STATISTICAL STRATEGIES FOR SCALING AND WEIGHTING VARIABLES FOR CLUSTER ANALYSIS

BY SRINIVAS P. MALOOR

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

David Madigan

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE DISSERTATION

Statistical Strategies for Scaling and Weighting Variables for Cluster Analysis

by Srinivas P. Maloor

Dissertation Director: David Madigan

Cluster analysis (CA) is a generic name for an array of quantitative methods, the applications of which are found in numerous fields ranging from astronomy and biology to finance and psychology. Though the intuitive idea of clustering is clear enough, the details of actually carrying out such an analysis entail many unresolved conceptual problems. Multivariate data, often poses a problem, in that the variables are not commensurate. Since the outcome of a CA is sensitive to the scales of measurement of the input data, many practitioners resort to standardizing the data prior to the analysis. Hence, the *scaling* of such multivariate data prior to CA is important as a preprocessing step. *Autoscaling*, is one such naïve approach. Although it is a widely used procedure to standardize variables in some major *point and click* statistical software packages, it ignores the inherent cluster structure and actually proves counterproductive.

This dissertation is broadly divided into two parts - Univariate and Multivariate strategies. The first part addresses some univariate scaling and weighting approaches. In an attempt to put all variables *on the same footing*, we propose some intuitive strategies which we call *equalizers*. In addition, we consider letting the data suggest weights or *highlighters* that emphasize those variables with most promise for revealing the latent cluster structure. The methods vary in degree of complexity from simple weights

based on order statistics to more complicated iterative ones. The results indicate that, in a variety of chosen simulated data as well as real data sets, the new methods are much better than the most popular method, *autoscaling*. Although these strategies are computationally appealing, they are at best suboptimal in their ability to unearth the latent clusters embedded in the multivariate structure of the data. Hence, the next part of this dissertation is devoted to multivariate scaling and weighting approaches. We perform a systematic study of the characteristics of a multivariate equalizer in both the null-cluster scenario and for a variety of cluster structures. In addition, we present a multivariate approach to perform variable highlighting that is validated by results from many simulated data sets as well as some real data sets. Taken together, our results indicate that simple and intuitive strategies to preprocess data sets render them amenable to superior cluster recovery.

Acknowledgements

As the old saying goes, "You are judged by the company you keep". By that qualitative measure, I believe that I am extremely blessed. Indeed, most graduate students consider themselves fortunate to have a likeable, insightful, and dedicated thesis advisor; I have had the good fortune of having three! This acknowledgement, therefore, begins with my deepest gratitude to a distinguished trio of advisors: Professors David Madigan, Jon Kettenring and Ram Gnanadesikan. In knowing these three individuals, I have been deeply inspired by their honesty, intelligence, and grace.

I would like to thank David for teaching his "Data Mining" course at the Statistics department, which kindled in me a love for research and motivated me to pursue a doctorate. His intelligence, drive and passion for the subject have been constant sources of inspiration. His kind, compassionate and extremely amenable personality has made all my interactions with him great fun and extremely motivating. Indeed, his innate ability to "demystify" any complex concept is unique. I will always be grateful to him for being a great teacher, guide, and a friend, and last but not the least, for introducing me to Ram and Jon.

It would be difficult to express in words the contribution of Ram and Jon to this work as well as the profundity of the impact they have had on my life; perhaps they might not realize it themselves. If there is one thing that I regret, it is that I did not meet them earlier in my life! As wise, gentle, caring and modest mentors and teachers, I could not have asked for anything better. I will never forget the summer of 2005, when I first met Ram and Jon, to work on a project that eventually became my first journal paper. I thank them for the many insightful conversations during the development of the ideas in this dissertation, and for their invaluable comments on the text. They have taught me to write well, present well, and most importantly, to develop intuition and do

good research. If I am able to achieve even a fraction of their stellar accomplishments, I would consider myself as having greatly succeeded in life. I owe a lot to them and I feel incredibly fortunate to have associated with them in my life.

Thanks to my committee members, Prof. Joe Wilder and Prof. Sophocles Orfanidis, who were willing to serve on my thesis committee despite their busy schedules, as well as members of the Statistics and ECE graduate faculty. Special thanks to Prof. Narindra Puri, Prof. Zoran Gajic and to Prof. Orfanidis for their belief and confidence in me and for providing timely advice and support when I needed it the most, and to Prof. Wilder for providing me with data for my thesis. Thanks to Drs. Dan Wartenberg, William Hallman and Howard Kipen for providing me with financial support to pursue my doctoral degree. I have always been in awe of Dan's brilliance and out-of-the-box thinking, and, hopefully, I have perhaps been imbued with a little of it. I thank him for all his advice and support over the years, which has been vital in helping me get to this stage.

Thanks to past and present colleagues at Rutgers - Diwakar Kedlaya, Suresh Gopalakrishnan, Paul Gong, Sherwin Montaño, Julio DaGraca, Hector Caban, Howard Bondell, Gerry Harris and Sujit Nair for making my stay at Rutgers an enjoyable and a memorable one. Special thanks to Dolores Rivera for always being ready to feed an ever-hungry graduate student with sandwiches and coffee!

Finally, I would like to end my acknowledgement where my happiness begins: with my family and friends. Their presence helped make the completion of my graduate work possible – my parents, my brother, and my friend Niyati, whose constant encouragement and support I have relied on throughout my stay at Rutgers. I would like to thank my parents for creating an environment in which following this path seemed so natural. Thanks to Niyati for being there for me through all the ups and downs in my graduate school life. Thanks also to my brother – having him here in the US over the past one year has been a pleasure. Many thanks to them for their unfailing encouragement, patience and love. Most of all, I sincerely thank God for giving me this wonderful opportunity as well as the strength to see it through to a successful conclusion.

Dedication

To Amma and Appa,

to my family and friends,

and to all my teachers, from kindergarten to Ph.D. and beyond

Table of Contents

Abstract ii

Acknowledgements iv

Dedication vi

List of Tables x

List of Figures xiv

1. Introduction 1

1.1. Cluster Analysis 1

1.2. Notation 2

1.3. Background and Motivation 3

1.4. Organization 9

2. Univariate Approach 11

2.1. Introduction 11

2.2. Univariate Scaling - "Equalizers" 12

2.3. Univariate Weighting - "Highlighters" 15

2.4. Choice of "m_1" 17

2.5. Issue of "missing constants" 18

2.6. Experiments 20

2.6.1. Description of data sets 20

2.6.2. Results 25

3. Multivariate $\mathbf{W}^*_{(m_1)}$ Algorithm - Null Cluster Structure 30

3.1. Introduction 30

3.2. $\mathbf{W}^*_{(m_1)}$ Algorithm 31
Diagonal version of $\mathbf{W}^*_{(m_1)}$ - $\mathbf{W}^*_{d(m_1)}$ algorithm 32
3.3. Sensitivity to Starting Point 35
3.3.1. Experimental design 36
3.3.2. Results 38
3.4. Quality of the $\mathbf{W}^*_{(m_1)}$ estimates 40
3.4.1. Experimental design 40
3.4.2. Results 42
3.4.3. "Missing" constant 43
3.4.4. Variability of $\mathbf{W}^*_{(m_1)}$ estimates 45
3.5. Conclusion 47

4. Multivariate $\mathbf{W}^*_{(m_1)}$ Algorithm - Presence Of Clusters 66
4.1. Introduction 66
4.2. Sensitivity to Starting Point 66
4.2.1. Experimental design 67
4.2.2. Description of data sets 68
4.2.3. Results 73
4.3. Quality of $\mathbf{W}^*_{(m_1)}$ estimates 74
4.3.1. Experimental design 74
4.3.2. Results 79
4.3.3. "Missing" constant 83
4.3.4. Variability of $\mathbf{W}^*_{(m_1)}$ estimates 85
4.3.5. Experiments using gaussian mixture model-based clustering – (MCLUST) 86
4.4. Conclusion 87

5. Multivariate Highlighters : Discriminant Analysis-Based Weighting 120
5.1. Introduction 120
5.2. DA and Pseudo-DA for Multivariate Highlighting 120

5.2.1. Eigenanalysis of $\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$ - Classical DA 122
5.2.2. Eigenanalysis of $\left[\mathbf{W}^{*}_{(m_1)}\right]^{-1}\left[\mathbf{B}^{*}_{(m_2)}\right]$ - Pseudo DA 124
Choice of "m_2" 125
5.3. Experimental design 126
5.4. Results 128
5.4.1. Experiments using gaussian mixture model-based clustering – (MCLUST) 131
5.5. Conclusion 132

6. Conclusions and Further Work 146

References 149

Vita 155

List of Tables

2.1. Univariate Scaling Strategies 15

2.2. Univariate Weighting Strategies 18

2.3. Data sets used in this study 24

2.4. Errors of misclassification (mismatches) across data sets 26

2.5. Scaling and Weighting factors (Diag of $\mathbf{M}$) - Normalized to sum to one . 27

3.1. Summary of Figures used to study sensitivity to starting point 38

3.2. Data sets with the different (n, p) combinations used in this study . . . 41

3.3. Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0$ 62

3.4. Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0$ 62

3.5. Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.5$ 62

3.6. Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.5$ 62

3.7. Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.9$ 62

3.8. Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.9$ 63

3.9. Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.95$ 63

3.10. Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.95$ 63

3.11. Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.99$ 63

3.12. Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.99$ 63

3.13. Variance of normalized eigenvalues of $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\text{avg } \mathbf{W}^*_{(\mathrm{m}_1)}\right]$, $\rho = 0$. 64

3.14. Variance of normalized eigenvalues of $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\text{avg } \mathbf{W}^*_{(\mathrm{m}_1)}\right]$, $\rho = 0.5$ 64

3.15. Variance of normalized eigenvalues of $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\text{avg } \mathbf{W}^*_{(\mathrm{m}_1)}\right]$, $\rho = 0.9$ 64

3.16. Least squares approximation of the unknown constant, $\rho = 0$ 64

3.17. Least squares approximation of the unknown constant, $\rho = 0.5$ 64

3.18. Least squares approximation of the unknown constant, $\rho = 0.9$ 64

3.19. $\mathcal{V}1$ variability ratios, $\rho = 0$ 64

3.20. $\mathcal{V}1$ variability ratios, $\rho = 0.5$ 65
3.21. $\mathcal{V}1$ variability ratios, $\rho = 0.9$ 65
3.22. $\mathcal{V}2$ variability ratios, $\rho = 0$ 65
3.23. $\mathcal{V}2$ variability ratios, $\rho = 0.5$ 65
3.24. $\mathcal{V}2$ variability ratios, $\rho = 0.9$ 65
4.1. Data sets with the different (n, p) combinations used in this study . . . 75
4.2. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices 110
4.3. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices 110
4.4. M.S.E between eigenvalues of $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices 110
4.5. Comparison of errors of misclassification after $\mathbf{W}$ and $\mathbf{W}^*_{(m_1)}$ scaling . . 111
4.6. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0$ 111
4.7. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0$ 111
4.8. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0.5$ 111
4.9. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0.5$ 112
4.10. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0.99$ 112
4.11. M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0.99$ 112
4.12. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0$ 112
4.13. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0$ 112
4.14. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0.5$ 113

4.15. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0.5$ 113
4.16. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0.99$ 113
4.17. Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0.99$ 113
4.18. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0$ 113
4.19. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0$ 114
4.20. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0.5$ 114
4.21. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0.5$ 114
4.22. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0.99$ 114
4.23. M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0.99$ . . . 114
4.24. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0$ 114
4.25. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0$ . . . 115
4.26. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0.5$ 115
4.27. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0.5$ . . 115
4.28. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0.99$ 115
4.29. Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0.99$ . . 116
4.30. Variance of normalized eigenvalues of $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\mathbf{W}^*_{(m_1)}\right]$ 117
4.31. Least squares approximation of the unknown constant 118
4.32. $\mathcal{V}1_{(1,2)}$ variability ratios 119
5.1. Comparison of number of significant CRIMCOORDS and pseudo-CRIMCOORDS retained 138

5.2. Comparison of errors of misclassification using different highlighter strategies 139
5.3. Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0$ 140
5.4. Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0$ 141
5.5. Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0.5$ 142
5.6. Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0.5$ 143
5.7. Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0.99$ 144
5.8. Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0.99$ 145

List of Figures

1.1. Clusters obtained using age and height measurements of people 5

1.2. Dendrogram using raw (unscaled) data 6

1.3. Dendrogram using *autoscaled* data 6

1.4. Elliptical clusters with cluster centroids after k-means clustering 8

1.5. "Transformed" data with cluster centroids after k-means clustering . . . 9

2.1. Cluster structure for $\mathcal{D}1$ in the space of structure variables 21

2.2. Presence of noise in $\mathcal{D}1$ 21

2.3. Scatter plot for data set $\mathcal{D}2$ (in the space of two structure variables) . . 22

2.4. Scatter plot for data set $\mathcal{D}3$ (in the space of two structure variables) . . 23

2.5. Scatter plots for data set $\mathcal{D}4$ 23

2.6. Scatter plots for data set $\mathcal{D}5$ 24

3.1. Convergence of $\mathbf{W}^*_{(m_1)}$-algorithm for a $\mathbf{N}(\mathbf{0}, \mathbf{I})$ sample 33

3.2. $\mathbf{W}^{*(t)}_{(m_1)}$ computations with each iteration, t 34

3.3. Comparison of diagonal elements - diagonal starting point - $\rho = 0$ . . . 48

3.4. Comparison of diagonal elements - SPD starting point - $\rho = 0$ 49

3.5. Comparison of diagonal elements - diagonal starting point - $\rho = 0.99$ . . 50

3.6. Comparison of diagonal elements - SPD starting point - $\rho = 0.99$ 51

3.7. Comparison of off-diagonal elements - diagonal starting point - $\rho = 0$ . . 52

3.8. Comparison of off-diagonal elements - SPD starting point - $\rho = 0$ 53

3.9. Comparison of off-diagonal elements - diagonal starting point - $\rho = 0.99$ 54

3.10. Comparison of off-diagonal elements - SPD starting point - $\rho = 0.99$ . . 55

3.11. Mean order statistics of off-diagonal elements-SPD starting point-ρ=0.99 56

3.12. Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$ and (average) $\mathbf{W}$ matrices, ρ=0 57

3.13. Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.5 57
3.14. Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.9 58
3.15. Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.95 58
3.16. Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.99 59
3.17. M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices,ρ=0 59
3.18. M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.5 60
3.19. M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.9 60
3.20. M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.95 61
3.21. M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.99 61
4.1. Cluster structure for $\mathcal{D}1$ 88
4.2. Cluster structure for $\mathcal{D}2$ 88
4.3. Cluster structure for $\mathcal{D}3$ 89
4.4. Cluster structure for $\mathcal{D}4$ 89
4.5. Cluster structure for $\mathcal{D}5$ 90
4.6. Cluster structure for $\mathcal{D}6$ 90
4.7. Cluster structure for $\mathcal{D}7$ 91
4.8. Cluster structure for $\mathcal{D}8$ 91
4.9. Cluster structure for $\mathcal{D}9$ 92
4.10. Cluster structure for $\mathcal{D}10$ 92
4.11. Cluster structure for $\mathcal{D}11$ 93

4.12. Cluster structure for $\mathcal{D}12$ 93
4.13. Cluster structure for $\mathcal{D}13$ 94
4.14. Cluster structure for $\mathcal{D}14$ 94
4.15. Cluster structure for *Iris* ($\mathcal{D}15$) data set 95
4.16. Cluster structure for *Crabs* ($\mathcal{D}16$) data set 95
4.17. *Wine* ($\mathcal{D}17$) data set scatter plot 1 96
4.18. *Wine* ($\mathcal{D}17$) data set scatter plot 2 96
4.19. *Wine* ($\mathcal{D}17$) data set scatter plot 3 97
4.20. *Cells* ($\mathcal{D}18$) data set scatter plot 1 97
4.21. *Cells* ($\mathcal{D}18$) data set scatter plot 2 98
4.22. *Cells* ($\mathcal{D}18$) data set scatter plot 3 98
4.23. *Cells* ($\mathcal{D}18$) data set scatter plot 4 99
4.24. Comparison of diagonal elements - SPD starting point 100
4.25. Comparison of diagonal elements - SPD starting point 101
4.26. Comparison of diagonal elements - SPD starting point 102
4.27. Comparison of diagonal elements - SPD starting point 103
4.28. Comparison of diagonal elements - SPD starting point 104
4.29. Comparison of off-diagonal elements - SPD starting point 104
4.30. Comparison of off-diagonal elements - SPD starting point 105
4.31. Comparison of off-diagonal elements - SPD starting point 106
4.32. Comparison of off-diagonal elements - SPD starting point 107
4.33. Comparison of off-diagonal elements - SPD starting point 108
4.34. Sample realization when clusters are well separated - from a simulated data set in group 2 109
4.35. Sample realization when clusters touch each other - from a simulated data set in group 2 109
5.1. Illustration of three groups and a single discriminant function 121
5.2. Sample *scree* plot 123
5.3. Dendrogram for data set $\mathcal{D}4$; using pseudo-DA with $m_2 = m_1$ 133

5.4. Dendrogram for data set $\mathcal{D}4$; using pseudo-DA with $m_2 = f$ 133

5.5. Scatter plot of *Iris* data in the space of the first two pseudo-CRIMCOORDS; $m_2 = m_1$ 134

5.6. Dendrogram for *Iris* data; using pseudo-DA with $m_2 = m_1$ 134

5.7. Scatter plot of *Crabs* data in the space of the first two pseudo-CRIMCOORDS; $m_2 = f_1$ 135

5.8. Dendrogram for *Crabs* data; using pseudo-DA with $m_2 = f_1$ 135

5.9. Scatter plot of *Crabs* data in the space of the first two CRIMCOORDS, from classical DA 136

5.10. Dendrogram for *Crabs* data; using classical DA 136

5.11. *Scree* plot for *Iris* data; $m_2 = m_1$ 137

5.12. *Scree* plot for *Crabs* data; $m_2 = m_1$ 137

5.13. Sample *scree* plot for a simulated data set ($p = 50$) in group 2 138

Chapter 1

Introduction

"Far better an approximate answer to the right question, which is often vague, than the exact answer to the wrong question, which can always be made precise" – John Wilder Tukey[1]

1.1 Cluster Analysis

The innate ability to form meaningful groups of objects is one of the most elementary modes of human intelligence. Understanding our complex world requires perceiving the similarities and differences between the entities that compose it. We do this subconsciously in our everyday lives with remarkable ease. In relatively early stages of the human cognitive process, one learns to distinguish, for example, between shapes of letters of the alphabet or between different colors. However (as also noted in [87]), enabling a computer to perform this deceptively simple task of automatically forming natural groupings is a hard and often ill-posed problem.

Cluster Analysis (CA) is the general logic, formulated as a procedure, for exploring the latent structure of data. It is an important technique in the rapidly growing field known as Data Mining and is being applied to a variety of scientific disciplines such as biology, physics, medicine, marketing, computer vision and remote sensing. The development of clustering methodology has been truly interdisciplinary. As an approach for analyzing multivariate data, it is widely used, and at an accelerated pace, in recent years. (See [57].) The goal of CA is to group objects into clusters such that objects within the same cluster are similar in some sense, and objects from different clusters are

[1] In "The future of data analysis". Annals of Mathematical Statistics 33(1), pp. 1 – 67

dissimilar. These objects could be described in terms of measurements (e.g., attributes, features) or by relationships with other objects (e.g., pairwise distances or similarity measures).

Discriminant analysis (DA) has been in the mainstream of the theory and practice of multivariate analysis for a long time. The key difference between DA and CA situations is that, in the former setup, one has random samples of data from known groups in hand and is interested in characterizing differences among the known groups (for *classification*) while, in the latter case, all one has is a set of multivariate observations and is interested in finding both the groups that might exist and an assignment of the individual observations to these groups. Hence, in CA situations, the groups and their members, and thus the numbers in the different groups, are all unknown. Thus, unlike *Classification*, which belongs to methods in *Supervised Learning*, CA is a form of *Unsupervised learning* in the Computer Science literature.

1.2 Notation

This dissertation contains many mathematical expressions and equations. As the context and convention permits, small Roman and Greek italicized letters are used for scalars, small Roman boldface letters for vectors and capital Roman boldface letters are used for matrices. $\| \bullet \|$ is the L_2 norm. The subscript d denotes either a diagonal matrix (as in upper case $\mathbf{W}_d^*$ or $\mathbf{B}_d^*$) or a scalar derived from a corresponding diagonal matrix (as in lower case $\mathbf{b}_d^*$). Bracketed subscripts such as (m_1), (m_2) or (f) denote the number of pairs of data points that were used in computing the corresponding measure. Specifically, (m_1) is used to represent the number of "closest neighbors", while the subscript (m_2) is used to represent the number of "farthest neighbors". For example, $\mathbf{B}_{d(m_2)}^*$ denotes that the diagonal matrix $\mathbf{B}_d^*$ was computed using (m_2) farthest-apart pairs of data points. Furthermore, the subscript (f) has a special definition; $f = \binom{n}{2} - m_1$. Hence, the measure $\mathbf{B}_{(f)}^*$ denotes that the matrix was computed using all but the m_1 closest pairs of data points, where the total number of point pairs equals $\binom{n}{2}$.

1.3 Background and Motivation

The process of CA (also see, [33]) could be broadly divided into three stages:

1. The input stage
2. The algorithm stage
3. The output stage

Of these three stages, the second stage, concerning the various algorithms for clustering, seems to have received the lion's share of attention in the literature. As noted in [33], there has been a deluge of "methods" for clustering, but very little is actually known about their relative statistical behavior. Among the methods of CA available currently, there are two broad categories of them (*Hierarchical* and *Partitioning*) and a plethora of specific algorithms within each category. (See, for example, [54].) Given a user-input choice for the number of clusters, both these methods produce non-overlapping clusters. However, they also differ in a variety of aspects. *Hierarchical* clustering algorithms can proceed *divisively* (top-down) or *agglomeratively* (bottom-up) (also, see [87]). In the top-down approach, one starts with all the objects as belonging to one cluster at the begining of the procedure and then this encompassing cluster is cut into successively smaller chunks. Conversely, in agglomerative algorithms, each object starts as a singleton cluster and the "clusters" are merged successively. Hierarchical algorithms output a rooted tree structure that can be represented as a dendrogram. In the case of *Partitioning* methods (or, flat clustering), *hard* clustering induces a partitioning into non-overlapping groups. In contrast, a *soft* clustering provides the probabilities for each object being a member of a cluster. Depending on the result of clustering, one can distinguish between *hierarchical* and *partitioning* approaches. In the literature search described in [57], the most often used techniques seem to be hierarchical ones. The type of method also seems to depend on the subject matter area of the application. The work of Fraley and Raftery (see [24], [25] and [26]) on model-based clustering provides an interesting framework for unifying many methods, and also includes approaches to inferences about the number of clusters, and software for

model-based clustering (MCLUST).

Despite the wide prevalence of CA as a tool for analyzing multivariate data, there are pitfalls in the methods used in many of the applications. To mention only one class of causes for such pitfalls, the ignoring of an inevitable "circularity" involved in CA situations is a feature of many of the widely used methods (also see [29]). Consequences of not taking the circularity into account can range from lack of effective discovery of the groups to even misleading conclusions. This is illustrated in Figures 1.2 and 1.3 respectively (there is more on this on page 7).

There are many types of data that are amenable to a CA. In this thesis, the data are presumed to be continuous multivariate data. With one important exception, the model underlying the data follows the usual assumptions of g-group DA or equivalently, one-way multivariate analysis of variance (MANOVA). The exception is that neither the number of groups nor the group identities of the observations are known in CA. Note that explicit use of multivariate normality is not actually necessary, but rather that the data be viewed as reasonably homogenous ellipsoidal point clouds primarily differing in location in a multivariate Euclidean space.

An important and hard-to-resolve issue, concerns the appropriate scaling or weighting of the variables prior to a CA (see, for example, [12] and [11].) Choices made at this stage would have an influence on the subsequent indicators of closeness among the objects to be clustered and as a result change the output of the CA. Most clustering methods form clusters based on the proximity between data points in the multi-dimensional space. A commonly used measure of proximity between a pair of data points (objects) is the *squared Euclidean Distance*. If we define the input data matrix ($\mathbf{Y}$) as a $p \times n$ matrix (each $\mathbf{y}_i$ being a p-length vector, $i = 1, 2 \ldots n$),then the objects ($\mathbf{y_i}$) could be viewed as n points scattered in a p-dimensional Euclidean space. In such a scenario, a popular criterion for clustering is to minimize the error sum of squares (ESS), or the sum of squared Euclidean distances between the objects of a cluster and its centroid as given by:

$$\sum_{j=1}^{g} \sum_{\mathbf{y}_i \in C_j} \|\mathbf{y}_i - \mathbf{m}_j\|^2, \tag{1.1}$$

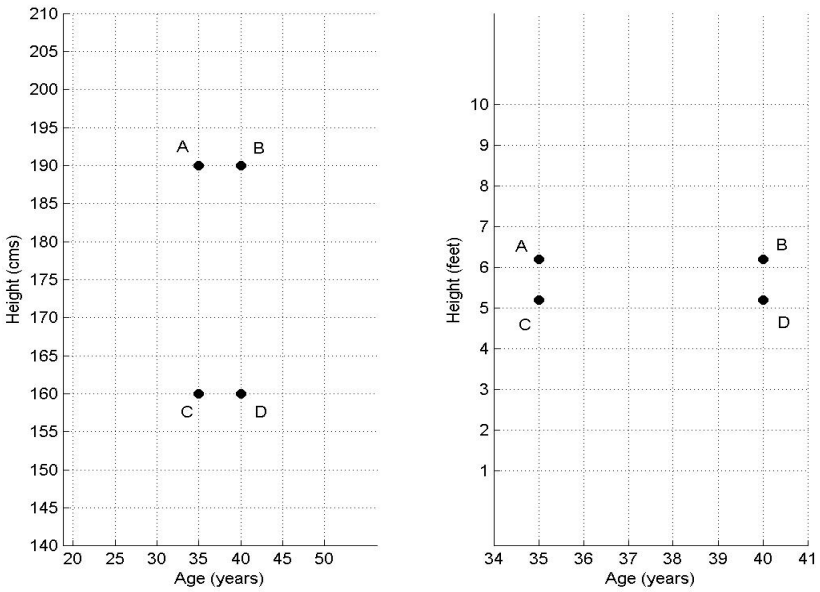

Figure 1.1: Clusters obtained using age and height measurements of people

where $\mathbf{m}_j$ is the p-length mean vector of data points in cluster C_j and g is the number of clusters (also, see section 5.3 in chapter 2 of [54].) Two popular clustering methods, k-means and Ward's hierarchical clustering, attempt to minimize this criterion. A major drawback of these and other traditional clustering methods is that the clustering results are sensitive to the units of measurement, that is, changing the units of measurement may lead to a different clustering result, as shown in Figure 1.1 (also see section 2.1 in chapter 2 of [54].)

The plot on the left of Figure 1.1 shows two natural clusters {A, B} and {C, D}. But when the variable "height" is expressed in feet, the obvious clusters are now {A, C} and {B, D}. This partition is completely different from the first. A commonly used naive-approach to fix this problem is to normalize the data to unit variance on each variable before clustering. This is termed as *autoscaling*. The objective in using this method of scaling the data is to "put all the variables on an equal footing". However, using a standard deviation of all the observations on each variable, instead of an estimate of the "within-clusters" standard deviation, ignores the fact that the former

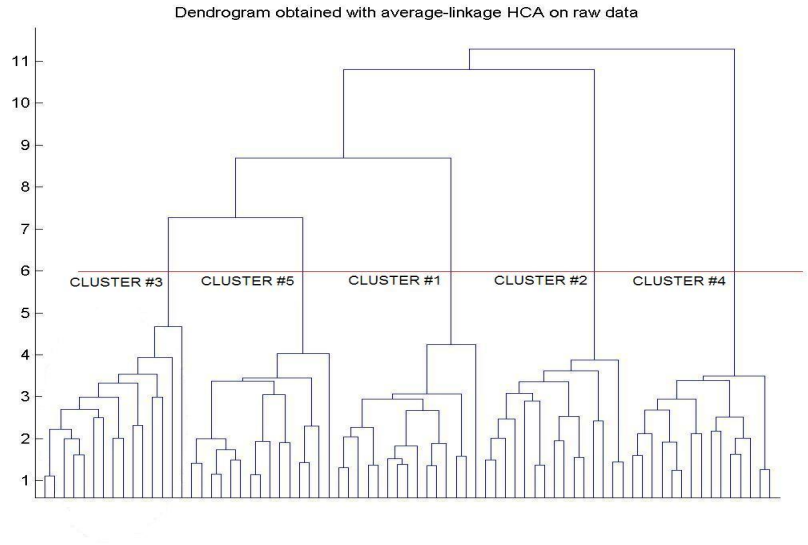

Figure 1.2: Dendrogram using raw (unscaled) data

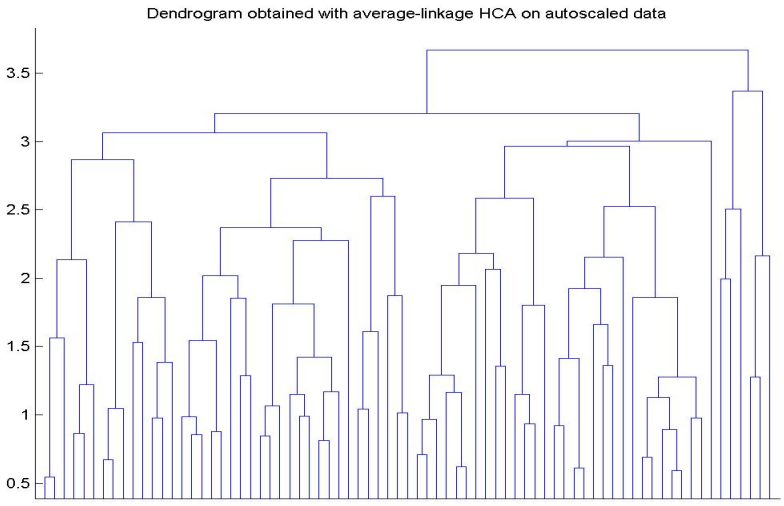

Figure 1.3: Dendrogram using *autoscaled* data

reflects both inter-cluster and within-cluster variation, thereby, potentially obscuring the latent cluster structure. As noted in [16], "such a normalization may be appropriate if the full data set arises from a single fundamental process (with noise), but inappropriate if there are several different processes." This is illustrated in Figures 1.2 and 1.3 respectively. Figure 1.2 displays the dendrogram obtained by a simple average-linkage hierarchical cluster analysis (HCA) on a raw (unscaled) five-cluster simulated data set (n=75 observations in p=5 dimensions with 5 spherical clusters of equal size). As shown in the dendrogram, a simple cut, at say, level 6 on the y-axis, would lead to perfect cluster recovery. Figure 1.3, however, shows the dendrogram obtained after *autoscaling* the same data set. Notice now that we are not able to cut the tree at any level to recover the original cluster structure!

Consequently, *autoscaling* is not only unsatisfactory conceptually, but may lead to misleading results due to its lack of sensitivity to clusters when they are present. Not knowing the clusters and the observations that belong to them ahead of the analysis is one reason for using the standard deviation of all the observations. This is thus an example of the circularity mentioned before.

Another technique, which shares the feature of circularity, is the use of a few leading Principal Components of the dispersion matrix of the entire data set, followed by a clustering of the data in the space of these Principal Components. The pitfalls of using Principal Components Analysis preliminary to a CA have been discussed and illustrated in the literature (see [9].) However, as in the case of *autoscaling*, the practice is still very common. A more appropriate approach would be to use DA and use the first few discriminant coordinates for a CA. Once again, it is not possible to carry out the more appropriate analysis directly because in the CA situation one does not know the groups ahead of time. However, it may be possible to find ways of incorporating the basic concept of DA, viz., weighting variables that contain information about the separations of the groups.

Apart from its sensitivity to scale, the Euclidean distance metric might be inappropriate for clustering data sets that have highly correlated variables. For instance, consider the scatter plot given in Figure 1.4. The five-dimensional data set with n=250

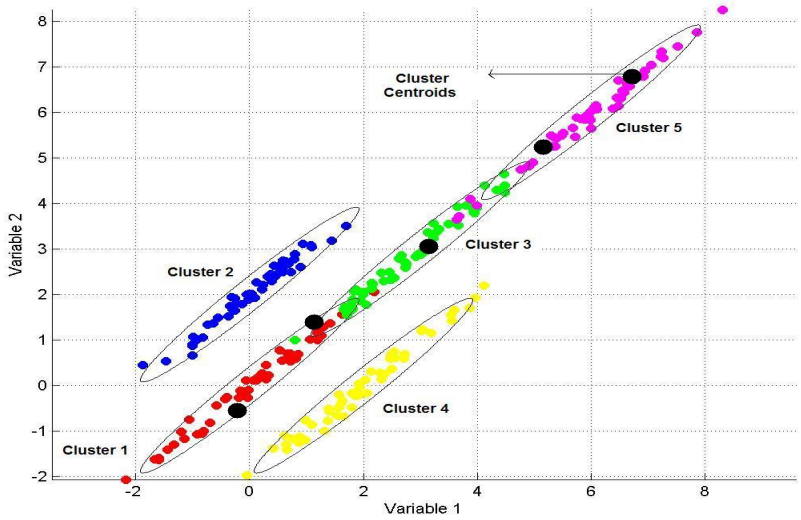

Figure 1.4: Elliptical clusters with cluster centroids after k-means clustering

observations, consists of five homogenous (of equal size and dispersion) elliptical clusters with high intra-cluster correlation ($\rho = 0.99$), in the space of the first two variables. The other three variables have no cluster structure. Each cluster is displayed using a unique color. Also, note that there is some degree of overlap among clusters 1, 3 and 5. If we were to use the simple k-means algorithm with the Euclidean distance metric (which is inherently biased to spherical clusters), we get about 117 - 142 errors of misclassification (computed by comparing the cluster memberships produced by the algorithm with the known cluster labels), depending on the initialization of the k-means algorithm (*Note*: The k-means algorithm requires the user to specify initial locations of the k cluster centroids. The algorithm used here initializes the centroids to k randomly chosen locations.) The final resulting positions of the cluster centroids after one such instance of the k-means initialization is also displayed in Figure 1.4 using boldface black dots. It is interesting to note here that clusters 1, 3 and 5 seem to dominate the analysis, being in closer Euclidean-proximity to all of the five cluster centroids, as shown.

For illustrative purposes, suppose we *sphericized* the elongated point clouds using the known pooled within-clusters covariance matrix ($\mathbf{W}$) of the data. If we then applied

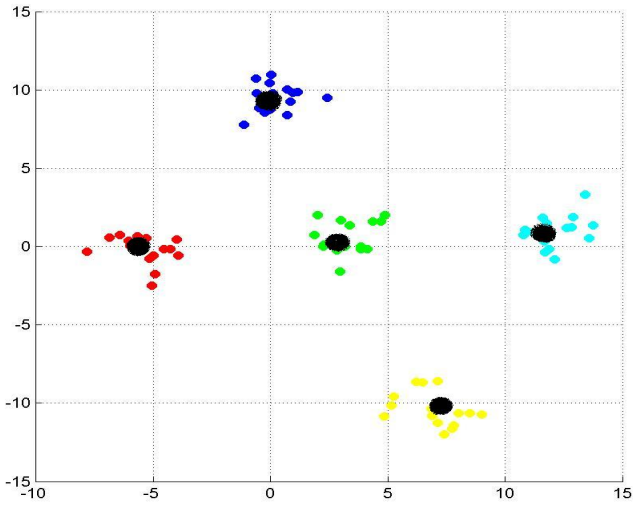

Figure 1.5: "Transformed" data with cluster centroids after k-means clustering

the k-means algorithm with the Euclidean distance metric in the transformed space (this is equivalent to using Mahalanobis distance in the original space), we obtain only 0 - 16 errors of misclassification (again, depending on the k-means initialization.) Figure 1.5 displays the clusters in the transformed space and the corresponding locations of the final cluster centroids obtained in the last k-means iteration. It is apparent here that the Euclidean distance metric is an ill-suited choice for clustering the given data. Hence, in some situations with high intra-cluster correlation, Mahalonobis distances might be better suited as a measure of proximity. Consequently, variable-weighting and scaling (univariate and multivariate) could also be thought of as finding alternative and more appropriate distance metrics for CA.

1.4 Organization

The variable scaling and weighting approaches discussed in this thesis are intuitive and recognize the possible existence of clusters without actually depending on a detailed knowledge about the clusters. The remainder of this work is organized as follows. Section 2.2 in chapter 2 focuses on and discusses some simple univariate, but potentially

more appropriate ways, for handling the scaling problems than presently widely used ones. Section 2.3 builds on section 2.2 by discussing methods for emphasizing variables that best separate the clusters. It is thus in the spirit of DA, which is a well-known method when the groups and their members are known *a priori*. In chapter 3, we study a multivariate approach to address the scaling problem. Though computationally more intensive than its univariate counterparts, this method will take into consideration the correlational structure of the data, which we lose, when using univariate methods. We systematically study what we call the multivariate $\mathbf{W}^*_{(m_1)}$ algorithm and characterize some of its features under the no-clusters scenario. Chapter 4 will extend the analyses carried out in chapter 3, but with data sets with varying cluster structures. Results from using a variety of simulated and some real data sets are presented. In chapter 5, we will introduce a multivariate scheme analogous to DA, but for the clustering context. We will study some of its characteristics and provide results using some simulated and real data. Discussion and potential direction for further research follow in chapter 6.

Chapter 2

Univariate Approach

2.1 Introduction

This chapter focuses on methods to scale and weight the variables, one variable at a time. Hence, the goal is to propose and study alternative methods that are intuitive and computationally simple, but more effective than *autoscaling*. This is based on work done in a joint paper [38].

Given a space for representing the objects, the prescription of a distance function or metric would then be the next step. One useful general class of squared distance functions is provided by a class of positive definite quadratic forms. Specifically, if d_{ij} is the inter-point distance between the p-dimensional point pairs $\mathbf{y}_i$ and $\mathbf{y}_j$, then their squares are all of the form

$$d_{ij}^2 = (\mathbf{y}_i - \mathbf{y}_j)' \, \mathbf{M} \, (\mathbf{y}_i - \mathbf{y}_j) \quad \forall \quad i < j = 1, 2, \ldots n, \tag{2.1}$$

where $\mathbf{M}$ $(p \times p)$ is a positive definite matrix to ensure that $d_{ij}^2 > 0$.

Each of the proposed univariate approaches would be determined by specifying a choice of a *diagonal* matrix for $\mathbf{M}$. Some data sets, where n and p are of the same order of magnitude might suffer from problems of multi-collinearity and ill-conditioned covariance structures thereby limiting the direct application of multivariate methods, without some sort of dimensionality reduction. However, an attractive feature of the univariate methods proposed in this chapter is that they do not require that the number of observations, n, be larger than the number of variables, p. This adds to their practical appeal.

2.2 Univariate Scaling - "Equalizers"

When variables are in different units of measurement, one would ideally want to standardize them in such a way as to put them on the same footing. Note that in equation (2.1), when $\mathbf{M} = \mathbf{I}$ (no scaling), one obtains the familiar Euclidean squared distance between all pairs of points. So as a first choice, we define,

$$\mathbf{M}_1 = \mathbf{I}.$$

It may still be an appropriate choice for certain classes of problems, but if standardizing variables is desirable then it is flawed for the purpose of CA as it pays no attention to the fact that changing the scales of measurement of the variables could lead to very different clusterings of the objects.

A second choice is *autoscaling*, as described before, and is specified as,

$$\mathbf{M}_2 = \mathbf{D}\left[\frac{1}{{s_i}^2}\right],$$

where $\mathbf{D}$ is a diagonal matrix with elements that are reciprocals of the usual total sample variances (${s_i}^2$) of each of the p variables.

Another popular scaling approach denoted $\mathbf{M}_3$, is to use the sample range instead of the standard deviation. This approach (which also ignores the cluster structure), is defined as,

$$\mathbf{M}_3 = \mathbf{D}\left[\frac{1}{{r_i}^2}\right],$$

where $\mathbf{D}$ is a diagonal matrix with elements that are reciprocals of squared sample ranges of each of the p variables. This approach was studied in [65].

The detrimental effects of scaling approaches based on the technique of equalizing variances without allowing for the possible presence of clusters is not limited to the presence of clusters in the data, but also aggravated by the presence of outliers. As such, the interquartile range (IQR) is one alternative to overcome this problem. Denoting $\mathbf{M}_4$ as IQR scaling, we have,

$$\mathbf{M}_4 = \mathbf{D}\left[\frac{1}{{q_i}^2}\right],$$

where $(q_i)^2$ represents the square of the interquartile range of each variable.

The technique of *autoscaling* as reported, involves standardizing all variables using their respective standard deviations. However, as stated earlier, variable scaling by the standard deviation ignores the fact that it involves both within-cluster and between-cluster variation, while what we would ideally want is to use only within-cluster variation. But not knowing the cluster structure in advance precludes us from doing this. Intuitively, although one may not know the clusters underlying the data, "nearest neighbors" are very likely to belong to the same cluster. This is the reasoning behind $\mathbf{M}_5$. To motivate this (also see [4]), we begin by giving below the standard one-way univariate ANOVA decomposition of total sums of squares of deviations into "within" and "between" components as:

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (y_{ij} - \overline{y})^2 = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (y_{ij} - \overline{y}_i)^2 \; + \; \sum_{i=1}^{g} n_i \, (\overline{y}_i \; - \overline{y})^2. \tag{2.2}$$

It is common practice to express this as:

$$t = w + b. \tag{2.3}$$

An alternative univariate decomposition could be made in terms of pairwise differences as:

$$\frac{1}{n} \sum_{i<j} (y_i - y_j)^2 = \frac{1}{n} \sum_{\substack{i<j \\ within}} (y_i - y_j)^2 + \frac{1}{n} \sum_{\substack{i<j \\ between}} (y_i - y_j)^2. \tag{2.4}$$

The first summation involves only within-group pairs while the second sum is over all between-group pairs. In short, this can be written as:

$$t = w^* + b^*. \tag{2.5}$$

Notice that the t's in equations (2.3) and (2.5) are algebraically equivalent. However, w^* and b^* are unknown since the cluster structure is unknown. Exploiting this "nearest neighbors" theme, we could try to obtain an approximate measure of the within-cluster variation, by basing it only on the closest pairs of observations. So we have,

$$\mathbf{M}_5=\mathbf{D}\left[\frac{1}{{s^*_{i(m_1)}}^2}\right],$$

where the quantity $({s^*_{i(m_1)}}^2)$ is defined as $\left(\frac{1}{n}\right)$ times sum of the m_1 smallest inter-point squared distances measured on each variable individually. So if the j^{th} observation on the i^{th} variable is denoted y_{ij}, then the absolute distance between the j^{th} and k^{th} observations on the i^{th} variable is,

$$d_i(j,k) = |y_{ij} - y_{ik}| \ \ \forall \ \ j < k = 1, \ldots n \ ; \ i = 1, \ldots p. \tag{2.6}$$

Then we have,

$${s^*_{i(m_1)}}^2 = \frac{1}{n} \sum_{1 \le j,k \le n} \delta_{jk}[d_i(j,k)^2]. \tag{2.7}$$

where

$$\delta_{jk} = \begin{cases} 1 & \text{if j}^{th} \text{ and k}^{th} \text{ observations are among the } m \text{ closest pairs} \\ 0 & \text{otherwise} \end{cases}$$

Note that the summation is taken only over the m_1 smallest pairwise distances (where m_1 is chosen conservatively small relative to the total number of within-cluster pairs, which is unknown *a priori*) which are likely to be ones associated with points belonging to the same cluster even though we do not know the clusters themselves (more on this in section 2.4). Also, the factor $(\frac{1}{n})$ in the above definition has no impact on the CA results.

Chapter 3 describes a full-fledged multivariate analogue of ${s^*_{(m_1)}}^2$. It is denoted $\mathbf{W}^*_{(m_1)}$ and is a measure of intra-cluster variability based on the m_1 nearest neighbors as determined by iteratively computed Mahalanobis inter-point distances in the p-dimensional space of all p variables (see section 3.2 of the thesis). A simpler version of the $\mathbf{W}^*_{(m_1)}$-algorithm which works in similar fashion but only computes and iterates on the diagonal entries of the matrix, leaving the others set to zero, leads to,

$$\mathbf{M}_6 = \left[\mathbf{W}^*_{d(m_1)}\right]^{-1},$$

where $\mathbf{W}^*_{d(m_1)}$ is the converged diagonal matrix. Thus, $\mathbf{M}_6$ yields a weighted Euclidean distance metric that is determined in a multivariate manner, but involving repeated

Notation	Scaling Strategies
$\mathbf{M}_1$	Euclidean distances (no scaling/weighting)
$\mathbf{M}_2$	Based on *autoscaling*
$\mathbf{M}_3$	Based on range scaling
$\mathbf{M}_4$	Based on IQR scaling
$\mathbf{M}_5$	Based on $s^*_{(m_1)}$ scaling
$\mathbf{M}_6$	Based on $\mathbf{W}^*_{d(m_1)}$ scaling

Table 2.1: Univariate Scaling Strategies

inversions of only a diagonal matrix. There is a more detailed description of the steps of this in section 3.2 of the thesis.

The univariate equalizer scaling strategies described so far are summarized in Table 2.1.

2.3 Univariate Weighting - "Highlighters"

To emphasize those variables that are important in revealing clusters, one could parallel the approach of DA in the context of clustering. One multivariate approach to this would be to develop a matrix, $\mathbf{B}^*$, paralleling the standard between-group sum of cross products matrix $\mathbf{B}$, as in DA. This is studied in chapter 5. But since our focus in this chapter is on computationally simpler univariate approaches, only methods for determining weights for the separate variables are considered.

In a sense, $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_4$, $\mathbf{M}_5$ and $\mathbf{M}_6$ are different methods to equalize the variables, *i.e.*, to standardize them so as to put them on the same footing. By contrast, we could let the data suggest weights that highlight those variables with most promise for revealing the latent cluster structure. A natural extension of the approaches to scaling considered earlier, would be to conceive of the farthest neighbors in the data as being more likely to belong to different clusters, although one does not know the detailed cluster structure underlying the data. Thus, basing estimates of the between-cluster spread on a subset of the largest inter-point distances is one strategy. In what follows, some intuitive univariate strategies are proposed to mimic a DA-like setup in the context of CA. As before, each method will be determined by specifying a choice of the positive definite matrix, $\mathbf{M}$.

In this vein, we define,

$$\mathbf{M}_7 = \mathbf{D}\left[\frac{b^*_{(f)}}{w^*_{(m_1)}}\right],$$

where $b^*_{(f)}$ is obtained by subtraction, using the identity among sum of squares "$t = b^* + w^*$", where t denotes the total sums of squares of pairwise differences and w^* denotes the within-clusters sums of squares. Using the earlier notation, $w^*_{(m_1)} = (s^*_{(m_1)})^2$. Then we have, $b^*_{(f)} = t - w^*_{(m_1)}$. In this setting, variables would tend to receive relatively large weights if their between-to-within sums of squares ratios are large. Since $s^*_{(m_1)}$ (and hence, $w^*_{(m1)}$) is based on only the m_1 closest neighbors $w^*_{(m_1)}$ would not be the same as w^*, as given in the identity, but would be a biased estimate of w^*. *Note*: In practice, $w^*_{(m1)}$ would be smaller than w^* as it is based on a smaller number of point-pairs.

Similarly, we could define,

$$\mathbf{M}_8 = \mathbf{D}\left[\frac{b^*_{(m_2)}}{w^*_{(m_1)}}\right],$$

where $b^*_{(m_2)}$ is computed in a manner similar to $w^*_{(m_1)}$, but is based on only the larger inter-point distances. If the clusters are reasonably homogenous in size (with $g \geq 2$), then the number of between-cluster point pairs will be greater than the number of within-cluster point pairs. As such, the number of largest pairwise distances, m_2 (corresponding to between-cluster point pairs) could be chosen to be greater than m_1, the number of smallest pairwise distances (corresponding to within-cluster point pairs). Hence, if m_1 of the smallest inter-point distances are used in computing $w^*_{(m_1)}$, $b^*_{(m_2)}$, analogous to $w^*_{(m_1)}$, could be computed as the sum of the m_2 ($m_2 > m_1$) largest inter-point squared distances measured on each variable. However, it may be noted that depending on the distribution of the n observations across the g clusters, in general, the number of between-cluster pairs of points could be greater than, equal to, or less than the number of within-cluster pairs. Hence, a prudent strategy here might be to study both cases, *i.e.*, $m_2 > m_1$ as well as, $m_2 < m_1$.

Also,

$$\mathbf{M}_9 = \mathbf{D}\left[\frac{b^*_{d(f)}}{w^*_{d(m_1)}}\right],$$

where $b^*_{d(f)}$ is an element of the diagonal matrix $\mathbf{B}^*_{d(f)} = \mathbf{T}_d - \mathbf{W}^*_{d(m_1)}$. $\mathbf{T}_d$ is a diagonal matrix whose elements are the total sums of squares for the p variables and $w^*_{d(m_1)}$ is the corresponding element of the diagonal matrix, $\mathbf{W}^*_{d(m_1)}$. Recall that we had defined $\left[\mathbf{W}^*_{d(m_1)}\right]^{-1}$ as $\mathbf{M}_6$ in the previous section.

Additionally, we have

$$\mathbf{M}_{10} = \mathbf{D}\left[\frac{b^*_{d(m_2)}}{w^*_{d(m_1)}}\right],$$

where $b^*_{d(m_2)}$ is the corresponding element of a diagonal matrix $\mathbf{B}^*_{d(m_2)}$, obtained using the m_2 farthest neighbors instead of the nearest neighbors, as measured by weighted Euclidean distances, weighted by $\left[\mathbf{W}^*_{d(m_1)}\right]^{-1}$.

A second category of *highlighters* consists of weights that derive from quantiles of the empirical distribution of the inter-point squared distances. In particular, we consider the ratio of averages of neighboring upper-tail quantiles to that of neighboring lower-tail quantiles. If $q(p)$ denotes the p^{th} quantile of the empirical distribution of the inter-point squared distances, then we could define the diagonal matrix $\mathbf{M}_{11}$ (ratio of quantiles) as,

$$\mathbf{M}_{11} = \mathbf{D}\left[\frac{(q(95) + q(92.5) + q(90))}{(q(5) + q(7.5) + q(10))}\right].$$

An attractive feature of $\mathbf{M}_{11}$ is its simplicity, both conceptually and computationally. It is important to note that the choice of the specific set of quantiles given above is intuitively appealing, and other similar sets of quantiles could be investigated.

Note: For the results reported for $\mathbf{M}_8$ and $\mathbf{M}_{10}$ in section 2.6.2, we have used $m_2 = m_1$, as we reported in [38]. The univariate weighting strategies described so far are summarized in Table 2.2.

2.4 Choice of "m_1"

The underlying intuition that nearest neighbors are likely to belong to the same cluster and farthest-apart neighbors to different clusters, requires us to essentially obtain a subset of the smallest and largest inter-point distances, to get at the closest m_1 or

Notation	Weighting Strategies
$\mathbf{M}_7$	Based on $\left[\frac{b^*_{(f)}}{w^*_{(m_1)}}\right]$ weighting
$\mathbf{M}_8$	Based on $\left[\frac{b^*_{(m_2)}}{w^*_{(m_1)}}\right]$ weighting
$\mathbf{M}_9$	Based on $\left[\frac{b^*_{d(f)}}{w^*_{d(m_1)}}\right]$ weighting
$\mathbf{M}_{10}$	Based on $\left[\frac{b^*_{d(m_2)}}{w^*_{d(m_1)}}\right]$ weighting
$\mathbf{M}_{11}$	Based on weights derived from averages of quantiles

Table 2.2: Univariate Weighting Strategies

farthest m_2 point pairs respectively. However, care needs to be taken to choose m_1 to be conservatively small to avoid contamination by between-cluster pairs or m_2 large enough to realistically reflect the between-cluster spread. In [34], Gnanadesikan *et al.* had developed a rule of thumb (in the multivariate context) for picking m_1 in cases where there is some idea of the number, g, of clusters to expect. The number of within-cluster pairs of points must be greater than or equal to $\left(\frac{n}{2}\right)\left(\frac{n}{g}-1\right)$, with equality, if and only if the groups are of equal size. They define $m_1 = \left(\frac{n}{3}\right)\left(\frac{n}{g}-1\right)$ as a conservative "$\frac{2}{3}^{rd}$ rule" for picking the number of closest pairs to work with "in order to guard against contamination by between-group pairs". For convenience, this rule could be applied in the cases where a choice of m_1 is needed. However, in practice it might be advisable to work with a range of m_1 values and compare results. As a related but different challenge, the highlighter weights $\mathbf{M}_8$ and $\mathbf{M}_{10}$ clearly require an appropriate choice of m_2, the number of farthest point pairs. This issue is open for further research and is discussed in chapter 6.

2.5 Issue of "missing constants"

Our goal for developing univariate "equalizers" is an attempt to essentially make the within-cluster variability the same across all variables. Conversely, with "highlighters", we are trying to intuitively develop weights that emphasize those variables that account for strong separation among clusters. Furthermore, since the highlighter weights involve

a ratio of between to within components, they would be invariant to scalar transformations of the variables. However, there are issues of missing constants lurking behind the proposed univariate equalizers and highlighters. The missing constants would tend to differ between variables that exhibit cluster structure and those that do not, as well as vary with the number and relative sizes of the clusters.

In particular, basing the measure of within-cluster variability on a conservatively small number of nearest neighbors (or likely within-cluster pairs) would yield estimates that in practice would be typically smaller than the *true* within-cluster variability. Hence, we might need to find appropriate constant multipliers to boost these measures to more realistically reflect the true within-cluster spread. On the other hand, our measure of between-cluster spread based on a fixed number of farthest-apart point pairs (or likely between-cluster pairs), could potentially be contaminated by a few within-cluster pairs. As such, we might again need suitable multiplicative factors (which are unknown), to make these estimates more nearly unbiased. Hence, not knowing the precise numbers and contents of the clusters, both the measures could be contaminated depending on the configurations of the clusters. The essence of our approaches have been to use the flexibility one has in choosing values of m_1 and m_2 to hopefully mitigate the effects of possible contaminations. By being "conservative" in choosing a small value of m_1 one could try to protect against *inflating* the measure of within-cluster variability by including too many inter-cluster pairs of points. Similarly, by choosing fewer of the farthest neighbors for a measure of inter-cluster variability, one could protect against *deflating* the measure of inter-cluster variability. Statistically, the concept in both choices is protecting against "bias" at the price of "efficiency". However, one of the objectives of this research is to investigate to what extent the complexities associated with these unknown constants can be completely ignored in executing an effective univariate weighting for CA.

2.6 Experiments

All the metrics suggested have been tested on five data sets to compare their relative effectiveness in recovering clusters. The first three are simulated ones, whose structures have been selected to incorporate specific interesting features. Two real data sets complete the set of examples. Real data can depart from the idealized assumptions underlying the methods described heretofore. Hence, they are valuable as test beds for the methods. The first real example used here is the well-known *Iris* data set from [2], which has been used as a test bed for methods of CA. The second is referred to as the *Crabs* data and pertains to five variables measured on samples of crabs from Australia. (See [8].) Brief descriptions of all the data sets are given in section 2.6.1. Results of using the methods discussed in sections 2.2 and 2.3 are discussed in section 2.6.2.

2.6.1 Description of data sets

The three simulated data sets, denoted $\mathcal{D}1, \mathcal{D}2$ and $\mathcal{D}3$, consist of $g = 5$ multivariate normal clusters each of size $n_i = 15$ in $p = 5$ dimensions. They are patterned after the test sets used in [40] which confined separation in cluster means to the first two dimensions, effectively leaving the other three as noise dimensions. Standard clustering techniques tend to be thrown off by (noise) variables that contain no cluster structure. The challenge is to thus select the structure variables or limit the noise variables through careful choice of "weights" in order to have a chance at satisfactory cluster recovery.

All of the simulated data sets were constructed from a single five-dimensional multivariate standard normal sample of 75 observations. In the case of $\mathcal{D}1$, the observations were shifted in groups of 15 along the first two (structure) coordinates by the following amounts: $(0, 0), (0, 10), (5, 5), (10, 0)$ and $(10, 10)$. Figure 2.1 displays a scatter plot for $\mathcal{D}1$, describing the cluster structure in the space of the structure variables and Figure 2.2 depicts the presence of noise in the space of the two noise dimensions. The five clusters are seen to be internally cohesive and well separated from each other on the first two variables.

For $\mathcal{D}2$, a transformation was applied to $\mathcal{D}1$ to induce a (population) correlation of

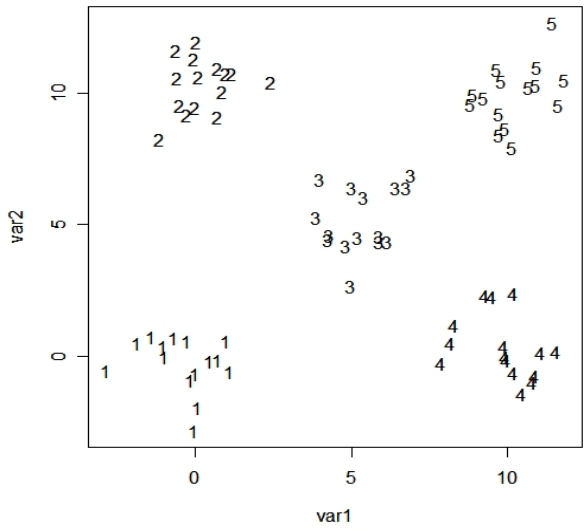

Figure 2.1: Cluster structure for $\mathcal{D}1$ in the space of structure variables

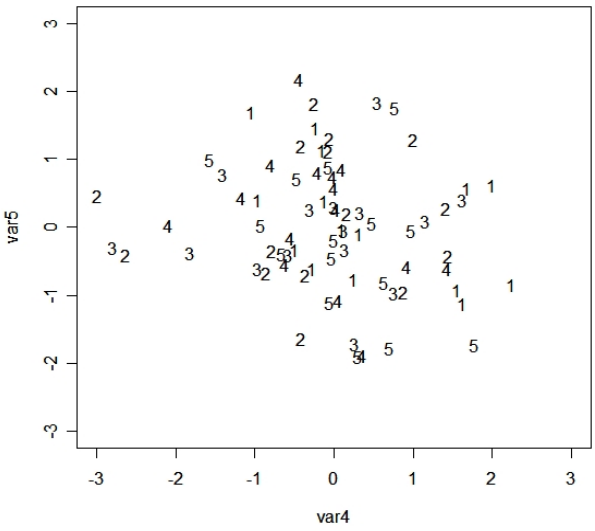

Figure 2.2: Presence of noise in $\mathcal{D}1$

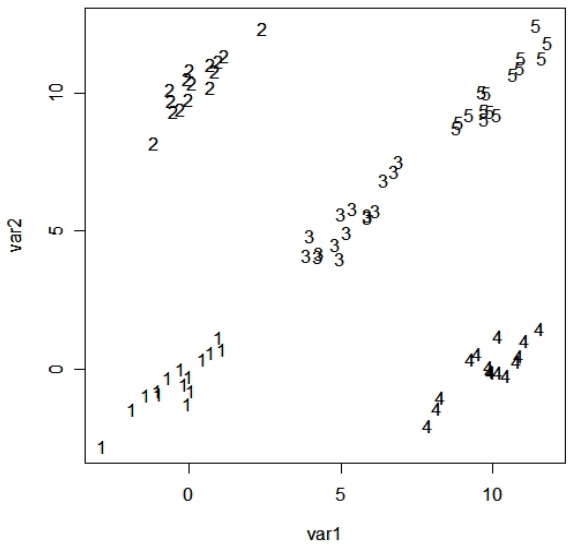

Figure 2.3: Scatter plot for data set $\mathcal{D}2$ (in the space of two structure variables)

0.9 between the first two variables. As in $\mathcal{D}1$, the data are cleanly separated into five clusters on the first two variables. Data set $\mathcal{D}3$ essentially has the same structure as $\mathcal{D}2$, but with the cluster centers of $\mathcal{D}2$ brought closer. The cluster means in $\mathcal{D}3$ measured along the first two coordinates are $(0,0), (0,7), (3,3), (7,0)$ and $(7,7)$ respectively. It is seen that clusters $1, 3$ and 5 have been pulled in much closer along the major axis of the within-cluster dispersion of the data. The difficulty that this feature causes for some of the methods is discussed further later. In the interest of parsimony, scatter plots of data sets $\mathcal{D}2$ and $\mathcal{D}3$ are displayed only in the space of the two structure variables in Figures 2.3 and 2.4 respectively.

The *Iris* data, denoted $\mathcal{D}4$, involves three groups of 50 observations each, measured on four variables. Each group represents a different type of iris. The variables are sepal length and width and petal length and width of the flowers. The latter two appear to be the more important ones for separating all three groups. This data set has been widely published and, hence, only two of the six scatter plots are displayed in Figure 2.5.

The *Crabs* data has 200 observations describing five morphological measurements

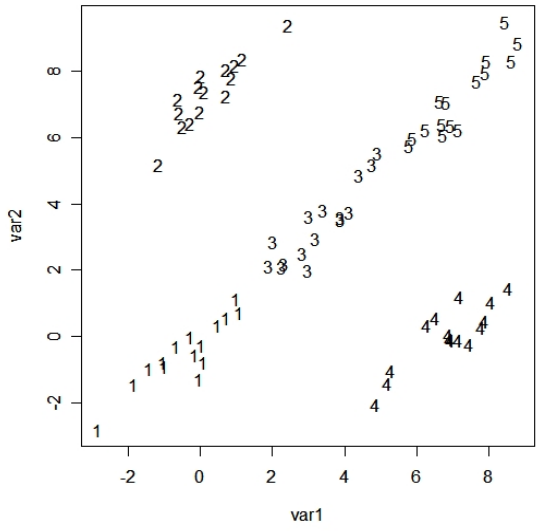

Figure 2.4: Scatter plot for data set $\mathcal{D}3$ (in the space of two structure variables)

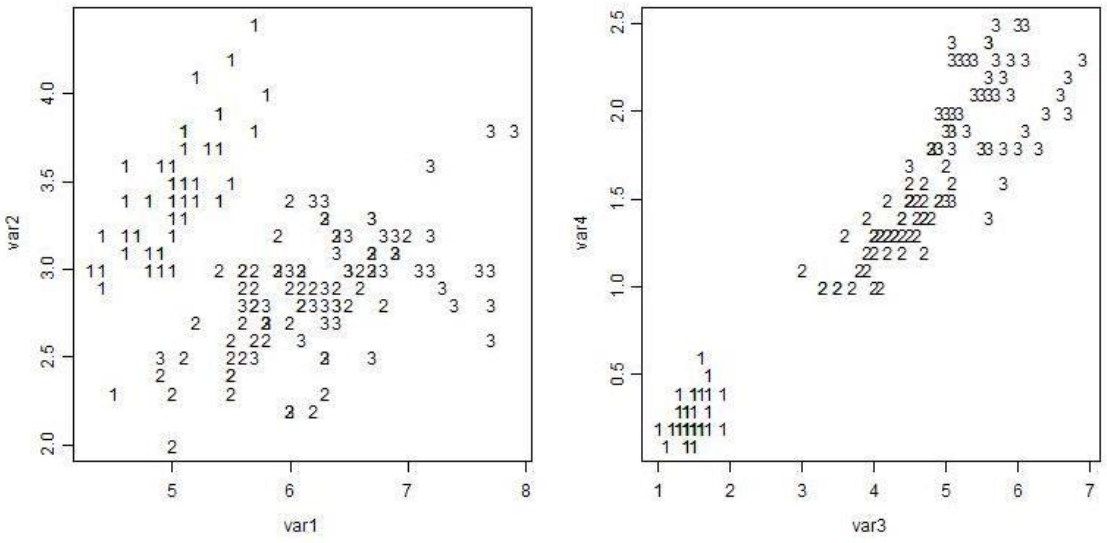

Figure 2.5: Scatter plots for data set $\mathcal{D}4$

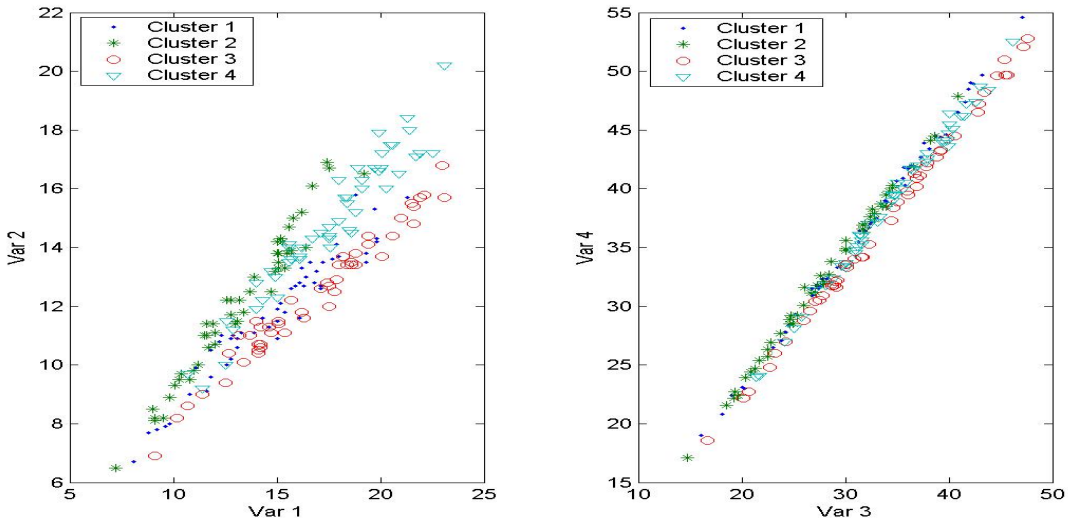

Figure 2.6: Scatter plots for data set $\mathcal{D}5$

Notation	g	n_i	p	Description
$\mathcal{D}1$	5	15	5	Spherical clusters in two dimensions
$\mathcal{D}2$	5	15	5	Elliptical clusters with high intra-cluster correlation
$\mathcal{D}3$	5	15	5	Elliptical clusters with cluster centers drawn closer
$\mathcal{D}4$	3	50	4	Anderson-Fisher *Iris* data
$\mathcal{D}5$	4	50	5	*Crabs* data

Table 2.3: Data sets used in this study

Note: The known groups are all of equal size and roughly or exactly homogeneous. These are ideal conditions under which a simple cut of the hierarchical cluster analysis tree based on average linkage could be used to exact a sensible partition.

(frontal lobe size, rear width, carapace length, carapace width and body depth) on 50 crabs, each of two color forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia. Each crab belongs to one of four groups based on their species (Blue or Orange) and sex (Male or Female) combination. This five-dimensional data set constitutes $\mathcal{D}5$. The scatter plots given in Figure 2.6 are a sample of the 10 possible ones. They exhibit a high level of intra-cluster correlation. The plots also display the close proximity of the clusters and the degree of overlap therein. Different symbols are used to depict the different clusters. A brief summary of the five data sets is provided in Table 2.3.

2.6.2 Results

In all the cases studied, squared distances among all pairs of points were computed using equation (2.1) with the 11 choices for $\mathbf{M}$ described earlier. Then the distances (d_{ij}'s) were fed as input to an average linkage hierarchical clustering program. The resulting dendrogram was cut to yield the known "correct" number of clusters. Hence, for each $(\mathbf{M}, \mathcal{D})$ combination, the data was partitioned into the "correct" number of clusters. The resulting groupings were compared to the known cluster memberships to get a count of number of mismatches. This provided a measure of performance for each $(\mathbf{M}, \mathcal{D})$ pair. The results are displayed in Table 2.4. For example, if metric $\mathbf{M}_2$ is applied to data set $\mathcal{D}1$, then 36 of the 75 observations are misclassified in the partition resulting from cutting the dendrogram. The numbers reported in the first five columns are based on results for the three carefully stylized cases of randomly generated data ($\mathcal{D}1$ – $\mathcal{D}3$) plus the two real data sets ($\mathcal{D}4$ and $\mathcal{D}5$). Among the simulated cases, $\mathcal{D}3$ is most delicate in terms of cluster structure and hence was subjected to further experimentation. The two rightmost columns give results obtained from 100 additional replications of the $\mathcal{D}3$ model. The numbers reported in these two columns are the mean and median error counts for these replicates. Note that in the results reported in this section, for methods requiring a choice of m_1 and/or m_2, m_1 and m_2 were set equal and based on the $\frac{2}{3}^{rd}$ rule (with an exception, as explained in the footnote to Table 2.4).

The diagonal elements of matrix $\mathbf{M}$ (univariate scales and weights) are shown in

	$\mathcal{D}1$	$\mathcal{D}2$	$\mathcal{D}3$	$\mathcal{D}4$	$\mathcal{D}5$	$\mathcal{D}3$-Mean	$\mathcal{D}3$-Median
$\mathbf{M}_1$	0	0	2	14	130	7	4
$\mathbf{M}_2$	36	35	39	47	134	45	47
$\mathbf{M}_3$	4	14	18	17	136	24	27
$\mathbf{M}_4$	55	53	50	48	133	52	52
$\mathbf{M}_5$	16	14	17	21	130	35	32
$\mathbf{M}_6$	0	0	2	4†	132	6	4
$\mathbf{M}_7$	0	0	2	14	131	6	4
$\mathbf{M}_8$	0	0	2	14	135	7	5
$\mathbf{M}_9$	0	0	3	5	136	5	3
$\mathbf{M}_{10}$	0	0	3	5	130	5	4
$\mathbf{M}_{11}$	0	0	2	14	135	6	4
n	75	75	75	150	200	75	75

Table 2.4: Errors of misclassification (mismatches) across data sets
†Choice of m_1 ($m_1 = 1850$) yielding the most favorable result was used;see discussion in text

Table 2.5 for $\mathbf{M}_1$ through $\mathbf{M}_{11}$. For comparability and ease of interpretation, the entries are normalized to sum to one. For example, the weights for $\mathbf{M}_1$ are all equal for each data set, as this corresponds to plain Euclidean distances ($\mathbf{M}_1 = \mathbf{I}$). Hence, for each $(\mathbf{M}, \mathcal{D})$ combination, the numbers denote the relative contributions of scale/weighting factors for each variable.

The results show that $\mathbf{M}_1$ leads to perfect cluster recovery for data sets $\mathcal{D}1$ and $\mathcal{D}2$ because of the substantial Euclidean spacing between clusters even in the presence of the three noise variables. The degree of proximity between cluster means dictates performance across most methods, as is especially evident in the overall weak results in the extreme case of data set $\mathcal{D}5$. $\mathbf{M}_2 - \mathbf{M}_5$ are methods intended to equalize the within-cluster variation of each variable. But this is not reflected in the corresponding entries given in Table 2.5. For example, $\mathbf{M}_4$, which ignores the cluster structure, seriously down-weights the first two structure variables for the three simulated data sets, relative to the noise dimensions. This results in its poor performance (55 errors) even in the case of data set $\mathcal{D}1$, where the cluster structure is conducive to perfect recovery. Apparently the "missing constants" (also see section 2.5) can't be ignored in these cases.

The intent of methods $\mathbf{M}_7$ through $\mathbf{M}_{11}$ is to facilitate more enlightened weighting based on preliminary estimates of within and between-cluster components. Generally,

	$\mathcal{D}1$	$\mathcal{D}2$	$\mathcal{D}3$	$\mathcal{D}4$	$\mathcal{D}5$
$\mathbf{M}_1$	0.20	0.20	0.20	0.25	0.20
	0.20	0.20	0.20	0.25	0.20
	0.20	0.20	0.20	0.25	0.20
	0.20	0.20	0.20	0.25	0.20
	0.20	0.20	0.20		0.20
$\mathbf{M}_2$	0.02	0.02	0.03	0.17	0.23
	0.02	0.01	0.03	0.60	0.43
	0.29	0.29	0.28	0.04	0.06
	0.30	0.30	0.29	0.20	0.05
	0.38	0.38	0.37		0.24
$\mathbf{M}_3$	0.03	0.03	0.05	0.17	0.26
	0.03	0.03	0.04	0.38	0.35
	0.26	0.26	0.25	0.06	0.06
	0.26	0.26	0.25	0.38	0.05
	0.42	0.42	0.40		0.28
$\mathbf{M}_4$	0.01	0.01	0.01	0.12	0.20
	0.01	0.01	0.01	0.78	0.50
	0.28	0.28	0.27	0.02	0.05
	0.40	0.40	0.39	0.09	0.05
	0.31	0.31	0.31		0.20
$\mathbf{M}_5$	0.03	0.03	0.04	0.12	0.22
	0.03	0.03	0.03	0.42	0.45
	0.27	0.27	0.27	0.09	0.05
	0.36	0.36	0.36	0.37	0.04
	0.31	0.31	0.31		0.24
$\mathbf{M}_6$	0.18	0.27	0.25	0.09	0.32
	0.17	0.26	0.25	0.13	0.15
	0.20	0.17	0.18	0.20	0.12
	0.23	0.15	0.15	0.58	0.07
	0.22	0.15	0.17		0.33

	$\mathcal{D}1$	$\mathcal{D}2$	$\mathcal{D}3$	$\mathcal{D}4$	$\mathcal{D}5$
$\mathbf{M}_7$	0.29	0.30	0.22	0.12	0.19
	0.31	0.29	0.22	0.12	0.22
	0.13	0.12	0.17	0.44	0.19
	0.16	0.17	0.23	0.32	0.20
	0.11	0.12	0.16		0.20
$\mathbf{M}_8$	0.26	0.26	0.20	0.13	0.19
	0.28	0.26	0.19	0.13	0.22
	0.14	0.15	0.19	0.42	0.19
	0.20	0.20	0.26	0.32	0.20
	0.13	0.13	0.16		0.20
$\mathbf{M}_9$	0.47	0.48	0.44	0.06	0.20
	0.45	0.48	0.48	0.02	0.05
	0.03	0.01	0.03	0.60	0.32
	0.03	0.01	0.02	0.32	0.23
	0.02	0.01	0.02		0.20
$\mathbf{M}_{10}$	0.50	0.49	0.45	0.06	0.20
	0.47	0.50	0.52	0.01	0.05
	0.01	0.01	0.01	0.60	0.32
	0.01	0.01	0.01	0.33	0.23
	0.01	0.01	0.01		0.20
$\mathbf{M}_{11}$	0.27	0.28	0.22	0.21	0.19
	0.30	0.27	0.21	0.09	0.22
	0.13	0.14	0.18	0.44	0.18
	0.18	0.18	0.24	0.26	0.20
	0.12	0.12	0.16		0.21

Table 2.5: Scaling and Weighting factors (Diag of $\mathbf{M}$) - Normalized to sum to one

the results are very promising, as observed in both Tables 2.4 and 2.5. For data sets $\mathcal{D}1$ and $\mathcal{D}2$, where the clusters are well separated, all methods perform perfectly. Furthermore, as shown in Table 2.5, the weights for $\mathbf{M}_7$ through $\mathbf{M}_{11}$ reflect the fact that taking into account between-cluster variability, significantly enhances those weights corresponding to the structure variables while simultaneously down-weighting the same for the noise variables. Hence, methods that incorporate the between-cluster component stand out when the original variables vary substantially in their discriminatory power. On the other hand, the univariate version of the multivariate approach of $\mathbf{M}_6$, which skirts the problem of missing constants, has done a much better job of equalizing the influence of all variables in $\mathcal{D}1-\mathcal{D}3$, and apparently $\mathcal{D}4$ too, as judged by the tremendous decreases in the number of misclassifications.

In case of the simulated data sets $\mathcal{D}1$, $\mathcal{D}2$ and $\mathcal{D}3$, the value of m_1 resulting from the $\frac{2}{3}^{rd}$ rule (m_1=350) was used for all methods described. Among the real data sets, for the error counts reported for $\mathcal{D}4$ using $\mathbf{M}_6$ in Table 2.4 (marked †), the choice of m_1 (m_1=1850) which yielded the most favorable result was used (*Note*: We tried a range of different m_1 values (1700 – 2200) and found m_1=1850 to yield the best result, although, the results did not vary much across the different m_1 values.) Hence for comparability across methods, the same value of m_1 was adopted for all scaling and weighting approaches for data set $\mathcal{D}4$. In the case of $\mathcal{D}5$, the error count reported (55) corresponds to using m_1 (m_1=3200) as obtained by the $\frac{2}{3}^{rd}$ rule. Hence, for consistency, the results reported for all methods using $\mathcal{D}5$, are also based on this m_1 value.

A practical advantage of methods $\mathbf{M}_3$ through $\mathbf{M}_{11}$ is that they serve as a useful preprocessing step that does not involve any cluster analysis. However, the above comparisons of their relative performance in terms of their error counts are all based on the average linkage HCA algorithm. The weights reported in Table 2.5 are, of course, not dependent on the particular method of clustering. To the extent that these scaling and weighting approaches perform as intuition suggests (i.e., equalize the within-cluster variability or assign more weight to the structure variables relative to noise), one might expect that the performances of these methods using other clustering algorithms with similar aims would lead to conclusions similar to those reported above. For instance,

with the complete linkage HCA algorithm, the results were very close, with the subtle variations being consistent across all methods and data sets. However, the relative rankings of performances remained the same.

Chapter 3

Multivariate $\mathbf{W}^*_{(m_1)}$ Algorithm - Null Cluster Structure

3.1 Introduction

With metric data being represented as n points in a p-dimensional space, if the interest is in grouping the n p-dimensional observations, one could use the $p \times n$ matrix, $\mathbf{Y}$, of raw data as the input. If one wants the results to be invariant under affine transformations of the initial variables, then one way to achieve this would be to use the transformed data, $\mathbf{Z} = \mathbf{A}\,\mathbf{Y}$, where $\mathbf{A}$ is the inverse of the triangular matrix from the Cholesky decomposition of an estimate of the covariance matrix. Ideally, we would like to use an estimate of the "within-clusters" covariance matrix analogous to the within-group covariance matrix used in DA. The difficulty posed by CA, is that the groups are not known *a priori*. However, various schemes have been proposed to handle this situation (see [4]; [34], [40] and references therein.)

A well-known way of securing "invariance" is, by using Mahalanobis's generalized distance so as to allow for differing variances of the variables and for inter-correlations among them (see section 4.3.1 of [33].) The literature on CA contains many algorithms that have been proposed as schemes for optimizing criteria that are invariant under certain classes of linear transformations (see, for example, [17] and [29].) For using Mahalanobis's generalized distance, an appropriate choice when the clusters are reasonably homogenous in their shapes (as in the idealized situation when they are considered as differing in location but having a common covariance matrix, $\mathbf{\Sigma}$), is the pooled within-clusters covariance matrix. In a real data problem, however, the clusters are unknown to start with, thus creating a dilemma which has been recognized by several authors. A way out of this quandary is to develop an appropriate metric iteratively from the data. The within-clusters estimated covariance matrix obtained in this way could also be

used for transforming the data to make the clusters look "spherical". In what follows, we provide a description of an iterative algorithm used to develop such a data-based metric.

3.2 $\mathbf{W}^*_{(m_1)}$ Algorithm

The motivation for the scheme proposed in [4] is the following decomposition of the total sum of cross products matrix in terms of pairwise differences among the observations:

$$\frac{1}{n}\sum_{i<j}(\mathbf{y}_i-\mathbf{y}_j)(\mathbf{y}_i-\mathbf{y}_j)' = \frac{1}{n}\sum_{\substack{i<j \\ within}}(\mathbf{y}_i-\mathbf{y}_j)(\mathbf{y}_i-\mathbf{y}_j)' + \frac{1}{n}\sum_{\substack{i<j \\ between}}(\mathbf{y}_i-\mathbf{y}_j)(\mathbf{y}_i-\mathbf{y}_j)'. \quad (3.1)$$

Similar to the standard breakdown of total sums of squares and cross products into the known within and between components, *i.e.*, the identity, $\mathbf{T} = \mathbf{W} + \mathbf{B}$, we can denote equation (3.1) as:

$$\mathbf{T} = \mathbf{W}^* + \mathbf{B}^*. \quad (3.2)$$

In the above equation, $\mathbf{W}^*$ is based only on the within-cluster pairs of observations while $\mathbf{B}^*$ is based only on the between-cluster pairs of observations. In the CA situation, since we lack advance information on the number of clusters and the cluster labels, we again use the intuitive reasoning that, despite this difficulty, if there are any clusters present in the data then the nearest neighbors are likely to belong to the same cluster. With this thinking in mind, an estimator similar to $\mathbf{W}^*$ was proposed in [4]. Given a $p \times n$ data matrix $\mathbf{Y}$, where $\mathbf{y}_i$, $i = 1, \ldots n$ are the n p-dimensional observations, the algorithmic steps are (also see [33]):

1. Set $\mathbf{W}^{*(0)}_{(m_1)} = \mathbf{I}$, the identity matrix, and set the iteration count $t = 1$.

2. Find the m_1 closest pairs of observations according to the squared generalized distance

$$(\mathbf{y}_i - \mathbf{y}_j)'\left[\mathbf{W}^{*(t-1)}_{(m_1)}\right]^{-1}(\mathbf{y}_i - \mathbf{y}_j) \ ; \ i < j = 1, \ldots n$$

3. Define:

$$\mathbf{W}^{*(t)}_{(m_1)} = \frac{1}{n}\sum_{\{A\}}(\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)',$$

where $\{A\}$ is the set of pairs of points points $(i, j), i < j$, corresponding to the closest pairs found in step (2).

4. Compute error:

$$\mathbf{E}^t = tr\big(\big[\mathbf{W}^{*(t-1)}_{(m_1)}\big]^{-1}\big[\mathbf{W}^{*(t)}_{(m_1)}\big] - \mathbf{I}\big)^2.$$

If the error $\mathbf{E}^t \leq \mathbf{E}$, a user-specified number (say, 0.001), or, if t=t_{max}, the maximum number of iterations allowed, STOP and set $\mathbf{W}^*_{(m_1)}$=$\mathbf{W}^{*(t)}_{(m_1)}$. Otherwise, set $t \leftarrow t + 1$ and return to step (2).

Note that the matrices $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}^*$ will be equal only if $\{A\}$ contains all of the within-cluster pairs and no others at the final step of the iterative process. In practice, the challenge is to keep m_1 "small enough" so that between-cluster pairs are not likely to contaminate the final $\{A\}$.

Diagonal version of $\mathbf{W}^*_{(m_1)}$ - $\mathbf{W}^*_{d(m_1)}$ algorithm

- The $\mathbf{W}^*_{(m_1)}$ algorithm given above, requires the inversion of a non-trivial matrix at each stage of the iteration. This could put a demand on computational resources as the dimensionality p, of the data set increases. To circumvent this problem, we propose a slight variation to the same algorithm, with every step remaining the same, except for the fact that we would now use only the diagonal entries of the $\mathbf{W}^*_{(m_1)}$ matrix at each step. So we would actually be computing weighted Euclidean distances at each iteration. This only involves the inversion of a diagonal matrix. The converged matrix would hence be a diagonal matrix ($\mathbf{W}^*_{d(m_1)}$), whose diagonal entries could be used as estimates of scale of the p variables. We had defined this as $\mathbf{M}_6$ in section 2.2. Hence, this could be thought of as a univariate version of the multivariate $\mathbf{W}^*_{(m_1)}$ algorithm.

Convergence of the $\mathbf{W}^*_{(m_1)}$ algorithm is monitored by the error function. As given in step 4 of the algorithm, when the error between successive iterations falls below a specified bound, the algorithm terminates. Figure 3.1 displays convergence of the $\mathbf{W}^*_{(m_1)}$-algorithm, as seen by reducing error with each iteration, for a $\mathbf{N}(\mathbf{0}, \mathbf{I})$ random

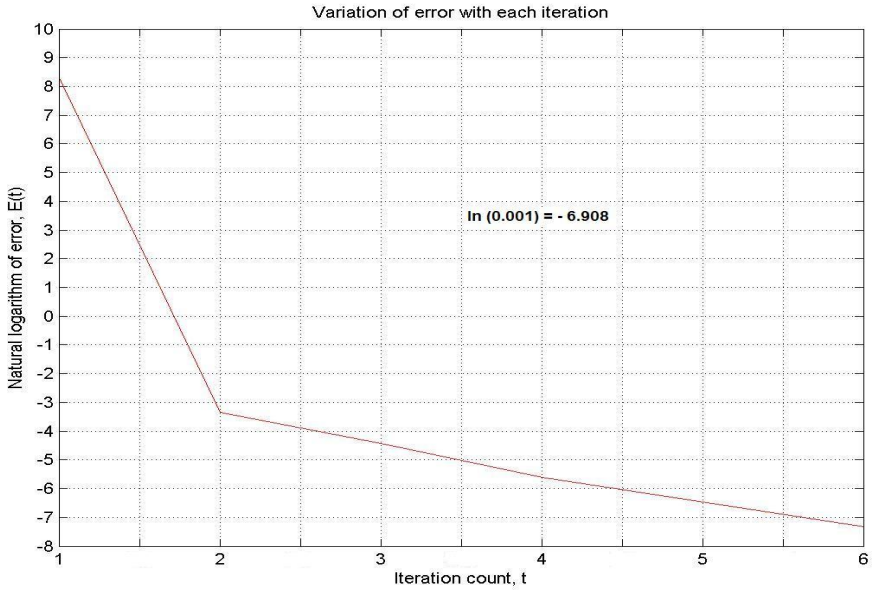

Figure 3.1: Convergence of $\mathbf{W}^*_{(m_1)}$-algorithm for a $\mathbf{N}(\mathbf{0}, \mathbf{I})$ sample

sample (n=75, p=5 and m_1=1850, based on the $\frac{2}{3}^{rd}$ rule). The error values on the y-axis are plotted in the logarithmic scale to make smaller changes in error more legible. Also, the algorithm converges in 6 iterations in this example, as shown in Figure 3.1.

To understand the $\mathbf{W}^*_{(m_1)}$ algorithm intuitively, consider the sample two-dimensional scatter plots given in Figure 3.2, for illustrative purposes. In the first iteration, since we computed Euclidean distances between every pair of points, we use the unit-circle to indicate the Euclidean distance metric. In subsequent iterations, since a Mahalanobis-like distance metric is used, we replace the unit circle with a gradually changing ellipse. As the algorithm reaches convergence, the orientation of this ellipse settles down to reasonably reflect the elliptical shape of the data.

The $\mathbf{W}^*_{(m_1)}$-algorithm just described, is fully defined, except for the value of m_1. As mentioned earlier, in section 2.4, care needs to be taken to choose an "appropriate" value of m_1 so that primarily within-cluster pairs are used to form $\mathbf{W}^*_{(m_1)}$. As explained in [4], even though the decomposition given in equation (3.2) "does not have as many nice statistical properties such as the independence of $\mathbf{W}$ and $\mathbf{B}$, a key point to note is

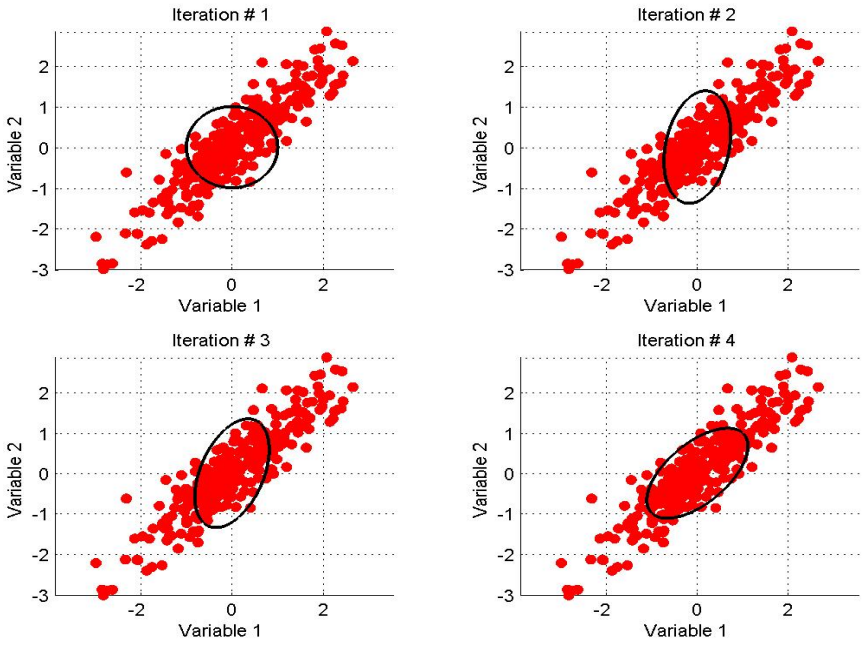

Figure 3.2: $\mathbf{W}^{*(t)}_{(m_1)}$ computations with each iteration, t

that both $\mathbf{W}^*$ and $\mathbf{W}$ have the same expected value, apart from a constant. The main difference is that $\mathbf{W}^*$ gives relatively more weight to large groups than does $\mathbf{W}$".

Using the inverse of the matrix, $\mathbf{W}^*_{(m_1)}$, obtained after convergence, one can compute a Mahalanobis type of metric to measure the inter-point distances (using the squared distance function defined in equation (2.1)) as inputs to hierarchical methods of clustering. Such a metric has the advantage that it is more appropriate for measuring distances when the variables are correlated and any clusters present are approximately homogenous and ellipsoidal in shape. Correspondingly, using Mahalanobis distance in the space of the original variables is equivalent to using Euclidean distance in the *sphericized* space of the variables obtained by a linear transformation of the original variables using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1/2}$. Hence, in the transformed space, the clusters, if there are any, will be more spherical in shape. Furthermore, the transformed data could also be used as input to non-hierarchical methods of CA such as k-means, using the Euclidean distance metric which is especially effective at detecting homogenous spherical clusters.

Although the $\mathbf{W}^*_{(m_1)}$ algorithm just described, is simple enough to implement, its performance characteristics have yet to be systematically studied. The primary focus of this chapter is to study some of the characteristics of the $\mathbf{W}^*_{(m_1)}$ algorithm, as given below.

We begin by studying the characteristics of the algorithm in the case where we have no clusters. It may be noted that this is also connected to the work on robust multivariate estimation of dispersion in the single sample situation (see [35]). Since it can be argued that many practical applications of clustering techniques are to situations for which there are no real statistical clusters, this could be regarded as a natural place to begin, and also the simplest model to study. To test the algorithm, we simulate data from a multivariate normal distribution with zero mean and correlation matrix, $\mathbf{P}$. A particularly convenient and important theoretical covariance structure is when all the pairwise correlations are equal. (If all the variances are equal, too, the covariance matrix is sometimes referred to as *compound symmetric*.) If the common correlation, ρ, is positive, then the corresponding ellipsoid of concentration will have $p-1$ dimensions with spherical structure, reflecting the $p-1$ smaller and equal eigenvalues ($\lambda_{(1)} = \lambda_{(2)} = \cdots = \lambda_{(p-1)} = 1-\rho$), and elongated structure along its major axis, dictated by the largest eigenvalue ($\lambda_{(p)} = 1+(p-1)\rho$). (Note: $\sum_{i=1}^{p} \lambda_{(i)} = p$). This ellipsoidal structure, when the common correlation is large and positive is sometimes described as "cigar shaped" and can be especially challenging for clustering algorithms that are implicitly or explicitly designed to discover spherical clusters. Hence, such a structure would be both a convenient and an appropriate challenge case for the $\mathbf{W}^*_{(m_1)}$ algorithm.

3.3 Sensitivity to Starting Point

Note that in the first iteration of the $\mathbf{W}^*_{(m_1)}$ algorithm, we set $\mathbf{W}^{*(0)}_{(m_1)} = \mathbf{I}$, the identity matrix, *i.e.*, Euclidean distance is used in the first iteration to obtain a rank ordering of pairwise distances. This prompts the question whether we could use a more "informed" *starting point* to better orient the ellipsoids computed in subsequent iterations of the algorithm. Along the same lines, we could ask the more general question - "How

sensitive is the algorithm to its starting point?"

One way to study the sensitivity of the $\mathbf{W}^*_{(m_1)}$ algorithm to its starting point is to consider different random diagonal and full (positive-definite) matrix starting points (instead of the identity matrix) and then compare the converged matrices arising from the different random starting points. This would, of course, need to be performed on data sets with different covariance structures. If the algorithm is not sensitive to its starting point, then no matter where we start, intuitively, the converged matrix would be the same, or, nearly the same. The following section delineates the experimental design used for this study.

3.3.1 Experimental design

Data sets were simulated by drawing random samples of $n = 75$ observations from a N($\mathbf{0}$, $\mathbf{P}$) distribution with correlation matrix $\mathbf{P}$ that satisfied the compound symmetry conditions. In addition, all variances were set equal to one. Hence, in this case, the corresponding covariance matrix was also equal to $\mathbf{P}$. We picked five different values for the common correlation ρ (0, 0.5, 0.9, 0.95 and 0.99) and in each case, we simulated data sets with three different dimensionalities p (5, 10 and 50). With five different common correlations and three different dimensionalities, we had 15 different random data sets, all with $n = 75$ observations. The idea behind such a setup was to evaluate the sensitivity issue under different conditions, ranging from the simplest (n, p, ρ) combination of (75, 5, 0) to the extreme combination of (75, 50, 0.99).

One simple way to create random diagonal matrix starting points is to generate diagonal matrices where the main diagonal elements are Uniformly distributed in the interval (0,1). We picked this method to generate such random diagonal matrices with positive diagonal elements, purely for its simplicity. Subsequently, to generate random symmetric positive-definite (SPD) matrix starting points, we applied the QR and eigenvalue decompositions in succession, as follows: We know that any $p \times p$ random matrix $\mathbf{A}$, of full rank, has the QR factorization (see [31])

$$\mathbf{A} \;=\; \mathbf{Q}.\mathbf{R}, \tag{3.3}$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper-triangular matrix.

If $\mathbf{\Lambda}$ is a random diagonal matrix with positive diagonal elements (generated as described earlier), using $\mathbf{Q}$ (from above), and the eigenvalue decomposition, we have,

$$\mathbf{F} \;=\; \mathbf{Q}.\mathbf{\Lambda}.\mathbf{Q}' \tag{3.4}$$

$\mathbf{F}$ is now an SPD matrix, which could be used as a starting point for the $\mathbf{W}^*_{(m_1)}$ algorithm. Again, it is important to note that this is only one way to generate random SPD matrices.

With m_1 fixed, based on the $\frac{2}{3}^{rd}$ rule (m_1=1850 for n=75), the $\mathbf{W}^*_{(m_1)}$ algorithm was executed on each of the 15 data sets, for 100 different random starting points (diagonal and SPD). The 100 converged matrices for each data set, were then compared. To gain more insight, the corresponding diagonal and off-diagonal elements from each converged $\mathbf{W}^*_{(m_1)}$ matrix were compared separately, described as follows.

Comparing diagonal elements:

The diagonal elements (w^*_{ii}, i=1$\ldots p$), were first extracted from each of the converged $\mathbf{W}^*_{(m_1)}$ matrices and rescaled to sum to p. This was done to neutralize the effect of any missing constant (there is more on the missing constant issue later in this chapter) in the $\mathbf{W}^*_{(m_1)}$ matrices and to put the sets of diagonal elements on the same footing. So we have,

$$w^*_{ii} \;\rightarrow\; w^{**}_{ii} \quad s.t \quad \sum_{i=1}^{p} w^{**}_{ii} \;=\; p.$$

To compare these sets of diagonal elements (p elements in each set) across the 100 matrices, some order statistics (maximum, upper quartile, median, lower quartile and minimum) were extracted from each set of normalized diagonal elements and plotted, as shown in Figures 3.3, 3.4, 3.5 and 3.6 respectively. Intuitively, if the diagonal elements are similar for all the converged $\mathbf{W}^*_{(m_1)}$ matrices, then we would expect low variability in the corresponding order statistics plots.

Comparing off-diagonal elements:

To compare the off-diagonal elements, all matrix components above the main diagonal were first extracted from each converged $\mathbf{W}^*_{(m_1)}$ matrix and normalized by the

	Comparison of diagonal elements	
	$\rho = 0$	$\rho = 0.99$
Diagonal starting point	Figure 3.3	Figure 3.5
SPD starting point	Figure 3.4	Figure 3.6
	Comparison of off-diagonal elements	
	$\rho = 0$	$\rho = 0.99$
Diagonal starting point	Figure 3.7	Figure 3.9
SPD starting point	Figure 3.8	Figure 3.10

Table 3.1: Summary of Figures used to study sensitivity to starting point

corresponding diagonal elements to obtain estimates of correlation, ρ. Thus we have,

$$\frac{w^*_{ij}}{\sqrt{w^*_{ii}} \cdot \sqrt{w^*_{jj}}} \;\rightarrow\; r^*_{ij}.$$

The normalized off-diagonal components of the converged $\mathbf{W}^*_{(m_1)}$ matrices were then compared by studying their distribution using the order statistics plots as described earlier. These are displayed in Figures 3.7, 3.8, 3.9 and 3.10 respectively.

3.3.2 Results

The plots in Figures 3.3 – 3.10 display the order statistics of the diagonal and off-diagonal elements of the converged $\mathbf{W}^*_{(m_1)}$ matrices. In all plots, the X-axis corresponds to the 100 random starting points while the Y-axis displays the corresponding order statistics. Solid lines are used to display the maximum, median and minimum values corresponding to each random starting point, while, dotted lines are used to display the corresponding upper and lower quartiles, respectively. Note that in the interest of parsimony, plots are displayed here only for the two extreme common-correlational structures of $\rho = 0$ and $\rho = 0.99$. Results for other values of the common correlation, ρ, are similar to those displayed. Also, it is important to note that in Figures 3.3, 3.4, 3.5 and 3.6, where the diagonal elements are compared, the order statistics corresponding to each starting point are based on only p elements as we have p diagonal elements in each $\mathbf{W}^*_{(m_1)}$ matrix. Similarly, in Figures 3.7, 3.8, 3.9 and 3.10, where the off-diagonal elements are compared, the order statistics corresponding to each starting point are based on $\binom{p}{2}$ elements. Table 3.1 lists the plots displayed.

An interesting common feature gleaned from all the plots is the strikingly low variability observed in the order statistics across the different random starting points. This suggests that the $\mathbf{W}^*_{(m_1)}$ algorithm converges to *almost* the same matrix, no matter what starting point is used, on a given one of the simulated data sets. Specifically, in Figures 3.3, 3.4, 3.5 and 3.6, in the plots that compares the diagonal elements of the 100 converged $\mathbf{W}^*_{(m_1)}$ matrices, the medians of the normalized diagonal elements are expected to lie close to one. This is exactly what is observed in the empirical plots. Similarly, with the off-diagonal elements, the respective medians are expected to lie close to the common correlation, ρ, in each case. Again, this is what is observed in the plots displayed in Figures 3.7, 3.8, 3.9 and 3.10 respectively. The algorithm converges on an average in 7 - 10 iterations for all the data sets studied. It is also important to note that for the purposes of this study, the convergence criterion is fixed at $\mathbf{E} = 0.001$. However, as a user defined lower bound, this constant could be relaxed slightly, as the dimensionality, p, increases.

Note that the plots displayed in Figures 3.3 – 3.10 are for one random sample corresponding to each (n, p, ρ) combination. To ascertain reliability of the results, we simulated 50 random data sets for each (n, p, ρ) combination and repeated the entire process on each data set. We again found similar results, with negligible variability in order statistics across the 100 random starting points, for each random replicate. Additionally, the plots given in Figure 3.11 are based on the *averaged* order statistics (corresponding to the off-diagonal elements, using random SPD matrix starting points), averaged over 50 replications of the ρ=0.99 model. As expected, averaging neutralizes the sampling biases apparent in the single sample experiments, thereby yielding a symmetric distribution of the order statistics, as displayed in Figure 3.11. Also, comparable results are obtained for the other values of ρ.

It may be noted that the results reported here correspond to data sets with a fixed $n = 75$. However, limited additional experiments on different values of n also yield similar results. Furthermore, the results do not change much between using either random diagonal matrix or SPD matrix starting points. This shows that the algorithm *forgets* its initial sets of rank orderings of nearest neighbors after the first few iterations,

converging to a common subset of nearest neighbors, in *Mahalanobis* sense, thereby having captured the embedded correlational structure in the data. In other words, the $\mathbf{W}^*_{(m_1)}$ algorithm is not sensitive to its starting point.

3.4 Quality of the $\mathbf{W}^*_{(m_1)}$ estimates

A crucial issue in the study of the $\mathbf{W}^*_{(m_1)}$ algorithm is evaluating the quality of the $\mathbf{W}^*_{(m_1)}$ estimates obtained under different experimental conditions. This could be done by comparing the converged $\mathbf{W}^*_{(m_1)}$ matrix with the known $\mathbf{\Sigma}$ matrix as well as the known $\mathbf{W}$ matrix for data sets with different correlational structures (ρ), as well as different n and p values. Additionally, we could study the adequacy of a single multiplicative constant for making the $\mathbf{W}^*_{(m_1)}$ matrix a more nearly unbiased estimate of the covariance structure of the data. This would also give us information on the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm. In what follows, we explain our design methodology, followed by some results that would assess the quality of the $\mathbf{W}^*_{(m_1)}$ estimates.

3.4.1 Experimental design

To evaluate the quality of the $\mathbf{W}^*_{(m_1)}$ matrices under different experimental conditions, data sets were first generated with varying sizes (n), dimensionality (p) as well as varying underlying common correlations (ρ). As before, this was done by drawing random samples from a multivariate normal distribution, corresponding to several different values of n (n = 75, 150, 200, 250, 300 and 400), p (p = 5, 10, 50, 100, 150, 200 and 300) and ρ (ρ = 0, 0.5, 0.9, 0.95 and 0.99). Table 3.2 displays the 30 different combinations of n and p used in this study. Note that in each case, n was always chosen greater than p, so as to guard against any potential singularity of the resulting $\mathbf{W}^*_{(m_1)}$ matrices. So, for instance, when n = 75, only three values of p (p = 5, 10 and 50) were used, as displayed in Table 3.2 (marked as "$\surd$" in the first row of Table 3.2.) Furthermore, this was repeated for all 5 values of the common correlation ρ, thereby yielding 150 different experimental conditions under which the quality of the $\mathbf{W}^*_{(m_1)}$ matrix was evaluated.

For each (n, p, ρ) combination, 100 random data sets were generated. For each of

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$
$n = 150$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$
$n = 200$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
$n = 250$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
$n = 300$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
$n = 400$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

Table 3.2: Data sets with the different (n, p) combinations used in this study

these 100 replicates, the $\mathbf{W}^*_{(m_1)}$ algorithm was executed, yielding 100 converged $\mathbf{W}^*_{(m_1)}$ matrices. Since the starting-point-invariance characteristic of the algorithm was already established, Euclidean distances were used in the first iteration of the algorithm in all cases. The 100 resulting matrices were then averaged so as to smooth out the random variation arising from the sampling process. A similar averaging was done with the 100 $\mathbf{W}$ matrices corresponding to the replicates. Hence, for each value of ρ, there were 30 (averaged) $\mathbf{W}^*_{(m_1)}$ matrices and 30 (averaged) $\mathbf{W}$ matrices, corresponding to the 30 (n, p) combinations of data sets.

To begin, as one way to evaluate the quality of the $\mathbf{W}^*_{(m_1)}$ estimates, at least in terms of tracking the correlational structure of the data, each $\mathbf{W}^*_{(m_1)}$ matrix was converted to its corresponding correlation matrix, ($\mathbf{R}^*$). The mean of the off-diagonal elements (correlations) were then compared with the known common correlation, ρ. Likewise, the known $\mathbf{W}$ matrices were also converted to corresponding correlation matrices, ($\mathbf{R}$), and a similar comparison with $\mathbf{R}^*$ was performed. The idea behind such a comparison is to assess the quality of the $\mathbf{W}^*_{(m_1)}$ estimates, in terms of providing a good measure of the true $\mathbf{\Sigma}$ and the known $\mathbf{W}$ matrices, respectively.

To further study the accuracy of the $\mathbf{W}^*_{(m_1)}$ matrix in approximating the known $\mathbf{W}$ matrix, the eigenvalues of the two matrices were examined. If the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices are approximately proportional (apart from an unknown constant of proportionality), then the respective eigenvalues of the two matrices would also be approximately proportional to each other. To eliminate the effect of the unknown constant, the eigenvalues were divided by their respective means so that they sum to one. This put the sets of eigenvalues on the same footing, ready for comparison. Subsequently,

the variances of the normalized sets of eigenvalues were computed. Hence, for instance, when the common correlation, $\rho = 0$, the true covariance matrix $\mathbf{\Sigma} = \mathbf{I}$, the identity matrix, and thus the eigenvalues of $\mathbf{\Sigma}$ are all equal to one. In this case, the variances of the normalized eigenvalues of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices would both be expected to lie close to zero. Such a variance-analysis provided a closer look into the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm in terms of tracking the known $\mathbf{W}$ matrix. Additionally, to compare the individual eigenvalues of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices with the known $\mathbf{\Sigma}$ matrix, the mean square error (M.S.E) between the eigenvalues of $\mathbf{W}^*_{(m_1)}$ and $\mathbf{\Sigma}$, and similarly, between $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, was computed. As before, to render the eigenvalues invariant to any unknown constant, the sets of eigenvalues were first normalized by their respective means before computing the M.S.E values. This analysis was done to specifically check for any inherent biases between the individual observed and true eigenvalues of the respective matrices.

3.4.2 Results

As described in the previous section, for each (n, p, ρ) combination, the resulting (averaged) $\mathbf{W}^*_{(m_1)}$ and (averaged) $\mathbf{W}$ matrices were first converted to their corresponding correlation matrices $\mathbf{R}^*$ and $\mathbf{R}$, respectively. The mean of the upper-diagonal elements from both the matrices were then compared with the known common correlation, ρ. Tables 3.3 and 3.4 provide this comparison for the 30 different (n, p) combinations of data sets, for the value of $\rho = 0$. Tables 3.5 – 3.12 display the same for the other values of the common correlation, ρ.

The results in Tables 3.3 – 3.12 indicate that the $\mathbf{W}^*_{(m_1)}$ algorithm performs creditably in terms of tracking the known $\mathbf{W}$ matrices as well as the underlying $\mathbf{\Sigma}$ matrices. In particular, for all values of ρ, the average correlations from the $\mathbf{R}^*$ matrices are not only close to the average correlations given by the $\mathbf{R}$ matrices, but are also close to the underlying ρ itself. For instance, in Table 3.3, which displays the average correlations given by the $\mathbf{R}^*$ matrices, the numbers are reasonably close to the underlying value of $\rho = 0$. This suggests that the performance of the algorithm is commendable in trying to capture the underlying correlational structure of the data.

Furthermore, Figures 3.12 – 3.16 display plots of variances of the normalized eigenvalues of the (averaged) $\mathbf{W}^*_{(m_1)}$ and (averaged) $\mathbf{W}$ matrices for each of the five values of the common correlation, ρ. Additionally, in Figures 3.13 – 3.16, the variances of the normalized eigenvalues of the corresponding $\mathbf{\Sigma}$ matrices are also plotted. In all Figures, each set of vertical curves corresponds to a particular value of n, as noted on the plots. Also, for each value of n, the *asterisk* (" * ") symbols on the curves represent the variances corresponding to increasing values of p. The different (n, p) combinations for which the variances are plotted are given in Table 3.2. Hence, for each (n, p, ρ) combination, the plots display the variances of the normalized eigenvalues of the (averaged) $\mathbf{W}^*_{(m_1)}$, (averaged) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices respectively, (except in Figure 3.12, where the variances from the $\mathbf{\Sigma}$ matrices are not plotted, as they are all equal to zero.) The plots displayed in Figures 3.12 – 3.16 show the pattern-similarity of the curves obtained for the $\mathbf{W}^*_{(m_1)}$, $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices. Furthermore, as expected, in all cases the variability increases as n and p approach the same order of magnitude. Nevertheless, it is important to note that for any given value of ρ, the corresponding variances are relatively the same for the $\mathbf{W}^*_{(m_1)}$, $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices.

Figures 3.17 – 3.21 display the M.S.E plots of the normalized eigenvalues of the (averaged) $\mathbf{W}^*_{(m_1)}$ and (averaged) $\mathbf{W}$ matrices for each of the five values of the common correlation, ρ. Note that in each case, the M.S.E's are computed with respect to the true eigenvalues, *i.e.*, the eigenvalues of the known $\mathbf{\Sigma}$ matrix. The small M.S.Es, coupled with the nearly overlapping profiles of the plots seen for all values of ρ, is reassuring. This again demonstrates that the (averaged) $\mathbf{W}^*_{(m_1)}$ matrices are very similar to the known (averaged) $\mathbf{W}$ and known $\mathbf{\Sigma}$ matrices, thereby reinforcing the quality of the $\mathbf{W}^*_{(m_1)}$ estimates.

3.4.3 "Missing" constant

To check if the expected value of the $\mathbf{W}^*_{(m_1)}$ estimates and their corresponding $\mathbf{\Sigma}$ matrices are approximately apart from only a single proportionality constant, we could consider the matrix product, $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\mathbf{W}^*_{(m_1)}\right]$. If only a single constant multiplier

(approximately) "separates" the two matrices, then the matrix $\mathbf{M}$ would be (approximately) proportional to the Identity matrix, *i.e.*, $\mathbf{M} \approx k\mathbf{I}$. Hence, in such a scenario, the eigenvalues of $\mathbf{M}$ would be expected to display low variability. To examine this, we used a subset of the different (n, p, ρ) combinations of data sets reported earlier. For each data set, the average $\mathbf{W}^*_{(m_1)}$ matrix was used to compute $\mathbf{M}$. The eigenvalues of $\mathbf{M}$ were then extracted and normalized so that they sum to one. Subsequently, in each case the variance of these normalized eigenvalues was computed, as displayed in Tables 3.13 – 3.15.

The small variance values in all cases suggests that, indeed, $\mathbf{M}$ could be approximated as being proportional to the Identity matrix, $\mathbf{I}$. This provides evidence that apart from a single constant multiplier, the $\mathbf{W}^*_{(m_1)}$ algorithm yields a reliable estimate of the underlying covariance matrix, $\mathbf{\Sigma}$. It is also interesting to note that the variances seem more or less constant as the common correlation, ρ, changes. Furthermore, as seen for all three values of ρ, the respective variances in each case decreases as n increases relative to p. It may also be noted that similar results were obtained for data sets with the other values of n, p and ρ.

Additionally, if we were to estimate the "unknown constant", k, at least in the least squares sense, one way to do this would be to minimize the function, $\text{trace}\big[\mathbf{M} - k\mathbf{I}\big]^2$. This could be expressed equivalently as,

$$\arg\min_k \ \text{trace}\big[\mathbf{M} - k\mathbf{I}\big]^2 \ = \ \arg\min_k \ \sum_{i=1}^{p} \big[(\lambda_i - k)\big]^2,$$

where, λ_i, $i = 1 \ldots p$ are the eigenvalues of the matrix, $\mathbf{M}$. It can be easily seen that the above function is least-squares-minimized when $k = \overline{\lambda}$, where, $\overline{\lambda}$ is the average eigenvalue of $\mathbf{M}$. Tables 3.16 – 3.18 provide the least squares estimates of the unknown constant computed as described above, that would make the $\mathbf{W}^*_{(m_1)}$ matrices more unbiased. Again, the average $\mathbf{W}^*_{(m_1)}$ matrix corresponding to each (n, p, ρ) combination was used to compute $\mathbf{M}$, in each case. As seen in the Tables, it is interesting to note that although one would expect the estimates of the constant to vary with m_1, the estimates seem to vary more with n and p and less so with ρ.

3.4.4 Variability of $\mathbf{W}^*_{(m_1)}$ estimates

The focus of the analyses done so far has been primarily on the average performance of the $\mathbf{W}^*_{(m_1)}$ algorithm, based on the average $\mathbf{W}^*_{(m_1)}$ matrix used in each case. Hence, the analysis has mainly concentrated on the issue of bias. However, it is also interesting to examine the variability of the $\mathbf{W}^*_{(m_1)}$ estimates. Specifically, comparing the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices for each of the 100 random replicates corresponding to each (n, p, ρ) combination would provide more insight into the variability of the $\mathbf{W}^*_{(m_1)}$ matrices. One way to do this is explained as follows. For each (n, p, ρ) combination we have 100 converged $\mathbf{W}^*_{(m_1)}$ matrices and 100 known $\mathbf{W}$ matrices. We could first convert each of these into correlation matrices to obtain 100 $\mathbf{R}^*$ and 100 $\mathbf{R}$ matrices, respectively. For each $(i,j)^{th}$ element of the corresponding matrices, $i < j = 1 \ldots p$, we could then consider the ratio, $\mathcal{V}1_{(ij)} = \dfrac{var[r^*_{(i,j)}]}{var[r_{(i,j)}]}$, and average this measure for all $i < j$. This would give us a succinct measure of the variability of the $\mathbf{W}^*_{(m_1)}$ matrices relative to the $\mathbf{W}$ matrices for each (n, p, ρ) combination of data sets. Tables 3.19 – 3.21 provide the $\mathcal{V}1$ variability measures, computed as described above, for a subset of the data sets.

As seen in Tables 3.19 – 3.21, although the variability ratios ($\mathcal{V}1$) are close to one, they are in general larger than one. This indicates that the $\mathbf{W}^*_{(m_1)}$ matrices are more variable than their corresponding $\mathbf{W}$ matrices, as one would expect. This is, in part, the price making a conservative choice for m_1. Also, for the same reason, one would expect the magnitude of the departure from unity to vary with the choice of m_1. However, it is interesting to note that the ratios seem independent of the common correlation, ρ. This, taken together with the near-unity variance ratios again indicate that the $\mathbf{W}^*_{(m_1)}$ algorithm provides reliable results, consistently tracking the known $\mathbf{W}$ matrix in each case.

Additionally, to specifically study the variability of the diagonal elements of the 100 converged $\mathbf{W}^*_{(m_1)}$ matrices, we could do the following. Given 100 $\mathbf{W}^*_{(m_1)}$ matrices and 100 known $\mathbf{W}$ matrices for each (n, p, ρ) combination, we could first extract the diagonal elements from each of the 100 matrices to get $w^{*(j)}_{(ii)}$ and $w^{(j)}_{(ii)}$ respectively, where

$i = 1 \ldots p$ and $j = 1 \ldots 100$. Here, $w_{(ii)}^{*(j)}$ corresponds to the 100 sets of diagonal elements from the 100 $\mathbf{W}_{(m_1)}^{*}$ matrices, while $w_{(ii)}^{(j)}$ corresponds to the 100 sets of diagonal elements from the 100 $\mathbf{W}$ matrices. We could then normalize each set separately, such that they sum to one. Hence we would have,

$$\sum_{i=1}^{p} w_{(ii)}^{*(j)} \;=\; 1,\; j = 1 \ldots 100$$

and,

$$\sum_{i=1}^{p} w_{(ii)}^{(j)} \;=\; 1,\; j = 1 \ldots 100$$

For each i^{th} diagonal element we could then compute the ratio $\mathcal{V}2_{(i)} = \dfrac{var[w_{(ii)}^{*}]}{var[w_{(ii)}]}$, where the variance for each i^{th} diagonal element is computed across the 100 samples. We could then average this ratios-of-variances measure over all p. This would provide a comparison of the variances of the normalized diagonal terms of the $\mathbf{W}_{(m_1)}^{*}$ matrices with the variances of the normalized diagonal terms of the $\mathbf{W}$ matrices, for each (n, p, ρ) combination. Tables 3.22 – 3.24 provide the $\mathcal{V}2$ variability measures just described.

As displayed in Tables 3.22 – 3.24, the variability ratios are all very close to one, albeit slightly larger than one. This is again consistent with intuition, as one would expect estimates of the diagonal elements from the $\mathbf{W}_{(m_1)}^{*}$ matrices to be more variable than those of the $\mathbf{W}$ matrices (also, recall that the diagonal terms of the underlying $\mathbf{\Sigma}$ matrices were all set equal to one). Furthermore, it may be noted that the $\mathcal{V}2$ variability ratios seem relatively smaller than the $\mathcal{V}1$ ratios for the off-diagonal elements. This may, in part, be due to the normalizations performed on the diagonal elements, constraining them to be relatively more alike. Nevertheless, the near-unity variability ratios indicate that the variances of the diagonal elements of the respective matrices are very similar. This result, taken together with the previous result which studied the variability of the off-diagonal terms of the $\mathbf{W}_{(m_1)}^{*}$ matrices, again reinforces the stability of the $\mathbf{W}_{(m_1)}^{*}$ algorithm. Hence, the $\mathbf{W}_{(m_1)}^{*}$ algorithm converges to relatively the same underlying covariance matrix across the 100 random replicates, with the small variability in the converged matrices attributable primarily to the sampling process.

3.5 Conclusion

This chapter studied some of the characteristics of the $\mathbf{W}^*_{(m_1)}$ algorithm under the no-clusters scenario. To begin, the starting point issue of the algorithm was studied. The results indicated that the algorithm is not sensitive to the choice of its starting point. This is an important finding for this iterative approach, demonstrating that the natural choice of the Euclidean distance metric (Identity matrix starting point) for computing nearest neighbors in the first iteration, is as good as any other metric, thereby providing evidence of simplicity as well as stability of the algorithm. In terms of performance of the $\mathbf{W}^*_{(m_1)}$ algorithm in yielding a good quality estimate of the covariance structure of the data, the analysis performed suggests that the algorithm not only converges to a good approximation of the known $\mathbf{W}$ matrix (our gold standard for comparison), but also provides a reasonable approximation of the underlying $\mathbf{\Sigma}$ matrix. Furthermore, an eigenanalysis of the respective matrices also revealed striking similarity of the converged $\mathbf{W}^*_{(m_1)}$, $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices. This, coupled with additional experiments to study the missing constant issue leads us to reason that in the absence of clusters, only a single constant multiplier would be needed to make the $\mathbf{W}^*_{(m_1)}$ matrix a more nearly unbiased estimate of the underlying covariance matrix, $\mathbf{\Sigma}$. In addition to the average-performance-characteristics of the algorithm, the variability of the $\mathbf{W}^*_{(m_1)}$ estimates was also studied. The results again established the relative stability of the $\mathbf{W}^*_{(m_1)}$ algorithm by converging to covariance estimates that were consistent with $\mathbf{W}$ as well as the underlying $\mathbf{\Sigma}$ matrices in each case. Also, it may be noted that the results reported in this chapter were based on random data sets with underlying *compound symmetric* covariance structures. However, limited experiments were also performed on data sets with other randomly chosen covariance structures and the conclusions drawn from them were generally consistent with those reported here. All these results suggests that the algorithm provides a reliable measure of the covariance structure, at least in the no-clusters scenario. The next chapter will provide a detailed study of the characteristics of the $\mathbf{W}^*_{(m_1)}$ algorithm for data sets with underlying cluster structure.

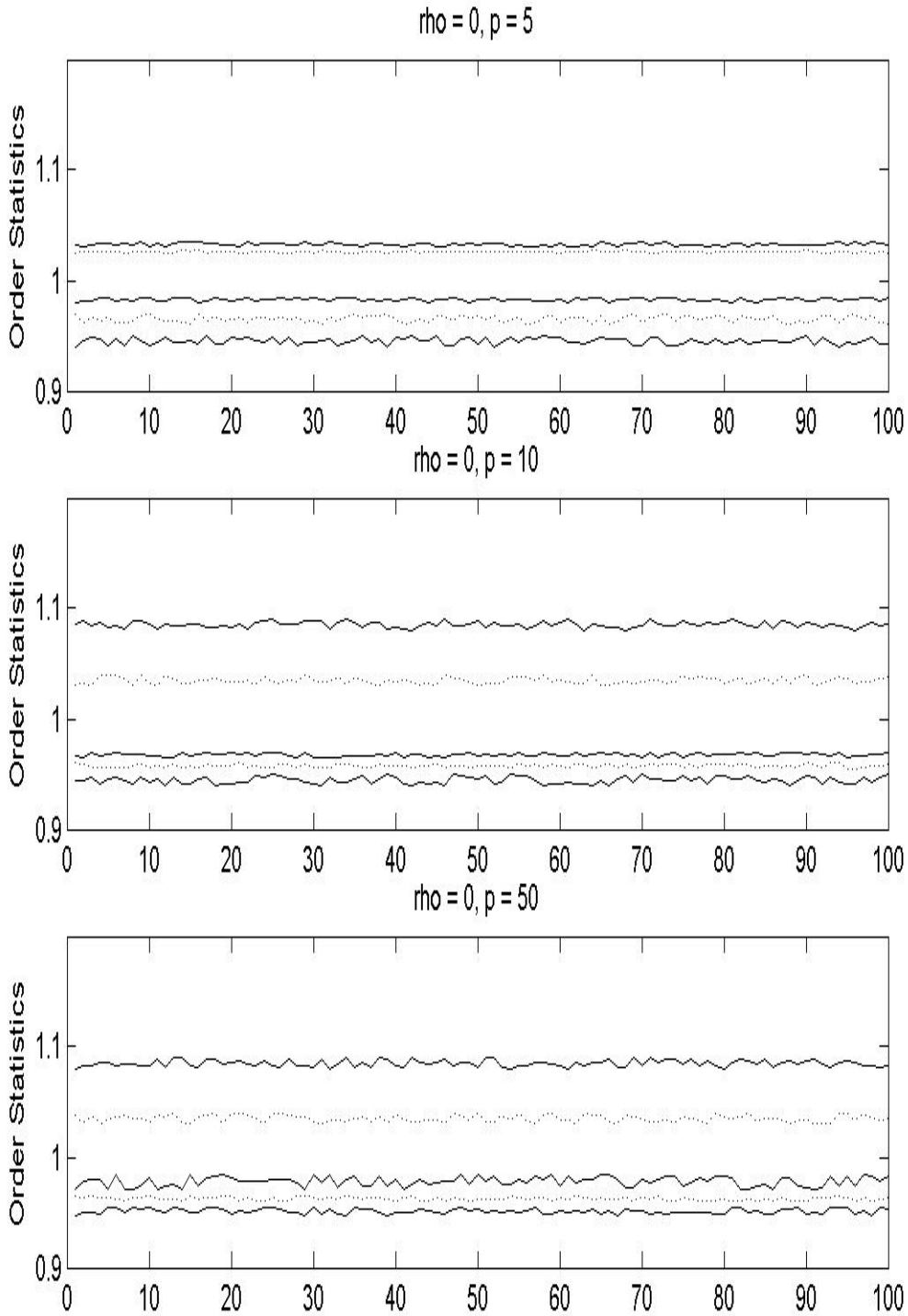

Figure 3.3: Comparison of diagonal elements - diagonal starting point - $\rho = 0$

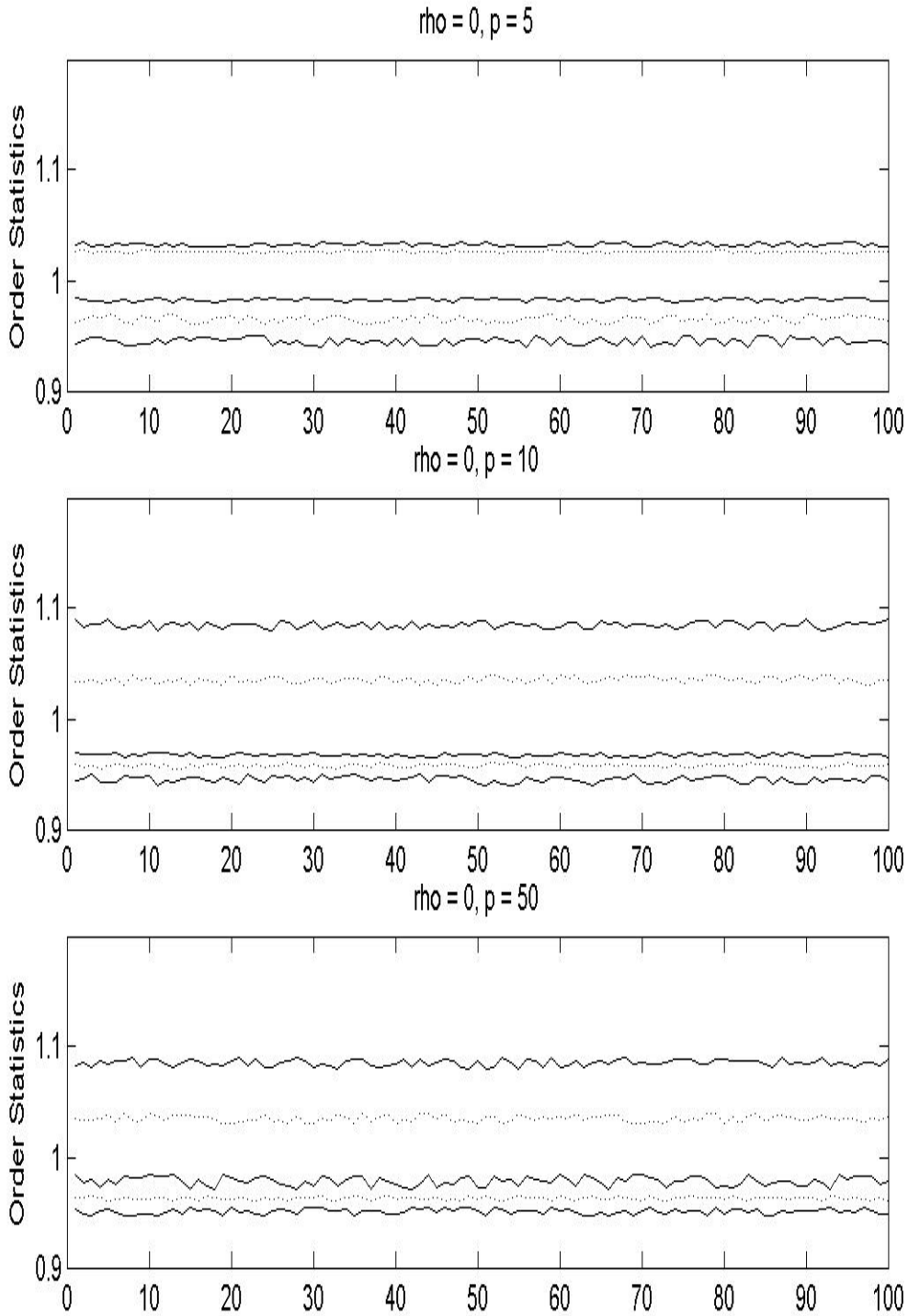

Figure 3.4: Comparison of diagonal elements - SPD starting point - $\rho = 0$

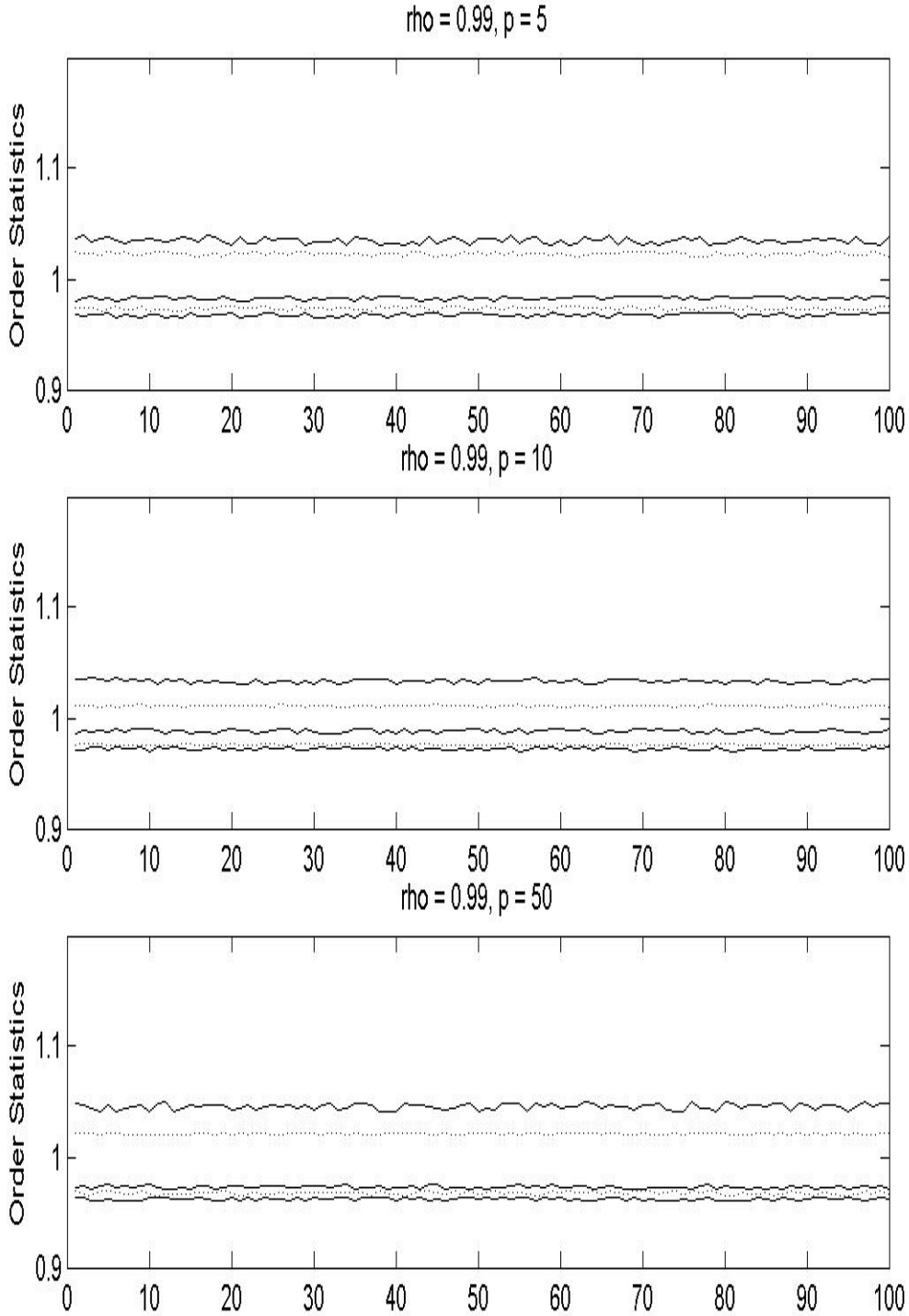

Figure 3.5: Comparison of diagonal elements - diagonal starting point - $\rho = 0.99$

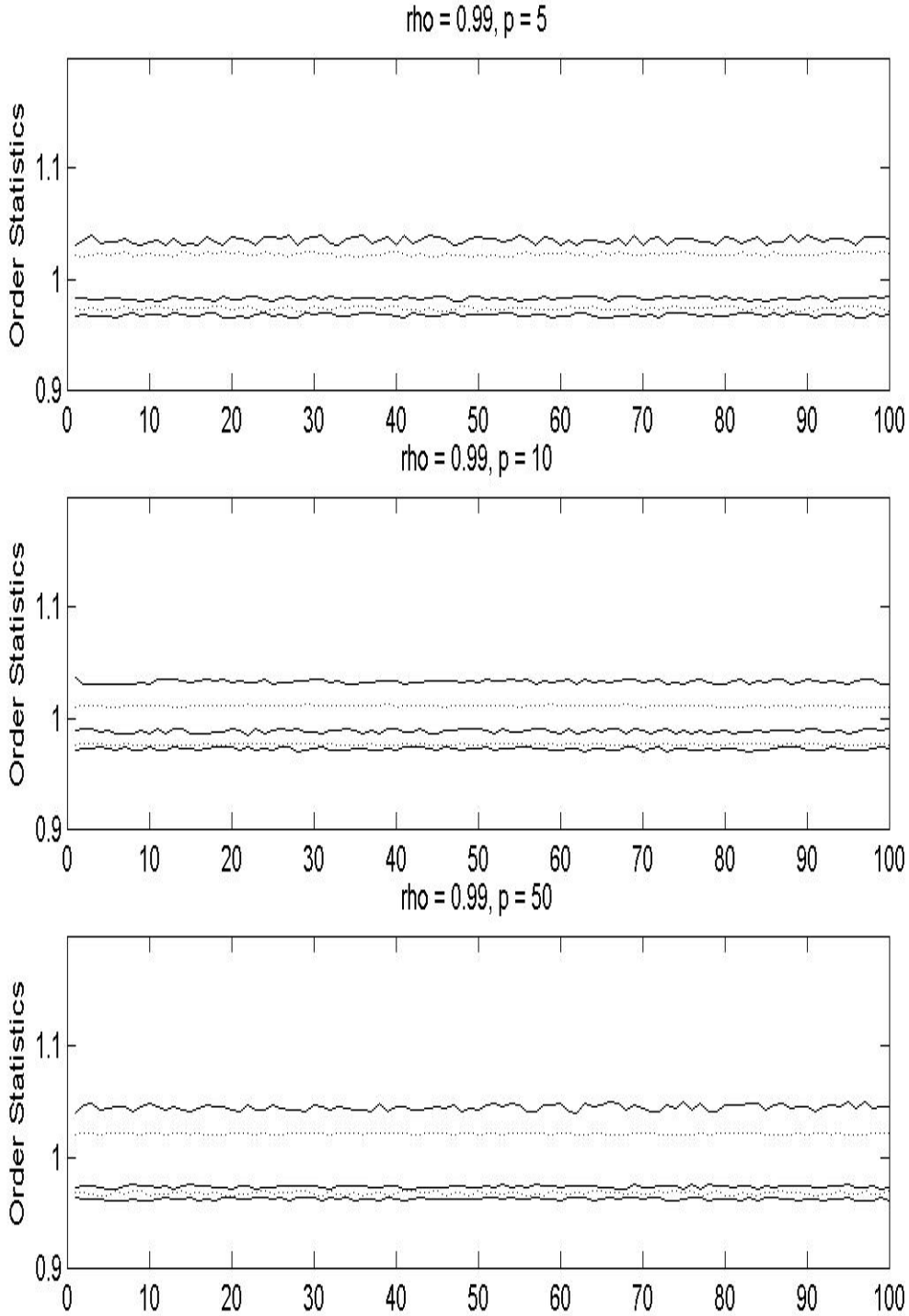

Figure 3.6: Comparison of diagonal elements - SPD starting point - $\rho = 0.99$

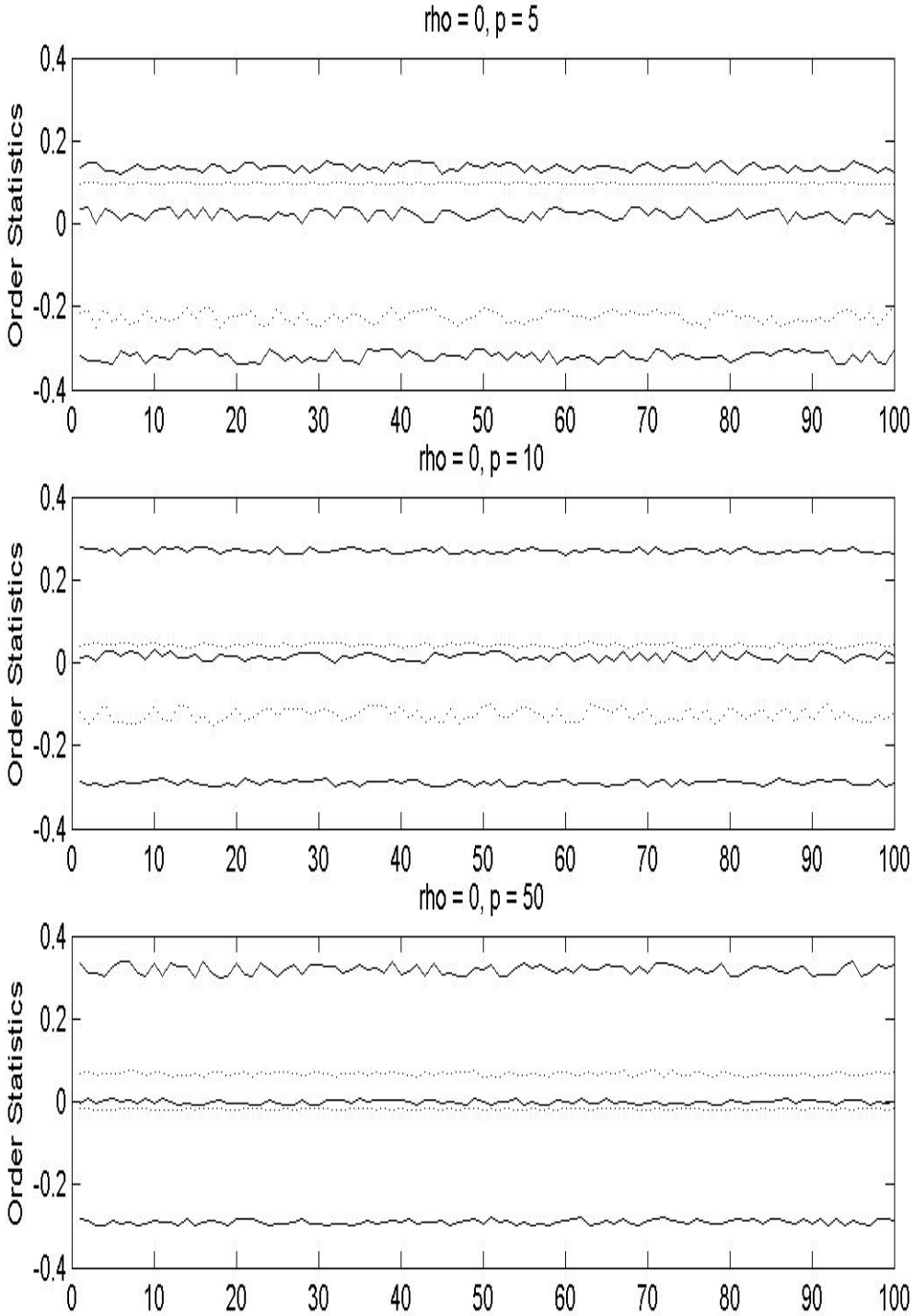

Figure 3.7: Comparison of off-diagonal elements - diagonal starting point - $\rho = 0$

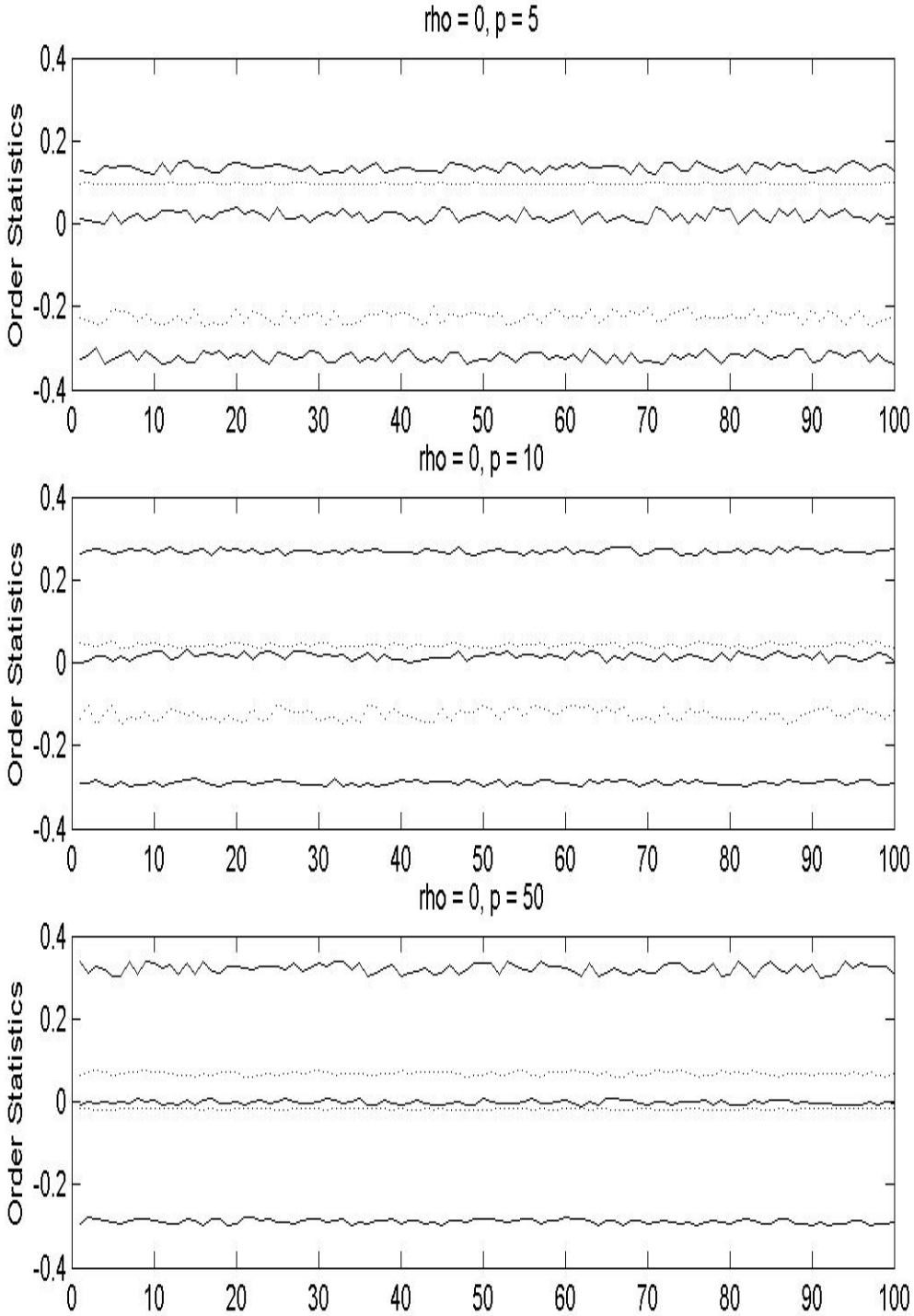

Figure 3.8: Comparison of off-diagonal elements - SPD starting point - $\rho = 0$

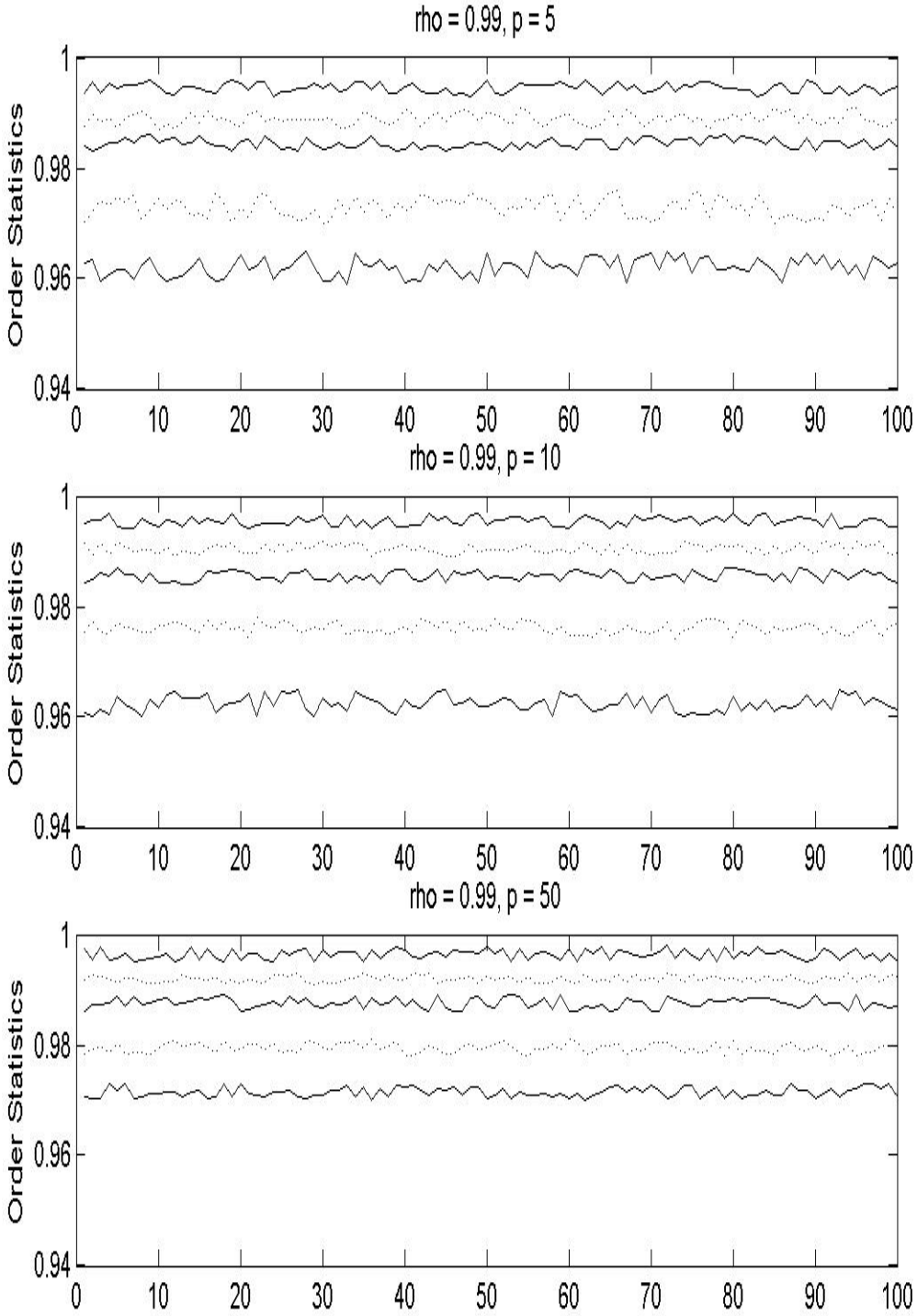

Figure 3.9: Comparison of off-diagonal elements - diagonal starting point - $\rho = 0.99$

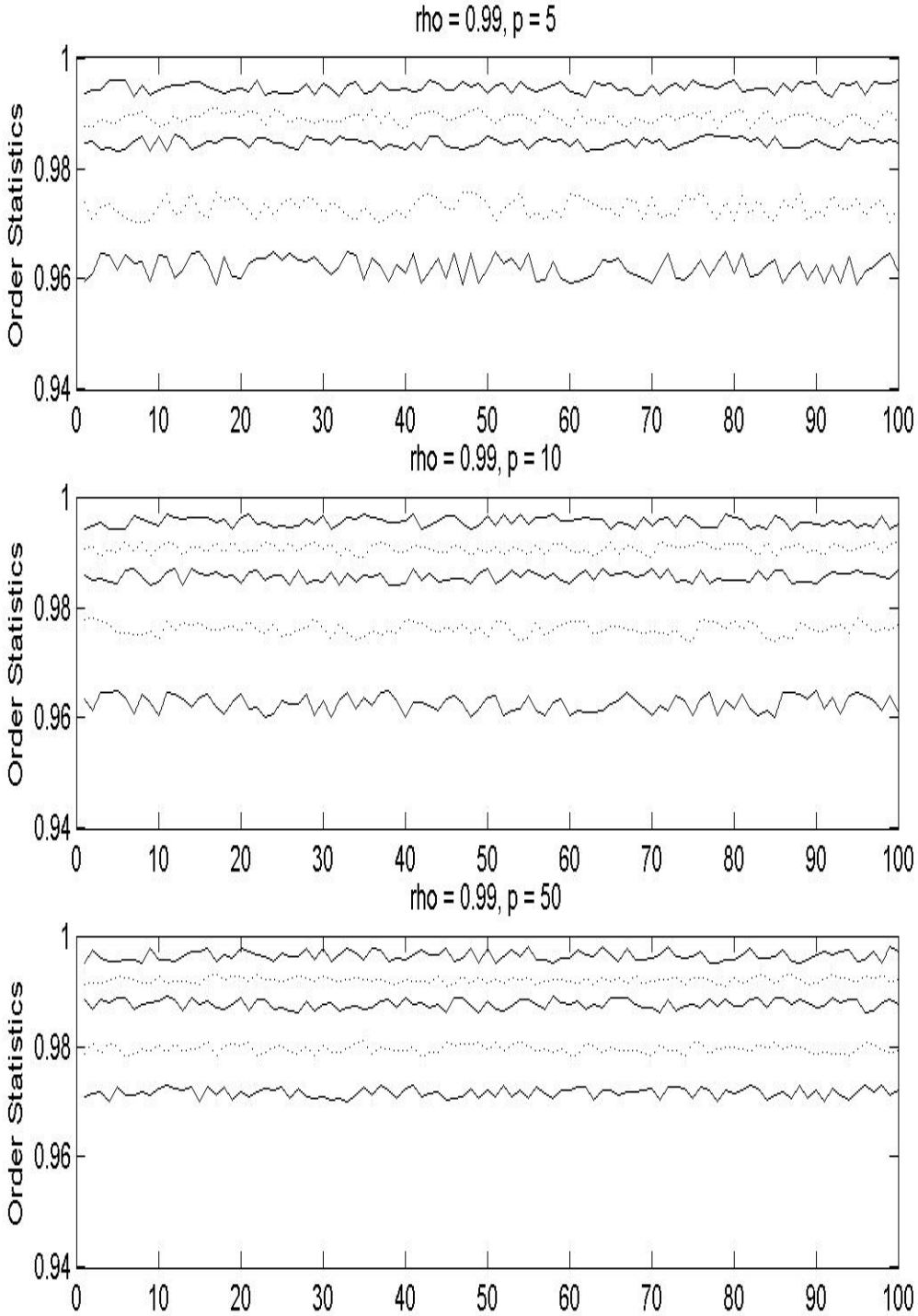

Figure 3.10: Comparison of off-diagonal elements - SPD starting point - $\rho = 0.99$

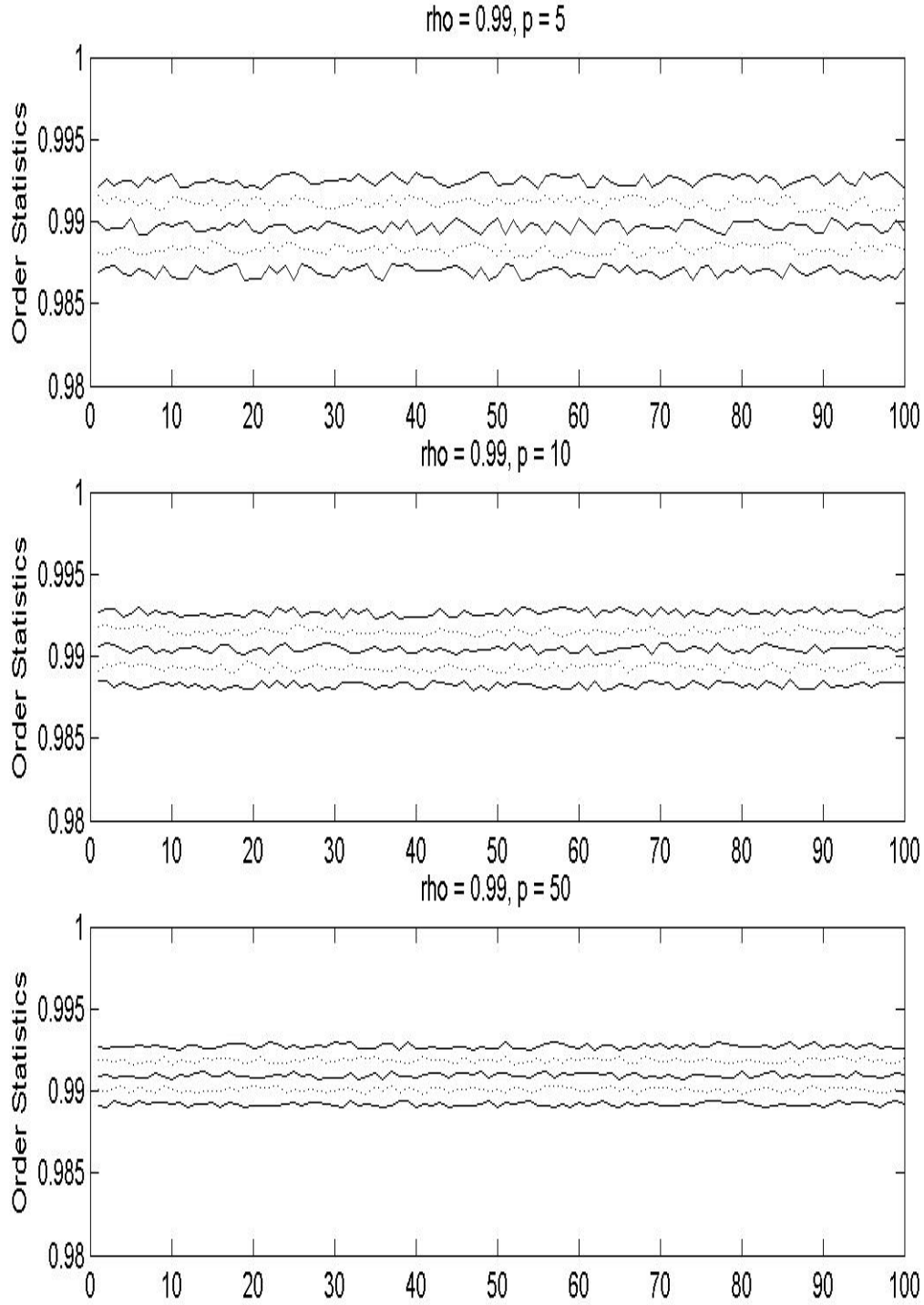

Figure 3.11: Mean order statistics of off-diagonal elements-SPD starting point-ρ=0.99

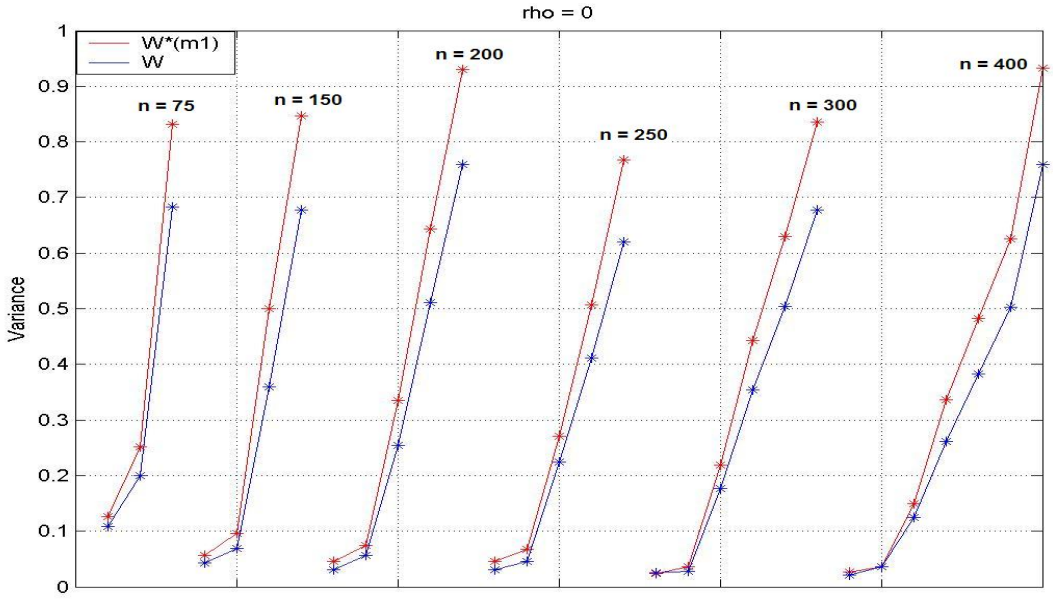

Figure 3.12: Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$ and (average) $\mathbf{W}$ matrices, ρ=0

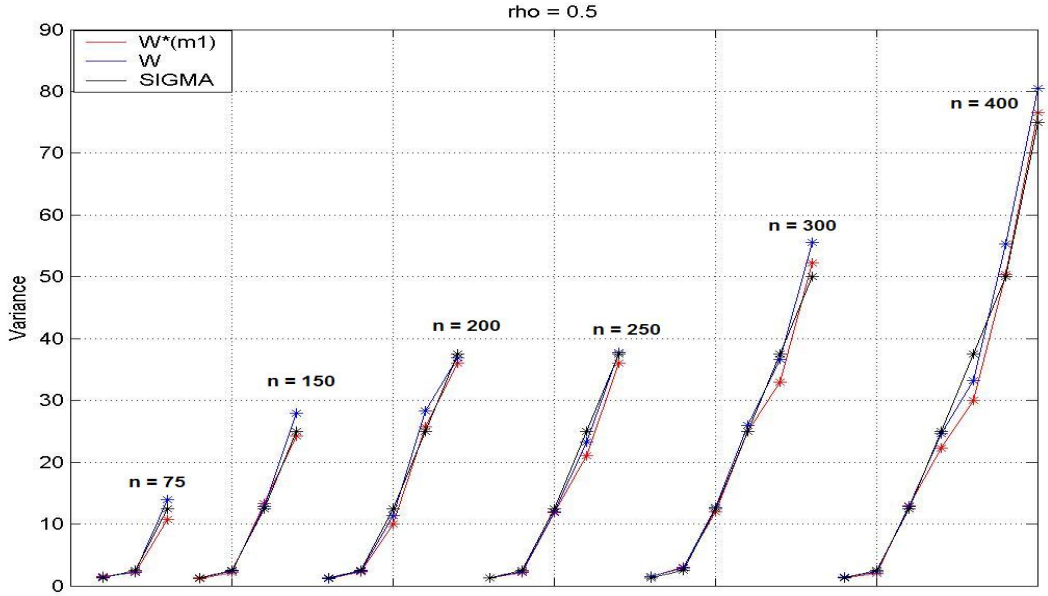

Figure 3.13: Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.5

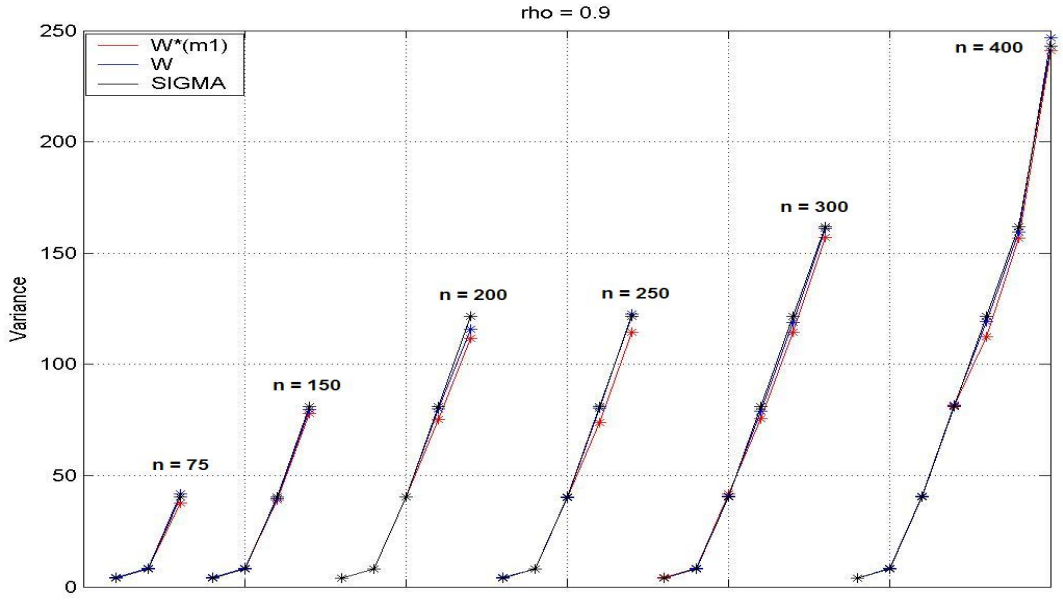

Figure 3.14: Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.9

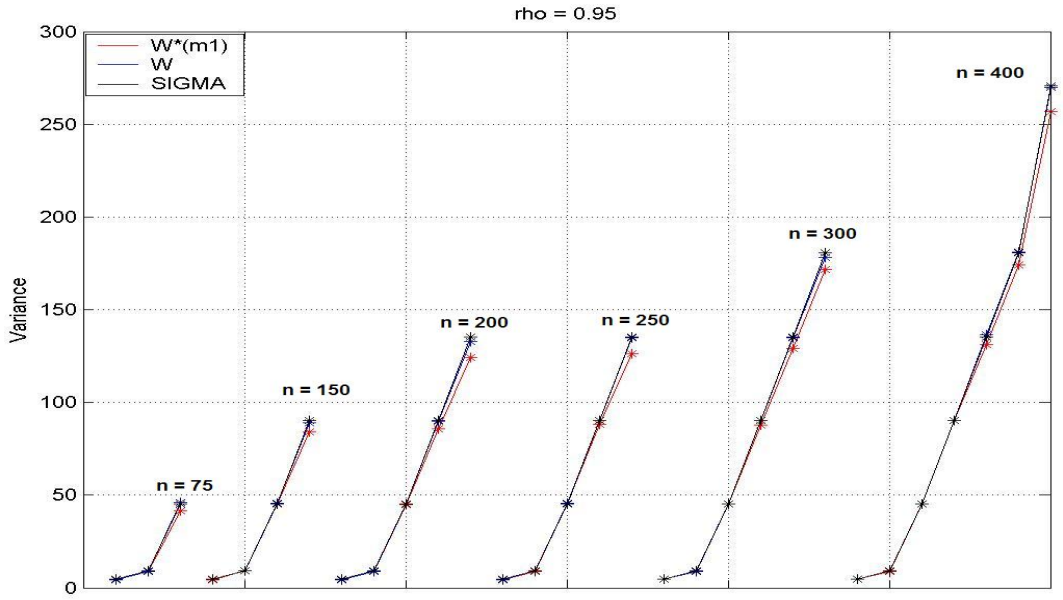

Figure 3.15: Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.95

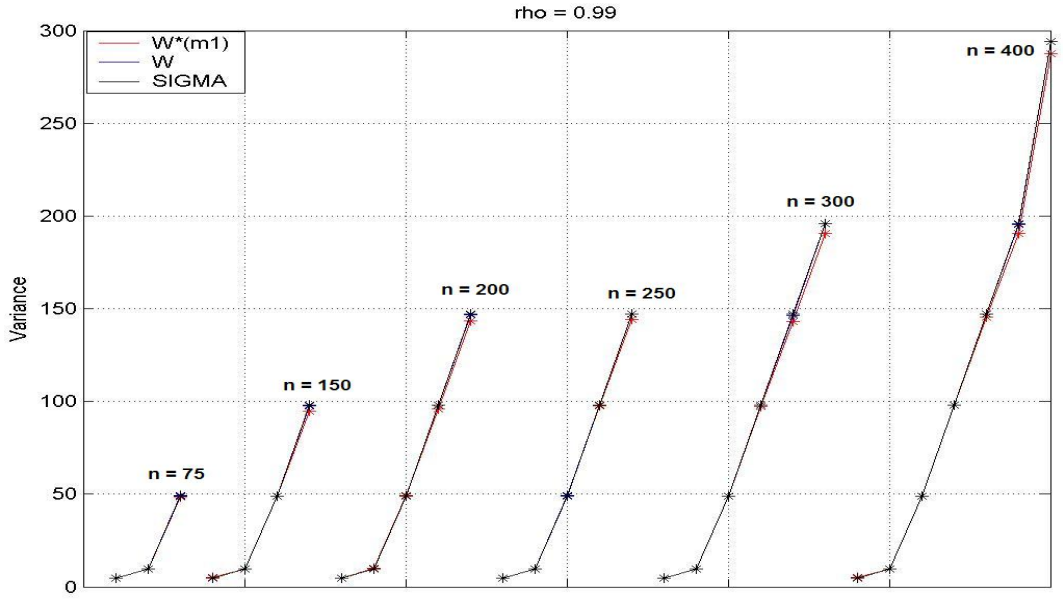

Figure 3.16: Variance of normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.99

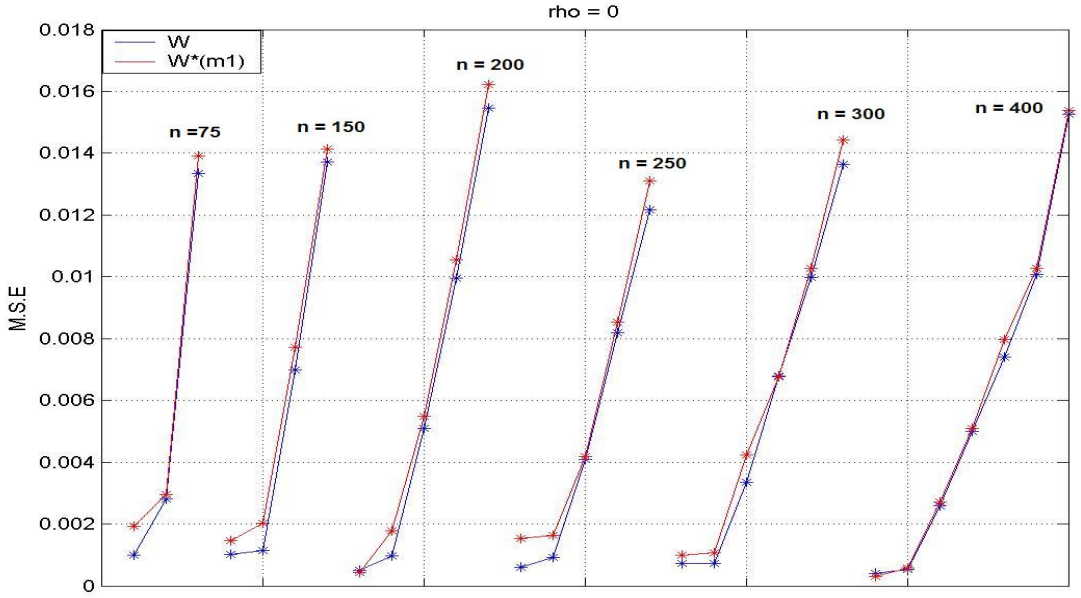

Figure 3.17: M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices,ρ=0

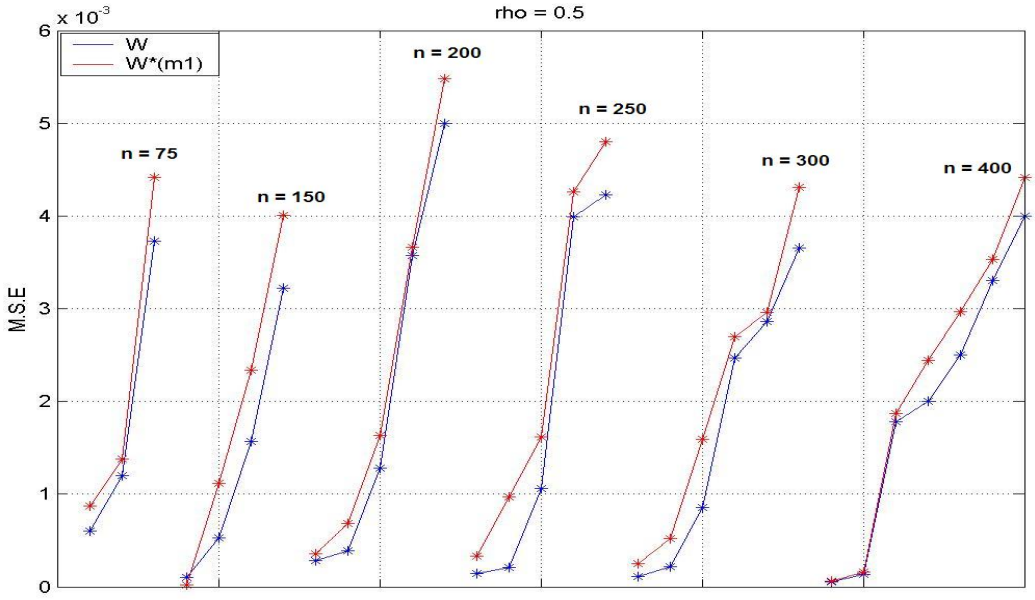

Figure 3.18: M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.5

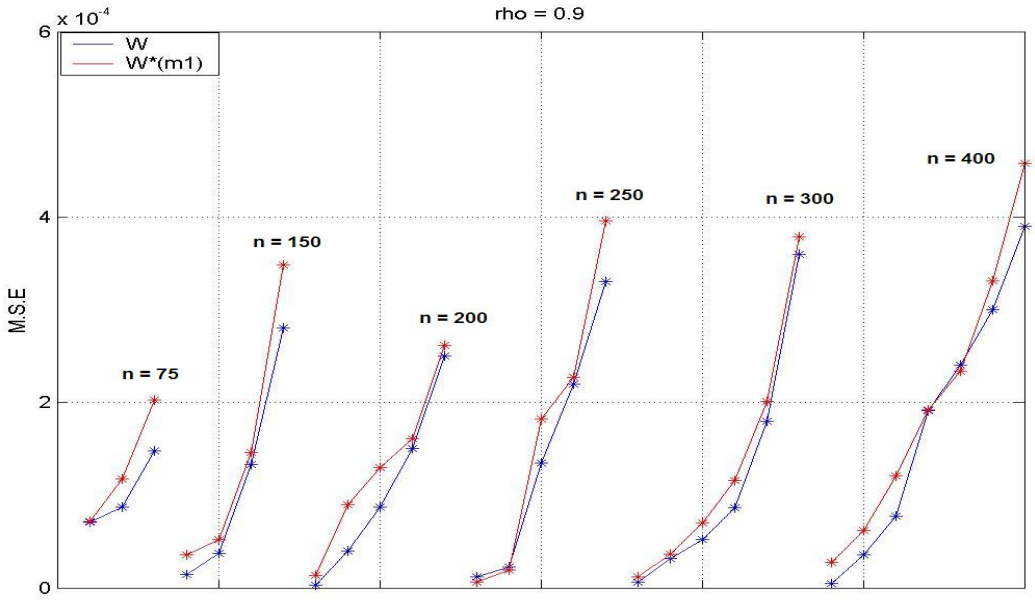

Figure 3.19: M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.9

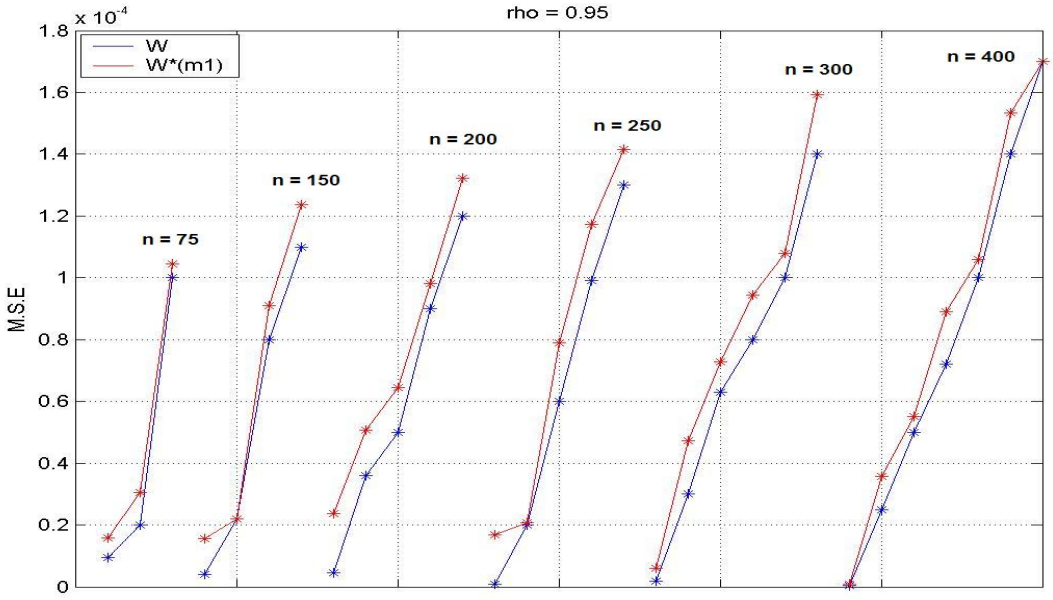

Figure 3.20: M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.95

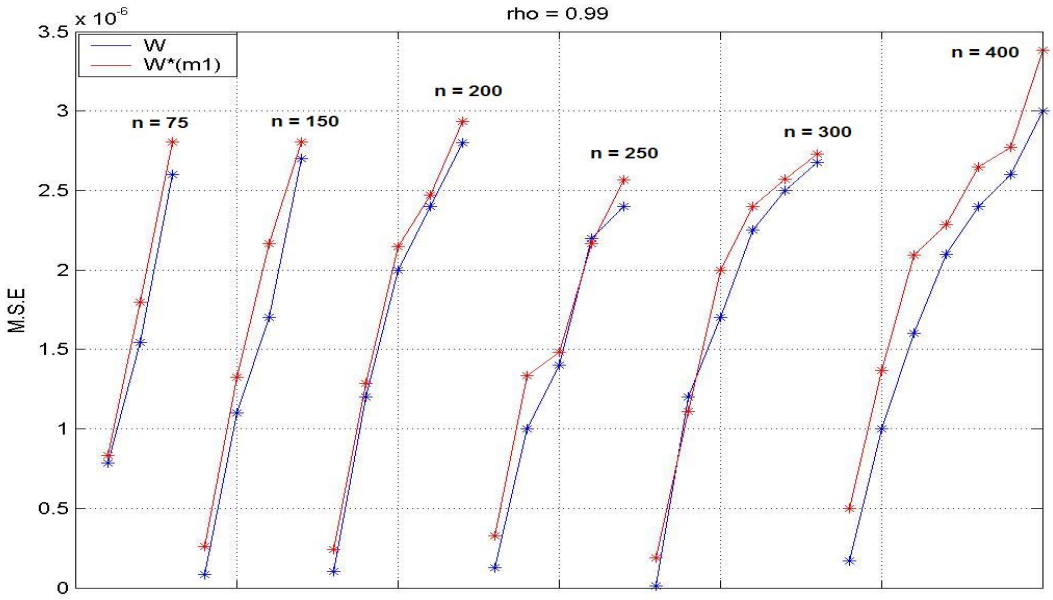

Figure 3.21: M.S.E between normalized eigenvalues of (average) $\mathbf{W}^*_{(m_1)}$, (average) $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices, ρ=0.99

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.02	-0.01	-0.005	×	×	×	×
$n = 150$	0.01	-0.006	-0.003	0	×	×	×
$n = 200$	-0.008	-0.005	0	0	0	×	×
$n = 250$	0.006	0.003	0	0	0	×	×
$n = 300$	0.005	0.003	0	0	0	0	×
$n = 400$	0.004	0.002	-0.001	0	0	0	0

Table 3.3: Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.03	-0.02	-0.005	×	×	×	×
$n = 150$	0.02	-0.006	-0.002	0	×	×	×
$n = 200$	-0.009	-0.004	0	0	0	×	×
$n = 250$	0.005	0.001	0	0	0	×	×
$n = 300$	0.004	0.003	-0.001	0	0	0	×
$n = 400$	0.004	0.002	0	0	0	0	0

Table 3.4: Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.52	0.46	0.46	×	×	×	×
$n = 150$	0.47	0.46	0.51	0.47	×	×	×
$n = 200$	0.47	0.48	0.47	0.46	0.47	×	×
$n = 250$	0.50	0.47	0.48	0.47	0.45	×	×
$n = 300$	0.51	0.52	0.50	0.48	0.47	0.46	×
$n = 400$	0.50	0.47	0.51	0.47	0.46	0.45	0.44

Table 3.5: Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.5$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.52	0.47	0.48	×	×	×	×
$n = 150$	0.50	0.49	0.50	0.51	×	×	×
$n = 200$	0.48	0.49	0.47	0.50	0.48	×	×
$n = 250$	0.51	0.47	0.48	0.48	0.50	×	×
$n = 300$	0.55	0.54	0.50	0.50	0.50	0.52	×
$n = 400$	0.52	0.48	0.51	0.49	0.47	0.52	0.52

Table 3.6: Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.5$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.93	0.91	0.88	×	×	×	×
$n = 150$	0.91	0.90	0.88	0.87	×	×	×
$n = 200$	0.88	0.89	0.89	0.86	0.86	×	×
$n = 250$	0.91	0.88	0.89	0.86	0.85	×	×
$n = 300$	0.90	0.90	0.89	0.87	0.85	0.84	×
$n = 400$	0.89	0.90	0.90	0.90	0.86	0.85	0.83

Table 3.7: Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.9$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.93	0.91	0.91	×	×	×	×
$n = 150$	0.91	0.91	0.89	0.89	×	×	×
$n = 200$	0.89	0.90	0.89	0.89	0.88	×	×
$n = 250$	0.91	0.89	0.89	0.89	0.90	×	×
$n = 300$	0.90	0.90	0.91	0.89	0.88	0.89	×
$n = 400$	0.90	0.91	0.90	0.90	0.89	0.89	0.90

Table 3.8: Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.9$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.95	0.94	0.93	×	×	×	×
$n = 150$	0.95	0.95	0.95	0.92	×	×	×
$n = 200$	0.94	0.95	0.95	0.93	0.92	×	×
$n = 250$	0.94	0.95	0.95	0.93	0.92	×	×
$n = 300$	0.95	0.95	0.95	0.94	0.92	0.92	×
$n = 400$	0.95	0.95	0.95	0.95	0.94	0.93	0.92

Table 3.9: Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.95$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.95	0.94	0.95	×	×	×	×
$n = 150$	0.95	0.95	0.95	0.94	×	×	×
$n = 200$	0.95	0.95	0.95	0.94	0.94	×	×
$n = 250$	0.95	0.95	0.95	0.94	0.94	×	×
$n = 300$	0.95	0.95	0.95	0.95	0.95	0.94	×
$n = 400$	0.95	0.95	0.95	0.95	0.95	0.95	0.94

Table 3.10: Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.95$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.99	0.99	0.99	×	×	×	×
$n = 150$	0.99	0.99	0.99	0.98	×	×	×
$n = 200$	0.98	0.99	0.99	0.98	0.98	×	×
$n = 250$	0.98	0.98	0.99	0.98	0.98	×	×
$n = 300$	0.98	0.99	0.99	0.98	0.98	0.98	×
$n = 400$	0.99	0.98	0.99	0.99	0.98	0.97	0.98

Table 3.11: Average of off-diagonal elements of $\mathbf{R}^*$, $\rho = 0.99$

	$p = 5$	$p = 10$	$p = 50$	$p = 100$	$p = 150$	$p = 200$	$p = 300$
$n = 75$	0.99	0.99	0.99	×	×	×	×
$n = 150$	0.99	0.99	0.99	0.98	×	×	×
$n = 200$	0.99	0.99	0.99	0.99	0.99	×	×
$n = 250$	0.99	0.99	0.99	0.99	0.99	×	×
$n = 300$	0.99	0.99	0.99	0.98	0.99	0.99	×
$n = 400$	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 3.12: Average of off-diagonal elements of $\mathbf{R}$, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.005	0.010	×	×
$n = 150$	0.004	0.008	0.012	×
$n = 200$	0.002	0.007	0.011	0.018

Table 3.13: Variance of normalized eigenvalues of $\mathbf{M} = \big[\mathbf{\Sigma}\big]^{-1}\big[\text{avg } \mathbf{W}^{*}_{(\mathrm{m}_1)}\big]$, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.006	0.012	×	×
$n = 150$	0.006	0.010	0.014	×
$n = 200$	0.003	0.009	0.013	0.020

Table 3.14: Variance of normalized eigenvalues of $\mathbf{M} = \big[\mathbf{\Sigma}\big]^{-1}\big[\text{avg } \mathbf{W}^{*}_{(\mathrm{m}_1)}\big]$, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.010	0.016	×	×
$n = 150$	0.008	0.010	0.018	×
$n = 200$	0.005	0.009	0.013	0.021

Table 3.15: Variance of normalized eigenvalues of $\mathbf{M} = \big[\mathbf{\Sigma}\big]^{-1}\big[\text{avg } \mathbf{W}^{*}_{(\mathrm{m}_1)}\big]$, $\rho = 0.9$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	27.86	44.76	×	×
$n = 150$	66.52	93.98	95.66	×
$n = 200$	83.76	111.10	120.44	125.22

Table 3.16: Least squares approximation of the unknown constant, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	28.92	45.17	×	×
$n = 150$	68.36	94.01	96.37	×
$n = 200$	86.21	115.14	123.42	127.98

Table 3.17: Least squares approximation of the unknown constant, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	30.11	47.21	×	×
$n = 150$	71.11	96.24	97.49	×
$n = 200$	88.76	117.20	125.35	130.93

Table 3.18: Least squares approximation of the unknown constant, $\rho = 0.9$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.12	1.21	×	×
$n = 150$	1.10	1.18	1.21	×
$n = 200$	1.05	1.11	1.16	1.32

Table 3.19: $\mathcal{V}1$ variability ratios, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.23	1.35	×	×
$n = 150$	1.16	1.27	1.31	×
$n = 200$	1.08	1.17	1.24	1.33

Table 3.20: $\mathcal{V}1$ variability ratios, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.30	1.37	×	×
$n = 150$	1.28	1.33	1.36	×
$n = 200$	1.24	1.29	1.30	1.40

Table 3.21: $\mathcal{V}1$ variability ratios, $\rho = 0.9$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.01	1.03	×	×
$n = 150$	1.00	1.01	1.02	×
$n = 200$	1.00	1.01	1.01	1.03

Table 3.22: $\mathcal{V}2$ variability ratios, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.01	1.02	×	×
$n = 150$	1.01	1.02	1.03	×
$n = 200$	1.00	1.01	1.02	1.02

Table 3.23: $\mathcal{V}2$ variability ratios, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.02	1.02	×	×
$n = 150$	1.01	1.03	1.04	×
$n = 200$	1.00	1.03	1.03	1.05

Table 3.24: $\mathcal{V}2$ variability ratios, $\rho = 0.9$

Chapter 4

Multivariate $\mathbf{W}^*_{(m_1)}$ Algorithm - Presence Of Clusters

4.1 Introduction

The previous chapter provided a systematic characterization of the $\mathbf{W}^*_{(m_1)}$ algorithm under the null-clusters scenario. We observed that the algorithm is not sensitive to the choice of its starting point thereby providing reassuring evidence that the Euclidean distance metric is as good as any other choice of distance measure in the first iteration of the algorithm. Furthermore, the analysis provided substantial indication that in the null-cluster structure scenario the resulting $\mathbf{W}^*_{(m_1)}$ estimates are reliable measures of the underlying covariance structures of the data, up to a constant scaling factor. The goal of this chapter is to extend the analysis carried out in the previous chapter, *i.e*, methodically study the starting point characteristic of the $\mathbf{W}^*_{(m_1)}$ algorithm, evaluate the quality of the covariance measure it provides, and also to study its efficacy in subsequent cluster recovery.

4.2 Sensitivity to Starting Point

A particularly problematic issue with many iterative algorithms is the sensitivity to initialization conditions. The stability of the algorithm for a large part depends on its ability to perform consistently, independent of the starting point. In this section we study the starting point characteristic of the $\mathbf{W}^*_{(m_1)}$ algorithm in the presence of clusters. As described in section 3.3, random diagonal and SPD matrix starting points are used (instead of the Identity matrix starting point) and the $\mathbf{W}^*_{(m_1)}$ algorithm is executed using the different random starting points. The diagonal and off-diagonal elements respectively of the converged $\mathbf{W}^*_{(m_1)}$ matrices corresponding to the different

starting points are then compared separately to evaluate the sensitivity of the algorithm. This will be done via simulations using a combination of simulated data sets (with carefully chosen cluster structures) and a few real data sets. The next three sections provide the experimental design, description of the data sets used and corresponding results.

4.2.1 Experimental design

A collection of simulated data sets with selectively chosen cluster structures and a few real data sets are used to evaluate the starting point issue. A brief description of the data sets used is provided in the next section. Referring back to section 3.3.1, for each data set, the $\mathbf{W}^*_{(m_1)}$ algorithm was executed using 100 random diagonal and SPD matrix starting points. The random SPD matrices were generated using equation 3.4. For each data set, the 100 resulting $\mathbf{W}^*_{(m_1)}$ matrices were then compared. As in section 3.3.1, this was done by using quantile plots of the normalized diagonal and off-diagonal elements.

The diagonal elements (w^*_{ii}, i=1...p), were first extracted from each of the converged $\mathbf{W}^*_{(m_1)}$ matrices and rescaled to sum to p. This was done, as before, to neutralize the effect of any missing constant in the $\mathbf{W}^*_{(m_1)}$ matrices and to put the sets of diagonal elements on the same footing. To then compare these sets of rescaled diagonal elements (p elements in each set) across the 100 matrices, some order statistics (maximum, upper quartile, median, lower quartile and minimum) were extracted directly from each set of (rescaled) diagonal elements and plotted. For the off-diagonal elements, as described on page 37 in section 3.3.1, all matrix components above the main diagonal were extracted from each converged $\mathbf{W}^*_{(m_1)}$ matrix and converted to obtain estimates of the within-cluster correlation, ρ. These $\binom{p}{2}$ correlation estimates, r^*_{ij}'s, were then directly compared by studying their distribution via quantile plots. As noted before, if the $\mathbf{W}^*_{(m_1)}$ algorithm is not sensitive to the starting point, then intuitively, the quantile plots (for diagonal and off-diagonal elements) would display "straight line" profiles. Also, note that in each case, m_1 is based on the $\frac{2}{3}^{rd}$ rule.

4.2.2 Description of data sets

The data sets used in this study are a combination of 14 simulated ones ($\mathcal{D}1$ - $\mathcal{D}14$), whose structures were selected to incorporate specific interesting features, and four real data sets ($\mathcal{D}15$ - $\mathcal{D}18$). All simulated data sets consist of random samples of size n and dimensionality p, drawn from a multivariate normal distribution. In each case, the cluster structure is confined to the space of the first two variables (structure dimensions), leaving the remaining $p-2$ dimensions as noise dimensions. Some of the simulated data sets involve spherically shaped clusters. To obtain data sets with ellipsoidal clusters, a transformation was applied to induce a (within-cluster) correlation of ρ=0.9 between the first two variables. Such a setup facilitates a systematic characterization of the $\mathbf{W}^*_{(m_1)}$ algorithm under a variety of carefully controlled scenarios.

Data set $\mathcal{D}1$ consists of $n = 75$ observations drawn from a spherical multivariate normal distribution. The data points were then shifted in groups of 15 only along the first two dimensions (structure dimensions), to yield five spherical homogenous clusters as shown in Figure 4.1. $\mathcal{D}2$ depicts the same cluster structure as seen in $\mathcal{D}1$, but with the cluster centers from $\mathcal{D}1$ drawn closer so that the five spherical clusters slightly overlap. For $\mathcal{D}3$, a correlation of $\rho = 0.9$ was induced between the structure dimensions of $\mathcal{D}1$, to obtain five homogenous elliptical clusters. Data set $\mathcal{D}4$ has the same cluster structure as $\mathcal{D}3$, but with the clusters from $\mathcal{D}3$ drawn closer only along its major axis of separation. Additionally, $\mathcal{D}5$, has the same structure as $\mathcal{D}4$, but with the clusters from $\mathcal{D}4$ drawn closer along its minor axis of separation, thereby causing a slight overlap among all of its five elliptical clusters. One of the reasons behind studying the specific cluster scenarios shown in data sets $\mathcal{D}2$, $\mathcal{D}4$ and $\mathcal{D}5$ is to evaluate the characteristics of the $\mathbf{W}^*_{(m_1)}$ algorithm when there is potential contamination by between-cluster point pairs caused by nearly overlapping clusters. Along similar lines, data set $\mathcal{D}6$ is an interesting test case of within-cluster outliers. Although from a clustering perspective, outliers could also be thought of as singleton clusters, it might be interesting to consider their effect when viewed as observations belonging to the individual clusters. This again considers the issue of "contamination" just discussed. Data sets $\mathcal{D}7$ - $\mathcal{D}10$ consider the

case of heterogenous clusters. Each data set is a sample from a mixture of three, equal in size, multivariate gaussians with different covariance matrices.

On the other hand, data sets $\mathcal{D}11$ - $\mathcal{D}14$ highlight perhaps an "extreme" case of singleton clusters. They all consist of n data points with $g = 5$ clusters distributed as four singleton clusters and one big cluster with the remaining $n-g+1$ data points. Data sets $\mathcal{D}11$ and $\mathcal{D}12$ consist of $n = 75$ data points with four singleton clusters and the one big cluster consisting of $n = 71$ data points. In $\mathcal{D}11$, the clusters are well separated from each other, while, in $\mathcal{D}12$, the four singleton clusters are drawn closer to the big cluster so as to cause the between-pairs to "mix in" with the within-pairs. Data sets $\mathcal{D}13$ and $\mathcal{D}14$ have the same cluster structure as $\mathcal{D}12$, except that the number of data points in the big cluster are now 146 and 196 respectively. The idea behind studying the structures in $\mathcal{D}12$ - $\mathcal{D}14$ is as follows. In such an extreme scenario of singleton clusters, the ratio of the number of between-cluster point pairs (b) to the number of within-cluster point pairs (w) is given by:

$$\frac{b}{w} = \frac{n(2g-2) - g^2 + g}{n^2 + n(1-2g) + g^2 - g} \;=\; O\big(1/n\big)$$

Hence, as n increases (from 75 data points in $\mathcal{D}12$ to 200 data points in $\mathcal{D}14$), we would expect the between-cluster point pairs to play a declining role in the $\mathbf{W}^*_{(m_1)}$ algorithm.

Among the real data sets, $\mathcal{D}15$ and $\mathcal{D}16$ are the *Iris* and *Crabs* data sets respectively. A description along with an accompanying sample of scatter plots (see Figures 2.5 and 2.6) was provided in chapter 2, in section 2.6.1. The *Wine* data set (denoted $\mathcal{D}17$) is a 13 dimensional data set ($n = 178$) consisting of three groups with 59, 71 and 48 observations in each group, respectively. It is based on the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The 13 corresponding variables are - Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline respectively. It is publicly available from the UCI Machine Learning Repository. The *Cells* ($\mathcal{D}18$) data set

(©Advanced Animal Diagnostics, NC, 2007) consists of three continuous geometrical measurements on a set of 250 cells found in milk. Each of the cells is of one of three types (Lymphocyte, Neutrophil or Macrophage). The three features were extracted using patented methods that involved computer vision techniques, optical imaging, image segmentation and feature extraction algorithms. The data set consists of 14 Macrophage, 153 Neutrophil and 83 Lymphocyte samples. The cluster structure in data sets $\mathcal{D}16$, $\mathcal{D}17$ and $\mathcal{D}18$ respectively, displays considerable cluster-overlap. Additionally, $\mathcal{D}16$ displays a high degree of intra-cluster correlation, while the variables in data set $\mathcal{D}18$ display widely differing variances.

A brief description of the 14 simulated data sets with corresponding cluster centers, and the four real data sets is given below.

- $\mathcal{D}1$ (n=75, p=5, g=5, m_1=350)

  Description: Five spherical clusters ($\rho = 0$) in two dimensions, clusters homogenous in size and dispersion, clusters well separated from each other

  Cluster centers: (0,0), (5,5), (10,10), (0,10), (10,0)

- $\mathcal{D}2$ (n=75, p=5, g=5, m_1=350)

  Description: Five spherical clusters ($\rho = 0$) in two dimensions, homogenous in size and dispersion, clusters nearly overlapping

  Cluster centers: (0,0), (2,2), (5,5), (0,5), (5,0)

- $\mathcal{D}3$ (n=75, p=5, g=5, m_1=350)

  Description: Five elliptical clusters ($\rho = 0.9$) in two dimensions, homogenous in size and dispersion, clusters well separated from each other

  Cluster centers: (0,0), (5,5), (10,10), (0,10), (10,0)

- $\mathcal{D}4$ (n=75, p=5, g=5, m_1=350)

  Description: Five elliptical clusters ($\rho = 0.9$) in two dimensions, homogenous in size and dispersion, clusters drawn closer to each other along the major of axis of separation

  Cluster centers: (0,0), (3,3), (7,7), (0,7), (7,0)

- $\mathcal{D}5$ (n=75, p=5, g=5, m_1=350)

  Description: Five elliptical clusters ($\rho = 0.9$) in two dimensions, homogenous in size and dispersion, clusters nearly overlapping

  Cluster centers: (0,0), (3,3), (6,6), (0,2), (2.5,0)

- $\mathcal{D}6$ (n=75, p=5, g=3, m_1=600)

  Description: Three spherical clusters ($\rho = 0$) in two dimensions, homogenous in size and dispersion, with few within-cluster outliers

  Cluster centers: (2,1), (5,5), (10,0)

- $\mathcal{D}7$ (n=75, p=5, g=3, m_1=600)

  Description: Three spherical clusters ($\rho = 0$) in two dimensions, homogenous in size but with different within-cluster dispersions

  Cluster centers: (0,2), (10,5), (20,0)

- $\mathcal{D}8$ (n=75, p=5, g=3, m_1=600)

  Description: Three elliptical clusters ($\rho = 0.9$) in two dimensions, homogenous in size but with different within-cluster dispersions

  Cluster centers: (0,0), (15,15), (15,0)

- $\mathcal{D}9$ (n=75, p=5, g=3, m_1=600)

  Description: One elliptical ($\rho = 0.9$) and two spherical clusters ($\rho = 0$), homogenous in size but with different within-cluster dispersions

  Cluster centers: (10,5), (15,15), (20,0)

- $\mathcal{D}10$ (n=150, p=5, g=3, m_1=2450)

  Description: Three elliptical clusters, homogenous in size and dispersion, one of the three clusters with different orientation ($\rho_1 = \rho_2 = 0.9$, ρ_3 = -0.9)

  Cluster centers: (0,0), (0,5), (5,5)

- $\mathcal{D}11$ (n=75, p=5, g=5, m_1=350)

  Description: Extreme case - Five clusters - one elliptical cluster (71 data points, $\rho = 0.9$) and four singleton clusters, well separated from each other

  Cluster centers: (5,5), (3,7), (6.5,10), (5,1.5), (7,3)

- $\mathcal{D}12$ (n=75, p=5, g=5, m_1=350)

  Description: Extreme case - Five clusters - one elliptical cluster (71 data points, $\rho = 0.9$) and four singleton clusters, cluster centers close enough to cause slight overlap of clusters

  Cluster centers: (5,5), (4,4.5), (6,6.5), (5,4), (6.5,5.5)

- $\mathcal{D}13$ (n=150, p=5, g=5, m_1=1450)

  Description: Extreme case - Five clusters - one elliptical cluster (146 data points, $\rho = 0.9$), four singleton clusters, cluster centers close enough to cause slight overlap of clusters

  Cluster centers: (5,5), (4,4.5), (6,6.5), (5,4), (6.5,5.5)

- $\mathcal{D}14$ (n=200, p=5, g=5, m_1=2600)

  Description: Extreme case - Five clusters - one elliptical cluster (196 data points, $\rho = 0.9$), four singleton clusters, cluster centers close enough to cause slight overlap of clusters

  Cluster centers: (5,5), (4,4.5), (6,6.5), (5,4), (6.5,5.5)

- $\mathcal{D}15$ - *Iris* data set (n=150, p=4, g=3, m_1=2450)

  Description: Three groups of 50 observations each, each observation measured on four variables. Clusters are homogenous in size with moderate heterogeneity of within-cluster dispersion

- $\mathcal{D}16$ - *Crabs* data set (n=200, p=5, g=4, m_1=3200)

  Description: Five measurements on 200 crabs, with four clusters of 50 observations each. Clusters are homogenous in size with high intra-cluster correlation

- $\mathcal{D}17$ - *Wine* data set (n=178, p=13, g=3, m_1=3460)

  Description: 13 continuous measurements on 178 wine samples belonging to three groups with 59, 71 and 48 samples in each group.

- $\mathcal{D}18$ - *Cells* data set (n=250, p=3, g=3, m_1=6860)

  Description: Three continuous measurements on 250 cells belonging to three groups with 14, 153 and 83 samples in each group.

Scatter plots of the simulated and real data sets are displayed in Figures 4.1 – 4.23 respectively. Note that for the simulated data sets, scatter plots are displayed only in the space of the structure variables. Furthermore, among the real data sets, for $\mathcal{D}15$, $\mathcal{D}16$ and $\mathcal{D}17$, in the interest of parsimony only a (randomly selected) sample of the scatter plots are displayed.

4.2.3 Results

The plots in Figures 4.24 – 4.33 display the order statistics of the rescaled diagonal and off-diagonal elements of the converged $\mathbf{W}^*_{(m_1)}$ matrices. As before, in all plots, the X-axis corresponds to the 100 random starting points while the Y-axis displays the corresponding order statistics. Solid lines are used to display the maximum, median and minimum values corresponding to each random starting point, while dotted lines are used to display the corresponding upper and lower quartiles, respectively.

It may be noted that in the interest of parsimony, only plots corresponding to the random SPD matrix starting points are displayed here. Results for the random diagonal matrix starting points are similar to those displayed. Also, it is important to note that in Figures 4.24 – 4.28, where the diagonal elements are compared, the order statistics corresponding to each starting point are based on only p elements as there are p diagonal elements in each $\mathbf{W}^*_{(m_1)}$ matrix. Similarly, in Figures 4.29 – 4.33, where the off-diagonal elements are compared, the order statistics corresponding to each starting point are based on $\binom{p}{2}$ elements.

The plots again display a strikingly low variablility among the order statistics across the different random starting points. This suggests that the $\mathbf{W}^*_{(m_1)}$ algorithm converges to *almost* the same matrix, regardless of what starting point is applied, even in the presence of clusters. Being an iterative algorithm, this is a particularly desirable characteristic that will be reassuring to practitioners who might be concerned about the dependence of the algorithm on the starting values. This observation along with the similar set of results observed under the null clusters scenario reported in the previous chapter, forms an important finding - the $\mathbf{W}^*_{(m_1)}$ algorithm is not sensitive to its

initialization, irrespective of the cluster structure.

4.3 Quality of $\mathbf{W}^*_{(m_1)}$ estimates

In section 3.4 we evaluated the quality of the $\mathbf{W}^*_{(m_1)}$ estimates under the null-clusters scenario by studying its proximity to the known $\mathbf{W}$ and $\mathbf{\Sigma}$ matrices. The intuition behind such an analysis was to examine to what extent the $\mathbf{W}^*_{(m_1)}$ estimate (apart from a constant multiplier) serves as an effective proxy for the known $\mathbf{W}$ matrix. The results provided reasonable evidence that the $\mathbf{W}^*_{(m_1)}$ matrix not only serves as a reliable substitute for the known $\mathbf{W}$ matrix, but also tracks the underlying covariance structure, $\mathbf{\Sigma}$, commendably. The goal in this chapter is to evaluate the extent to which these results carry over to the situation where clusters are present in the data. In what follows, we explain the experimental design, supplemented by results from simulations.

4.3.1 Experimental design

To evaluate the quality of the $\mathbf{W}^*_{(m_1)}$ estimates in the presence of clusters, a combination of real and simulated data sets was used. We started with the simulated data sets $\mathcal{D}1$–$\mathcal{D}14$. Section 4.2.2 provided a complete description of these data sets. They ranged from the simple case of homogenous or near homogenous cluster structures to more complex heterogenous cluster structures. As noted earlier, the goal is to evaluate the quality of the $\mathbf{W}^*_{(m_1)}$ matrices under a variety of controlled cluster scenarios, including those with departures from the idealized assumptions of homogeneity.

Additionally, it may be noted that for a given number of observations (n), dimensionality (p), number of clusters (g), underlying covariance structure ($\mathbf{\Sigma}$) and cluster centers, as we vary the numbers of within-cluster observations among clusters, n_i, $i = 1 \ldots g$, the least favorable case for the $\mathbf{W}^*_{(m_1)}$ algorithm would be one with equal-sized clusters, *i.e.*, $(n_1 = n_2 \ldots = n_g)$, because this is the situation when the total number of within-pairs (w), is at its lowest, $\left(w = \frac{n(n-g)}{2g}\right)$. Thus, the risk of contamination of the algorithm by between-pairs would be the greatest.

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\surd$	$\surd$	$\times$	$\times$
$n = 150$	$\surd$	$\surd$	$\surd$	$\times$
$n = 200$	$\surd$	$\surd$	$\surd$	$\surd$

Table 4.1: Data sets with the different (n, p) combinations used in this study

Hence, in addition to data sets $\mathcal{D}1$ – $\mathcal{D}14$, to systematically study the issue of potential contamination of within-cluster point pairs by between-cluster pairs, we simulated data sets with a fixed number of equal sized clusters ($g = 5$, and $n_i = \frac{n}{g}$, i=1 ... g), but varied n (75, 150 and 200), p (5, 50, 100 and 150) and $\mathbf{\Sigma}$. As in the previous chapter, the $\mathbf{\Sigma}$ matrix was *compound symmetric*, with three different values of the common correlation ρ (0, 0.5 and 0.99), while the variances were all set equal to one. Moreover, to include the effects of *noise*, the data sets were patterned after the test sets used in [40], where the cluster structure is confined to the space of only a few dimensions, leaving the remaining dimensions as noise coordinates. To do this, for each (n, p, ρ) combination as given above, random samples of equal size (n_i), were drawn from five, $\frac{p}{2}$-dimensional multivariate Gaussian distributions. Subsequently, an additional $\frac{p}{2}$ coordinates were added to each data set by including $\frac{p}{2}$ independent univariate standard normal random variables. Hence, approximately half the number, p, of variables had cluster structure (structure variables), while the others were devoid of any cluster structure and served purely as noise variables (Note: When $p = 5$, the number of structure dimensions was set equal to three, leaving two noise dimensions). Table 4.1 displays the nine different combinations of n and p used for the simulated data sets. Note that in each case, n was always chosen greater than p, so as to guard against any potential singularity of the resulting $\mathbf{W}^*_{(m_1)}$ matrices. So, for instance, when $n = 75$, only two values of p ($p = 5$ and 50) were used, as displayed in Table 4.1 (marked as "$\surd$" in the first row of Table 4.1)

Furthermore, to specifically evaluate the impact of potential contamination by between-cluster pairs of points, two different sets of data were generated using the random simulation model just described. In the first set (Set 1), the cluster means were randomly set far apart from each other, so that the possibility of contamination

by between-group pairs is minimized. In the second set (Set 2), the cluster means were randomly set close to each other, causing the clusters to nearly overlap. These two conditions could be viewed as two possible extreme scenarios within an entire spectrum of data sets with shrinking cluster separations. A pair of scatter plots (in the space of two structure variables) of a sample data set from each the two sets is displayed in Figures 4.34 and 4.35 respectively. Notice the two cases when the clusters are well separated and when they nearly overlap.

Hence, with nine different (n, p) combinations, three different values of the common correlation ρ and two sets of data (*i.e.*, Set 1 - well separated clusters and Set 2 - nearly overlapping clusters), we had $9 \times 3 \times 2 = 54$ different experimental conditions to evaluate the quality of the $\mathbf{W}^*_{(m_1)}$ matrix.

Finally, to supplement the results obtained from simulations, four real data sets were also used. Data sets $\mathcal{D}15$ - $\mathcal{D}18$, constituted the real data sets and served as a useful test bed for evaluation. Section 4.2.2 provided a brief description of the real data sets along with a few accompanying scatter plots. The following summarizes the data sets described so far:

1. Simulated data
    - Group 1 - Data sets $\mathcal{D}1$ - $\mathcal{D}14$, displaying a variety of cluster scenarios ranging from simple homogenous clusters to heterogenous structures.
    - Group 2 - Data sets from random simulations with different (n, p, ρ) combinations and two types of cluster separations
        - Set 1: 27 data sets with five well separated clusters
        - Set 2: 27 data sets with five nearly overlapping clusters
2. Real data
    - $\mathcal{D}15$ (*Iris* data)
    - $\mathcal{D}16$ (*Crabs* data)
    - $\mathcal{D}17$ (*Wine* data)

- $\mathcal{D}18$ (*Cells* data)

To study the quality of the $\mathbf{W}^*_{(m_1)}$ estimates, the $\mathbf{W}^*_{(m_1)}$ algorithm was first executed for each data set. Since the starting point invariance characteristic of the algorithm was already established, Euclidean distance was the choice of the metric used in the first iteration of the algorithm. Also, in all cases the choice of m_1, was based on the $\frac{2}{3}^{rd}$ rule, while the convergence constant $\mathbf{E}$, was set equal to 0.001. It may be noted that for the real data sets, $\mathcal{D}17$ and $\mathcal{D}18$, which have unequal cluster sizes, the corresponding values for m_1 (using the $\frac{2}{3}^{rd}$ rule) were based on an assumption of equal cluster sizes. Additionally, the known $\mathbf{W}$ matrix was also computed for each data set. In the case of the simulated data sets in group 2, for each of the 54 (n, p, ρ, cluster means) combinations, 100 random replicates were generated and the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices were computed for each of the 100 random replicates. The 100 resulting $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices were then averaged respectively, so as to smooth out the random variation arising from the sampling process. Hence, we have 14 $\mathbf{W}^*_{(m_1)}$ and 14 $\mathbf{W}$ matrices corresponding to the simulated data sets in group 1, 54 (averaged) $\mathbf{W}^*_{(m_1)}$ and 54 (averaged) $\mathbf{W}$ matrices corresponding to the simulated data sets in group 2, and finally, 4 $\mathbf{W}^*_{(m_1)}$ and 4 $\mathbf{W}$ matrices corresponding to the real data sets. Comparison of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices could then be performed in the four following ways:

- Convert each $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrix to corresponding correlation matrices $\mathbf{R}^*$ and $\mathbf{R}$ respectively. Compute the mean square error (M.S.E) between the off-diagonal elements (correlation estimates) of $\mathbf{R}^*$ and $\mathbf{R}$. This would give an idea of the extent to which the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices compare in capturing the correlational structure of the data. Specifically, for the data sets in group 2, we could work with the averaged matrices in each case.

- Study the eigenvalues of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices. As described in section 3.4.1, if the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices are approximately proportional (apart from an unknown constant), then the respective eigenvalues of the two matrices would also be approximately proportional to each other. To make the eigenvalues of the two matrices invariant to the unknown constant, the eigenvalues could be

normalized by their respective means. This would put the sets of eigenvalues on the same footing, ready for comparison. Subsequently, the variances of the normalized sets of eigenvalues could be compared. Specifically, for the group 2 data sets, variance of the normalized eigenvalues of the averaged matrices could be compared.

- To additionally check for any inherent biases between the eigenvalues of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices, the M.S.E between the eigenvalues of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices could be computed. As before, to render the eigenvalues invariant to any unknown constant, the sets of eigenvalues could be normalized by their respective means prior to the M.S.E computations. Again, specifically for the simulated data sets in group 2, the average $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices could be used for the analysis.

- Finally, to evaluate the performance of the $\mathbf{W}^*_{(m_1)}$ estimates in aiding recovery of the clusters, each data set could first be scaled (sphericized) by its corresponding $\mathbf{W}^*_{(m_1)}$ matrix. The sphericized data could then be applied as input to a hierarchical cluster analysis (HCA) using average linkage, as described earlier in section 2.6.2. The dendrograms could be examined prior to cutting the tree and the resulting cluster labels could be compared with the known cluster memberships to obtain an errors of misclassification count. For the simulated data sets in group 2, this could be automated to be done separately on each of the 100 random replicates, using the $\mathbf{W}^*_{(m_1)}$ matrix specific to each random replicate. This would lead to an average error count (averaged over the 100 replicates) in each case. This procedure could then be repeated for each of the 54 (n, p, ρ, cluster centers) combinations of data sets and an average error count for each of the 54 combinations could be obtained. The same process could also be repeated using the respective $\mathbf{W}$ matrices and the performance of the $\mathbf{W}^*_{(m_1)}$ estimates could be compared to that of the $\mathbf{W}$ matrices by examining their relative error counts. Note that the application of HCA involves cutting the resulting dendrograms to yield a partition of the data. However, with data sets that have unequal cluster sizes (such as $\mathcal{D}11$ - $\mathcal{D}14$, $\mathcal{D}17$ and $\mathcal{D}18$), it is known that "tree cutting" could

be tricky. Nevertheless, since the goal of this analysis is to primarily perform a *comparative* study of the characteristics of the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices, we are only interested in examining the *relative* performances of the two methods using *any* clustering procedure.

4.3.2 Results

As described in the previous section, for each data set, the M.S.E between the correlation estimates from the $\mathbf{R}^*$ and $\mathbf{R}$ matrices were first computed. Table 4.2 displays the M.S.E values for data sets $\mathcal{D}1$ - $\mathcal{D}18$, *i.e.*, the simulated data sets in group 1 and the four real data sets. The small M.S.E values are indicative of the fact that the resulting $\mathbf{R}^*$ estimates are *close* to the known $\mathbf{R}$ matrices in all cases, including the cases with heterogenous cluster structures. Even in the case of $\mathcal{D}10$, where one of the three ellipsoidal point clouds has a different orientation, the signs of the correlation estimates in the $\mathbf{R}^*$ matrix are the same as those in the corresponding $\mathbf{R}$ matrix, indicating that the $\frac{2}{3}^{rd}$ rule has worked well in selecting the appropriate point pairs to compute the $\mathbf{W}^*_{(m_1)}$ matrix. Furthermore, Table 4.3 provides additional evidence that the $\mathbf{W}^*_{(m_1)}$ matrices do a good job in tracking the known $\mathbf{W}$ matrices for most of the data sets. However, as one would expect, with more challenging cluster structures such as in $\mathcal{D}2$, $\mathcal{D}4$ and $\mathcal{D}5$, where the clusters centers are close enough to nearly cause cluster overlap, the $\mathbf{W}^*_{(m_1)}$ estimates tend to deteriorate. In such cases, choosing a smaller value of m_1 yields $\mathbf{W}^*_{(m_1)}$ estimates that approximates the known $\mathbf{W}$ matrices more accurately. The merits of choosing a smaller value of m_1 in cases where the clusters are closer, is discussed next. Similar results are observed in the M.S.E values given in Table 4.4.

Among the error counts shown in Table 4.5, we observe that the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm in leading to cluster recovery is closely tied with the performance of the known $\mathbf{W}$ matrix. It is important to note that in data sets where there is sufficient separation among clusters, the $\mathbf{W}^*_{(m_1)}$ algorithm is extremely efficient as a preprocessing step. However, as the cluster separation decreases, the inclusion of many between-cluster point pairs might be problematic. Even so, as mentioned earlier, in data sets $\mathcal{D}2$, $\mathcal{D}4$ and $\mathcal{D}5$, where the clusters nearly overlap, choosing a smaller value of

m_1 remarkably improves results. A case in point is data set $\mathcal{D}4$, where three elliptical clusters lie very close to each other along the major axis of separation. The value of m_1 ($m_1 = 350$) based on the $\frac{2}{3}^{rd}$ rule leads to poor performance of the $\mathbf{W}^*_{(m_1)}$ algorithm, resulting in 29 errors of misclassification. However, setting m_1 to any smaller value in the range 150 - 225 lowers the error count to two, comparable to that given by the known $\mathbf{W}$ matrix. One possible geometrical explanation for this is as follows. It is known that the volume of an ellipsoid is directly proportional to the determinant of the matrix describing it. Hence, when computing the $\mathbf{W}^*_{(m_1)}$ matrices, as m_1 increases, the volume of the corresponding ellipsoid it describes also increases as point pairs lying farther apart are included in the computation, thereby leading to smaller Mahalanobis inter-point distances. Additionally, the orientation of the ellipsoid (described by the off-diagonal elements of $\mathbf{W}^*_{(m_1)}$) could change depending on the point pairs entering the $\mathbf{W}^*_{(m_1)}$ computation. This sensitivity in Mahalanobis distances might have been emphasized for the cluster structure in data set $\mathcal{D}4$, wherein, the choice of $m_1 = 350$ results in smaller Mahalanobis inter-point distances for the data points distributed among the three clusters along the major axis of separation. This tendency to emphasize diminution of the distances might have caused the confusion in subsequent cluster recovery. As a result, all of the 29 errors arise from the three elliptical collinear clusters. Furthermore, recall that the true underlying correlation between the structure variables in $\mathcal{D}4$ is, $\rho = 0.9$. However, the correlation estimate given by the $\mathbf{W}^*_{(m_1)}$ algorithm using $m_1 = 350$ (based on the $\frac{2}{3}^{rd}$ rule), is $r^*_{(1,2)}(350) = 0.9799$, while using $m_1 = 150$ yields $r^*_{(1,2)}(150) = 0.8971$, which is closer to the underlying ρ. Hence, it is evident that the "bias" in using the estimate from the $\frac{2}{3}^{rd}$ rule is larger compared to using the smaller value of m_1. This would be a major reason for the poor performance of the $\mathbf{W}^*_{(m_1)}$ algorithm in the case of $\mathcal{D}4$, when the $\frac{2}{3}^{rd}$ rule is used to pick m_1. Hence, setting $m_1 = 150$ mitigates this problem, yielding near perfect cluster recovery. Similar improvements in error are observed for data sets $\mathcal{D}2$ and $\mathcal{D}5$, where smaller values of m_1 leads to more accurate $\mathbf{W}^*_{(m_1)}$ estimates, thereby resulting in error counts even lower than using the known $\mathbf{W}$ matrix. In $\mathcal{D}2$, for instance, setting $m_1 = 150$, lowers the error count from 36 to eight. Similarly, in data set $\mathcal{D}5$, $m_1 = 150$, reduces the error

count of 31, down to 12.

Likewise, among the real data sets, in case of $\mathcal{D}15$ (*Iris* data), using a value of m_1 ($m_1 = 1850$), smaller than that given by the $\frac{2}{3}^{rd}$ rule, lowers the error count from 39 (as reported in Table 4.5) to four (equal with $\mathbf{W}$). Here again, the correlation given by the known $\mathbf{W}$ matrix, $r_{(3,4)}$, between variable 3 (petal length) and variable 4 (petal width) is 0.4845. Using the $\frac{2}{3}^{rd}$ rule value of m_1 (2450), the correlation estimate from the $\mathbf{W}^*_{(m_1)}$ matrix is, $r^*_{(3,4)}(2450) = 0.6380$. However, using a smaller value of m_1 (1850), yields a more accurate, $r^*_{(3,4)}(1850) = 0.5461$. Hence, as mentioned earlier, while the $\frac{2}{3}^{rd}$ rule provides some guidance in picking m_1, it is always prudent to try a range of different values. Furthermore, in situations where the clusters are close to each other, a more conservative, smaller choice for m_1 would prove beneficial in order to avoid the "bias" caused by "contamination" by between-cluster pairs of data points. In case of data sets $\mathcal{D}17$ (*Wine* data) and $\mathcal{D}18$ (*Cells* data), as noted earlier, the value of m_1 (using the $\frac{2}{3}^{rd}$ rule), was based on a naive assumption of *equal* cluster sizes. However, since in these data sets, the clusters actually vary significantly in size, the $\frac{2}{3}^{rd}$ rule value of m_1 might not be an appropriate choice. Additionally, as mentioned earlier, cluster extraction based on tree cutting could be tricky with unequal cluster sizes. These two reasons partly explain the poor results of the $\mathbf{W}^*_{(m_1)}$ algorithm compared to using the known $\mathbf{W}$ matrix in $\mathcal{D}17$ and $\mathcal{D}18$, as shown in Table 4.5.

In data sets $\mathcal{D}9$ and $\mathcal{D}10$ that display heterogeneity of dispersion and orientation, the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm is commensurate with that of the known $\mathbf{W}$ matrices, as long as there is good cluster separation. However, based on limited experiments, if the clusters in $\mathcal{D}9$ and $\mathcal{D}10$ are brought any closer, the $\mathbf{W}^*_{(m_1)}$ algorithm performs poorly.

Tables 4.6 - 4.29 display results for the simulated data sets in group 2. A common feature gleaned from the tables is the good performance of the $\mathbf{W}^*_{(m_1)}$ algorithm as measured by its *closeness* with the known (average) $\mathbf{W}$ matrix, especially when the clusters are well separated from each other. This is first seen in the small M.S.E values between the corresponding (average) $\mathbf{R}^*$ and (average) $\mathbf{R}$ matrices. However, even when the clusters nearly overlap (Set 2), reasonably accurate correlations are captured by the

$\mathbf{W}^*_{(m_1)}$ algorithm. This indicates that the $\frac{2}{3}^{rd}$ rule performs well when there is good separation among clusters, notwithstanding the presence of noise. This is also observed in Tables 4.12 - 4.17 which compares the variances of the normalized eigenvalues of the (average) $\mathbf{W}^*_{(m_1)}$ and (average) $\mathbf{W}$ matrices. The variability in the numbers, however, does increase as n and p approach nearly the same order of magnitude.

Results from a subsequent clustering of the data sets (Tables 4.24 - 4.29) indicates that even in the presense of noise variables, the $\mathbf{W}^*_{(m_1)}$ algorithm performs creditably when the clusters are well separated. However, the difficulty caused when the clusters are drawn closer to each other is reflected in the higher error counts, across all values of the common correlation, ρ. Careful choice of a smaller value of m_1 would prove beneficial in such situations.

The algorithm converges in 7 - 31 iterations across all data sets. Additionally, it is important to point out that on selected experiments (not reported here) where the clusters overlap to a higher degree than those reported here in group 2, the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm seems to progressively worsen, leading to a "break-down" of the method due to the inclusion of an increasing number of between-cluster pairs of points. This is a very challenging scenario for any clustering procedure and, indeed, such difficulties of the real world have no simple solutions. However, it is important to note that the "break-down" point as it relates to the degree of overlap among the clusters, is dependent on the choice of m_1. This again points to the importance of using a range of values for m_1 in practice.

Furthermore, it may be noted that for a fixed number of data points n, and a fixed number of clusters g, with n_i, $i = 1 \ldots, g$ data points in each cluster, the number of within-cluster point-pairs increases as the *variability* in the distribution of n_i increases. Intuitively in such a scenario, the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm would improve since the number of between-cluster pairs is progressively minimized. This was also verified via limited simulations performed for random data (n=200, p=5, g=5) as n_i was varied from when the clusters were of equal size to the case when the clusters were of widely differing sizes. Performance of the algorithm (as measured by the accuracy of the $\mathbf{W}^*_{(m_1)}$ estimates and subsequent misclassification error) improves as the degree

of *discordance* among n_i, increases.

4.3.3 "Missing" constant

In section 3.4.3, we studied the adequacy of a single multiplicative constant to make the expected value of the $\mathbf{W}^*_{(m_1)}$ matrix an estimate of $\mathbf{\Sigma}$ that is more "comparable" to $\mathbf{W}$, in the null-clusters scenario. We observed that in the simple null-clusters case, apart from a single constant multiplier, the $\mathbf{W}^*_{(m_1)}$ matrix indeed, provided a very reliable estimate of $\mathbf{\Sigma}$. We could perform a similar analysis to check if the adequacy of only a single, albeit unknown, constant holds true in the more complex, clustering context. Although in the presence of clusters the characteristics of the missing constant would be more complex to study, in part due to its unknown dependency on parameters such as $n_{(i)}$, p, Σ and cluster means, it would nevertheless be interesting to verify if our intuition regarding the adequacy of only a single multiplier is indeed, true.

Given the average $\mathbf{W}^*_{(m_1)}$ matrix corresponding to each (n, p, ρ) combination, we could consider the matrix product, $\mathbf{M} = \big[\mathbf{\Sigma}\big]^{-1}\big[\mathbf{W}^*_{(m_1)}\big]$. As discussed in section 3.4.3, except for a single constant multiplier, intuitively if the average value of the $\mathbf{W}^*_{(m_1)}$ matrix is close to $\mathbf{\Sigma}$, then the matrix $\mathbf{M}$ would be (approximately) proportional to the identity matrix, *i.e.*, $\mathbf{M} \approx k\mathbf{I}$. To examine this, we could extract the eigenvalues of $\mathbf{M}$, normalize them to sum to one, and then compute their variance as done in the previous chapter. If only a single constant multiplier is adequate, then the variances as computed above would all be expected to lie close to zero. Table 4.30 displays the corresponding variances of the normalized eigenvalues of $\mathbf{M}$ for the simulated data sets in group 2 (see page 76 for a description of the data sets). Recall that all the simulated data sets in group 2 have five homogenous, equal sized clusters.

The relatively small variances seen in Table 4.30 are indicative of the fact that even in the presence of clusters (at least when there is a sufficient degree of inter-cluster separation), only a single multiplicative constant might be necessary to render $\mathbf{W}^*_{(m_1)}$ more comparable to $\mathbf{\Sigma}$. In the case where the clusters overlap slightly (second half of Table 4.30), as one would expect, the variance values are higher. However, it is important to note that all the results given here correspond to m_1 based on the $\frac{2}{3}^{rd}$

rule, while, using a smaller value of m_1 might improve results in such scenarios. This is a powerful feature of the $\mathbf{W}^*_{(m_1)}$ algorithm. Although the challenges associated with real data might be more complex to handle, it is nevertheless reassuring to know that at least in the simple homogenous clustering case just studied, only a single constant multiplier might suffice to make the $\mathbf{W}^*_{(m_1)}$ matrix an approximately unbiased estimate of the within-clusters covariance structure.

Furthermore, as reported in section 3.4.3, if we were to use the least squares estimate of the unknown constant, k, we would set $k = \overline{\lambda}$, where $\overline{\lambda}$ is the mean eigenvalue of the matrix, $\mathbf{M}$ (see page 44 in section 3.4.3 for more details). Table 4.31 provides the least squares estimates of the unknown constant. Again, the average $\mathbf{W}^*_{(m_1)}$ matrix corresponding to each (n, p, ρ) combination is used to compute $\mathbf{M}$ in all cases.

There are two sources of bias in the estimate, $\mathbf{W}^*_{(m_1)}$, of the single underlying covariance structure, $\mathbf{\Sigma}$. Their effects are important depending on the nature of the clusters that are present. The two intertwined sources are: (a) smallness of m_1 relative to the unknown number of within-cluster pairs, and (b) the contamination of $\mathbf{W}^*_{(m_1)}$ by inappropriate inclusion of between-cluster pairs, which is a consequence of choosing m_1 to be large.

When there are no clusters (the null case considered in chapter 3), or when they are well separated (in some subspace of the variables) and relatively homogenous in their dispersions, the results show that a single constant multiplier would be adequate to adjust for bias due to (a). This is a very useful finding in and of itself. The results also indicate that, for a given m_1, the constant varies with n and p, and to a lesser extent, with ρ. Although the least squares estimates displayed in Table 4.31 provides a first cut approximation for the unknown constant, if one were actually interested in the value of the constant, further work assisted by some theory and asymptotics would be needed. This is, however, beyond the scope of this thesis.

The second source of bias becomes more critical when the cluster structure in the data is not well defined. As the results demonstrate, when the clusters are moved closer together in the space of the structure variables, the choice of m_1 becomes very critical. With real data and unknown cluster structure, one should expect to have difficulty with

misclassifying data anyway. However, a single constant might not adjust for this type of difficulty. Nevertheless, one important finding is that in such situations, a smaller choice of m_1 than one based on any rule of thumb (such as the $\frac{2}{3}^{rd}$ rule) would be valuable to handle this case.

4.3.4 Variability of $\mathbf{W}^*_{(m_1)}$ estimates

To study the variability of the $\mathbf{W}^*_{(m_1)}$ matrices in the presence of clusters, we could again parallel the design procedure described in section 3.4.4, by comparing the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices for each of the 100 random replicates belonging to each (n, p, ρ) combination. As done before, we could first convert these to corresponding correlation matrices, $\mathbf{R}^*$ and $\mathbf{R}$ respectively. Subsequently, for every $(i,j)^{th}$ pair of structure variables, we could consider the ratio, $\mathcal{V}1_{(ij)} = \dfrac{var[r^*_{(i,j)}]}{var[r_{(i,j)}]}$, where the variances are computed across the 100 random samples. Such a statistic would provide more insight into the variability of the correlation estimates (between any pair of strucure variables) in the $\mathbf{W}^*_{(m_1)}$ matrices relative to the $\mathbf{W}$ matrices. Table 4.32 provides the $\mathcal{V}1$ variability ratio just described. In all cases we have computed the variability measure for the correlation estimate between structure variables 1 and 2, *i.e*, using $r^*_{(1,2)}$ and $r_{(1,2)}$. Hence, the results in Table 4.32 are specifically for $\mathcal{V}1_{(1,2)}$.

The relatively near unity variance ratios in the first half of Table 4.32 indicates that when there is sufficient separation among the clusters, the variability of the correlation estimates between structure variables 1 and 2 in the $\mathbf{W}^*_{(m_1)}$ matrices is almost the same as those in the known $\mathbf{W}$ matrices. An identical analysis on the other structure variables also resulted in the same conclusion. Also, when the cluster centers are brought closer, as expected, the nearly overlapping clusters contributes to a marginal increase in the variability of the $\mathbf{W}^*_{(m_1)}$ matrices.

4.3.5 Experiments using gaussian mixture model-based clustering – (MCLUST)

It is important to note that all the error counts reported so far are based on a HCA. However, since the $\mathbf{W}^*_{(m_1)}$ algorithm is not tied to any particular method of clustering, we also studied its performance using the gaussian mixture model-based approach, MCLUST (also see [24], [25] and [26]). Experiments were performed in concert on the four real data sets using the $\mathbf{W}^*_{(m_1)}$ estimates to preprocess the raw data before their input to MCLUST. We forced MCLUST to extract the known number of clusters, while we let it pick the best covariance matrix parameterization in each case. Subsequently, the resulting cluster labels were compared with the known labels to get the misclassification error counts as given. The first number on each row indicates the error count using the raw data that was directly input to MCLUST, while the second number (within parentheses) indicates the error count from MCLUST after $\mathbf{W}^*_{(m_1)}$ preprocessing. It may be noted that the error counts given below are based on choices of m_1 that yielded the most favorable result in each case. As can be seen, the results clearly establish superior performance when the data is preprocessed using $\mathbf{W}^*_{(m_1)}$.

- $\mathcal{D}15$ (*Iris*): 4 (4)
- $\mathcal{D}16$ (*Crabs*): 79 (15)
- $\mathcal{D}17$ (*Wine*): 5 (1)
- $\mathcal{D}18$ (*Cells*): 121 (37)

4.4 Conclusion

This chapter studied some of the characteristics of the $\mathbf{W}^*_{(m_1)}$ algorithm in the presence of cluster structure. The issue of the algorithm's sensitivity to its starting point was systematically studied under a variety of cluster scenarios. The results in section 4.2.3 indicated that the $\mathbf{W}^*_{(m_1)}$ algorithm is relatively invariant to its initialization. This result, coupled with the null-clusters results presented in section 3.3.2 provided substantial evidence that the $\mathbf{W}^*_{(m_1)}$ algorithm is not dependent on its starting point, irrespective of the cluster structure of the data. Section 4.3 subsequently provided a methodical study of the performance of the algorithm in the presence of homogeneity as well as moderate heterogeneity in cluster structure. Although, as one would expect, the $\mathbf{W}^*_{(m_1)}$ algorithm worked best in the homogenous case, its performance was commendable in the presence of moderately heterogenous clusters as well.

Additionally, the $\frac{2}{3}^{rd}$ rule worked remarkably well when there was good separation among clusters. However, when the clusters were brought closer together, the inclusion of between-cluster pairs of observations into the $\mathbf{W}^*_{(m_1)}$ iterations led to slight deterioration of the $\mathbf{W}^*_{(m_1)}$ estimates. However, as discussed earlier, in such situations, choosing smaller values of m_1 than the one arrived at by the $\frac{2}{3}^{rd}$ rule, could help avoid the contamination caused by the inclusion of between-cluster pairs and decrease the bias of the estimate.

Regarding the practical feasibility of the method, in most of the situations studied, using the estimate of the within-clusters covariance matrix provided by $\mathbf{W}^*_{(m_1)}$ works as well as using the known pooled within-groups covariance matrix, $\mathbf{W}$. Overall, the performance of the $\mathbf{W}^*_{(m_1)}$ algorithm, including its starting point invariance characteristic, is concluded to work well in the presence of clusters and the intercorrelations among the variables.

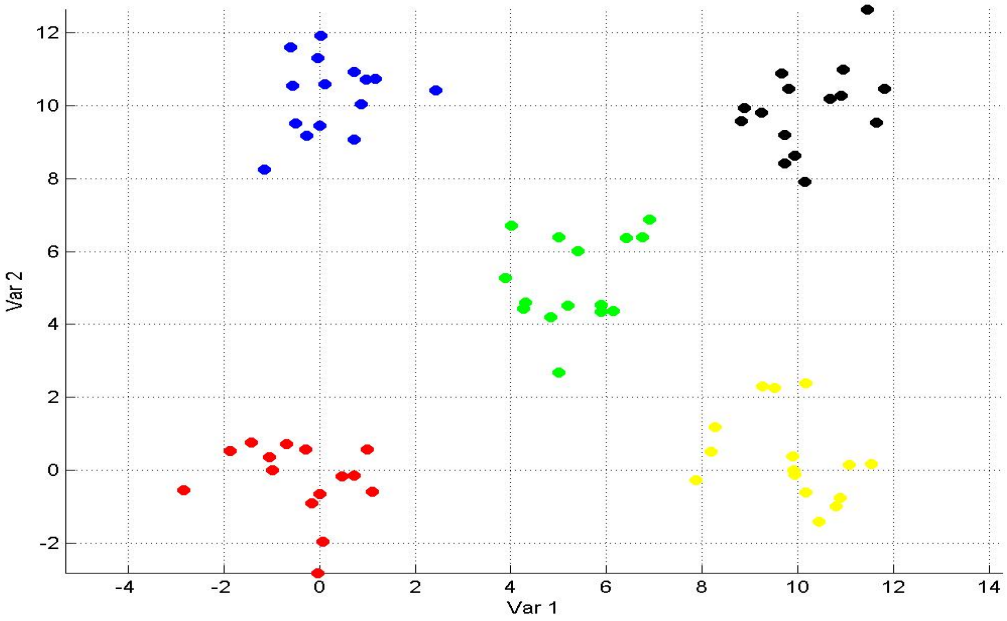

Figure 4.1: Cluster structure for $\mathcal{D}1$

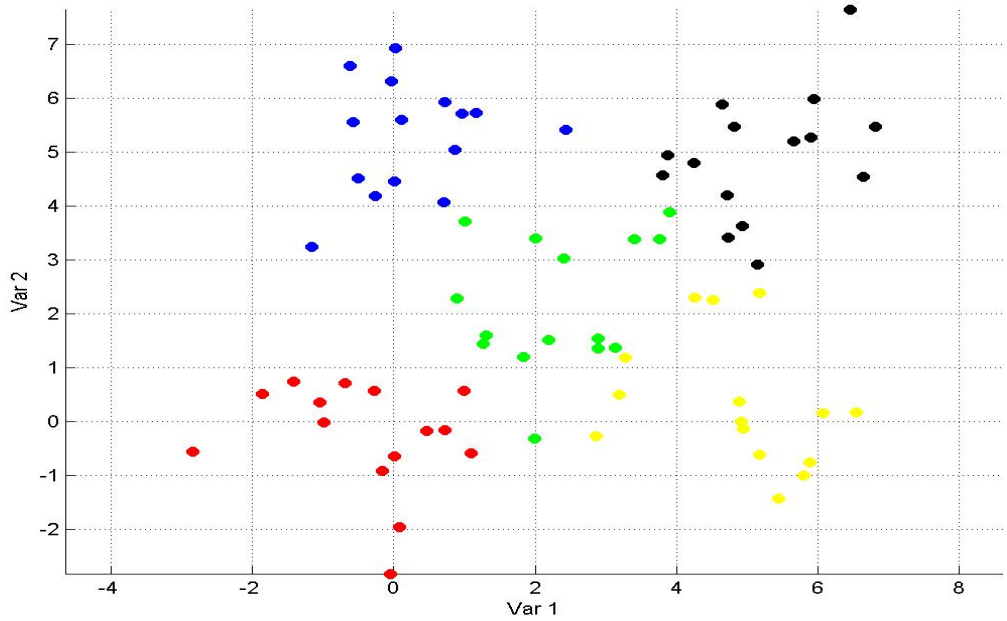

Figure 4.2: Cluster structure for $\mathcal{D}2$

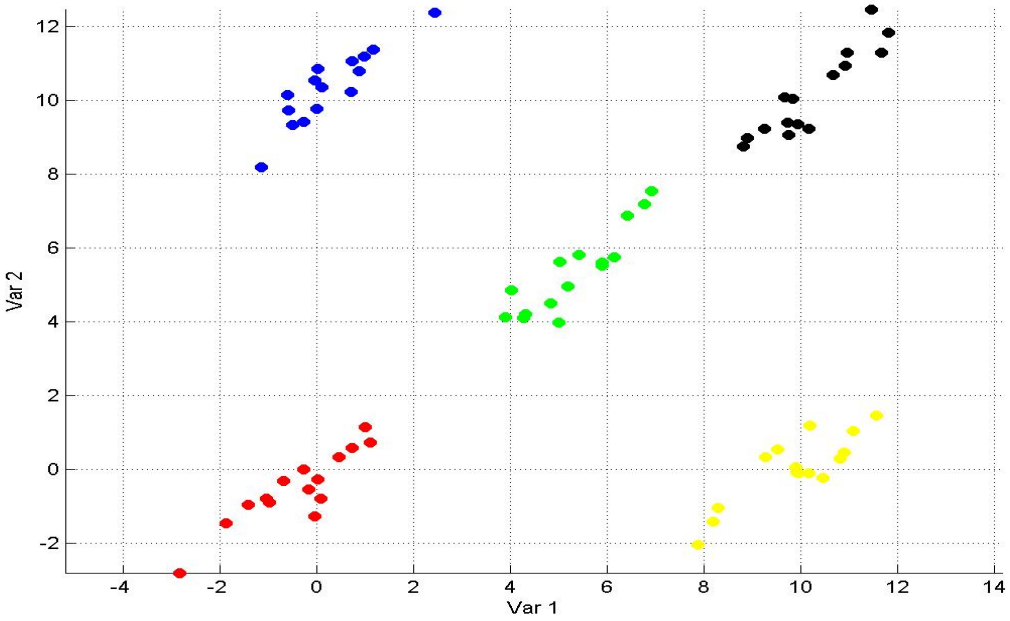

Figure 4.3: Cluster structure for $\mathcal{D}3$

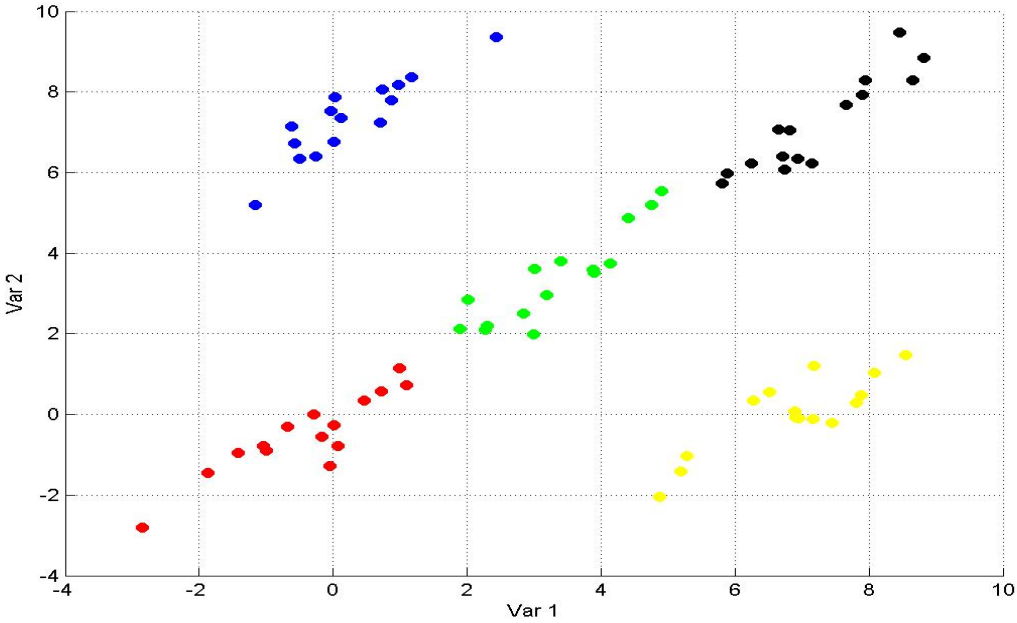

Figure 4.4: Cluster structure for $\mathcal{D}4$

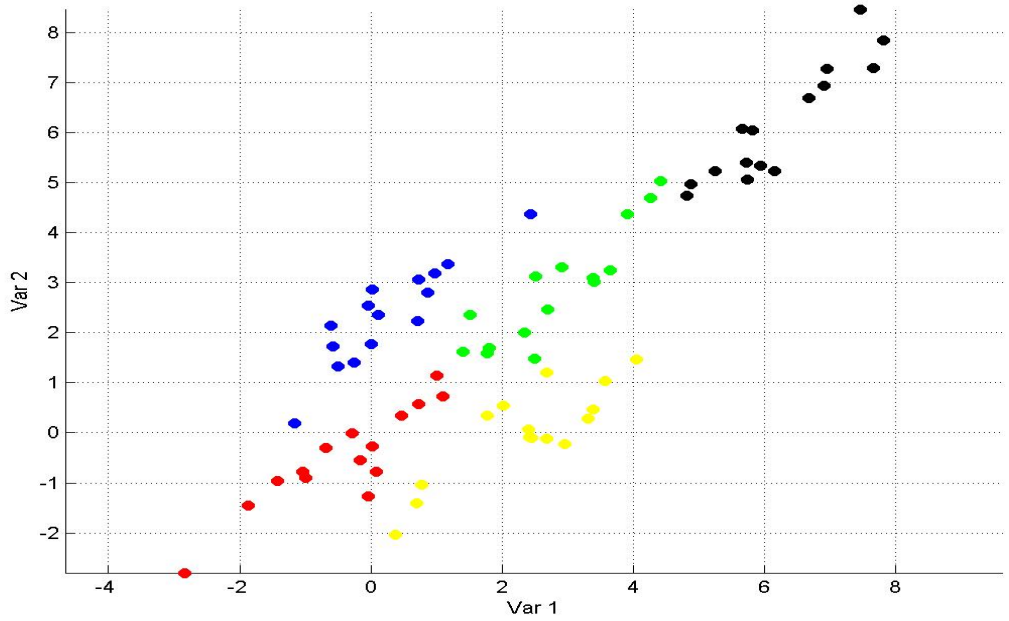

Figure 4.5: Cluster structure for $\mathcal{D}5$

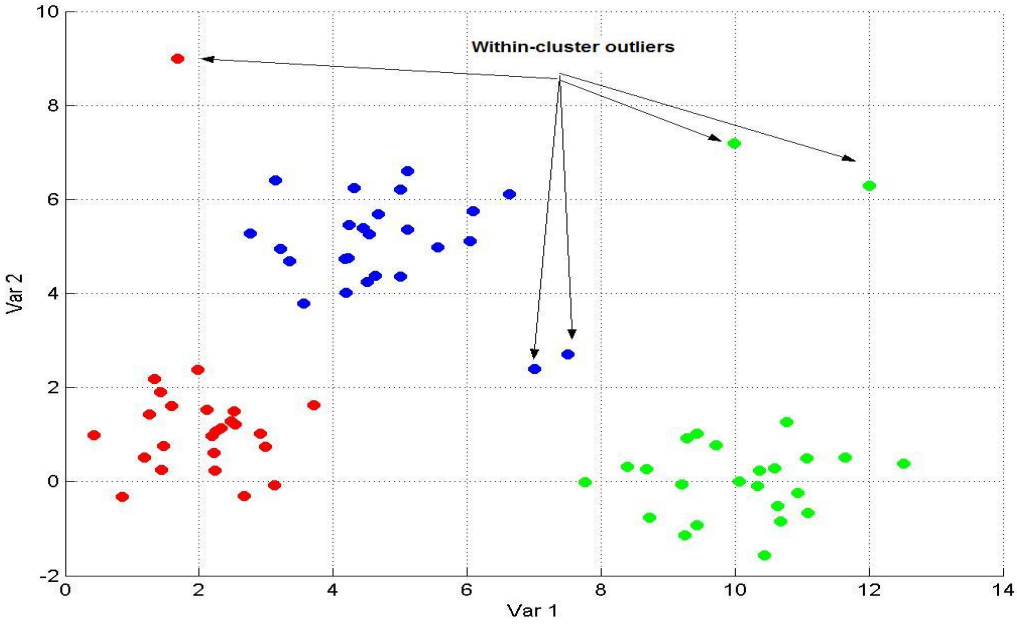

Figure 4.6: Cluster structure for $\mathcal{D}6$

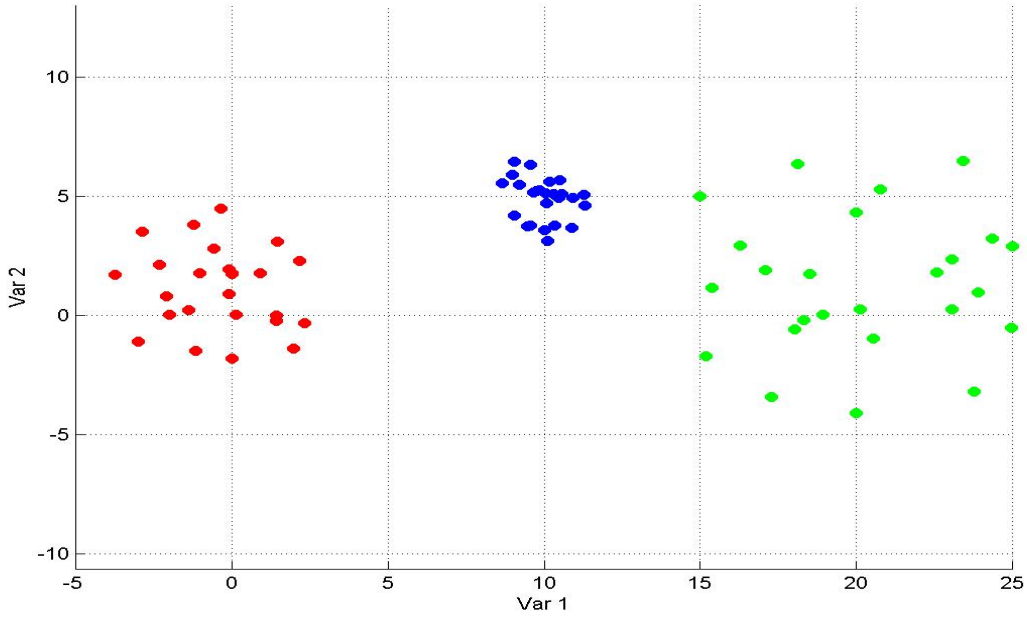

Figure 4.7: Cluster structure for $\mathcal{D}7$

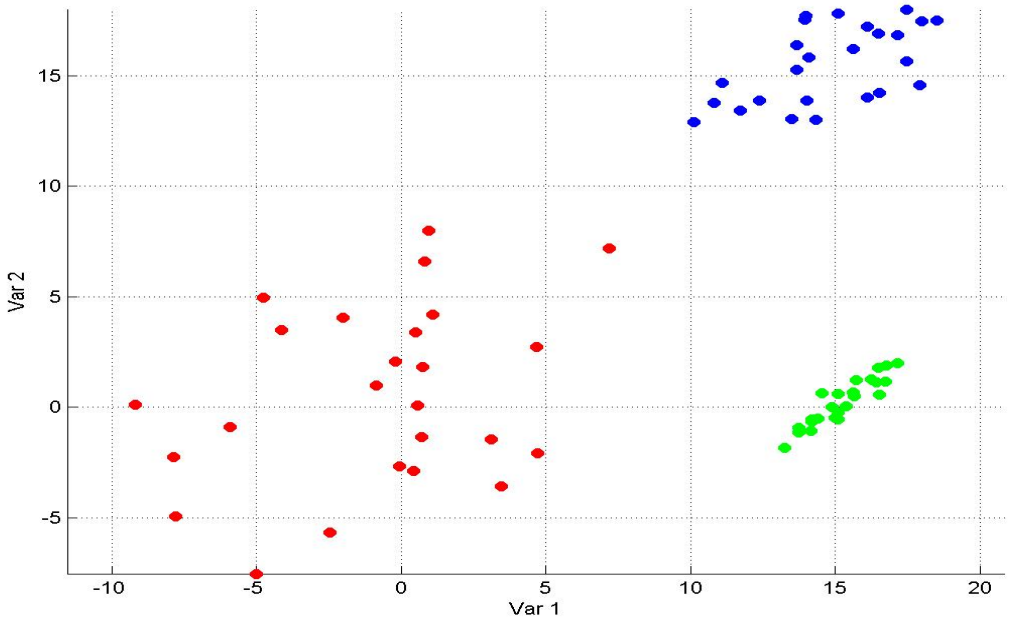

Figure 4.8: Cluster structure for $\mathcal{D}8$

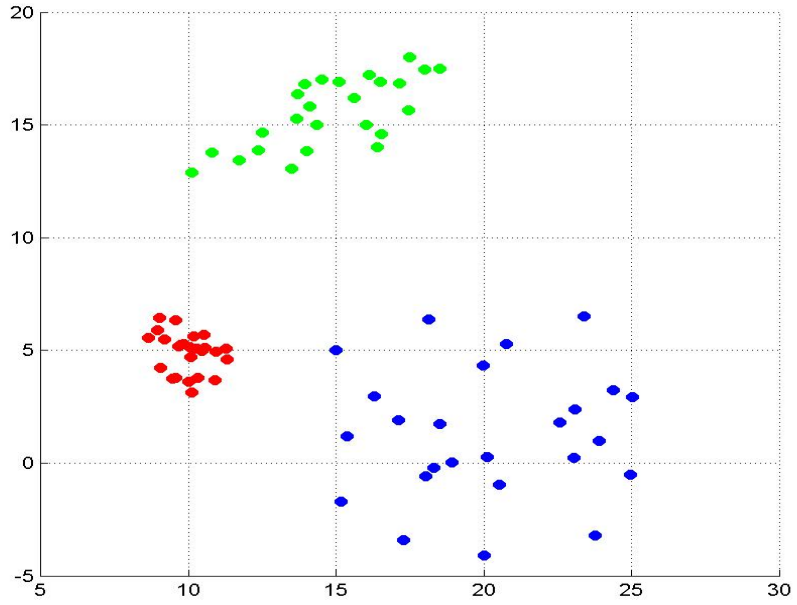

Figure 4.9: Cluster structure for $\mathcal{D}9$

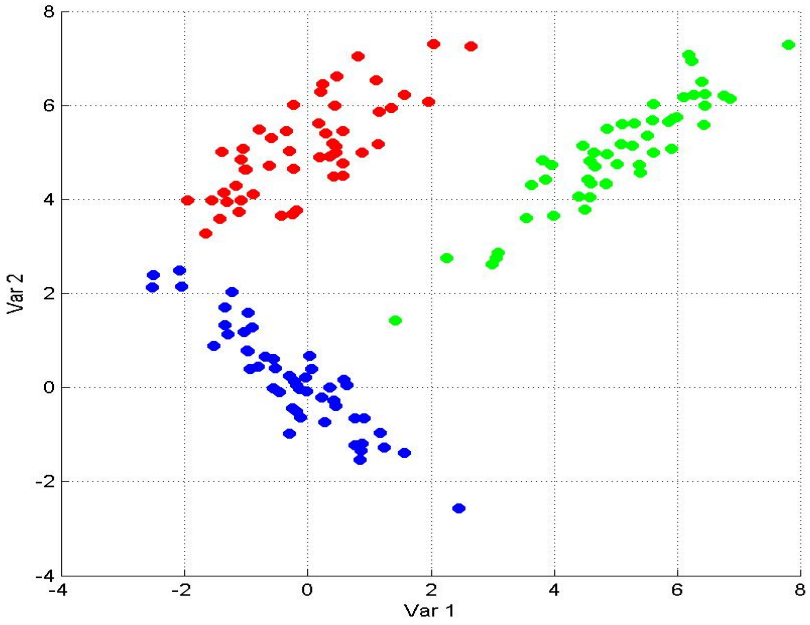

Figure 4.10: Cluster structure for $\mathcal{D}10$

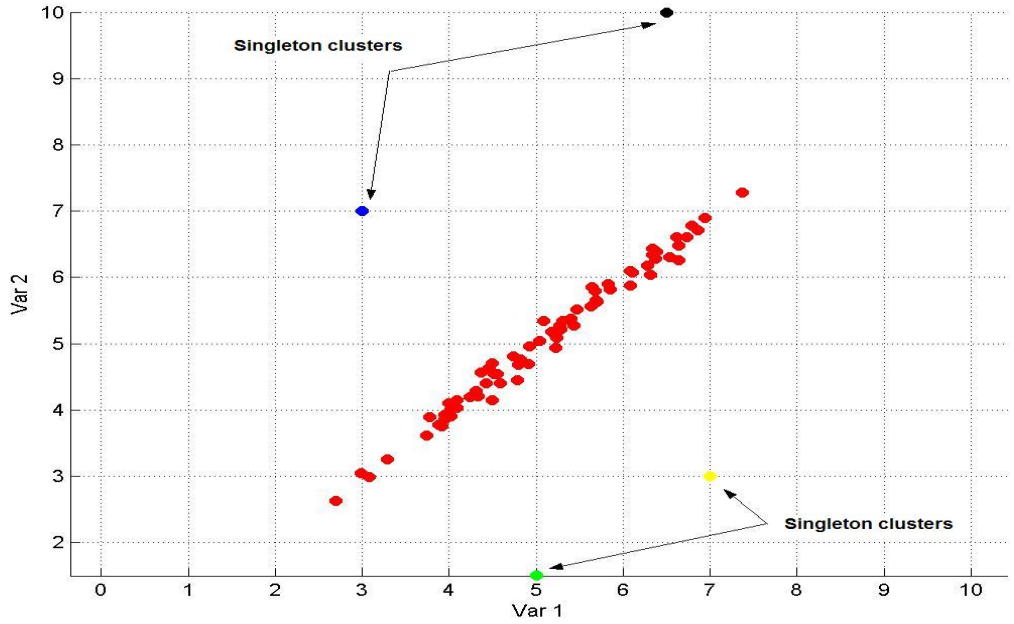

Figure 4.11: Cluster structure for $\mathcal{D}11$

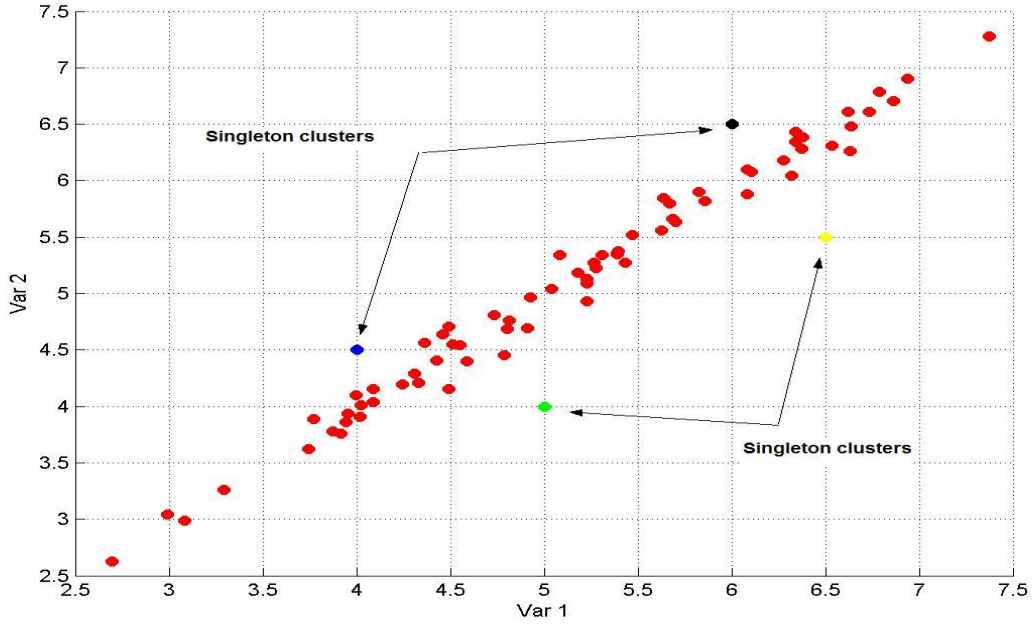

Figure 4.12: Cluster structure for $\mathcal{D}12$

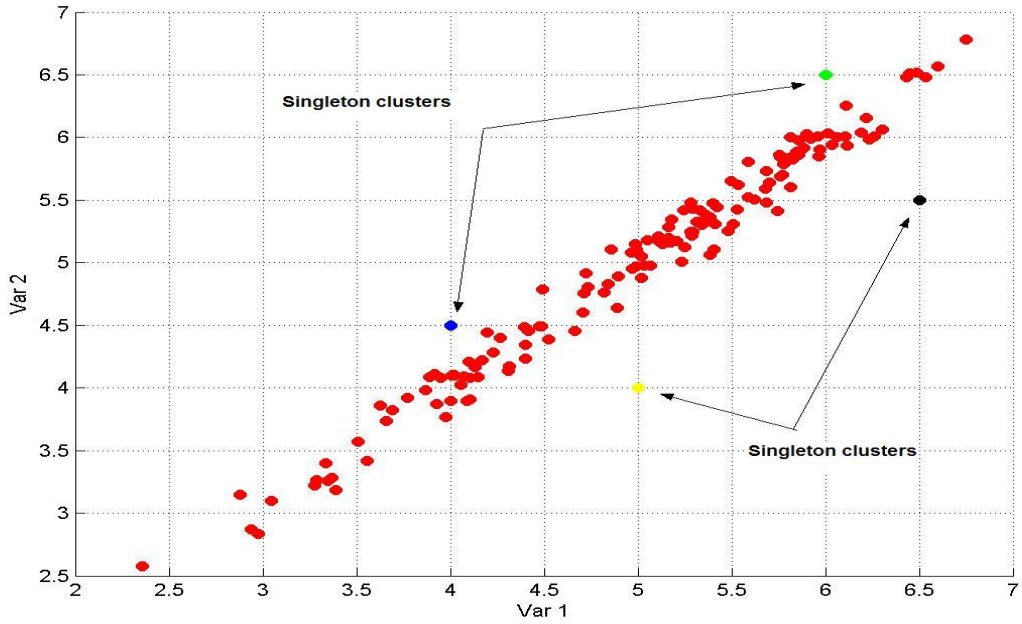

Figure 4.13: Cluster structure for $\mathcal{D}13$

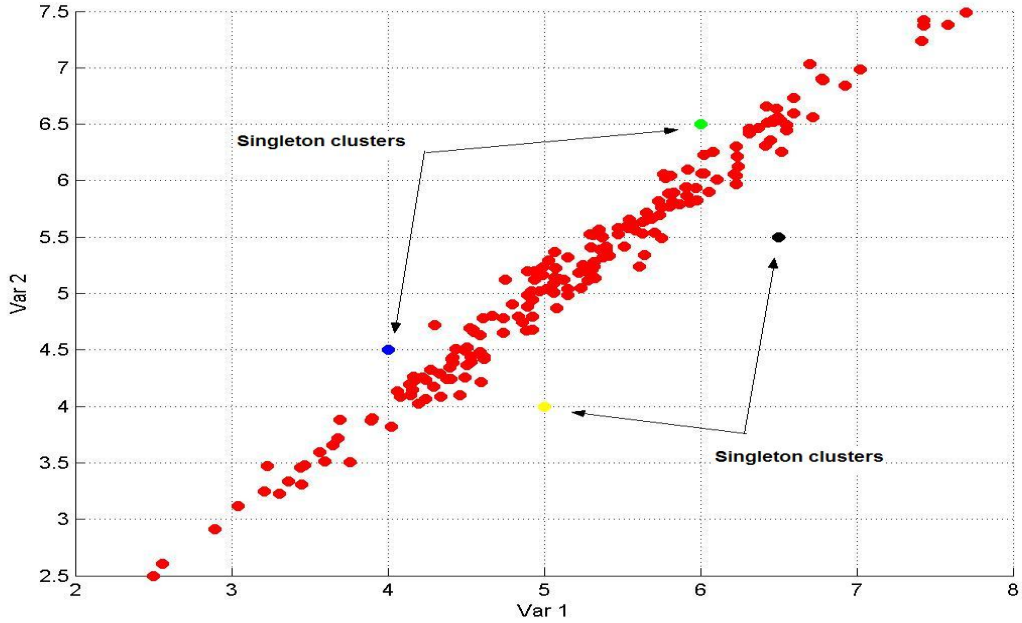

Figure 4.14: Cluster structure for $\mathcal{D}14$

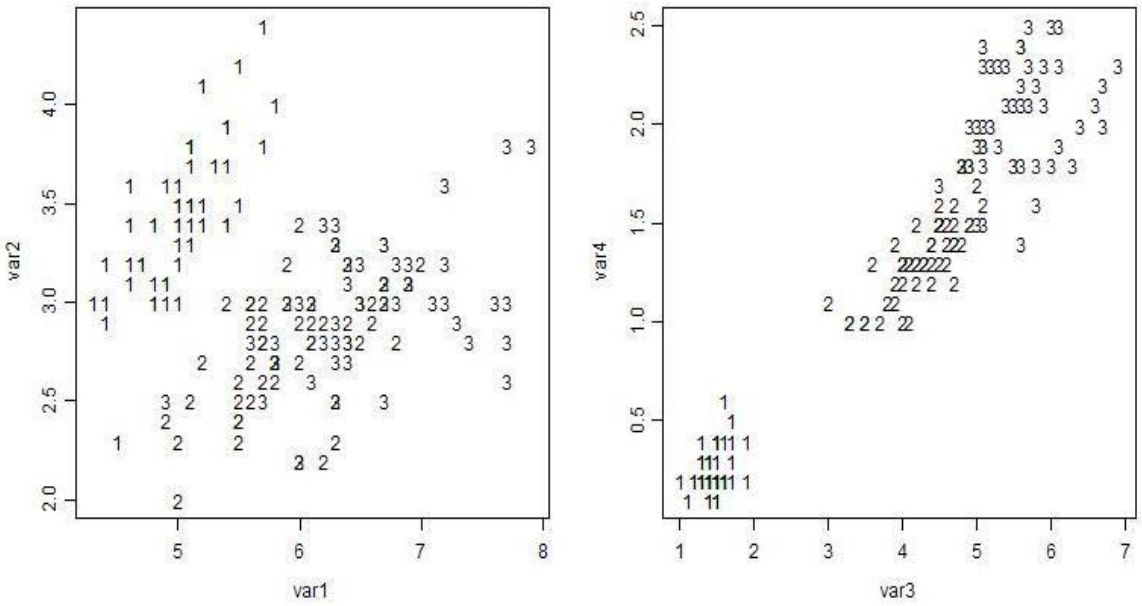

Figure 4.15: Cluster structure for *Iris* ($\mathcal{D}15$) data set

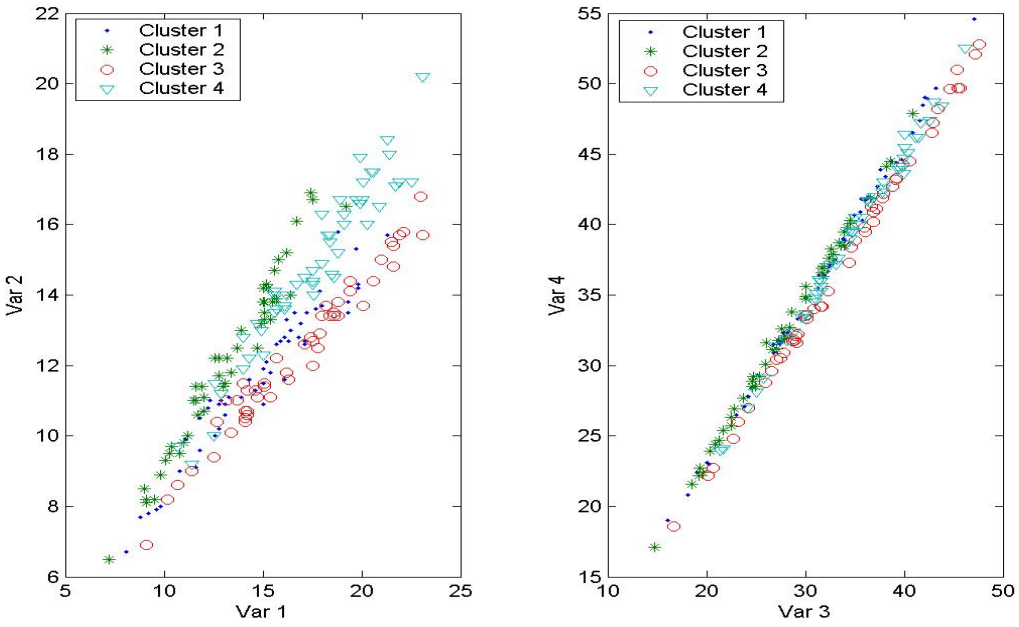

Figure 4.16: Cluster structure for *Crabs* ($\mathcal{D}16$) data set

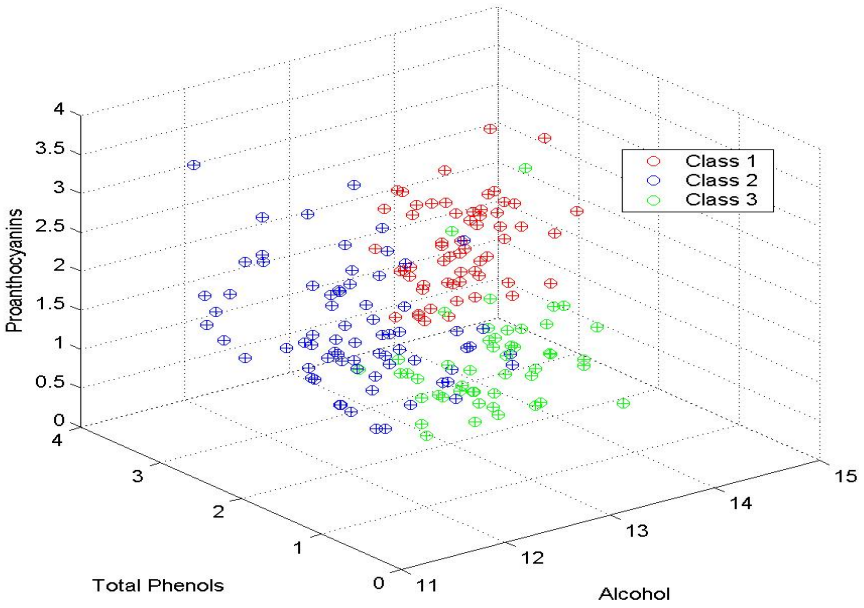

Figure 4.17: *Wine* ($\mathcal{D}17$) data set scatter plot 1

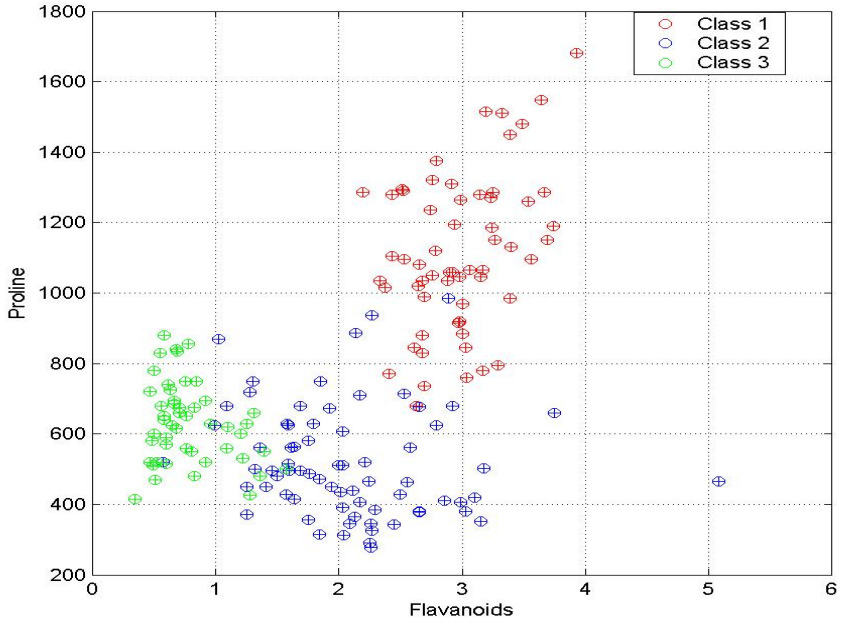

Figure 4.18: *Wine* ($\mathcal{D}17$) data set scatter plot 2

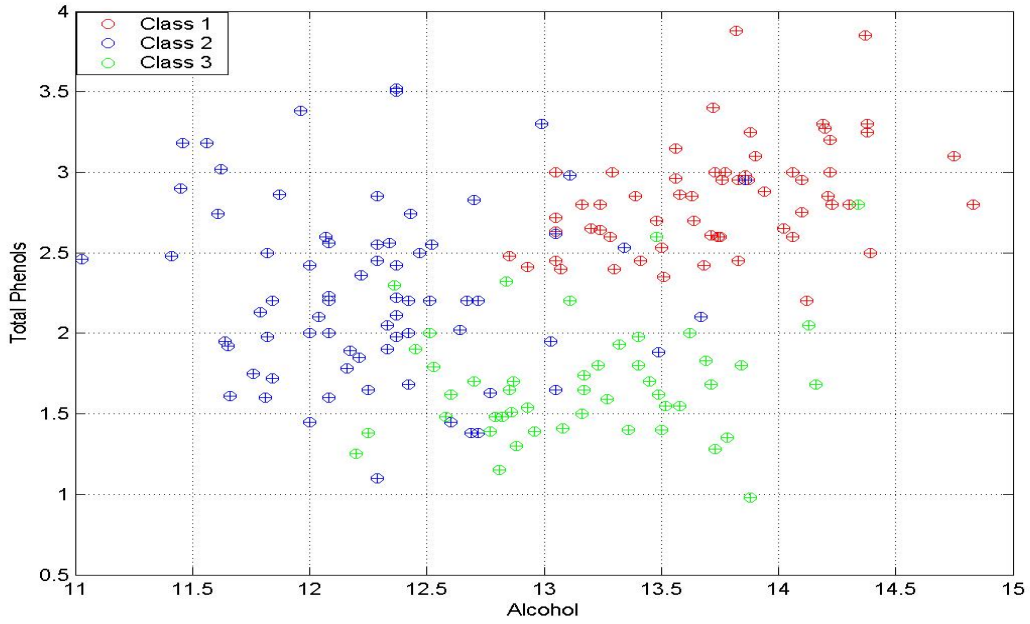

Figure 4.19: *Wine* ($\mathcal{D}17$) data set scatter plot 3

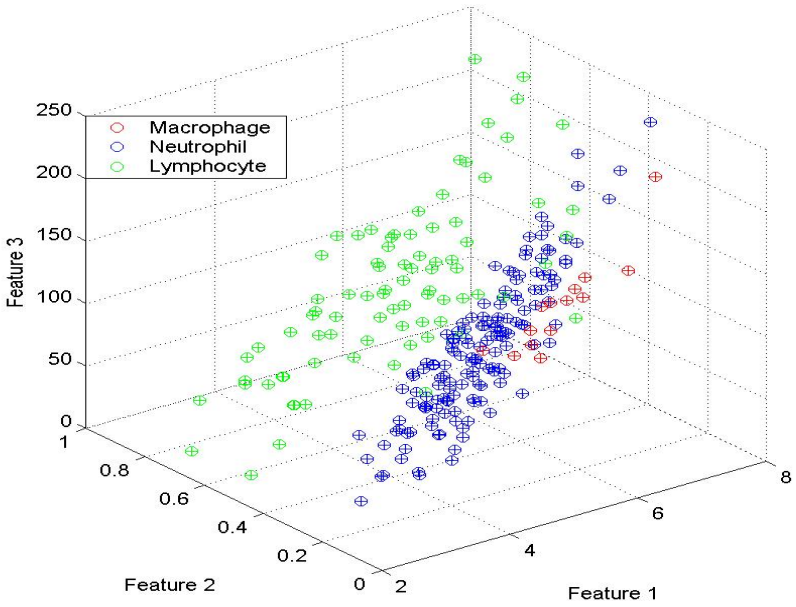

Figure 4.20: *Cells* ($\mathcal{D}18$) data set scatter plot 1

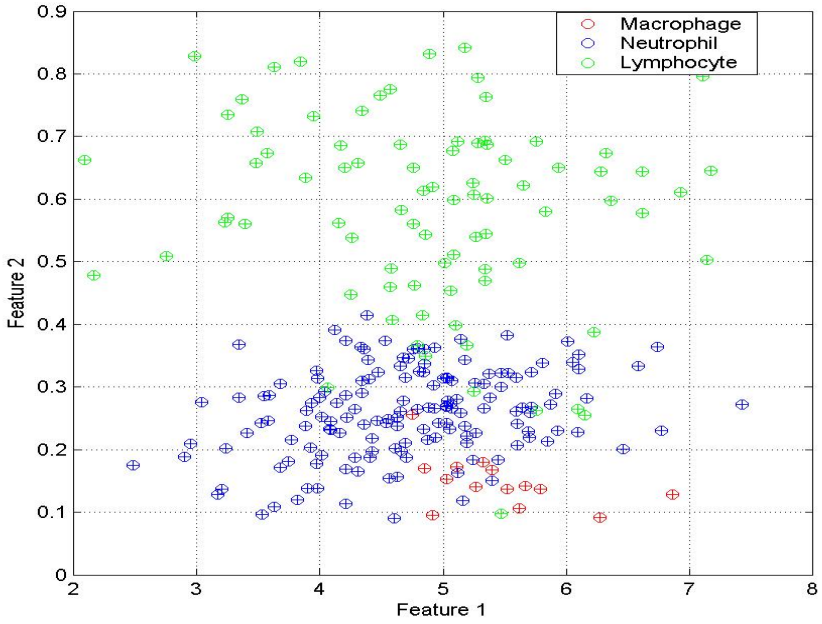

Figure 4.21: *Cells* ($\mathcal{D}18$) data set scatter plot 2

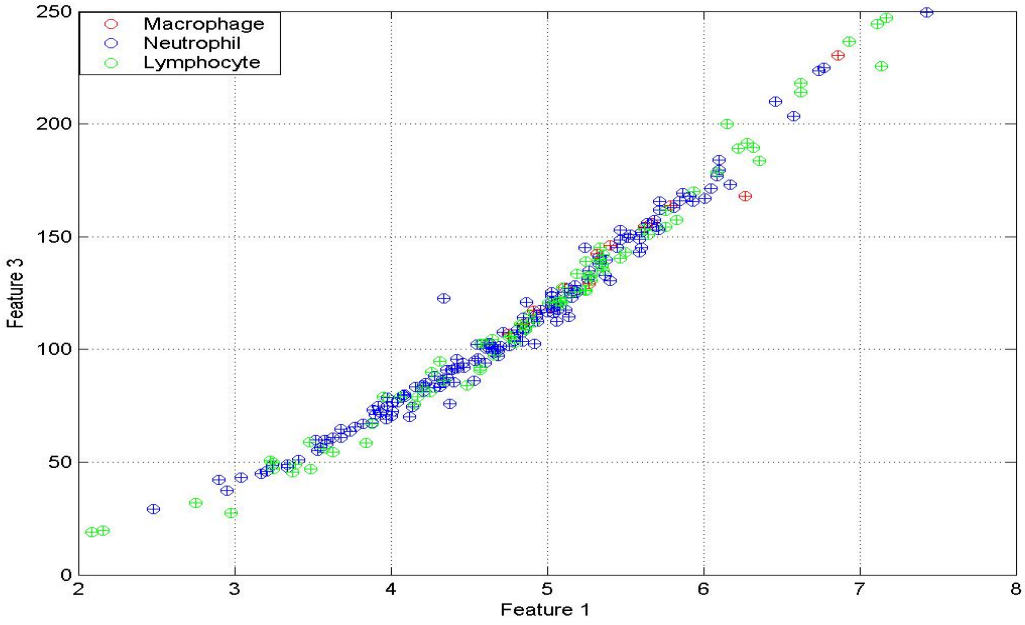

Figure 4.22: *Cells* ($\mathcal{D}18$) data set scatter plot 3

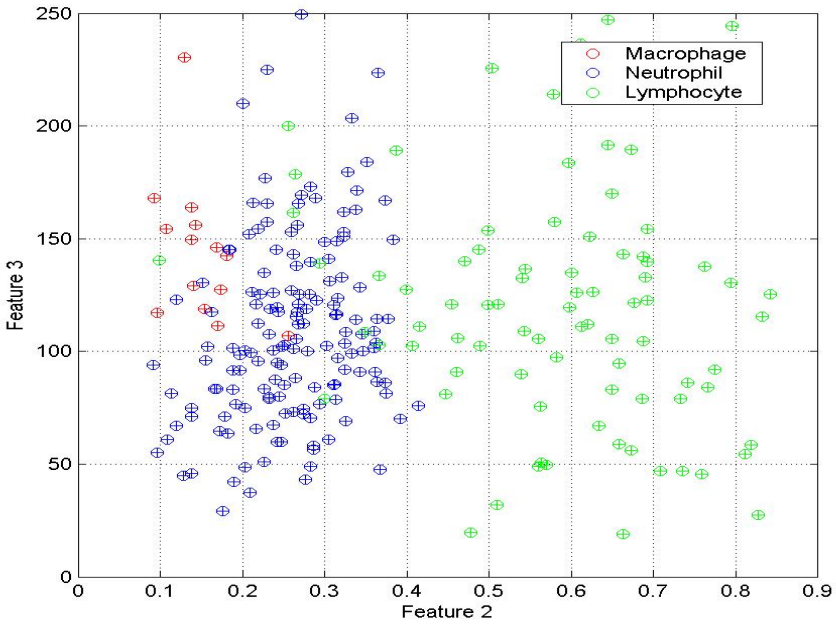

Figure 4.23: *Cells* ($\mathcal{D}18$) data set scatter plot 4

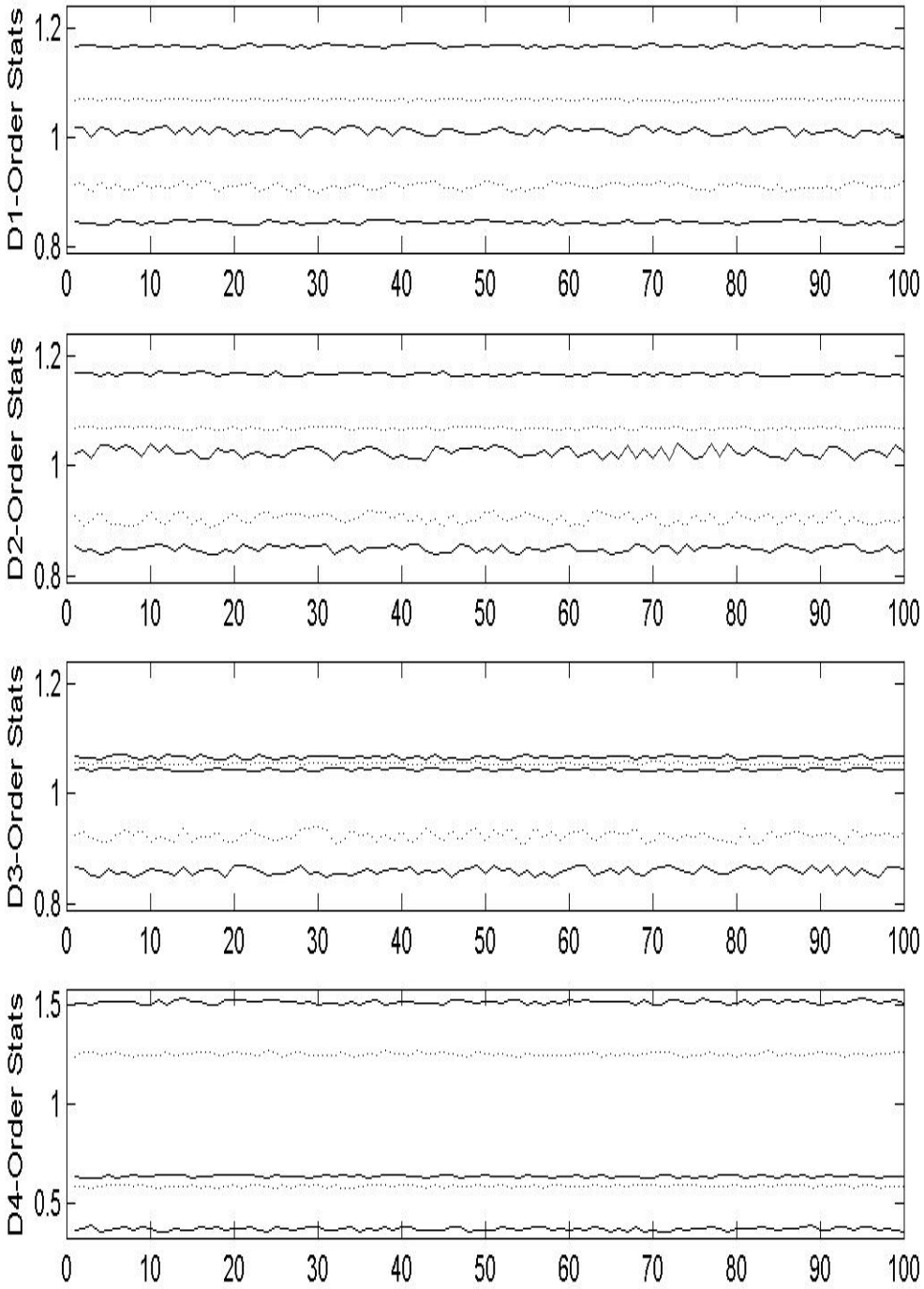

Figure 4.24: Comparison of diagonal elements - SPD starting point

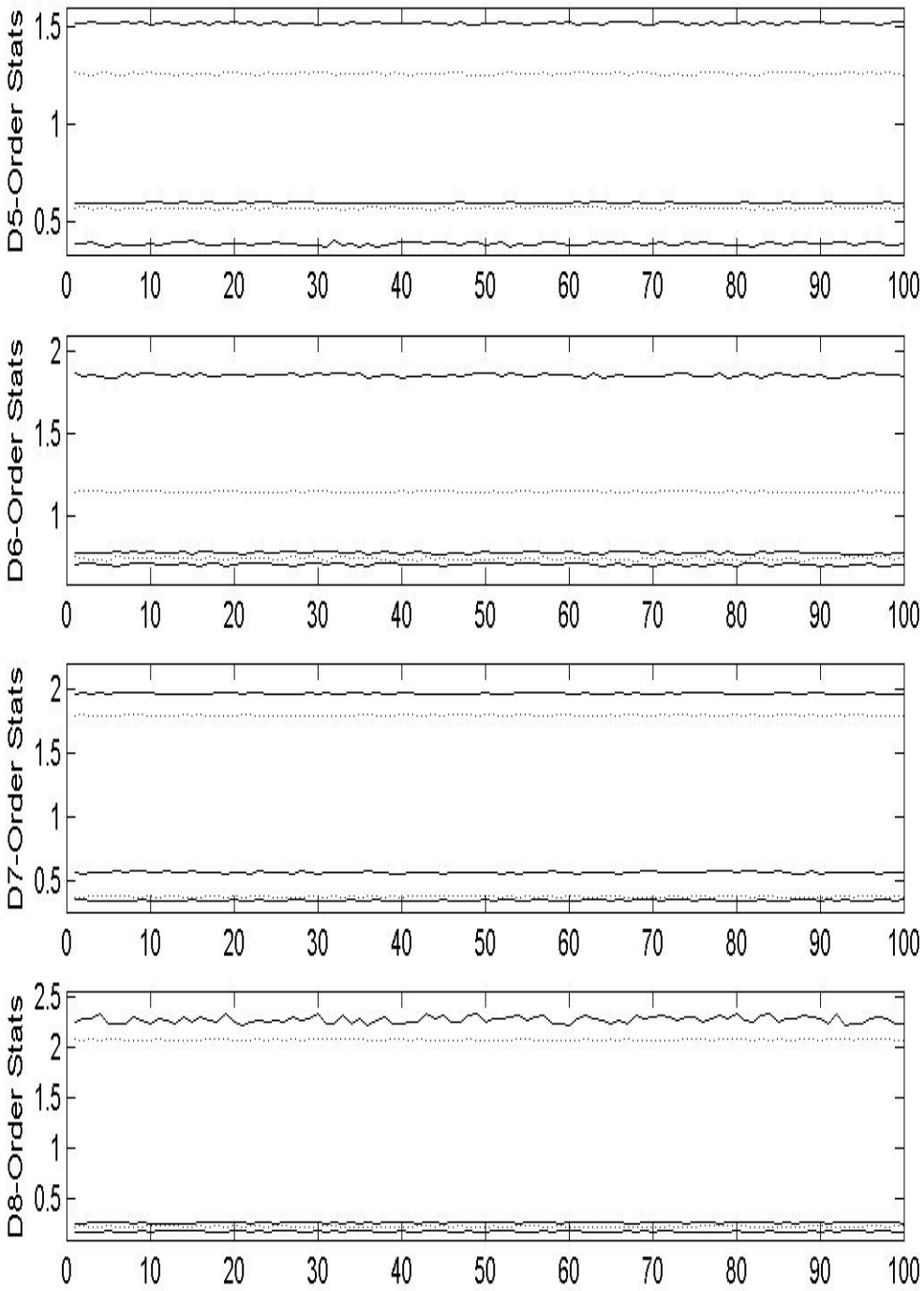

Figure 4.25: Comparison of diagonal elements - SPD starting point

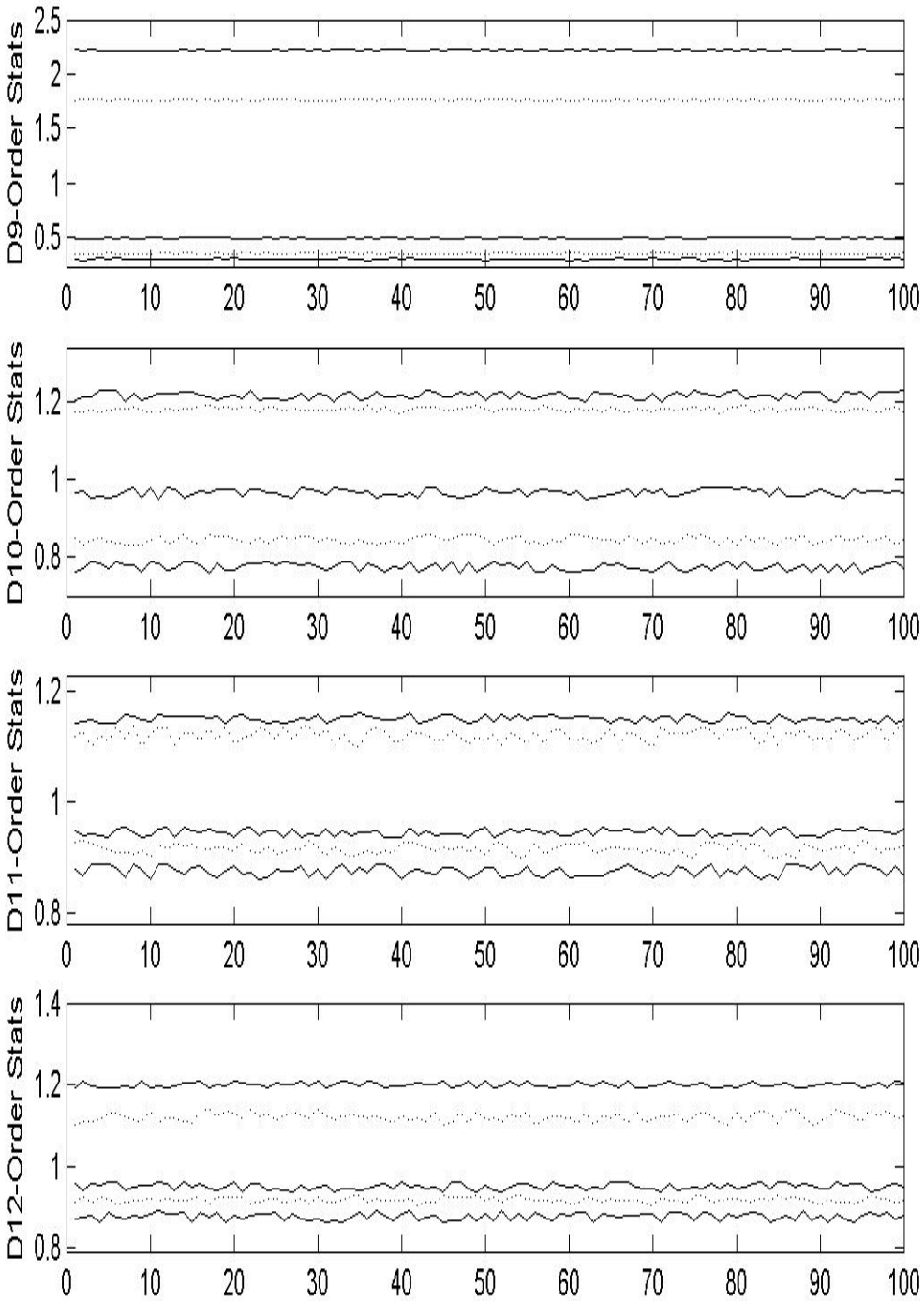

Figure 4.26: Comparison of diagonal elements - SPD starting point

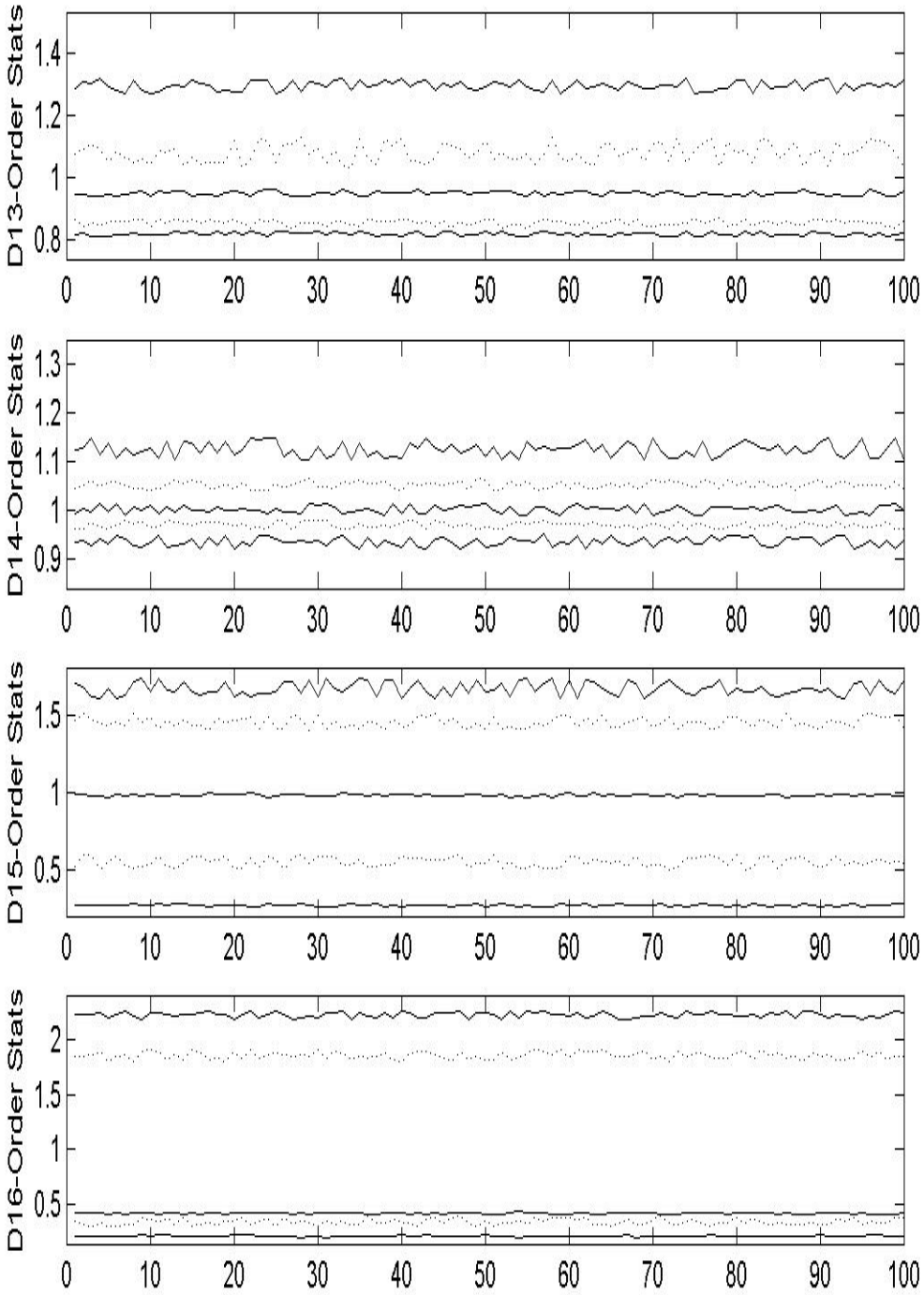

Figure 4.27: Comparison of diagonal elements - SPD starting point

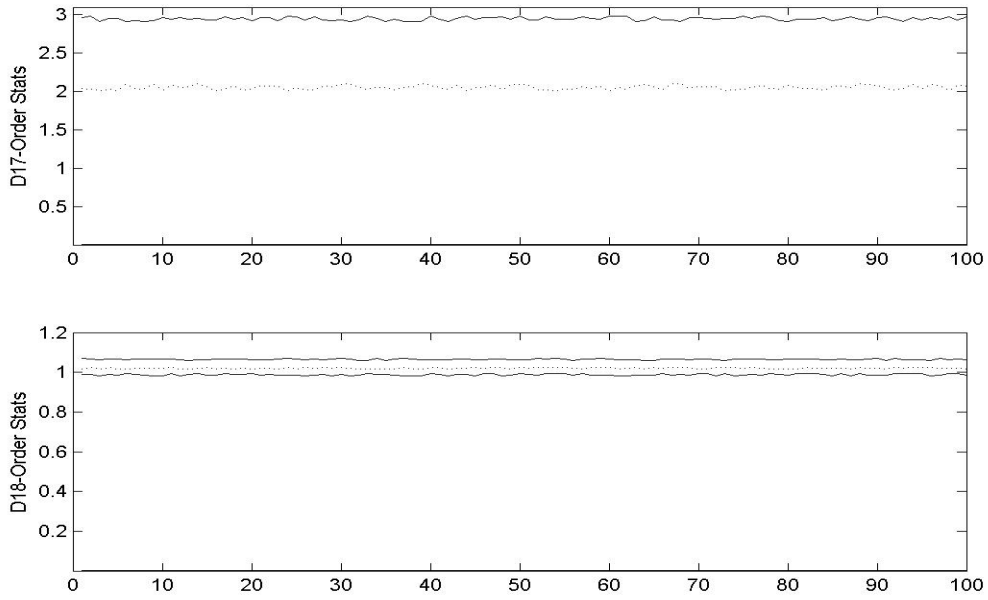

Figure 4.28: Comparison of diagonal elements - SPD starting point

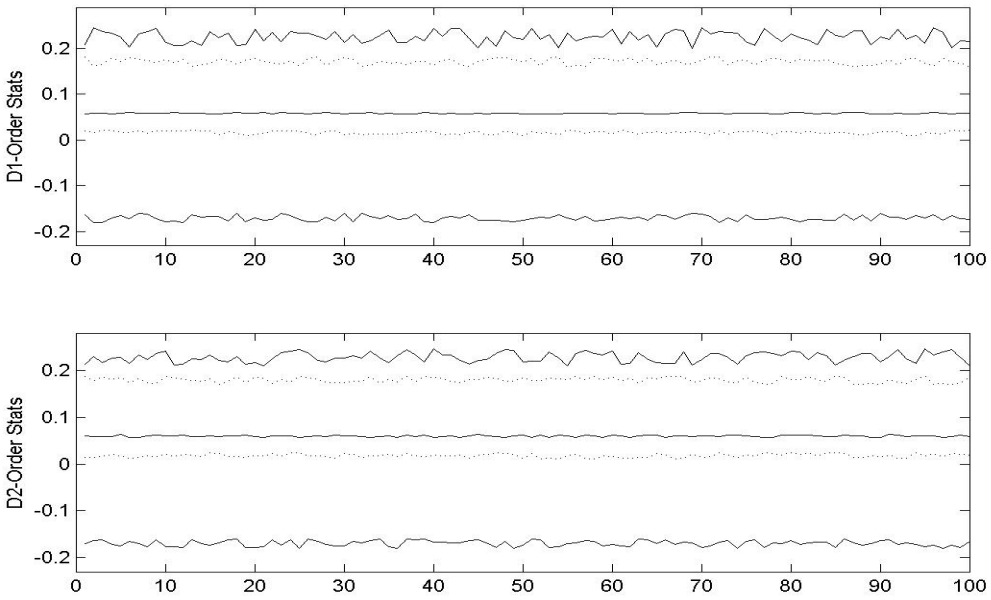

Figure 4.29: Comparison of off-diagonal elements - SPD starting point

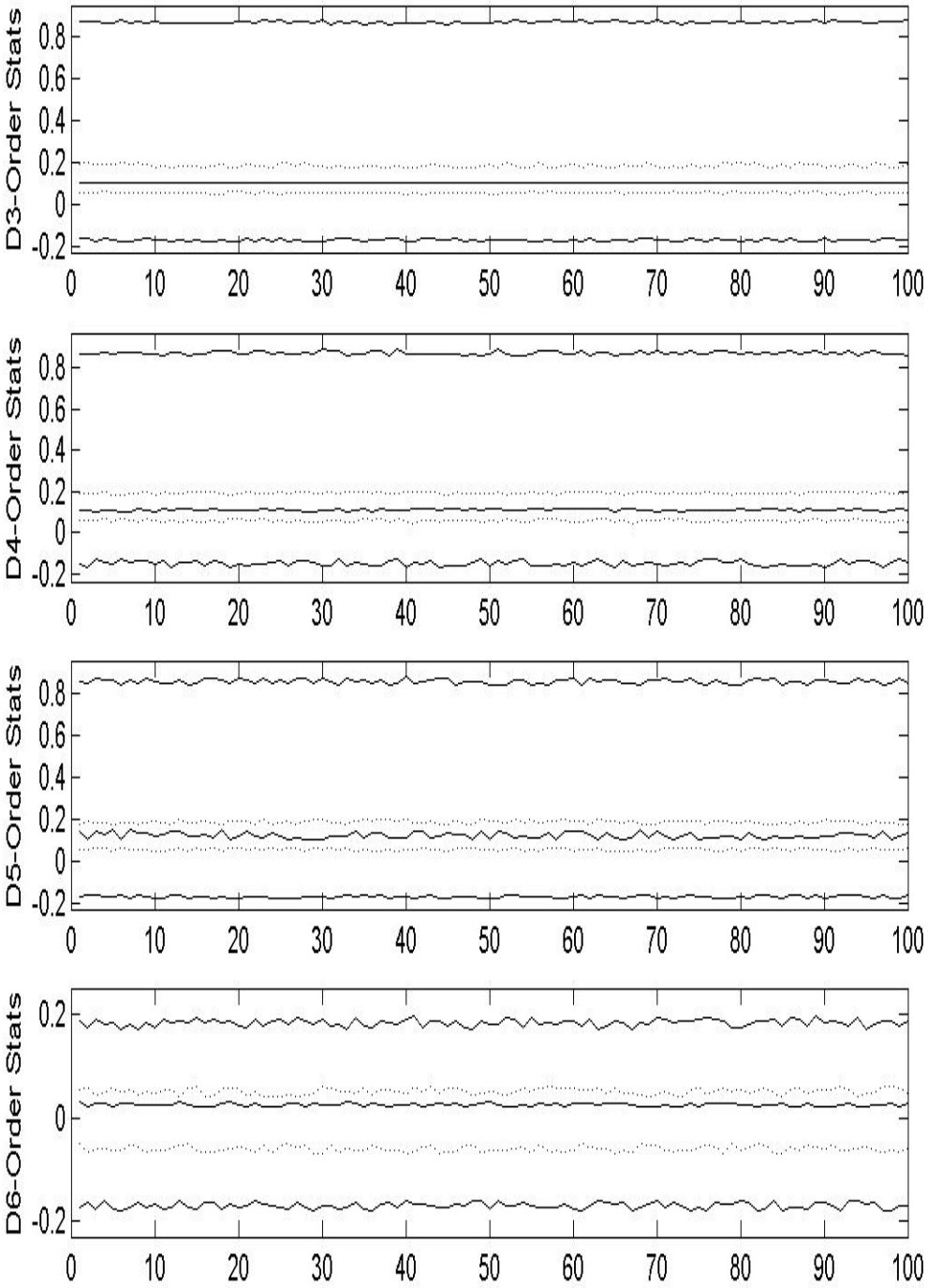

Figure 4.30: Comparison of off-diagonal elements - SPD starting point

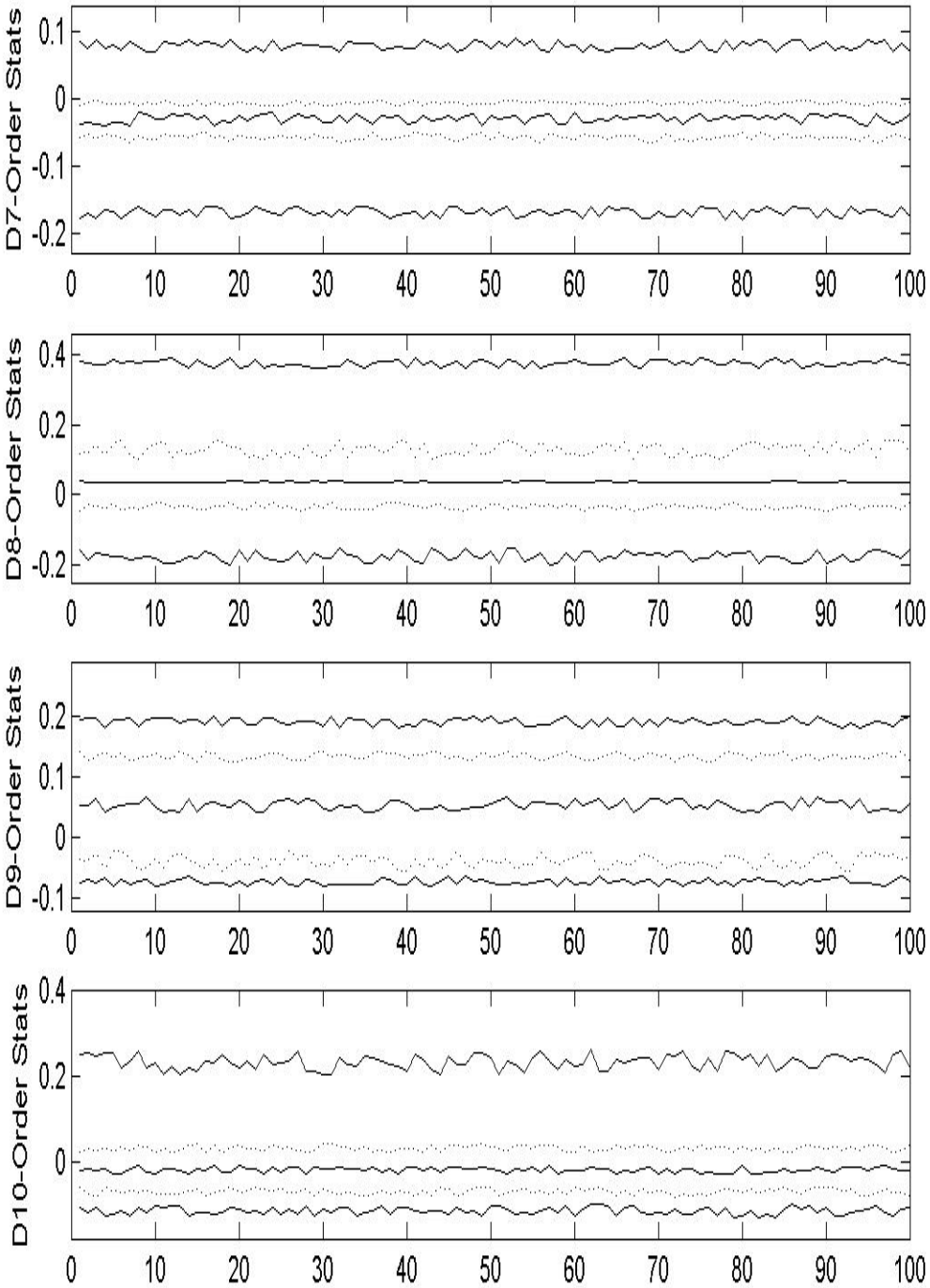

Figure 4.31: Comparison of off-diagonal elements - SPD starting point

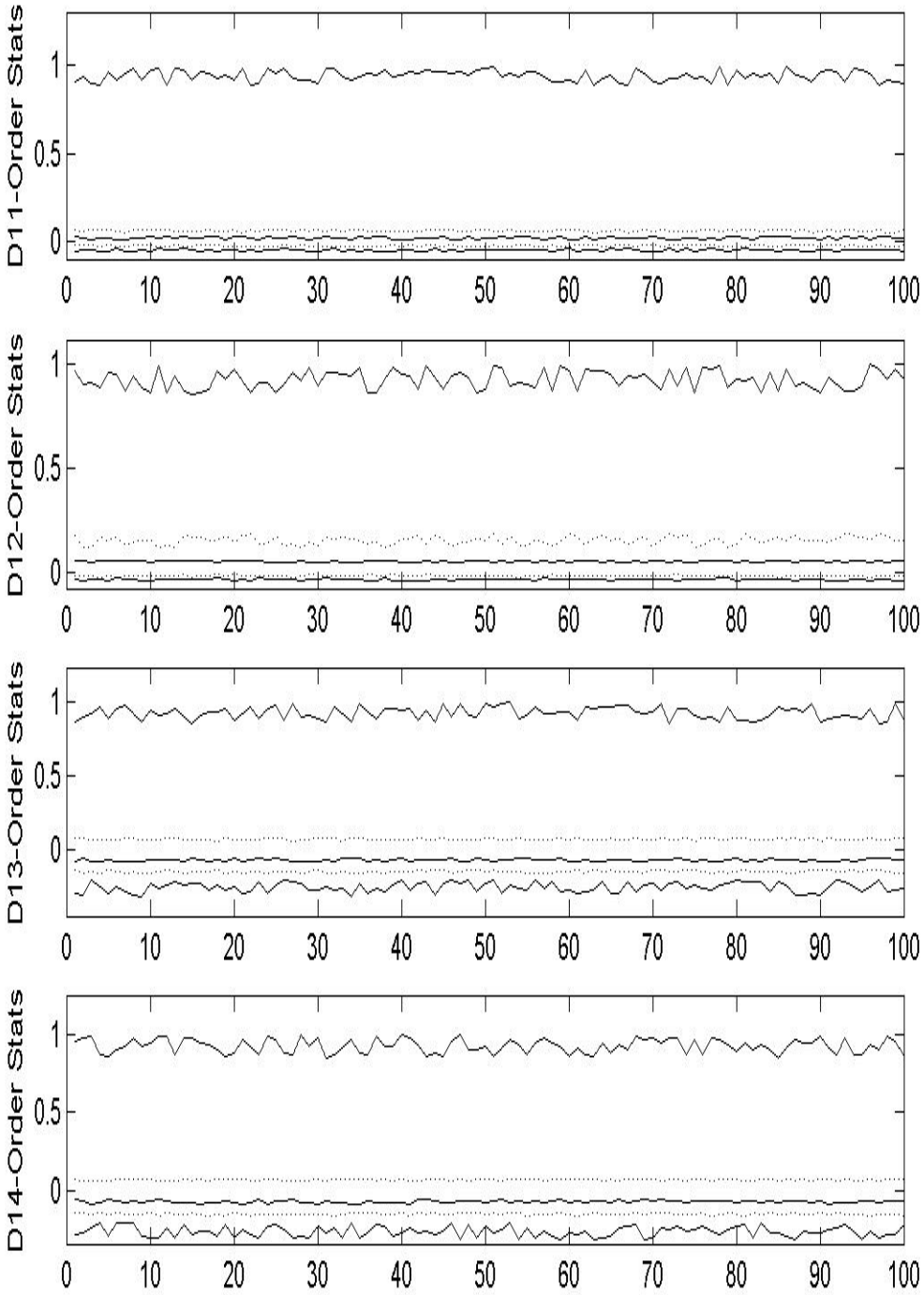

Figure 4.32: Comparison of off-diagonal elements - SPD starting point

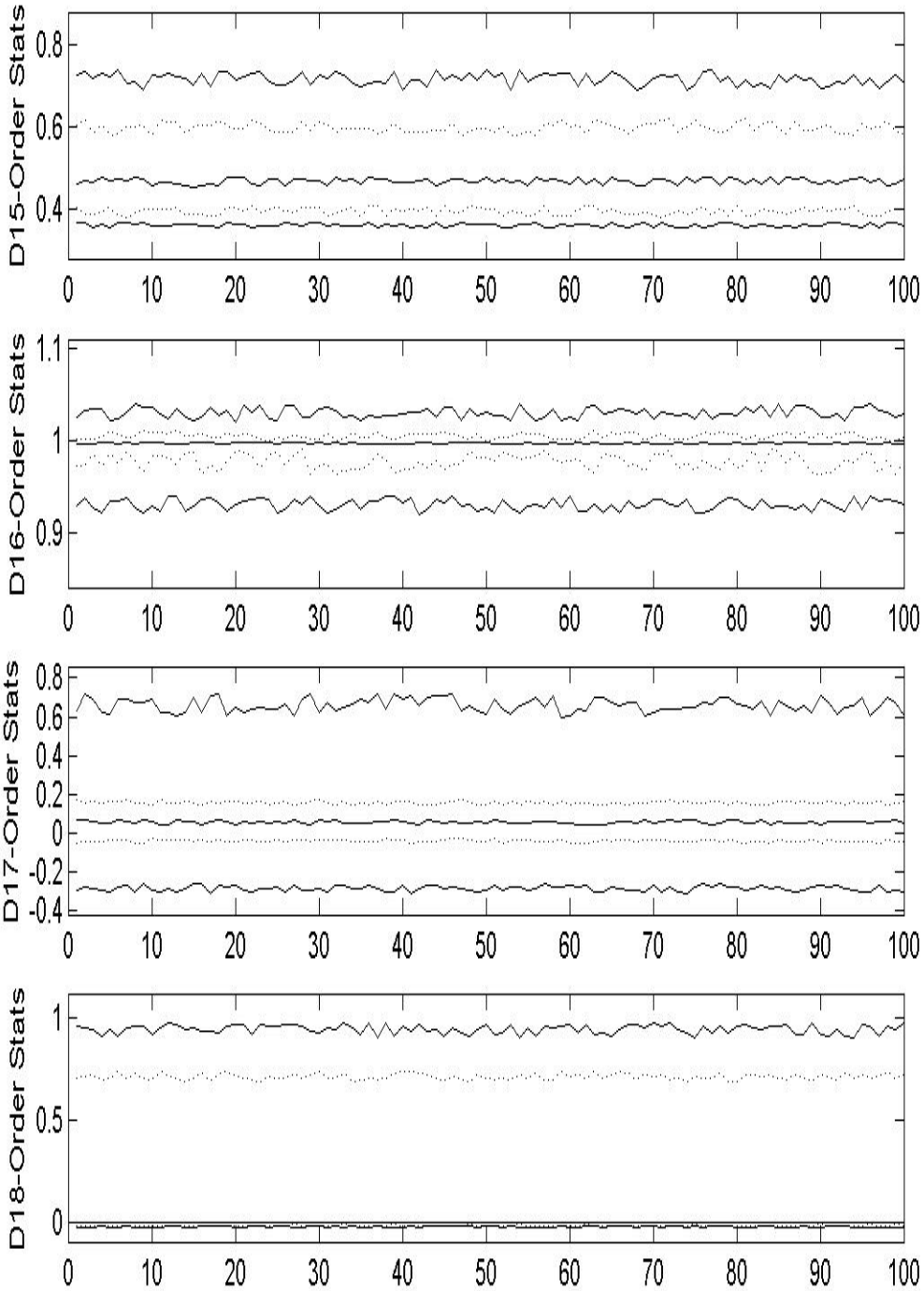

Figure 4.33: Comparison of off-diagonal elements - SPD starting point

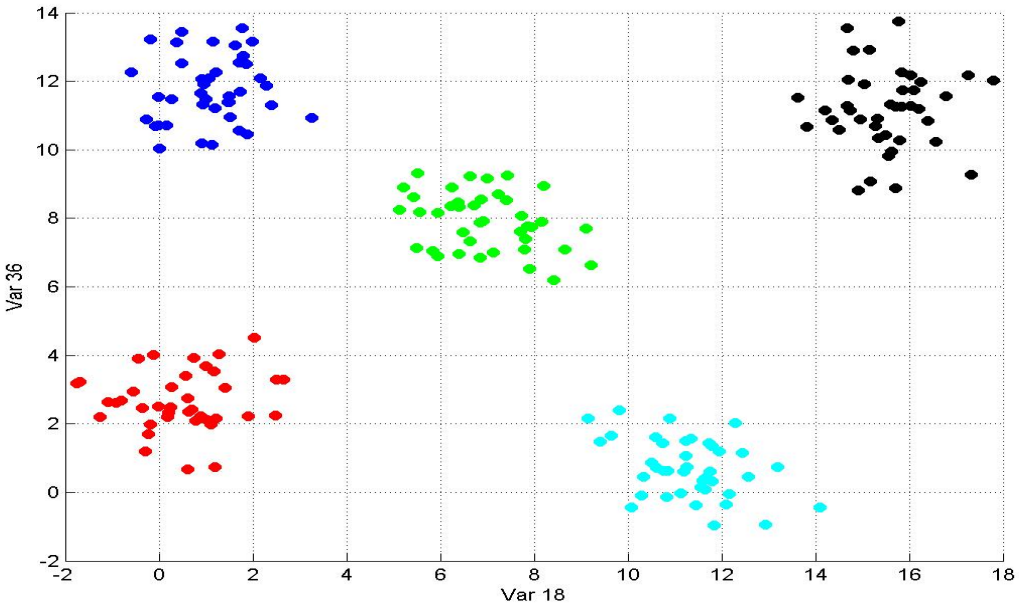

Figure 4.34: Sample realization when clusters are well separated - from a simulated data set in group 2

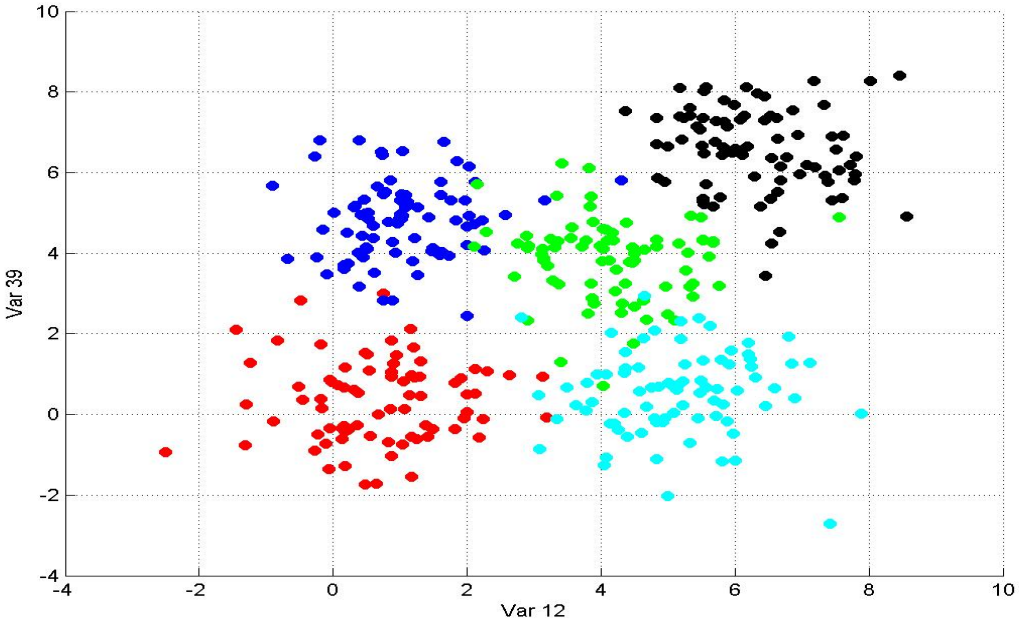

Figure 4.35: Sample realization when clusters touch each other - from a simulated data set in group 2

Data set	M.S.E
$\mathcal{D}1$	0.0061
$\mathcal{D}2$	0.0202
$\mathcal{D}3$	0.0110
$\mathcal{D}4$	0.0207
$\mathcal{D}5$	0.0331
$\mathcal{D}6$	0.0104
$\mathcal{D}7$	0.0120
$\mathcal{D}8$	0.0251
$\mathcal{D}9$	0.0240

Data set	M.S.E
$\mathcal{D}10$	0.0190
$\mathcal{D}11$	0.0032
$\mathcal{D}12$	0.0340
$\mathcal{D}13$	0.0201
$\mathcal{D}14$	0.0090
$\mathcal{D}15$	0.0111
$\mathcal{D}16$	0.0005
$\mathcal{D}17$	0.0141
$\mathcal{D}18$	0.0030

Table 4.2: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices

Data set	Var[eigs($\mathbf{W}^*_{(m_1)}$)]	Var[eigs($\mathbf{W}$)]
$\mathcal{D}1$	0.10	0.09
$\mathcal{D}2$	0.31	0.09
$\mathcal{D}3$	0.60	0.49
$\mathcal{D}4$	0.75	0.49
$\mathcal{D}5$	1.08	0.49
$\mathcal{D}6$	0.11	0.26
$\mathcal{D}7$	0.32	0.63
$\mathcal{D}8$	1.43	1.51
$\mathcal{D}9$	0.45	0.92
$\mathcal{D}10$	0.25	0.11
$\mathcal{D}11$	0.61	0.66
$\mathcal{D}12$	0.36	0.66
$\mathcal{D}13$	0.43	0.47
$\mathcal{D}14$	0.44	0.45
$\mathcal{D}15$	1.34	1.66
$\mathcal{D}16$	4.90	4.93
$\mathcal{D}17$	12.97	12.82
$\mathcal{D}18$	2.99	2.99

Table 4.3: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices

Data set	M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)]
$\mathcal{D}1$	0.0101
$\mathcal{D}2$	0.3400
$\mathcal{D}3$	0.0501
$\mathcal{D}4$	0.1236
$\mathcal{D}5$	0.5117
$\mathcal{D}6$	0.0322
$\mathcal{D}7$	0.1006
$\mathcal{D}8$	0.0211
$\mathcal{D}9$	0.1210

Data set	M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)]
$\mathcal{D}10$	0.0402
$\mathcal{D}11$	0.0141
$\mathcal{D}12$	0.0730
$\mathcal{D}13$	0.0081
$\mathcal{D}14$	0.0010
$\mathcal{D}15$	0.0121
$\mathcal{D}16$	0.0001
$\mathcal{D}17$	0.0007
$\mathcal{D}18$	0.0000

Table 4.4: M.S.E between eigenvalues of $\mathbf{W}^*_{(m_1)}$ and $\mathbf{W}$ matrices

Data set	After $\mathbf{W}$ scaling	After $\mathbf{W}^*_{(m_1)}$ scaling
$\mathcal{D}1$	0	0
$\mathcal{D}2$	20	36
$\mathcal{D}3$	0	1
$\mathcal{D}4$	2	29
$\mathcal{D}5$	23	31
$\mathcal{D}6$	12	16
$\mathcal{D}7$	0	1
$\mathcal{D}8$	0	0
$\mathcal{D}9$	2	6
$\mathcal{D}10$	3	7
$\mathcal{D}11$	2	0
$\mathcal{D}12$	12	22
$\mathcal{D}13$	27	13
$\mathcal{D}14$	42	16
$\mathcal{D}15$	4	39
$\mathcal{D}16$	54	55
$\mathcal{D}17$	62	79
$\mathcal{D}18$	35	87

Table 4.5: Comparison of errors of misclassification after $\mathbf{W}$ and $\mathbf{W}^*_{(m_1)}$ scaling

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.007	0.006	$\times$	$\times$
$n = 150$	0.002	0.002	0.009	$\times$
$n = 200$	0.001	0.003	0.006	0.010

Table 4.6: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.014	0.018	$\times$	$\times$
$n = 150$	0.015	0.013	0.017	$\times$
$n = 200$	0.013	0.011	0.011	0.016

Table 4.7: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.003	0.019	$\times$	$\times$
$n = 150$	0.003	0.003	0.016	$\times$
$n = 200$	0.002	0.001	0.009	0.015

Table 4.8: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.006	0.028	×	×
$n = 150$	0.005	0.009	0.018	×
$n = 200$	0.003	0.006	0.010	0.020

Table 4.9: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.003	0.007	×	×
$n = 150$	0.001	0.004	0.008	×
$n = 200$	0.001	0.003	0.006	0.010

Table 4.10: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters well separated, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.004	0.008	×	×
$n = 150$	0.003	0.005	0.009	×
$n = 200$	0.001	0.004	0.008	0.012

Table 4.11: M.S.E between correlation estimates from $\mathbf{R}^*$ and $\mathbf{R}$ matrices, clusters touching each other, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.87	1.32	×	×
	0.84	1.21		
$n = 150$	0.68	0.88	1.25	×
	0.66	0.75	1.05	
$n = 200$	0.66	0.73	1.08	1.50
	0.64	0.67	0.91	1.22

Table 4.12: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.03	1.39	×	×
	0.92	1.23		
$n = 150$	0.76	0.91	1.29	×
	0.67	0.81	1.08	
$n = 200$	0.71	0.84	1.12	1.54
	0.65	0.79	0.98	1.31

Table 4.13: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.63	11.19	×	×
	1.08	5.00		
$n = 150$	1.53	10.90	26.30	×
	1.02	4.84	14.92	
$n = 200$	1.44	9.99	22.19	31.81
	0.96	3.98	12.87	15.03

Table 4.14: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.99	14.25	×	×
	1.81	6.04		
$n = 150$	1.69	12.43	28.11	×
	1.64	5.99	16.02	
$n = 200$	1.53	11.12	24.81	33.72
	1.10	4.79	17.20	18.19

Table 4.15: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	4.66	44.09	×	×
	4.63	41.04		
$n = 150$	4.32	43.61	66.31	×
	4.21	40.91	58.82	
$n = 200$	4.11	37.91	62.47	71.21
	4.01	33.24	55.44	69.94

Table 4.16: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters well separated, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	4.88	47.83	×	×
	4.79	43.09		
$n = 150$	4.66	46.29	69.82	×
	4.49	45.08	61.33	
$n = 200$	4.31	40.25	67.61	90.19
	3.96	39.67	59.95	79.97

Table 4.17: Comparison of variance of eigenvalues of $\mathbf{W}^*_{(m_1)}$ (on first line) and $\mathbf{W}$ matrices, clusters touching each other, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.008	0.009	×	×
$n = 150$	0.003	0.011	0.014	×
$n = 200$	0.013	0.012	0.013	0.022

Table 4.18: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.087	0.015	×	×
$n = 150$	0.064	0.014	0.012	×
$n = 200$	0.068	0.007	0.009	0.020

Table 4.19: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.010	1.145	×	×
$n = 150$	0.006	0.117	1.420	×
$n = 200$	0.001	0.003	1.370	1.710

Table 4.20: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.018	1.210	×	×
$n = 150$	0.005	0.077	1.550	×
$n = 200$	0.001	0.021	1.404	1.900

Table 4.21: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.002	0.129	×	×
$n = 150$	0.000	0.006	0.216	×
$n = 200$	0.000	0.001	0.115	0.306

Table 4.22: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters well separated, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.006	0.060	×	×
$n = 150$	0.003	0.002	0.087	×
$n = 200$	0.001	0.002	0.091	0.228

Table 4.23: M.S.E[eig($\mathbf{W}^*_{(m_1)}$)-eig($\mathbf{W}$)], clusters touching each other, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0	3	×	×
	0	0		
$n = 150$	1	0	3	×
	1	0	1	
$n = 200$	0	2	4	4
	0	4	9	1

Table 4.24: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	6	1	×	×
	3	7		
$n = 150$	21	18	20	×
	16	17	14	
$n = 200$	24	21	25	23
	20	18	16	9

Table 4.25: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0	0	×	×
	0	0		
$n = 150$	0	0	3	×
	0	0	1	
$n = 200$	1	0	2	3
	2	0	2	5

Table 4.26: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	4	10	×	×
	1	4		
$n = 150$	20	13	16	×
	11	12	12	
$n = 200$	25	23	19	21
	19	19	18	10

Table 4.27: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0.5$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0	4	×	×
	0	1		
$n = 150$	1	2	1	×
	1	3	1	
$n = 200$	0	1	2	3
	0	2	3	1

Table 4.28: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters well separated, $\rho = 0.99$

	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	7	11	$\times$	$\times$
	8	2		
$n = 150$	16	16	17	$\times$
	9	12	11	
$n = 200$	25	21	20	24
	17	14	16	13

Table 4.29: Comparison of average errors of misclassification after $\mathbf{W}^*_{(m_1)}$ scaling (on first line) and after $\mathbf{W}$ scaling, clusters touching each other, $\rho = 0.99$

Clusters well separated				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.10	0.53	×	×
$n = 150$	0.07	0.31	0.48	×
$n = 200$	0.04	0.21	0.50	0.63
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.11	0.52	×	×
$n = 150$	0.10	0.30	0.46	×
$n = 200$	0.03	0.29	0.47	0.62
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.12	0.54	×	×
$n = 150$	0.09	0.31	0.50	×
$n = 200$	0.07	0.27	0.49	0.64

Clusters touching each other				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.20	0.86	×	×
$n = 150$	0.13	0.81	0.92	×
$n = 200$	0.11	0.66	0.81	0.93
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.24	0.88	×	×
$n = 150$	0.16	0.82	0.94	×
$n = 200$	0.12	0.69	0.86	0.92
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	0.28	0.89	×	×
$n = 150$	0.16	0.83	0.93	×
$n = 200$	0.14	0.71	0.88	0.94

Table 4.30: Variance of normalized eigenvalues of $\mathbf{M} = \left[\mathbf{\Sigma}\right]^{-1}\left[\mathbf{W}^{*}_{(m_1)}\right]$

Clusters well separated				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	6.41	8.94	$\times$	$\times$
$n = 150$	9.66	12.00	15.01	$\times$
$n = 200$	14.02	16.17	18.03	21.44
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	7.00	8.97	$\times$	$\times$
$n = 150$	9.81	13.08	16.20	$\times$
$n = 200$	13.79	17.01	18.11	20.68
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	8.15	9.36	$\times$	$\times$
$n = 150$	10.04	14.11	17.04	$\times$
$n = 200$	14.52	17.00	19.26	21.09

Clusters touching each other				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	7.74	8.99	$\times$	$\times$
$n = 150$	10.11	13.03	14.98	$\times$
$n = 200$	15.72	18.83	19.97	22.54
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	7.71	9.17	$\times$	$\times$
$n = 150$	10.81	13.09	16.26	$\times$
$n = 200$	16.01	19.01	20.13	23.18
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	9.21	9.72	$\times$	$\times$
$n = 150$	11.14	14.89	18.14	$\times$
$n = 200$	17.01	19.98	21.21	24.00

Table 4.31: Least squares approximation of the unknown constant

Clusters well separated				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.27	1.46	×	×
$n = 150$	1.10	1.34	1.46	×
$n = 200$	1.00	1.07	1.31	1.47
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.25	1.57	×	×
$n = 150$	1.20	1.39	1.52	×
$n = 200$	1.00	1.11	1.41	1.46
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.27	1.60	×	×
$n = 150$	1.23	1.40	1.51	×
$n = 200$	1.01	1.17	1.36	1.50

Clusters touching each other				
	$\rho = 0$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.48	1.83	×	×
$n = 150$	1.39	1.75	1.84	×
$n = 200$	1.40	1.70	1.78	1.94
	$\rho = 0.5$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.54	1.87	×	×
$n = 150$	1.44	1.79	1.91	×
$n = 200$	1.40	1.78	1.86	1.98
	$\rho = 0.9$			
	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	1.63	1.91	×	×
$n = 150$	1.56	1.83	1.94	×
$n = 200$	1.51	1.83	1.90	1.99

Table 4.32: $\mathcal{V}1_{(1,2)}$ variability ratios

Chapter 5

Multivariate Highlighters : Discriminant Analysis-Based Weighting

5.1 Introduction

In section 2.6 we introduced the idea of "highlighters" in the univariate context. These were weights (obtained prior to performing any cluster analysis) that emphasized variables which possessed strong cluster structure. The results in section 2.6.2 showed that univariate highlighter strategies that took into account the latent cluster structure led to improved performance across most of the data sets studied. In contrast, the challenging example of the *Crabs* data set underscored the limitations of univariate highlighting, pointing to the need for a multivariate highlighting approach to tackle such difficult cluster structures.

In this chapter, we will address highlighting in the multivariate context. The ideas will draw on the intuition underlying the well-known statistical technique of Linear Discriminant Analysis (DA). Experimental results from the application of the methods on a variety of simulated data sets (with specific interesting cluster structures) and a few real data sets will be provided.

5.2 DA and Pseudo-DA for Multivariate Highlighting

DA is a widely applied statistical technique used to study the differences between two or more groups of objects simultaneously. It is used to "discriminate" between known groups on the basis of a set of variables (or features), to study how well the variables discriminate among the groups, and also to help in identifying which variables are the most powerful discriminators. Another purpose of DA is to use *training* data to derive

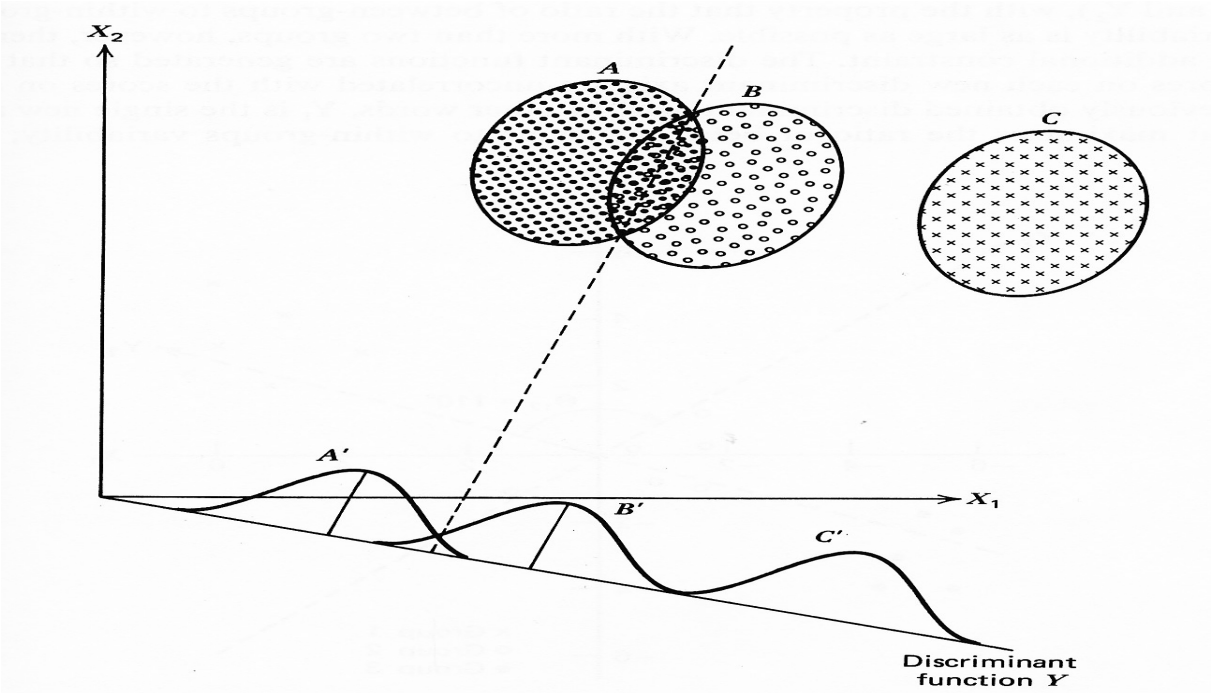

Figure 5.1: Illustration of three groups and a single discriminant function

one or more mathematical functions called "discriminant functions" (or discriminant variables), for the purpose of *classification* of *test* data.

To illustrate the idea behind classical DA, consider the three group problem shown in the two-dimensional scatter plot in Figure 5.1 (also see, [14]). The goal is to identify a new axis (*i.e.*, a discriminant coordinate) such that the projection of all the data points onto the new axis would maximize the differences among the three group means. However, as we can see in Figure 5.1, although groups A and B overlap, a single discriminant function might not distinguish between all three groups. Hence, with three or more groups, a single axis may not satisfactorily distinguish the groups. In general, with g groups and p dimensional data, there are $k = min(p, g - 1)$ possible discriminant axes (typically, $p > g$.) However, not all of the k axes might display statistically significant variation among the groups and in practice, fewer than $g - 1$ discriminant functions might be needed. This is the intuition underlying classical DA.

Considering the same g group problem in the clustering context (where the groups or clusters are unknown to begin with), we could think of the discriminant axes to

be defining a new space or representation of the data points wherein the discriminant variables "highlight" the differences among the latent clusters. This is the unsupervised analogue of classical DA which we call, "pseudo-DA". Hence, parallel to classical DA, pseudo-DA could be thought of as a technique to find optimal linear combinations of the original variables that define a new space, where the differences among the latent clusters are maximized. In section 5.2.1 we describe the computations involved in classical DA and in section 5.2.2 we discuss the framework for pseudo-DA.

5.2.1 Eigenanalysis of $\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$ - Classical DA

This section describes the classical DA framework when the groups are known *a priori*. It may be noted that this description is purely to illustrate the computations involved in classical DA and is not directly applicable in the clustering context, as in the latter, the groups are not known ahead of time. Rather the goal is to unearth them. As described in [33], with g groups and n_i, $i = 1 \ldots g$ observations in each group, we could compute the $p \times p$ pooled within-groups covariance matrix,

$$\mathbf{W} \;=\; \frac{1}{n-g} \sum_{i=1}^{g} (n_i - 1)\mathbf{S}_i, \tag{5.1}$$

where $\mathbf{S}_i$ is the sample covariance matrix of the i^{th} group.

Furthermore, we could define the $p \times p$ between-groups covariance matrix,

$$\mathbf{B} \;=\; \frac{1}{g-1} \sum_{i=1}^{g} n_i \, (\overline{\mathbf{y}}_i - \overline{\mathbf{y}})(\overline{\mathbf{y}}_i - \overline{\mathbf{y}})', \tag{5.2}$$

where $\overline{\mathbf{y}}_i$ is the sample mean vector of the i^{th} group and $\overline{\mathbf{y}}$ is the overall mean vector.

If $\mathbf{z} \;=\; \mathbf{a}'\mathbf{y}$ denotes a linear combination of the original variables, a one-way ANOVA for the derived variable $\mathbf{z}$ will lead to the following F-ratio of the between-groups mean square to the within-groups mean square:

$$F_a \;=\; \frac{\mathbf{a}'\mathbf{B}\mathbf{a}}{\mathbf{a}'\mathbf{W}\mathbf{a}}. \tag{5.3}$$

If one were to choose $\mathbf{a}$ so as to maximize this F_a-ratio (*i.e.*, maximize the group differences), the required $\mathbf{a}$ would be the eigenvector, $\mathbf{a}_1$, corresponding to the largest eigenvalue, c_1, of $\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$. This would give the first discriminant coordinate (or,

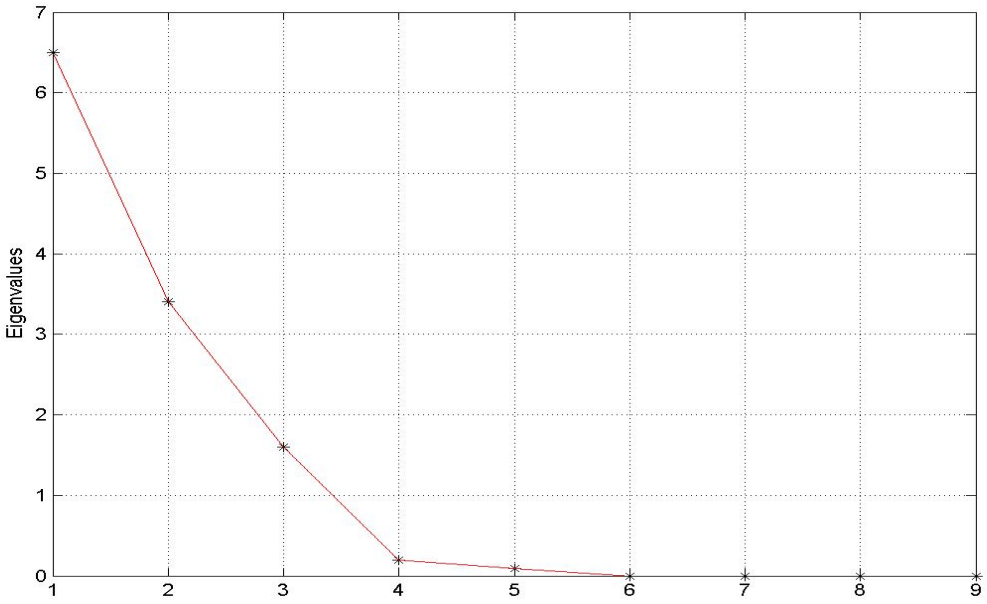

Figure 5.2: Sample *scree* plot

CRIMCOORD), $\mathbf{z}_1 = \mathbf{a}_1'\mathbf{y}$. Having determined $\mathbf{a}_1$, we could seek a second linear combination of the original variables, $\mathbf{z}_2$, which has the next largest F_a-ratio, subject to the condition that it is uncorrelated to $\mathbf{z}_1$, within groups. The required solution for the coefficients in the second linear combination would be the eigenvector, $\mathbf{a}_2$, corresponding to the second largest eigenvalue, c_2, of $\big[\mathbf{W}\big]^{-1}\big[\mathbf{B}\big]$. Hence, all that is involved computationally is an eigenanalysis of $\big[\mathbf{W}\big]^{-1}\big[\mathbf{B}\big]$, leading to the ordered eigenvalues $c_1 \geq c_2 \geq \ldots c_k$, where, $k = min(p,\ g\ \text{-}\ 1)$. (The remaining p - k eigenvalues are all zero.)

As mentioned earlier, not all of the k CRIMCOORDS might be significant in their discriminatory power. Hence, to obtain a reduced-rank model to parsimoniously, but effectively, describe the measured differences among the groups, we would need a descriptive index of importance of the discriminant variables. One approach to do this is proposed in [7], and is called the *scree* plot. In this approach (named after the rubble at the bottom of a cliff), the eigenvalues of each CRIMCOORD are plotted in successive

order and then an "elbow" in the curve is identified. Figure 5.2 depicts this *scree*-analogy ($k = 5$), where the first four eigenvalues show the "cliff", and the rest, the "rubble". Hence, only the first four CRIMCOORDS are retained. The rationale for the *scree* plot is that since the DA procedure extracts CRIMCOORDS in successive order of magnitude, the substantive axes appear first, followed by the remaining trivial axes which account for only a small proportion of the total variability. The space defined by these k CRIMCOORDS or a subset of them using the first t ($t \leq k$) CRIMCOORDS is called the *discriminant space*. Additionally, it may be noted that Mahalanobis distance (using $\left[\mathbf{W}\right]^{-1}$) in the space of the original variables would now be equivalent to using Euclidean distance in the discriminant space spanned by the CRIMCOORDS.

In the clustering context, since the clusters are unknown, the $\mathbf{W}$ and $\mathbf{B}$ matrices are also unknown. Nevertheless, we could still apply the classical DA intuition described above, to obtain linear combinations of the original variables that highlight the latent cluster structure. This is the idea behind pseudo-DA, described next.

5.2.2 Eigenanalysis of $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_2)}\right]$ - Pseudo DA

Equation (3.2) in section 3.2 provided the standard multivariate decomposition of the total sums of squares and cross products matrix into the within-groups and between-groups components when the groups were known *a priori*. In the clustering context we could develop similar "within" and "between" measures by using some of the methods described in this thesis. When the clusters are unknown, we could first obtain a within-clusters measure similar to the $\mathbf{W}^*$ matrix using the $\mathbf{W}^*_{(m_1)}$ algorithm described in section 3.2. Subsequently, a between-clusters measure, $\mathbf{B}^*_{(m_2)}$, could be developed using the set $\{\mathsf{K}\}$ of m_2 farthest-apart neighbors (farthest-apart in Mahalanobis sense, computed using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$) as,

$$\mathbf{B}^*_{(m_2)} \;=\; \frac{1}{n} \sum_{\substack{i<j \\ i,j \in \mathsf{K}}} (\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)'. \tag{5.4}$$

We could then use the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{B}^*_{(m_2)}$ matrices as substitutes for the unknown

$\mathbf{W}$ and $\mathbf{B}$ matrices, leading to the ratio (analogous to equation 5.3),

$$s_v \;=\; \frac{\mathbf{v}'\mathbf{B}^*_{(m_2)}\mathbf{v}}{\mathbf{v}'\mathbf{W}^*_{(m_1)}\mathbf{v}}. \tag{5.5}$$

Maximizing this ratio with respect to $\mathbf{v}$ could again be accomplished by an eigen-analysis of the matrix product $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_2)}\right]$, leading to the ordered eigenvalues, $h_1 \geq h_2 \geq \ldots h_p$, and the corresponding set of eigenvectors, $\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_p$. It may be noted here that since the $\mathbf{W}^*_{(m_1)}$ and $\mathbf{B}^*_{(m_2)}$ matrices would both tend to be of full rank, the (non-symmetric) matrix product $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_2)}\right]$, would also be of full rank. In such a setting, the subset r ($r < p$) of significant discriminating axes could be chosen by setting r equal to the number of "dominant" eigenvalues of the matrix product $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_2)}\right]$. We could use the *scree* plot described in the previous section to accomplish this. Using the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_r$, we could then determine the r linear combinations of the variables (or, pseudo-CRIMCOORDS), $\mathbf{q}_i = \mathbf{v}_i'\mathbf{y},\ i = 1, 2, \ldots r$, so that they account for cluster separation in decreasing order. The $\mathbf{q}_i$'s would be similar in spirit to *discriminant coordinates* as in classical DA and the space defined by these axes would be similar to a *discriminant space*. Also, note that this subset of pseudo-CRIMCOORDS could directly be applied as input to a CA procedure. Furthermore, since the pseudo-CRIMCOORDS are by definition designed to pull the groups apart, pairwise plots of the first few of them would graphically depict distinct groupings. We call this procedure, analogous to classical DA, as pseudo-DA.

Choice of "m_2"

In [34], Gnanadesikan *et al.*, developed a statistic (q-statistic) and a graphical aid for picking the m_1 nearest neighbors in the context of the $\mathbf{W}^*_{(m_1)}$ algorithm. Results from simulations did suggest stability of the $\mathbf{W}^*_{(m_1)}$ measure over a range of different m_1 values. But for $\mathbf{B}^*_{(m_2)}$ however, the choice of the number m_2 of farthest-apart pairs is an important feature. In general, when the clusters are of equal size, with $n_i = \frac{n}{g},\ i = 1 \ldots g$, observations in each cluster, the number of between-cluster pairs would be larger than the number of within-cluster pairs, with the number of between-cluster pairs increasing as the number of clusters, g, increases. On the other hand, with

unequal cluster sizes, for a given n and g, it is possible that the number of within-cluster pairs is equal to, or larger than the number of between-cluster pairs. Hence, to obtain a reliable measure of between-clusters dispersion based on the m_2 farthest-apart point pairs, a prudent strategy would be to try a few alternative choices for m_2. Hence, we consider four different choices for m_2, as given below:

- $m_2 = m_1$ (m_1 based on the $\frac{2}{3}^{rd}$ rule)
- $m_2 = f$ (recall from section 1.2, that $f = \binom{n}{2}$ - m_1)
- $m_2 = f_1 = \frac{m_1}{3}$ (m_1 based on the $\frac{2}{3}^{rd}$ rule)
- $m_2 = f_2 = \frac{\binom{n}{2}}{2}$

Hence, starting from the simple choice of m_2 equal to m_1 based on the $\frac{2}{3}^{rd}$ rule, we explore three other choices for m_2 ranging from one that is much larger than m_1, to another where m_2 is set much smaller than m_1. The next section provides the experimental design followed by corresponding results and conclusions.

5.3 Experimental design

To study the performance of the multivariate highlighting (pseudo-DA) procedure under a variety of cluster structures, we used the same set of simulated and real data sets that were reported earlier in sections 4.2.2 and 4.3.1 respectively. Also, it may be noted here that the experiments on the simulated data sets in group 2 were based on only one random sample corresponding to each of the 54 (n, p, ρ, cluster centers) combinations. A summary of the data sets used (also see section 4.3.1 on page 76 of the thesis) is given below:

1. Simulated data
    - Group 1 - Data sets $\mathcal{D}1$ - $\mathcal{D}14$, displaying a variety of cluster scenarios ranging from simple homogenous clusters to heterogenous structures.
    - Group 2 - Data sets with different (n, p, ρ) combinations and two types of cluster separations

  - Set 1: 27 data sets with five far-apart clusters
  - Set 2: 27 data sets with five nearly overlapping clusters

2. Real data

   - $\mathcal{D}15$ (*Iris* data)
   - $\mathcal{D}16$ (*Crabs* data)
   - $\mathcal{D}17$ (*Wine* data)
   - $\mathcal{D}18$ (*Cells* data)

In each case the $\mathbf{W}^*_{(m_1)}$ matrix was first computed using the $\mathbf{W}^*_{(m_1)}$ algorithm with m_1 based on the $\frac{2}{3}^{rd}$ rule. Subsequently, the m_2 farthest-apart point pairs were identified (farthest-apart in Mahalanobis sense, using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$). The between-clusters measure $\mathbf{B}^*_{(m_2)}$ was then computed using these m_2 farthest-apart pairs of points. This was done for all four choices of m_2. Additionally, since we know the the groups *a priori* for all data sets, we also computed their corresponding $\mathbf{W}$ and $\mathbf{B}$ matrices. Hence, for each data set, we have five matrix products - four $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_2)}\right]$ matrices corresponding to the four choices of m_2 and one $\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$ matrix, our gold standard for comparison.

Subsequently, for each data set an eigenanalysis was done on each of the five matrix products, resulting in five sets of corresponding eigenvalues and eigenvectors. This defined one set of CRIMCOORDS and four sets of pseudo-CRIMCOORDS respectively, for each data set. For each set, the *scree* plot was then used to determine by eye, the subset of significant coordinates, *i.e.*, those with most discriminatory power. Cluster analysis was then performed in the space of the significant pseudo-CRIMCOORDS using hierarchical cluster analysis (HCA) with average linkage. As before, the resulting dendrograms were cut to partition the respective data sets into the known number of clusters. Performance of the pseudo-DA procedure was then evaluated by studying the misclassification error counts for all data sets.

5.4 Results

Table 5.2 displays the misclassification error counts for data sets $\mathcal{D}1$ – $\mathcal{D}18$. Results from classical DA as well as pseudo-DA using the four different choices of m_2 (m_1, f, f_1 and f_2) are provided. Additionally, previous results from scaling the data sets by $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ (*i.e.*, HCA applied to the sphericized data) are also displayed. This is done to study the differential advantage of using pseudo-DA by incorporating between-cluster information.

Among the simulated data sets in group 1 where the clusters are well separated (*viz.*, $\mathcal{D}1$, $\mathcal{D}3$, $\mathcal{D}7$, $\mathcal{D}8$ and $\mathcal{D}11$), the pseudo-DA procedure results in perfect cluster recovery for all four choices of m_2. It may also be noted that these data sets are handled almost as well by using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone. However, in data sets $\mathcal{D}2$ and $\mathcal{D}5$ where the clusters nearly overlap, the error counts from using pseudo-DA are lower, albeit marginally, than using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone. Also, similar marginal improvements are observed for data sets $\mathcal{D}6$, $\mathcal{D}9$, $\mathcal{D}10$, $\mathcal{D}12$, $\mathcal{D}13$, and $\mathcal{D}14$.

It is interesting to note that in case of data set $\mathcal{D}4$, setting $m_2 = m_1$ results in 21 errors of misclassification, while setting $m_2 = f$ results in only one error, comparable to that of classical DA. Figure 5.3 displays the corresponding dendrogram obtained using $m_2 = m_1$, while Figure 5.4 displays the dendrogram obtained when $m_2 = f$. Notice that the dendrogram displayed in Figure 5.4 depicts a more "organized" hierarchical tree structure (five major branches corresponding to the five clusters), conducive to near perfect cluster recovery (with only one misclassified observation) by cutting the dendrogram as shown.

For all the simulated data sets in group 1, either the first two or three pseudo-CRIMCOORDS are retained after a visual inspection of the corresponding *scree* plots. Overall, the performance of the pseudo-DA procedure appears promising. Although for a few data sets the pseudo-DA procedure seems to offer only a marginal improvement in results, it consistently yields improved cluster recovery when compared to using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone. This observation suggests that the inclusion of the between-clusters measure, $\mathbf{B}^*_{(m_2)}$, can improve the accuracy of the subsequent clustering step.

Among the real data sets, the biggest improvement in results is observed in the *Iris* ($\mathcal{D}15$) data set. The error count of five (for all choices of m_2), is comparable with that obtained using classical DA. Furthermore, this is a substantial improvement in performance compared with 39 errors that is obtained using just $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$. Hence pseudo-DA works very well in this case, as also seen in Figures 5.5 and 5.6. Figure 5.5 shows a scatter plot of $\mathcal{D}15$ in the space of the first two pseudo-CRIMCOORDS for the case when $m_2 = m_1$. While one of the clusters is well-separated from the other two, only two pseudo-CRIMCOORDS would suffice to discriminate among the three clusters. Figure 5.6 subsequently shows the corresponding dendrogram obtained using a HCA in the space of the first two pseudo-CRIMCOORDS. Notice that cutting the dendrogram at the level shown results in three distinct clusters.

In the *Crabs* data set ($\mathcal{D}16$) however, the pseudo-DA procedure does not seem to provide much improvement in results. The lowest error count obtained is when $m_2 = f_1$, when the error count drops to 50, compared to 54 errors obtained using classical DA. Figure 5.7 depicts the scatter plot in the space of the first two pseudo-CRIMCOORDS for the case when $m_2 = f_1$, while Figure 5.8 displays the corresponding dendrogram.

Also, for illustrative purposes, *scree* plots for the *Iris* and *Crabs* data sets are given in Figures 5.11 and 5.12, respectively. The *scree* plots, taken together with the scatter plots of the *Iris* and *Crabs* data in the space of the first two pseudo-CRIMCOORDS (see Figures 5.5 and 5.7), clearly show the decreasing returns for using the second pseudo-CRIMCOORD in addition to the first. The first pseudo-CRIMCOORD is more helpful in pulling at least some of the clusters apart while the second seems to be not useful in further distinguishing the clusters. For the *Crabs* data the plot in the space of the first two CRIMCOORDS from classical DA seems to indicate the same phenomenon. However, it is important to point out that although pseudo-DA does not appear to have contributed much in the case of the *Crabs* data set, the scatter plot displayed in Figure 5.7 (in the space of the first two pseudo-CRIMCOORDS) bears a striking resemblance to the CRIMCOORD plot shown in Figure 5.9, that is obtained using classical DA, with a majority of the 50 misclassified observations arising from the clusters labeled with blue and red solid dots in Figure 5.7. Furthermore, Figure 5.10 displays the dendrogram

obtained using the three CRIMCOORDS from classical DA (Note: It may be noted that similar results are obtained even when only the first two CRIMCOORDS are used). This poor tree structure configuration might explain the high error rate resulting from the HCA notwithstanding the fact that it is based on classical DA.

For the *Wine* and *Cells* data sets, although the pseudo-DA procedure results in much lower error counts compared to using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone, it does not seem to perform well in comparison to classical DA. As seen in Table 5.2, for the *Wine* data set ($\mathcal{D}17$), for all choices of m_2, the error counts are in the range 46 – 50, while a HCA on data scaled by $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone, results in 79 errors of misclassification. Similarly, with the *Cells* data set ($\mathcal{D}18$), the error count from pseudo-DA ranges from 50 – 64 for the four choices of m_2, while using $\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$ alone results in 87 errors.

Table 5.1 displays the number of significant CRIMCOORDS and pseudo-CRIMCOORDS retained based on the respective *scree* plots for the simulated data sets in group 1 and the four real data sets. Notice that in almost all cases the numbers of discriminant axes retained by DA and pseudo-DA are the same.

Tables 5.3 – 5.8 display corresponding results for the simulated data sets in group 2. An interesting observation gleaned from all the tables is the consistent improvement in performance when the between-clusters information is included in the analysis. As a result, the pseudo-DA approach performs well for all the data sets studied. Furthermore, as one would expect, the performance of pseudo-DA is excellent when the clusters are well separated from each other, as seen in the small error counts in tables 5.3, 5.5 and 5.7. Among the data sets where the clusters overlap slightly, in most of the cases, the performance of pseudo-DA is no worse than the classical DA approach. It may be noted that although for all the data sets studied in this chapter, m_1 is based on the $\frac{2}{3}^{rd}$ rule, as discussed in the previous chapter, in situations where the clusters nearly overlap, setting m_1 smaller than that given by the $\frac{2}{3}^{rd}$ rule might lead to more accurate $\mathbf{W}^*_{(m_1)}$ estimates, and hence, improved cluster recovery using the pseudo-DA approach. Also, interestingly, the results show that the pseudo-DA procedure generally does not seem too sensitive to the choice of m_2. Additionally, recall that all of the simulated data sets in group 2 consist of five clusters. However, *scree* plots suggest retaining either three or

four significant CRIMCOORDS across the 54 data sets in group 2, while the number of pseudo-CRIMCOORDS retained ranges from two to four. Figure 5.13 displays a sample *scree* plot for a simulated data set ($p = 50$) in group 2.

5.4.1 Experiments using gaussian mixture model-based clustering – (MCLUST)

All the error counts reported in this chapter are based on a HCA. However, since the pseudo-DA approach is not tied to any particular method of clustering, as we did in section 4.3.5, we also studied its performance using the gaussian-mixture approach, MCLUST (also see [24], [25] and [26]). We applied the significant pseudo-CRIMCOORDS from each of the four real data sets respectively, as input to MCLUST. As done in the previous chapter, we programmed MCLUST to extract the known number of clusters, while we let it pick its best covariance matrix parameterization in each case. Subsequently, the resulting cluster labels were compared with the known labels to get misclassification error counts. The results are given as shown.

- $\mathcal{D}15$ (*Iris*): 4 (3)
- $\mathcal{D}16$ (*Crabs*): 79 (12)
- $\mathcal{D}17$ (*Wine*): 5 (1)
- $\mathcal{D}18$ (*Cells*): 121 (27)

The first number on each row indicates the error count using the raw data, while the second number (within parentheses) indicates the error count from MCLUST based on the pseudo-CRIMCOORDS input. It may be noted that in all cases, m_2 was set equal to m_1 based on the $\frac{2}{3}^{rd}$ rule. Notice that MCLUST consistently performs better when applied in the discriminant space spanned by the pseudo-CRIMCOORDS.

5.5 Conclusion

This chapter described a framework for multivariate "highlighting" following the DA-based intuition to maximize separations among the latent clusters. Overall, the application of the multivariate strategy resulted in improving the performance for most of the data sets studied. This provided evidence that incorporating between-clusters information took things in the right direction. Although the multivariate strategy required a user-input number m_2, for the number of largest pairwise distances, the results indicated that for most of the data sets studied, the approach did not seem too sensitive to the choice of m_2, given the fixed value for m_1 based on the $\frac{2}{3}^{rd}$ rule.

Overall, the pseudo-DA framework, as a technique to perform multivariate "highlighting", performed commendably, and indeed, merits further investigation.

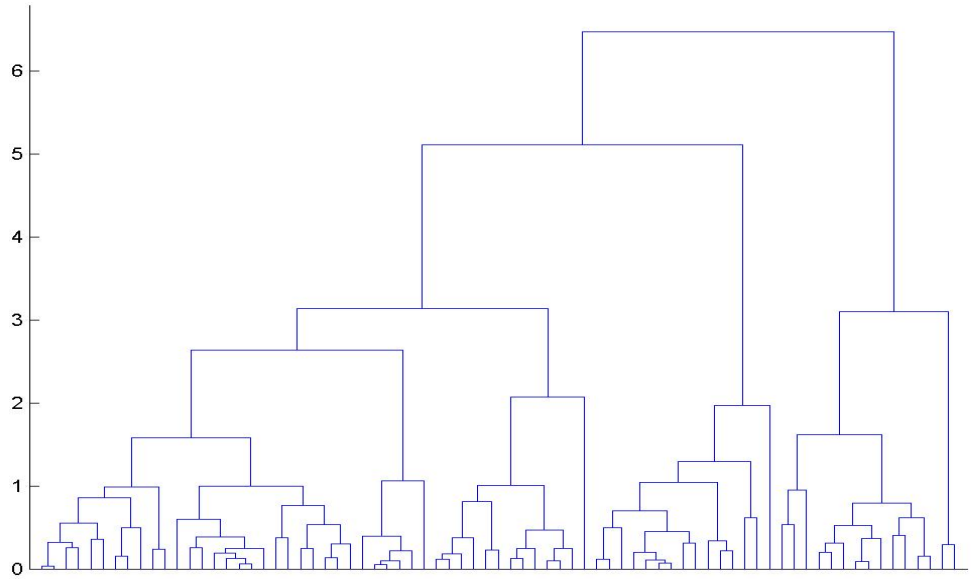

Figure 5.3: Dendrogram for data set $\mathcal{D}4$; using pseudo-DA with $m_2 = m_1$

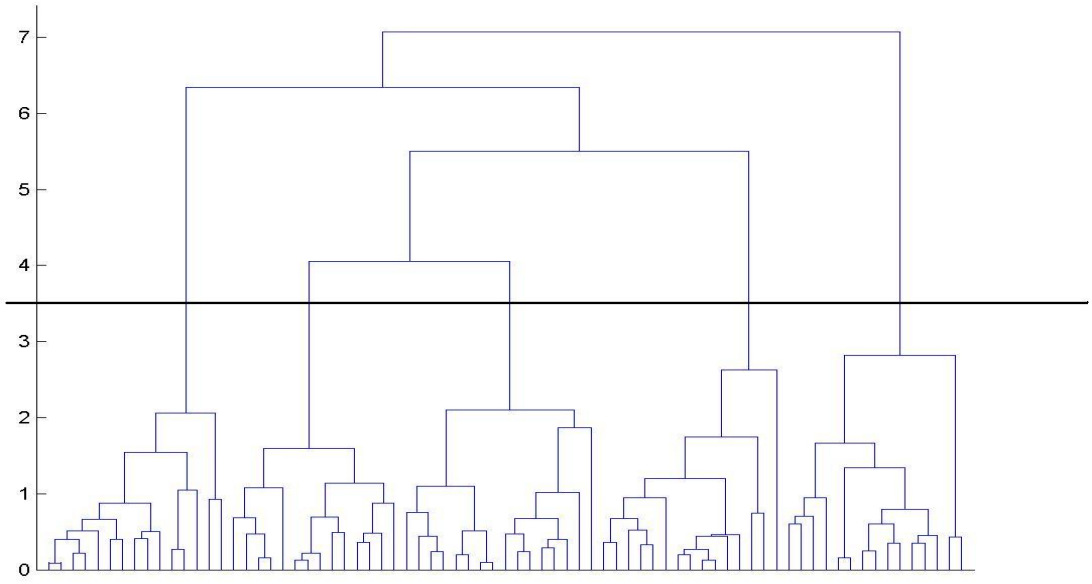

Figure 5.4: Dendrogram for data set $\mathcal{D}4$; using pseudo-DA with $m_2 = f$

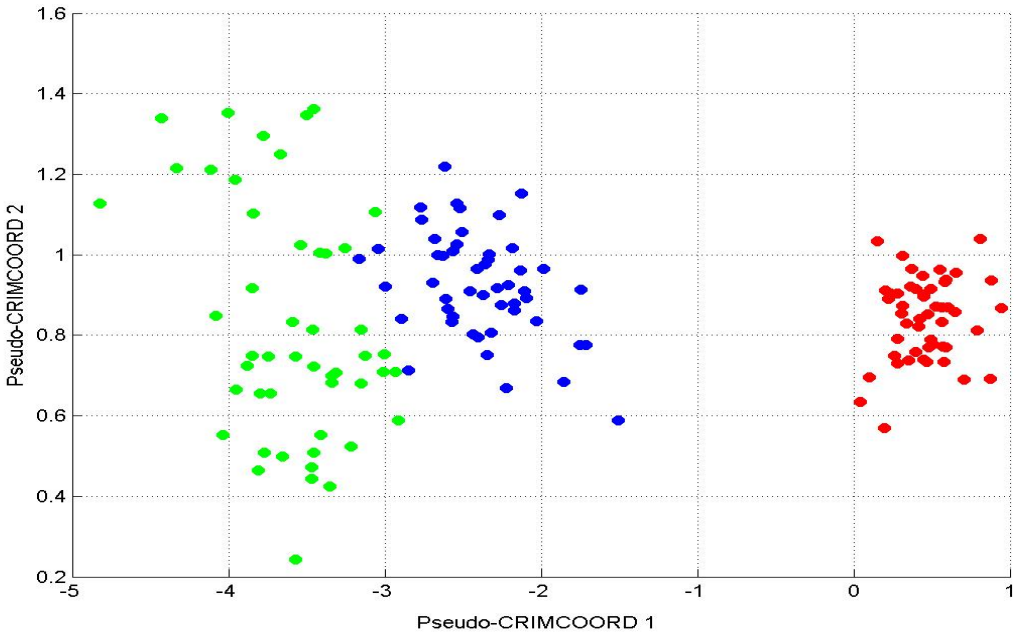

Figure 5.5: Scatter plot of *Iris* data in the space of the first two pseudo-CRIMCOORDS; $m_2 = m_1$

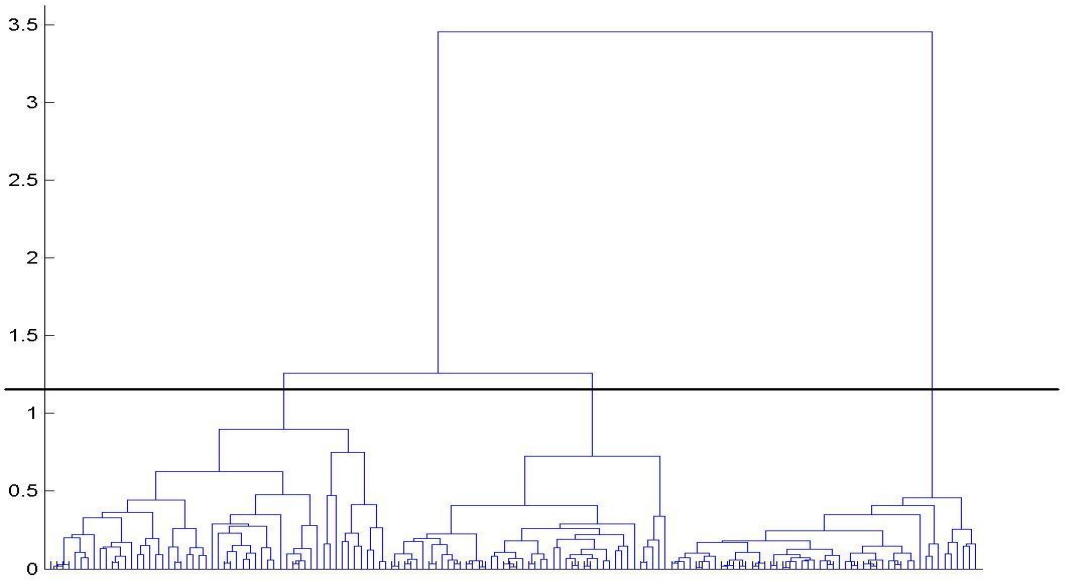

Figure 5.6: Dendrogram for *Iris* data; using pseudo-DA with $m_2 = m_1$

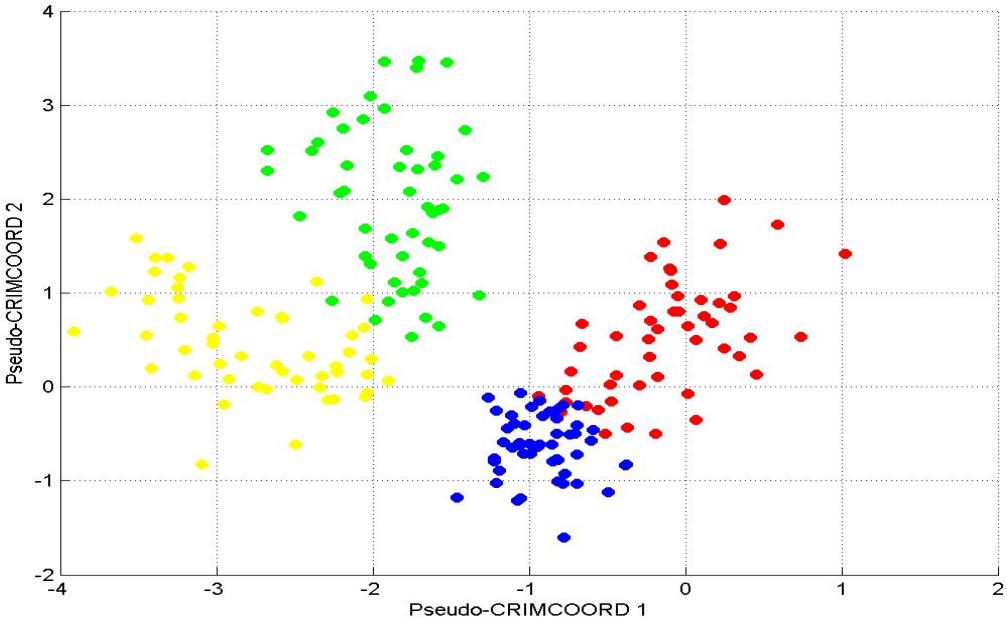

Figure 5.7: Scatter plot of *Crabs* data in the space of the first two pseudo-CRIMCOORDS; $m_2 = f_1$

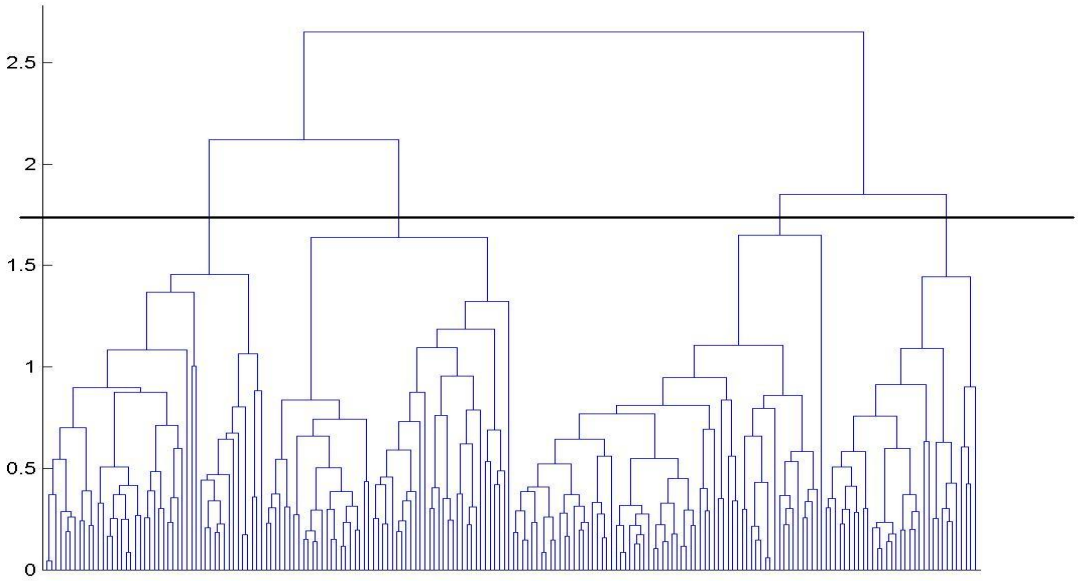

Figure 5.8: Dendrogram for *Crabs* data; using pseudo-DA with $m_2 = f_1$

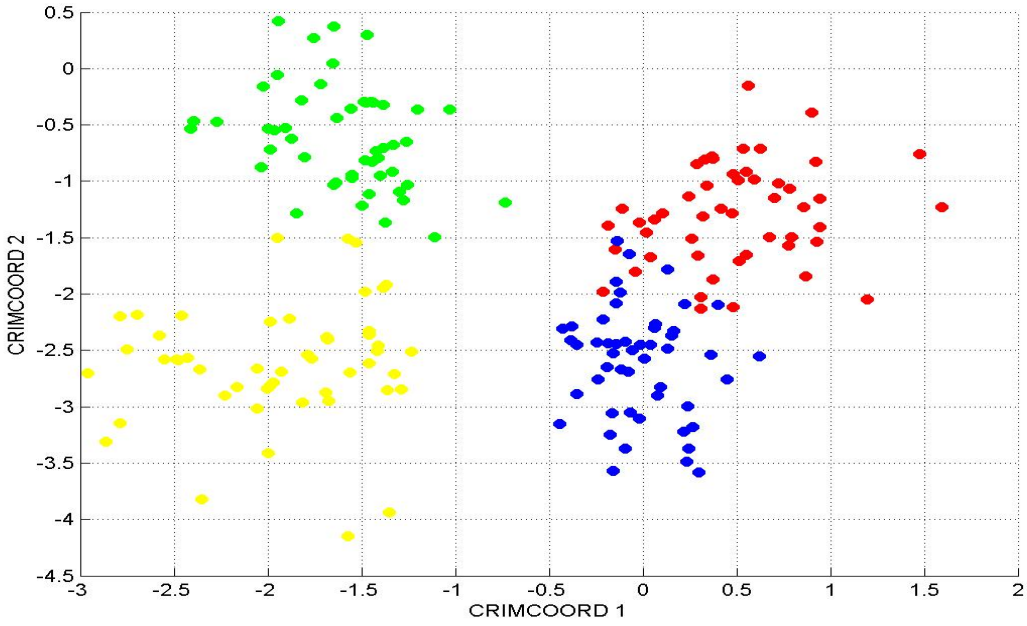

Figure 5.9: Scatter plot of *Crabs* data in the space of the first two CRIMCOORDS, from classical DA

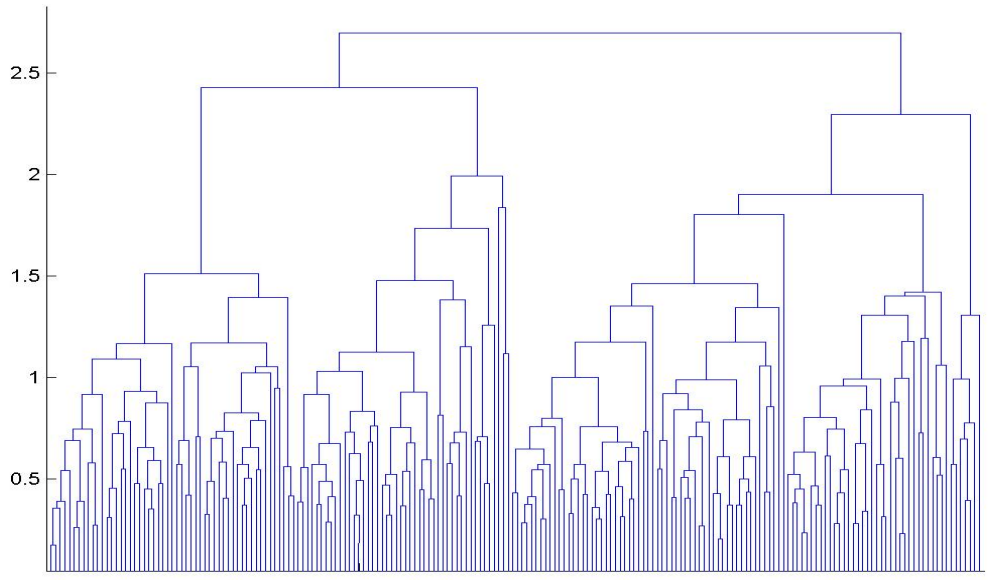

Figure 5.10: Dendrogram for *Crabs* data; using classical DA

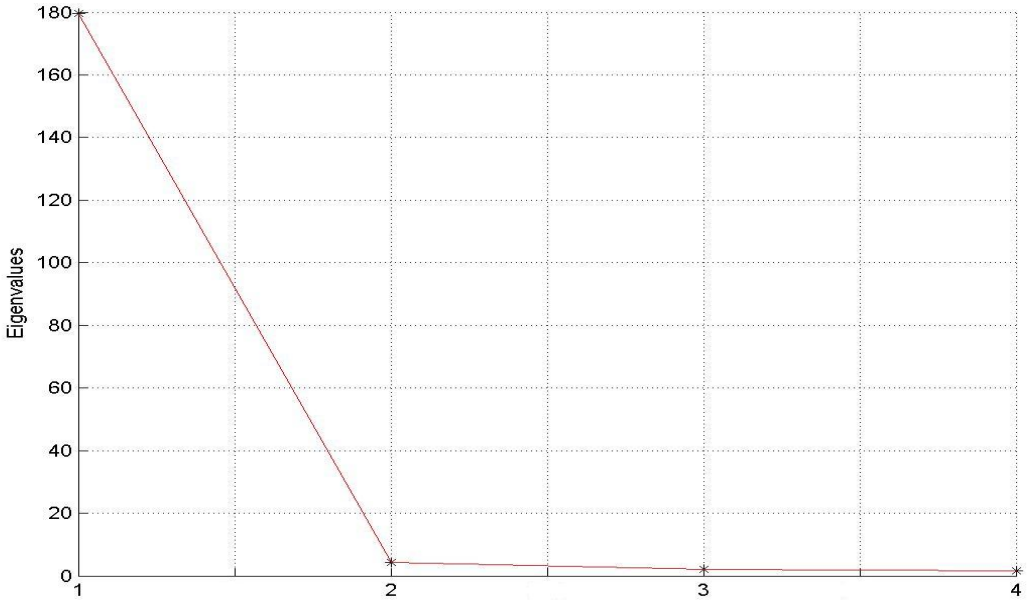

Figure 5.11: *Scree* plot for *Iris* data; $m_2 = m_1$

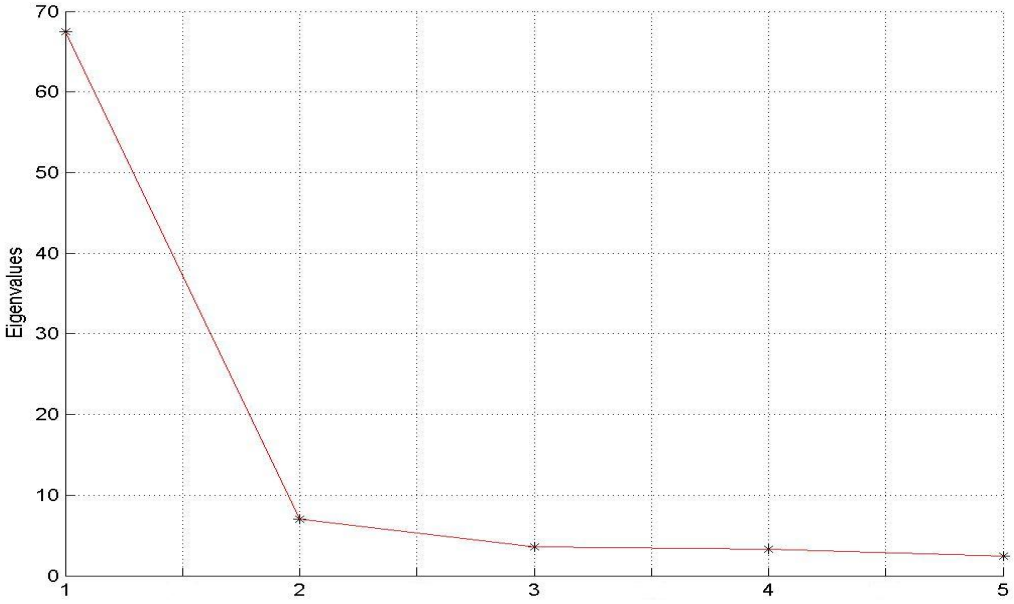

Figure 5.12: *Scree* plot for *Crabs* data; $m_2 = m_1$

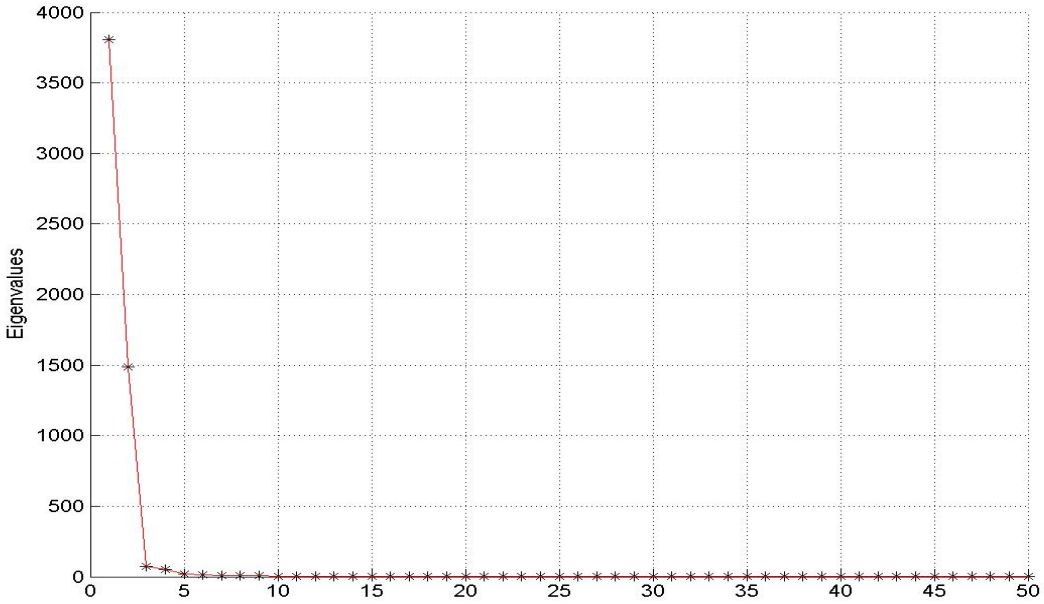

Figure 5.13: Sample *scree* plot for a simulated data set ($p = 50$) in group 2

Data set	p	g	# CRIMCOORDS	# pseudo-CRIMCOORDS
$\mathcal{D}1$	5	5	3	3
$\mathcal{D}2$	5	5	3	3
$\mathcal{D}3$	5	5	3	3
$\mathcal{D}4$	5	5	3	3
$\mathcal{D}5$	5	5	3	3
$\mathcal{D}6$	5	3	2	2
$\mathcal{D}7$	5	3	2	2
$\mathcal{D}8$	5	3	2	2
$\mathcal{D}9$	5	3	2	2
$\mathcal{D}10$	5	3	2	2
$\mathcal{D}11$	5	5	2	2
$\mathcal{D}12$	5	5	2	2
$\mathcal{D}13$	5	5	2	2
$\mathcal{D}14$	5	5	2	2
$\mathcal{D}15$	4	3	2	2
$\mathcal{D}16$	5	4	3	2
$\mathcal{D}17$	13	3	2	3
$\mathcal{D}18$	3	3	2	2

Table 5.1: Comparison of number of significant CRIMCOORDS and pseudo-CRIMCOORDS retained

Strategy	$\mathcal{D}1$	$\mathcal{D}2$	$\mathcal{D}3$	$\mathcal{D}4$	$\mathcal{D}5$	$\mathcal{D}6$	$\mathcal{D}7$	$\mathcal{D}8$	$\mathcal{D}9$
$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	0	36	1	29	31	16	1	0	6
$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	18	0	2	21	10	0	0	1
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	27	0	21	30	16	0	0	5
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	28	0	1	29	12	0	0	4
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	21	0	20	23	15	0	0	5
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	25	0	16	28	13	0	0	5

Strategy	$\mathcal{D}10$	$\mathcal{D}11$	$\mathcal{D}12$	$\mathcal{D}13$	$\mathcal{D}14$	$\mathcal{D}15$	$\mathcal{D}16$	$\mathcal{D}17$	$\mathcal{D}18$
$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	7	0	22	13	16	39	55	79	87
$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	2	0	11	11	13	4	54	4	35
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	6	0	18	12	13	5	54	50	60
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	7	0	16	11	12	5	54	52	64
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	5	0	14	10	10	5	50	46	50
$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	6	0	15	12	12	5	53	48	55

Table 5.2: Comparison of errors of misclassification using different highlighter strategies

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	0	4	$\times$	$\times$
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	1	0	5	$\times$
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	0	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	0	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0	2	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0	2	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	2	3	4	5
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	1	0	1	0
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	1	0	0	1
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	1	0	0	1
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0	1	1
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	1	0	0	1

Table 5.3: Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0$

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	9	4	×	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	3	2		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	4	4		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	6	4		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	2	1		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	4	4		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	5	19	21	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	3	9	12	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	3	10	13	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	4	9	12	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	2	6	9	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	3	7	11	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	25	22	24	25
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	12	12	13	13
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	11	10	10	14
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	15	13	13	14
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	9	8	9	8
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	10	12	11	12

Table 5.4: Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0$

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	0	3	$\times$	$\times$
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	0	0	5	$\times$
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	0	2	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	0	2	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0	2	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0	2	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	2	2	4	4
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	0	3	4
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	0	2	4
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0	0	0
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	0	3	4
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0	2	3

Table 5.5: Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0.5$

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	5	12	×	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	2	6		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	3	7		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	4	8		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	1	4		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	3	6		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	22	13	17	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	13	10	10	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	14	10	11	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	15	11	11	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	9	8	7	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	11	11	10	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	25	23	19	21
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	13	12	10	9
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	12	13	11	10
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	13	13	11	10
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	8	9	7	6
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	11	11	11	9

Table 5.6: Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0.5$

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	1	6	×	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	2		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	2		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	1		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	3		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	2		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	2	3	3	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	0	1	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	1	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	0	0	0	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	1	1	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0	0	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	3	4	4	5
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	1	1	2	3
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	0	1	2	2
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	1	0	0	1
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	0	1	1	2
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	0	0	0	0

Table 5.7: Comparison of errors of misclassification using different highlighter strategies, clusters well separated, $\rho = 0.99$

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 75$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	7	12	×	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	5	9		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	5	8		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	6	9		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	2	4		
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	4	7		

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 150$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	18	16	19	×
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	9	10	11	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	7	10	10	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	8	11	11	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	3	7	7	
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	5	11	9	

	Strategy	$p = 5$	$p = 50$	$p = 100$	$p = 150$
$n = 200$	$\left[\mathbf{W}^*_{(m_1)}\right]^{-1}$	26	21	21	25
	$\left[\mathbf{W}\right]^{-1}\left[\mathbf{B}\right]$	12	12	12	14
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(m_1)}\right]$	10	9	9	12
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f)}\right]$	11	10	10	12
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_1)}\right]$	8	6	7	8
	$\left[\mathbf{W}_{(m_1)}\right]^{-1}\left[\mathbf{B}^*_{(f_2)}\right]$	9	8	9	10

Table 5.8: Comparison of errors of misclassification using different highlighter strategies, clusters touching each other, $\rho = 0.99$

Chapter 6

Conclusions and Further Work

In this dissertation we have explored some ideas for preprocessing data prior to doing a CA. This chapter summarizes the findings of this work, while identifying further research that it suggests. Scaling variables to place them on an equal footing or to emphasize those most likely to aid detection of clusters is a critical preprocessing step in CA. This dissertation compared the performances of a number of alternatives. They varied in degree of complexity from very simple weights based on the order statistics of the inter-observation distances to more complicated iteratively determined ones. The results presented confirmed that some of the most popular choices are seriously flawed, while other newer ones offer distinct advantages although none are perfect for all occasions. Indeed, a prudent strategy in practice is to experiment with different choices of weights, just as one should also try different clustering algorithms.

The scaling challenge could be tackled from a multivariate perspective, as in the $\mathbf{W}^*_{(m_1)}$ method, or a univariate one, which yields a set of weights to be applied individually to the different variables, as in *autoscaling* or the $\mathbf{W}^*_{d(m_1)}$ method. One of our objectives was to demonstrate that the popular *autoscaling* approach is counterproductive and well-known alternatives, range and interquartile range scaling, have their own major limitations. Range scaling breaks down in the presence of extreme outliers (as pointed out by Milligan and Cooper, 1988). Interquartile range scaling, which has been suggested as a way of mitigating this flaw, performed even worse in our experiments.

Among the univariate methods tested that aim to equalize the influence of individual variables (chapter 2), $\mathbf{M}_6$ stood out for its superior performance. While iterative in nature and requiring the user to choose the number of pairs of closest points to work with, this multivariate approach to finding the best univariate scale factors is

simple conceptually and more effective overall than the alternatives studied. There is little doubt, however, that they are worthy replacements for the tarnished favorite of *autoscaling*. All the univariate highlighter methods (chapter 2), $\mathbf{M}_7 - \mathbf{M}_{11}$, produced impressive results. However, none of the eleven methods helped in the extreme case of the *Crabs* data, wherein the clusters overlap considerably. All of these schemes are in the spirit of giving larger weight to variables that exhibit greater between relative to within variability. It would be premature to project a winner among the several alternatives proposed.

The multivariate (equalizer) $\mathbf{W}^*_{(m_1)}$ algorithm was systematically studied in both the "null" clusters scenario as well as in the presence of clusters (chapters 3 and 4). The results demonstrated the starting point invariance characteristic of the algorithm irrespective of the presence of cluster structure, thereby providing evidence of both simplicity and stability of the algorithm. Additionally, when the clusters are reasonably separated in in their locations, the results also showed that the algorithm worked commendably in terms of providing a reliable measure of the within-clusters covariance structure of the data (apart from a single multiplicative constant, which does not impact CA). When the clusters overlap noticeably, the misclassification errors increased as one might expect. The algorithm converged in 7 – 31 iterations across all the data sets studied. Chapter 5 then explored an approach to perform highlighting in the multivariate context. As an unsupervised analogue of classical DA, the pseudo-DA approach to multivariate highlighting displayed promising results. In particular, when used in conjunction with gaussian mixture model based clustering (MCLUST), the pseudo-DA approach led to a dramatic improvement in results. Overall, the inclusion of the multivariate between-clusters measure of dispersion led to superior cluster recovery.

The work in chapters 2, 3, 4 and 5 suggest some natural extensions.

- For the univariate equalizers and highlighters (chapter 2) the issue of missing constants needed to make the within-clusters and between-clusters measures of dispersion more nearly unbiased, was discussed in section 2.5. Furthermore, it was noted that the constants would tend to differ between structure variables and

noise variables, as well as vary with the number of clusters. This is a hard-to-resolve problem. However, it would be interesting to study this further to find statistical approaches to estimate these unknown constants.

- The multivariate $\mathbf{W}^*_{(m_1)}$ algorithm studied in chapters 3 and 4, is an appealing approach to obtain an approximation to $\mathbf{W}$ that can be used to perform a multivariate equalization that reflects the within-cluster variation in the data. However, for "large n" (and "large p") data sets, due to its iterative nature, the algorithm could put a high demand on computational resources. One way to alleviate this problem is via random sampling. The key idea is to apply the $\mathbf{W}^*_{(m_1)}$ algorithm to random samples drawn from the data set rather than the entire data set. Consequently, significant improvements in execution time could be realized. Efficient algorithms for drawing a sample randomly from data in a file in one-pass and using constant space are discussed in [90]. Subsequently, one way to *merge* the $\mathbf{W}^*_{(m_1)}$ matrices from the random samples, would be to average them. This would however, still be intuitively suboptimal to using the entire data set at once, but it does provide a solution when dealing with large data sizes.

- As discussed in chapter 5, the application of the pseudo-DA approach to multivariate highlighting was studied using a few different choices for m_2, the number of farthest-apart point pairs. In all cases, m_1 was fixed based on the $\frac{2}{3}^{rd}$ rule, while only m_2 was varied. However, it might be interesting to study ways of picking m_1 and m_2 simultaneously, to potentially increase the sensitivity of the highlighters to the latent cluster structure. This would be useful in the context of both univariate as well as multivariate highlighting.

This dissertation has explored various strategies to tackle the ticklish problem of how to scale or weight variables effectively for cluster analysis. The goal has been to suggest intuitive alternatives that would provide significantly improved performance relative to current practice. While the tools developed and studied here have proved promising and have already demonstrated their value, they could definitely be refined more with further research.

References

[1] Anderberg, M.R.,1971. Cluster Analysis for Applications. Academic Press, New York.

[2] Anderson, E., 1935. The irises of the Gaspe Peninsula. Bull.Amer. Iris Soc., 59, 2-5.

[3] Anderson, T. W., 1958. An Introduction to Multivariate Statistical Analysis, New York: Wiley. (3rd ed., 2003)

[4] Art, D., Gnanadesikan, R., Kettenring, J.R., 1982. Data-based metrics for cluster analysis. Utilitas Mathematica, 21A, 75-99.

[5] Banfield, J. and Raftery, A., 1993. Model-based Gaussian and Non-Gaussian Clustering. Biometrics, 49, 803-821.

[6] C. Blake and C. Merz., 1998. UCI repository of machine learning databases.

[7] Cattell, R.B., 1966. The screen test for the number of factors. Multivariate Behavioral Research., 1, 140-161.

[8] Campbell, N.A., Mahon, R.J., 1974. A multivariate study of variation in two species of rock crab of the genus Leptograpsus. Australian J. Zoology, 22, 417-425.

[9] Chang, W.C., 1983. On using principal components before separating a mixture of two multivariate normal distributions. App.Statist., 32, 267-275.

[10] Chernoff, H. 1970. Metric Considerations in Cluster Analysis. Technical Report, Dept. of Statistics, Stanford University, CA.

[11] Cohen, A., Gnanadesikan, R., Kettenring, J.R., Landwehr, J.M., 1977. Methodological developments in some applications of clustering. Applications of Statistics, edited by P.R. Krishnaiah, Amsterdam: North-Holland, 141-162.

[12] Cormack, R.A., 1971. A review of classification. J.R. Statist. Soc. A 134, 321-367.

[13] Devlin, S.J., Gnanadesikan, R., Kettenring, J.R., 1975. Robust estimation and outlier detection with correlation coefficients. Biometrika, 62, 531-545.

[14] Dillon, W.R., Goldstein, M., 1984. Multivariate Analysis. John Wiley & Sons, Inc.

[15] Dillon, W.R., Mulani, N., Frederick, D.G., 1989. On the Use of Component Scores in the Presence of Group Structure. J. Consumer Research, 16, 106-112.

[16] Duda, R.O., Hart, P.E, Stork, D.G, 2000. Pattern Classification. Second Ed., Wiley, New York.

[17] Edwards, A.W.F., Cavalli-Sforza, L.L., 1965. A method for cluster analysis. Biometrics 21, 362-375.

[18] Everitt, B. S., Landau, S., and Leese, M., 2001. Cluster Analysis, 4th ed., New York: Arnold Publishers.

[19] B. S. Everitt and D. J. Hand., 1981. Finite Mixture Distributions. Chapman and Hall, London.

[20] Farnstrom, F., Lewis, J., Elkan, C., 2000. Scalability of Clustering Algorithms Revisited. SIGKDD Explorations, July, 2000.

[21] U. M. Fayyad, C. Reina, and P. S. Bradley., 1998. Initialization of iterative refinement clustering algorithms. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 194-198. AAAI Press.

[22] Fowlkes, E.B., Mallows, C.L., 1983. A method for Comparing Two Hierarchical Clusterings. J.Amer.Statist.Assn., Volume 78, Number 383.

[23] Fraley, C., 1998. Algorithms for Model-based Gaussian Hierarchical Clustering. SIAM Journal of Scientific Computing, 20, 270-281.

[24] Fraley, C., Raftery, A.E., 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer J., 41, 578-588.

[25] Fraley, C., Raftery, A.E., 1999. MCLUST: Software for model-based cluster analysis. J. Classification, 16, 297-306.

[26] Fraley, C., Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. J.Amer.Statist.Assn., 97, 611-631.

[27] Fraley, C., Raftery, A.E., Wehrens, R., 2005. Incremental model-based clustering for large datasets with small clusters. J. of Computational and Graphical Statistics, 14, 1-18.

[28] Fraley, C., Burns, P.J., 1995. Large-Scale Estimation of Variance and Covariance Components. SIAM J. on Scientific Computing, 16, 192-209.

[29] Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. J. Amer. Statist. Assoc. 62, 1159-1178.

[30] Ganti, V., Gehrke, J., Ramakrishnan, R., 1999. Mining very large databases. Computer, 32(8):38-45.

[31] Strang, G., 1988. Linear Algebra and its Applications. Third Ed., Brooks Cole.

[32] Gnanadesikan, R., 1970. S.N. Roy's interests in and contributions to the analysis and design of certain quantitative multiresponse experiments. In: Bose, R.C., Chakravarti, I.M., Mahalanobis, P.C., Rao, C.R., Smith, K.J. (Eds.), Essays in Probability and Statistics, University of North Carolina Press, Chapel Hill, 293-310.

[33] Gnanadesikan, R., 1997. Methods for Statistical Data Analysis of Multivariate Observations. Second Ed., Wiley, New York.

[34] Gnanadesikan, R., Harvey, J.W., Kettenring, J.R., 1993. Mahalanobis metrics for cluster analysis. Sankhya, A55, 494-505.

[35] Gnanadesikan, R., Kettenring, J.R., 1972. Robust estimates, residuals and outlier detection with multiresponse data. Biometrics 28, 81-124.

[36] Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2006. Strategies for Scaling and Weighting Variables in Cluster Analysis. Classification Society of North America – Annual Meeting, DIMACS, NJ.

[37] Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2006. Strategies for Scaling and Weighting Variables in Cluster Analysis. Joint Statistical Meetings, Seattle, WA.

[38] Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2007. Better Alternatives to Current Methods of Scaling and Weighting Data for Cluster Analysis. Journal of Statistical Planning and Inference, S.N. Roy Volume, 137, 3483–3496.

[39] Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2007. Equalizing or Highlighting Variables for Cluster Analysis. In proceedings of 56th Session of International Statistical Institute, Lisboa, Portugal.

[40] Gnanadesikan, R., Kettenring, J.R., Tsao, S.L., 1995. Weighting and selection of variables for cluster analysis. J. Classification, 12, 113-136.

[41] Green, P.E., Carmone, F.J., Kim, J., 1990. A Preliminary Study of Optimal Variable Weighting in k-Means Clustering. J. of Classification, 7, 271-285.

[42] Gordon, A.D., 1990. Constructing Dissimilarity Measures. J. of Classification, 7, 257-269.

[43] S. Guha, R. Rastogi, and K. Shim., 1998. Cure: An efficient clustering algorithm for large databases. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 73-84, New York.

[44] J. Han and M. Kamber., 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann

[45] Hand, D., Mannila, H., and Smyth, P., 2001. Principles of Data Mining, Cambridge, MA: The MIT Press.

[46] Hartigan, J.A., 1975. Clustering Algorithms. Wiley, New York.

[47] Hastie, T., Tibshirani, R., and Friedman, J., 2001. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, New York: Springer-Verlag.

[48] L. Hubert and P. Arabie., 1985. Comparing partitions. Journal of Classification, 2(1):193 218.

[49] A. K. Jain, M. N. Murty, and P. J. Flynn., 1999. Data clustering: a review. ACM Computing Surveys, 31(3):264-323.

[50] Jain, A.K., Dubes, R.C., 1988. Algorithms for clustering data. Prentice Hall, Englewood Cliffs, New Jersey.

[51] Johnson, R. A. and Wichern, D. W., 1982. Applied Multivariate Statistical Analysis, Englewood Cliffs, NJ: Prentice-Hall. (5th ed., 2002)

[52] Jolion, J.M., Meer, P., Bataouche, S., 1991. Robust clustering with applications in computer vision. IEEE Trans. Pattern Anal. Machine Intell. 13, 791-802.

[53] R. Kannan, S. Vempala, and A. Vetta., 2000. On clusterings: Good, bad and spectral. In 41st Annual Symposium on Foundations of Computer Science, FOCS, pages 367377.

[54] Kaufman, L., Rousseeuw, P.J., 1990. Finding Groups in Data. Wiley, New York.

[55] Teuvo Kohonen., 1990. The self-organizing map. Proc. IEEE, 78(9):1464-80.

[56] A. Lika, N. Vlassis, and J. J. Verbeek., 2003. The global k-means clustering algorithm. Pattern Recognition, 36(2):451461.

[57] Kettenring, J.R., 2005. The practice of cluster analysis. Special invited article. J. Classification, 23, 3-30.

[58] Liu, T., Moore, A., Gray, A., Yang, K., 2004. An Investigation of Practical Approximate Nearest Neighbor Algorithms. NIPS.

[59] J. MacQueen., 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkley Symposium on Mathematical Sciences and Probability, pages 281297.

[60] Maronna, R., Jacovkis, P.M., 1974. Multivariate Clustering Procedures with Variable Metrics. Biometrics, 30, 499-505.

[61] Martinez, W.L., Martinez, A.R., 2002. Computational Statistics Handbook with Matlab. Boca Raton, FL: Chapman & Hall/CRC.

[62] Meila, M., Heckerman, D., 1998. An experimental comparison of several clustering and initialization methods. Technical Report MSR-TR-98-06, Microsoft Research, Redmond, WA.

[63] Natick, M.A., 1999. MATLAB [Computer Software]. The Mathworks, Inc.

[64] G. W. Milligan and M. C. Cooper., 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika, 58(2):159179.

[65] Milligan, G.W., Cooper, M.C., 1988. A study of standardization of variables in cluster analysis. J. Classification, 5, 181-204.

[66] Tom M. Mitchell., 1997. Machine Learning. McGraw-Hill.

[67] D. Pelleg and A. Moore., 1999. Accelerating exact k-means algorithms with geometric reasoning. In Knowledge Discovery and Data Mining, pages 277281.

[68] Moore, A., 1991. A tutorial on kd-trees. University of Cambridge Computer Laboratory Technical Report No. 209.

[69] Murtagh, F., 2002. Clustering in Massive Data Sets. In Handbook of Massive Data Sets, eds. J. Abello, P. M. Pardalos, and M. G. Resende, New York: Kluwer, pp. 501-543.

[70] Pelleg, D., Moore, A., 2000. X-means: Extending k-means with Efficient Estimation of the Number of Clusters. In Proceedings of the Seventeenth International Conference on Machine Learning, 727-734, Morgan Kaufmann.

[71] J. Pen, J. Lozano, and P. Larranaga., 1999. An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters, 20:10271040.

[72] W. M. Rand., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66(336):846850.

[73] Rajeev Rastogi and Kyuseok Shim., 1999. Scalable algorithms for mining large databases. In Jiawei Han, editor, KDD-99 Tutorial Notes. ACM.

[74] Rao, C. R., 1965. Linear Statistical Inference and Its Applications, New York: Wiley. (2nd ed., 1973)

[75] S. Ray and R. H. Turi., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In Proceedings of the ICAPRDT99, pages 137143.

[76] Raychaudhuri, S., Stuart, J.M., Altman, R.B., 2000. Principal Components Analysis to summarize microarray experiments: application to sporulation time series. In Pacific Symposium on Biocomputing, 452-483.

[77] Rocke, D. and Dai, J., 2003. Sampling and Subsampling for Cluster Analysis I Data Mining: With Applications to Sky Survey Data. Data Mining and Knowledge Discovery, 7, 215-232.

[78] Roy, S.N., Gnanadesikan, R., Srivastava, J.N., 1971. Analysis and design of certain quantitative multiresponse experiments. Pergamon Press, Oxford.

[79] Seber, G.A.F., 1984. Multivariate Observations. John Wiley & Sons, Inc.

[80] S. Z. Selim and M. Ismail., 1984. K-means-type algorithms: A generalyzed convergence theorem and characterization of local optimality. IEEE Transactions on pattern analysis and machine intelligence, 6(1):8187.

[81] Shannon, W., Culverhouse, R., Duncan, J., 2003. Analyzing Microarray data using Cluster Analysis. Pharmacogenomics, 4(1), 41-51.

[82] Padhraic Smyth., 1996. Clustering using Monte Carlo cross-validation. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, pages 126-133. AAAI Press.

[83] Sneath, P. H. A. and Sokal, R.R., 1973. Numerical Taxonomy, San Francisco: Freeman

[84] Steinley, D., 2003. Local Optima in K-Means Clustering: What You Don't Know May Hurt You. Psychological Methods, 8, 294-304.

[85] Stoddard, A.M., 1979. Standardization of Measures Prior to Cluster Analysis. Biometrics, 35, 765-773.

[86] Strang, G., 1988. Linear algebra and its applications. Third Ed., Brooks/Cole.

[87] Strehl, A., 2002. Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. Ph.D. Thesis. The University of Texas at Austin.

[88] T. Su and J. G. Dy., 2004. A deterministic method for initializing k-means clustering. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI04).

[89] Vesanto, J., 2001. Importance of Individual Variables in the k-Means Algorithm. In Proceedings of the Pacific-Asia Conference in Knowledge Discovery and Data Mining.

[90] Vitter, J., 1985. Random sampling with a reservoir. ACM transactions on mathematical software. 11(1):37-57.

[91] Ward, J.H., 1963. Hierarchical Grouping to Optimize an Objective Function. J. American Statistical Soc, 58, 236-244.

[92] Xu, R. and Wunsch, D., 2005. Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16, 645-678.

[93] Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L., 2001. Model-based Clustering and Data Transformations for Gene Expression Data. Bioinformatics, 17, 977-987.

[94] Yeung, K. Y. and Ruzzo, W. L., 2001. Principal Component Analysis for Clustering Gene Expression Data. Bioinformatics, 17, 763-774.

[95] Young, I.T., 1969. Automated Leukocyte Recognition. In Proceedings of 8th ICMBE, Chicago.

[96] Young, I.T., 1970. Automated Leukocyte Recognition. In Automated Cell Identification and Sorting. Academic Press, 187-194.

[97] T. Zhang, R. Ramakrishnan, and M. Livny., 1997. BIRCH: A new data clustering algorithm and its applications. Data Mining and Knowledge Discovery, 1(2):141-182.

Vita

Srinivas P. Maloor

Education

2007	Ph.D. Electrical and Computer Engineering, Rutgers University
2004	M.S. Applied and Mathematical Statistics, Rutgers University
2002	M.S. Electrical and Computer Engineering, Rutgers University
1998	B.E. Telecommunications Engineering, Bangalore University

Conferences and Publications

- Wartenberg, D., Kipen, H., Hallman, W., Harris, G., Maloor, S., 2002. Quantitative methods for studying medically unexplained symptoms. Epidemiology, 13:4 (Supplement): S129.
- Wartenberg, D., Kipen, H., Hallman, W., Fiedler, N., Maloor, S., Brewer, N., 2003. A new name for previously characterized complaints? Epidemiology, 14(5):S85.
- Wartenberg, D., Kipen, H., Hallman, W., Fiedler, N., Maloor, S., Brewer, N., 2003. Is Gulf War Illness just a new name for old complaints? In 15th ISEE conference, Perth, Australia.
- Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2006. Strategies for Scaling and Weighting Variables in Cluster Analysis. Classification Society of North America – Annual Meeting, DIMACS, NJ.
- Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2006. Strategies for Scaling and Weighting Variables in Cluster Analysis. Joint Statistical Meetings, Seattle, WA.
- Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2007. Better Alternatives to Current Methods of Scaling and Weighting Data for Cluster Analysis. Journal of Statistical Planning and Inference, S.N. Roy Volume, 137, 3483–3496.

- Gnanadesikan, R., Kettenring, J.R., Maloor, S., 2007. Equalizing or Highlighting Variables for Cluster Analysis. In proceedings of 56th Session of International Statistical Institute, Lisboa, Portugal.