		0.3465	0.2795	0.2377
Final band. using Segment 3		**0.2837**	0.2863	0.2420
Final band. using Segment 4		0.3447	0.2788	0.2327
Final band. using Segment 5		0.3805	0.3134	0.2578
Final band. using Segment 6		0.3397	0.2799	0.2353
Banding of Segment 1	**2006**	0.2718	**0.1978**	**0.1482**
Banding of Segment 2		0.3860	0.2229	0.1661
Banding of Segment 3		0.2888	0.2139	0.1581
Banding of Segment 4		0.2726	0.2123	0.1573
Banding of Segment 5		0.2944	0.2214	0.1628
Banding of Segment 6		**0.2653**	0.2036	0.1497
Final band. using Segment 1		**0.3154**	0.2261	**0.1869**
Final band. using Segment 2		0.3405	0.2437	0.1981
Final band. using Segment 3		0.3296	0.2264	0.1987
Final band. using Segment 4		0.3211	0.2224	0.1987
Final band. using Segment 5		0.3402	0.2358	0.1989
Final band. using Segment 6		0.3165	**0.2251**	0.1953

C.4 Effectiveness Results in Terms of GBS Using the MD-ABPM Algorithm

This sub-appendix presents the results obtained using the MD-ABPM algorithm. The results, with respect to each county, are presented in Tables C.9, C.10, C.11 and C.12.

TABLE C.9: GBS results using the MD-ABPM algorithm for Aberdeenshire

Data segment id	**Year id**	**MD-ABPM**		
		3D	**4D**	**5D**
Aberdeenshire				
Banding of Segment 1	2003	0.3242	0.3415	0.3318
Banding of Segment 2		0.3332	0.3526	0.3429
Banding of Segment 3		0.3244	0.3502	0.3354
Banding of Segment 4		**0.3049**	**0.3398**	**0.3208**
Banding of Segment 5		0.3623	0.4301	0.4206
Banding of Segment 6		0.3083	0.3511	0.3299
Final band. using Segment 1		0.3607	**0.4037**	0.4306
Final band. using Segment 2		0.3677	0.4127	0.4380
Final band. using Segment 3		0.3528	0.4221	0.4459
Final band. using Segment 4		0.3643	0.4043	0.4336
Final band. using Segment 5		0.3863	0.4318	**0.4175**
Final band. using Segment 6		**0.3479**	0.4557	0.4781
Banding of Segment 1	2004	0.3744	0.3824	0.3791
Banding of Segment 2		0.3539	0.3684	0.3622
Banding of Segment 3		0.3318	0.3481	0.3423
Banding of Segment 4		0.3299	0.3593	0.3466
Banding of Segment 5		0.3165	0.3514	0.3328
Banding of Segment 6		**0.3023**	**0.3411**	**0.3257**
Final band. using Segment 1		0.3842	0.4009	0.4224
Final band. using Segment 2		0.3785	0.3843	0.4092
Final band. using Segment 3		0.3507	0.3601	**0.3429**
Final band. using Segment 4		0.3707	0.3678	0.3970
Final band. using Segment 5		0.3570	0.3613	0.3892
Final band. using Segment 6		**0.3403**	**0.3401**	0.3763
Banding of Segment 1	2005	0.2899	**0.3087**	**0.3056**
Banding of Segment 2		0.3206	0.3509	0.3446
Banding of Segment 3		0.3237	0.3573	0.3452
Banding of Segment 4		0.2986	0.3370	0.3179
Banding of Segment 5		0.3506	0.4294	0.4145
Banding of Segment 6		**0.2890**	0.3335	0.3119
Final band. using Segment 1		**0.3278**	0.3826	0.4116
Final band. using Segment 2		0.3468	0.3785	0.4121
Final band. using Segment 3		0.3440	0.3759	**0.4018**
Final band. using Segment 4		0.3923	**0.3680**	0.4561
Final band. using Segment 5		0.3573	0.3884	0.4240
Final band. using Segment 6		0.3403	0.3401	0.3763
Banding of Segment 1	2006	**0.3106**	**0.3213**	**0.3219**
Banding of Segment 2		0.3116	0.3304	0.3259
Banding of Segment 3		0.3427	0.3588	0.3461
Banding of Segment 4		0.3182	0.3366	0.3213
Banding of Segment 5		0.3413	0.3655	0.3343
Banding of Segment 6		0.3213	0.3559	0.3351
Final band. using Segment 1		**0.3322**	**0.3285**	**0.3591**
Final band. using Segment 2		0.3359	0.3306	0.3602
Final band. using Segment 3		0.3588	0.3646	0.3877
Final band. using Segment 4		0.3533	0.3531	0.3748
Final band. using Segment 5		0.3677	0.3706	0.3671
Final band. using Segment 6		0.3653	0.3568	0.3822

Table C.10: GBS results using the MD-ABPM algorithm for Cornwall

Data segment id	**Year id**	**MD-ABPM**		
		3D	**4D**	**5D**
Cornwall				
Banding of Segment 1	2003	0.3442	0.3702	0.3559
Banding of Segment 2		0.3341	**0.3606**	**0.3421**
Banding of Segment 3		0.3483	0.3851	0.3624
Banding of Segment 4		0.3297	0.3699	0.3441
Banding of Segment 5		0.3676	0.4410	0.4242
Banding of Segment 6		**0.3263**	0.3746	0.3422
Final band. using Segment1		0.3735	0.4092	0.4355
Final band. using Segment 2		0.3717	0.4122	0.4376
Final band. using Segment 3		0.3715	0.4221	0.4509
Final band. using Segment 4		**0.3572**	0.4098	0.3271
Final band. using Segment 5		0.3996	0.4404	0.4626
Final band. using Segment 6		0.3628	0.4057	0.4334
Banding of Segment 1	2004	0.3324	**0.3855**	0.3826
Banding of Segment 2		0.3528	0.4103	0.4007
Banding of Segment 3		0.3335	0.4005	0.3851
Banding of Segment 4		0.3358	0.4028	0.3889
Banding of Segment 5		0.3307	0.4035	0.3822
Banding of Segment 6		**0.3140**	0.3863	**0.3650**
Final band. using Segment 1		0.3566	0.4050	0.4317
Final band. using Segment 2		0.3779	0.3455	0.4428
Final band. using Segment 3		0.3710	0.4128	0.4389
Final band. using Segment 4		0.3762	0.3867	0.4224
Final band. using Segment 5		0.3661	0.4068	0.4395
Final band. using Segment 6		**0.3524**	0.3372	0.4248
Banding of Segment 1	2005	0.3276	0.3550	0.3395
Banding of Segment 2		0.3202	0.3524	0.3296
Banding of Segment 3		0.3206	0.3552	0.3306
Banding of Segment 4		**0.3073**	**0.3427**	**0.3196**
Banding of Segment 5		0.3864	0.4532	0.4337
Banding of Segment 6		0.3144	0.3644	0.3316
Final band. using Segment 1		0.3657	0.4050	0.4291
Final band. using Segment 2		0.3594	0.4041	0.4264
Final band. using Segment 3		0.3679	0.4098	0.4318
Final band. using Segment 4		0.3635	0.3943	0.4219
Final band. using Segment 5		0.3880	0.3284	0.4729
Final band. using Segment 6		**0.3580**	0.4013	0.4259
Banding of Segment 1	2006	0.3186	**0.3971**	0.3951
Banding of Segment 2		0.3246	0.4011	0.3980
Banding of Segment 3		0.3269	0.4095	0.4007
Banding of Segment 4		**0.3160**	0.4033	**0.3876**
Banding of Segment 5		0.3550	0.4447	0.4243
Banding of Segment 6		0.3322	0.4235	0.4042
Final band. using Segment 1		**0.3449**	0.4049	0.4292
Final band. using Segment 2		0.3520	0.4101	0.4399
Final band. using Segment 3		0.3524	0.4127	0.4415
Final band. using Segment 4		0.3514	0.4022	0.4363
Final band. using Segment 5		0.3837	0.4494	0.4684
Final band. using Segment 6		0.3697	0.4274	0.4541

Table C.11: GBS results using the MD-ABPM algorithm for Lancashire

Data segment id	**Year id**	**MD-ABPM**		
		3D	**4D**	**5D**
Lancashire				
Banding of Segment 1	2003	0.3556	**0.3966**	0.3920
Banding of Segment 2		0.3755	0.4316	0.4213
Banding of Segment 3		0.3604	0.4174	0.4088
Banding of Segment 4		0.3779	0.4328	0.4178
Banding of Segment 5		0.3673	0.4244	0.4024
Banding of Segment 6		**0.3394**	0.4027	**0.3777**
Final band. using Segment 1		**0.3747**	**0.4176**	**0.4482**
Final band. using Segment 2		0.3949	0.4361	0.4625
Final band. using Segment 3		0.3834	0.4302	0.4739
Final band. using Segment 4		0.4025	0.4458	0.4712
Final band. using Segment 5		0.4002	0.4434	0.4660
Final band. using Segment 6		0.3830	0.4237	0.4493
Banding of Segment 1	2004	0.2465	**0.3190**	0.3188
Banding of Segment 2		0.2568	0.3342	0.3341
Banding of Segment 3		0.3345	0.4058	0.3890
Banding of Segment 4		0.3538	0.4116	0.4104
Banding of Segment 5		**0.2417**	0.3302	**0.3180**
Banding of Segment 6		0.3101	0.3520	0.3267
Final band. using Segment 1		0.3197	0.3593	0.3975
Final band. using Segment 2		**0.3025**	**0.3449**	0.3897
Final band. using Segment 3		0.3585	0.4216	0.4431
Final band. using Segment 4		0.3767	0.4261	0.4568
Final band. using Segment 5		0.3105	0.3497	**0.3886**
Final band. using Segment 6		0.3696	0.4252	0.4555
Banding of Segment 1	2005	0.3586	0.3861	0.3845
Banding of Segment 2		0.3475	**0.3830**	0.3799
Banding of Segment 3		0.3631	0.4067	0.3989
Banding of Segment 4		0.3526	0.3933	0.3868
Banding of Segment 5		0.3608	0.4121	0.3944
Banding of Segment 6		**0.3354**	0.3891	**0.3703**
Final band. using Segment 1		0.3843	0.4101	0.4308
Final band. using Segment 2		**0.3797**	**0.3589**	**0.4255**
Final band. using Segment 3		0.3866	0.4144	0.4476
Final band. using Segment 4		0.3834	0.4155	0.4419
Final band. using Segment 5		0.3953	0.4182	0.4421
Final band. using Segment 6		0.3862	0.4057	0.4351
Banding of Segment 1	2006	**0.3346**	0.3882	0.3927
Banding of Segment 2		0.3569	**0.3875**	0.3822
Banding of Segment 3		0.3613	0.4004	0.3918
Banding of Segment 4		0.3451	0.3909	**0.3770**
Banding of Segment 5		0.3619	0.4253	0.4083
Banding of Segment 6		0.3586	0.4108	0.3860
Final band. using Segment 1		**0.3729**	0.4049	0.4423
Final band. using Segment 2		0.3816	0.4182	0.4430
Final band. using Segment 3		0.3916	0.4247	0.4472
Final band. using Segment 4		0.3854	0.4060	**0.4385**
Final band. using Segment 5		0.3901	0.4284	0.4623
Final band. using Segment 6		0.3951	**0.3252**	0.4571

Table C.12: GBS results using the MD-ABPM algorithm for Norfolk

Data segment id	**Year id**	**MD-ABPM**		
		3D	**4D**	**5D**
Norfolk				
Banding of Segment 1	2003	**0.3192**	0.3333	0.3256
Banding of Segment 2		0.3277	0.3440	0.3323
Banding of Segment 3		0.3239	0.3477	0.3369
Banding of Segment 4		0.2889	**0.3171**	**0.3004**
Banding of Segment 5		0.3509	0.4232	0.4099
Banding of Segment 6		0.3207	0.3602	0.3364
Final band. using Segment 1		0.3576	0.4101	0.4326
Final band. using Segment 2		0.3717	0.4220	0.4543
Final band. using Segment 3		0.3600	0.4213	0.4465
Final band. using Segment 4		**0.3399**	**0.3930**	**0.4206**
Final band. using Segment 5		0.3881	0.4247	0.4564
Final band. using Segment 6		0.3591	0.4147	0.4437
Banding of Segment 1	2004	0.3201	0.3920	0.3877
Banding of Segment 2		0.3085	**0.3715**	0.3681
Banding of Segment 3		0.3065	0.3893	0.3679
Banding of Segment 4		0.3092	0.3941	0.3792
Banding of Segment 5		0.2980	0.3796	0.3648
Banding of Segment 6		**0.2894**	0.3867	**0.3612**
Final band. using Segment 1		0.3427	0.4015	0.4368
Final band. using Segment 2		0.3336	0.3840	0.4241
Final band. using Segment 3		0.3568	0.4191	0.4472
Final band. using Segment 4		0.3508	0.3918	0.4389
Final band. using Segment 5		0.3476	0.3964	0.4298
Final band. using Segment 6		0.3698	0.4208	0.4459
Banding of Segment1	2005	0.2985	**0.3137**	0.3066
Banding of Segment 2		0.3219	0.3163	0.3088
Banding of Segment 3		0.2911	0.3419	0.3290
Banding of Segment4		0.2910	0.3138	**0.3016**
Banding of Segment 5		0.3381	0.4302	0.4098
Banding of Segment 6		0.2933	0.3247	0.3033
Final band. using Segment 1		0.3387	**0.3832**	**0.4154**
Final band. using Segment 2		0.3465	0.3859	0.4179
Final band. using Segment 3		**0.2837**	0.3981	0.4262
Final band. using Segment 4		0.3447	0.3843	0.4212
Final band. using Segment 5		0.3805	0.4274	0.4585
Final band. using Segment 6		0.3397	0.3846	0.4156
Banding of Segment 1	2006	**0.2718**	**0.2835**	**0.2824**
Banding of Segment 2		0.3860	0.3189	0.3144
Banding of Segment 3		0.2888	0.3069	0.2984
Banding of Segment 4		0.2726	0.3015	0.2948
Banding of Segment 5		0.2944	0.3195	0.3057
Banding of Segment 6		0.2653	0.2933	0.2798
Final band. using Segment 1		**0.3154**	0.3075	0.3476
Final band. using Segment 2		0.3405	0.3319	0.3665
Final band. using Segment 3		0.3296	0.3216	0.3538
Final band. using Segment 4		0.3211	0.3171	0.3523
Final band. using Segment 5		0.3402	0.3328	0.3723
Final band. using Segment 6		0.3165	**0.3068**	**0.3460**

Appendix D

Some Additional Analysis Concerning Number of Iterations to Arrive at a Best Banding

D.1 Introduction

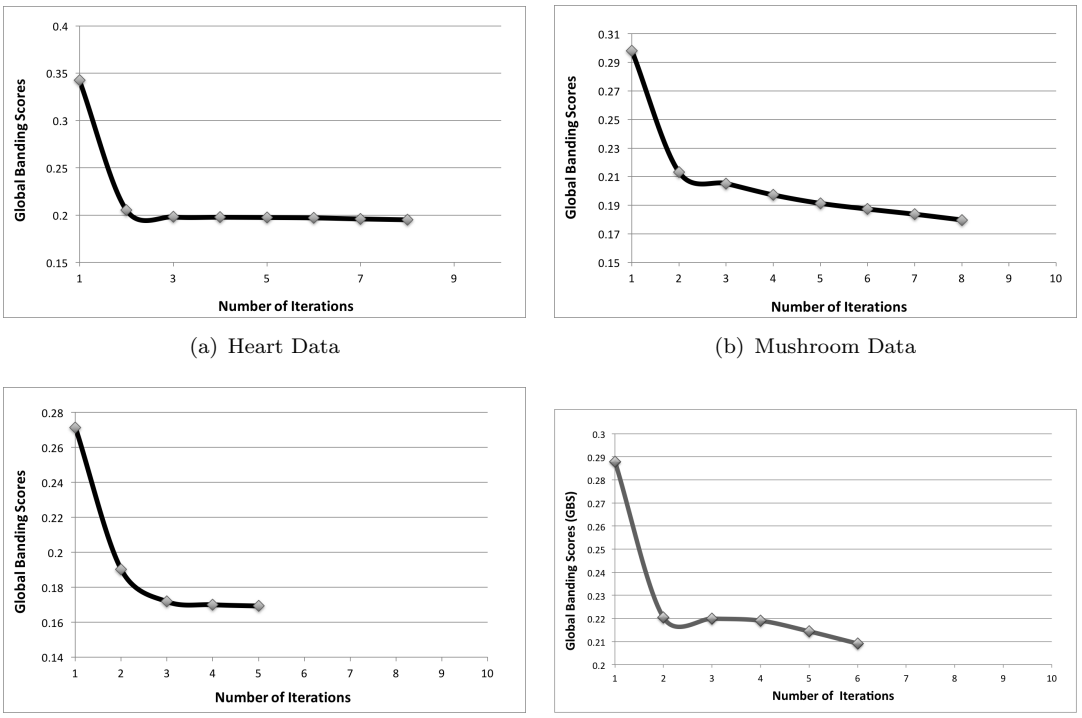
In this appendix some additional results to those presented in Chapters 4 and 9 are presented. More specifically in Chapter 4 analysis concerning the operation of the proposed 2D-BPM algorithm was presented in terms of the number of iterations required to arrive at a best banding; some further results in this respect are presented here. Similarly in Chapter 9, analysis with respect to the effectiveness of the proposed sampling and segmentation techniques in terms of the quality of the bandings produced when using MD-BPM algorithms. Further evaluation is presented here concerning the number of iterations that the MD-BPM algorithms require to arrive at a best banding.

The appendix is organised as follows. Sub-appendix D.2, presents the further analysis concerning the 2D-BPM algorithm in terms of number of iterations; while Sub-appendix D.3, presents the further analysis of the operation of the MD-BPM algorithms (MD-EBPM and MD-ABPM) in terms of the number of iterations required to arrive at a best banding.

D.2 Further Analysis of 2D-BPM Algorithm in Terms of Number of Iterations

This sub-appendix provides some additional analysis of the operation of the 2D-BPM algorithm proposed in Chapter 4 in terms of the number of iterations required to arrived at a best banding. Recall that in Chapter 4, graphs were presented indicating the number of iterations required for the 2D-BPM algorithm to find a best configuration (a best GBS value). This was done using eight of the twelve UCI data sets considered in this thesis. However, because of space limitations, four of the UCI data sets were excluded. This

sub-appendix presents the graphs associated with the four remaining UCI data sets. The relevant graphs are presented Figure D.1, in the given plots the x axis represents the number of iterations and the y axis the GBS values. Inspection of the figures further confirms the significant improvement of GBS values after the first few iterations and that the best GBS value (the minimal GBS value) is reached before the prescribed maximum number of iterations of "10" is reached.



(a) Heart Data

(b) Mushroom Data

(c) LetRecognition Data

(d) Waveform Data

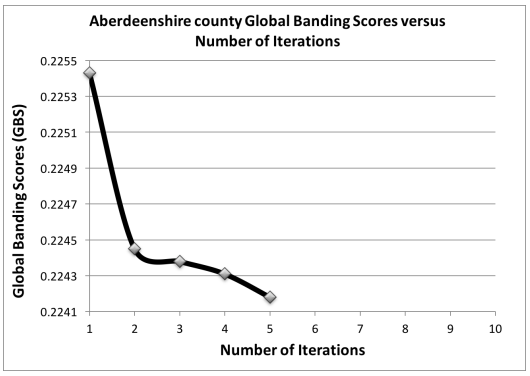
FIGURE D.1: GBS values per number of iterations obtained for the remaining four UCI data sets using the 2D-BPM algorithm.

D.3 Further Analysis of MD-BPM Algorithm in Terms of Number of Iterations

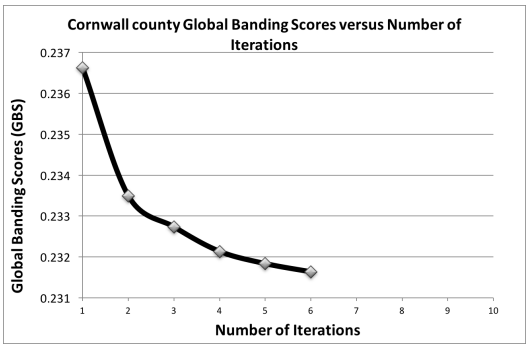
In this sub-appendix some graphs are presented indicating the number of iterations required using the Euclidean MD-EBPM, Manhattan MD-EBPM and MD-ABPM Algorithms to identify a best configuration. In the body of the thesis this was reported only in the context of the 2D-BPM algorithm. The objective was thus to analyse the operation of the MD-BPM (MD-EBPM and MD-ABPM) algorithms discussed in Chapter 8 in terms of the number of iterations required to identify a banding.

The result are presented in Figures D.2 and D.3 which show how the GBS value decreases with the number of iterations. In the graphs the iteration number is given on the X-axis and the GBS value on the Y-axis. In Figure D.2 graphs (a), (b), (c) and (d) shows the behaviour using Euclidean MD-EBPM, and graphs (e), (f),(g) and (h) shows

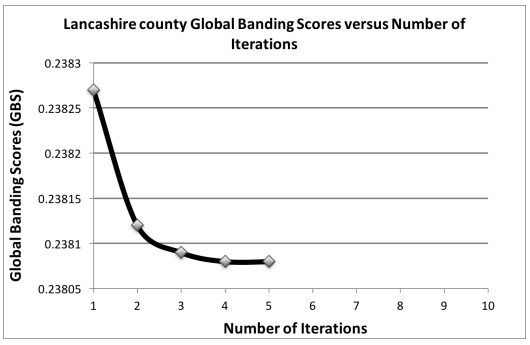
the behaviour using Manhattan MD-EBPM, while in Figure D.3, graphs (a), (b), (c) and (d) shows the behaviour with respect to the four identified counties (Aberdeenshire, Cornwall, Lancashire and Norfolk using the MD-ABPM algorithms From the graphs, the GBS values improved (tend towards zero) as the MD-BPM algorithms progresses. Closer inspection indicates that significance improvements were made after the first few iterations. This results corroborates the results presented previously in Chapter 4 indicating that the MD-BPM algorithms operate in a similar manner.



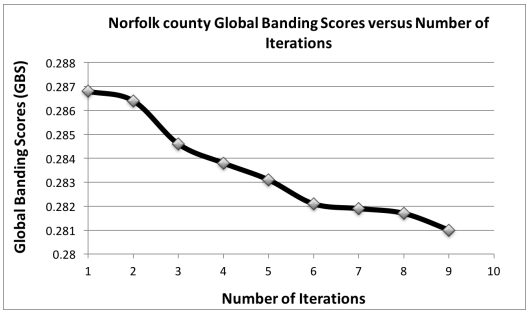
(a)



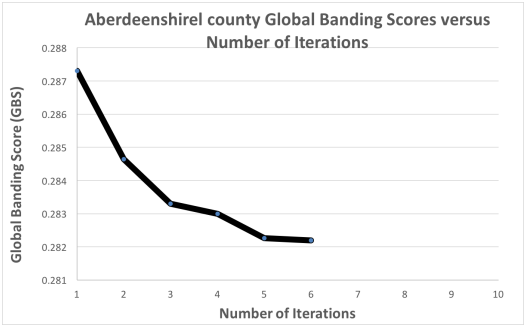
(b)



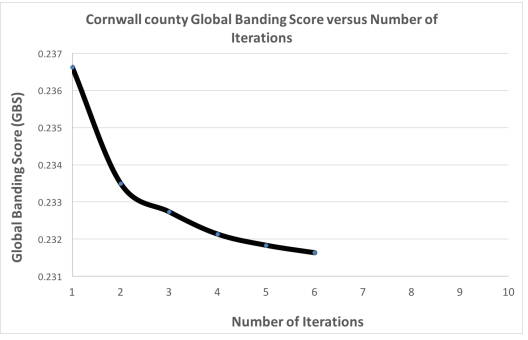
(c)



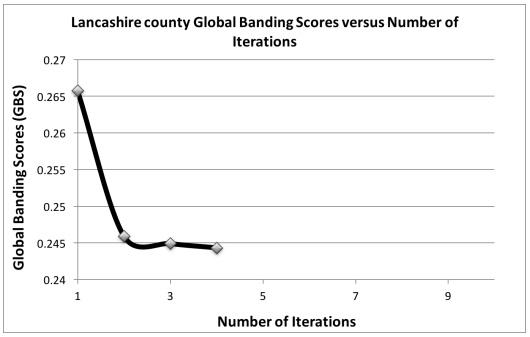
(d)



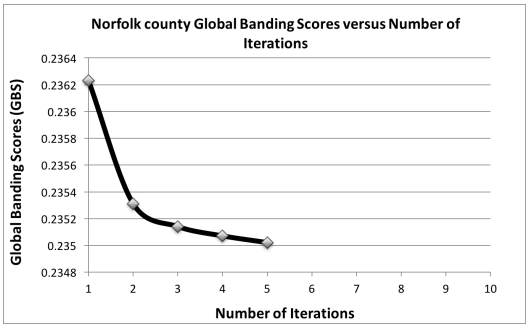
(e)



(f)

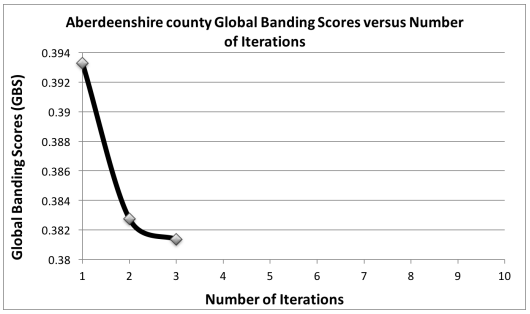


(g)

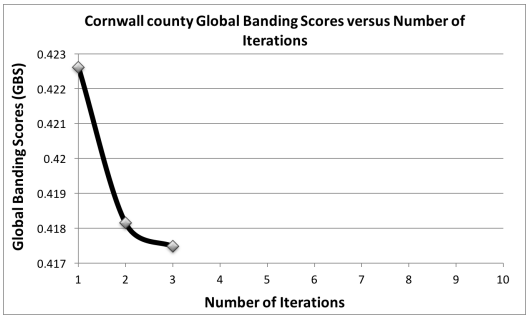


(h)

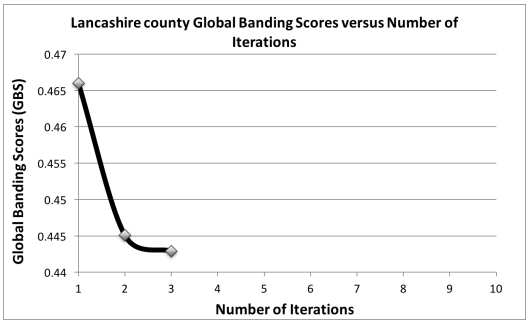
FIGURE D.2: GBS values versus the number of iterations using the Euclidean MD-EBPM algorithm for the four counties: (a) Aberdeenshire, (b) Cornwall, (c) Lancashire and (d) Norfolk, and Manhattan MD-EBPM Algorithm for: (e) Aberdeenshire, (f) Cornwall, (g) Lancashire and (h) Norfolk.



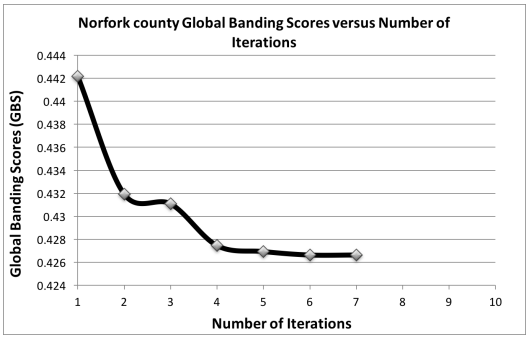
(a)



(b)



(c)



(d)

FIGURE D.3: GBS values versus the number of iterations using the MD-ABPM algorithm for the four counties: (a) Aberdeenshire, (b) Cornwall, (c) Lancashire and (d) Norfolk.