VARIABLE-WEIGHTED ULTRAMETRIC OPTIMIZATION FOR MIXED-TYPE DATA: CONTINUOUS, ORDINAL, NOMINAL, BINARY SYMMETRIC AND BINARY ASYMMETRIC

by

Eric C. Sayre
M.Sc., Simon Fraser University, 2005

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

In the
Department of Statistics and Actuarial Science

SIMON FRASER UNIVERSITY

Summer 2009

APPROVAL

Name: **Eric C. Sayre**

Degree: **Doctor of Philosophy**

Title of Thesis: **Variable-Weighted Ultrametric Optimization for Mixed-Type Data: Continuous, Ordinal, Nominal, Binary Symmetric and Binary Asymmetric**

Examining Committee:

Chair: **Jiguo Cao**
Assistant Professor

_____

Richard Lockhart
Senior Supervisor, Professor and Chair

_____

K. Laurence Weldon
Supervisor, Associate Professor

_____

Charmaine Dean
Supervisor, Professor

_____

Jacek A. Kopec
Supervisor, Associate Professor
University of British Columbia

_____

Derek Bingham
Internal Examiner, Associate Professor

_____

Rollin Brant
External Examiner, Professor
University of British Columbia

Date Defended/Approved: _____

ABSTRACT

Scientific research begins with hypothesis generation, for which cluster analysis (CA) can be used. Traditionally, CA involves continuous variables weighted equally, and the subjective choice of linkage and stopping rules. Variable weighting for cluster analysis (VWCA), beginning with De Soete (1985/6), produces weights that may be useful for hypothesis generation. De Soete's VWCA optimized ultrametricity, a property of better separated clusters, without requiring CA.

We developed variable-weighted ultrametric optimization for mixed-type data (VWUO-MD), starting with a variable-weighted, multivariate distance for data with any number of continuous, ordinal, nominal, binary symmetric and binary asymmetric (e.g., rare disease) variables. In Monte Carlo simulations we found that weights are consistent with *a priori* relationships between variables, under several distributions. On some relationships (e.g., single group linear), the method performs poorly. Compared to De Soete, VWUO-MD better penalizes for 0-weights, and better ensures a unique solution with a strategic random restart procedure. The bootstrap covariance matrix is slightly conservative. For mixtures of at least four continuous/nominal variables, a U-statistic-based covariance matrix performs well. Point estimates and covariances are invariant to column/category/record order and affine transformations.

We analyzed of a subset of the Joint Canada/United States Survey of Health: working, mature students 50+ years old who received health services in the past year (*n*=167), split into training and testing segments. Prescreening within types and backwards elimination with VWUO-MD reduced the space. The final 14 variable weights were plotted as a scree plot. On the testing segment, a model was fit from the upper scree plot variables. Similar models were fit from the lower scree plot, prescreening and backwards elimination reject variables. Models were ordered on overall statistical significance and the upper model had the best fit, indicating that VWUO-MD had successfully mined these data for hypotheses. We learned that reduction in activities due to a long term health condition was associated with consultations with a mental health professional in the past year (odds ratio=12.25, 95% CI=1.67, 90.02).

While needing additional research, in its present form VWUO-MD produces variable weights that may be informative for hypothesis generation on data with varied mixtures of data types.

Keywords:
Hypothesis generation, ultrametric optimization, data mining, cluster analysis

ACKNOWLEDGEMENTS

I wish to thank Larry Weldon, who supported my study of statistics since my second undergraduate statistics course. I could not have achieved a PhD (or even a BSc) in statistics without his support. On this thesis specifically, Larry's data analysis course gave me skills in the graphical exploration of data that were invaluable to exploring the complex surfaces encountered in this research.

I wish to thank Richard Lockhart, my senior supervisor after Larry retired. Richard’s theoretical expertise in statistics was a critical resource. His suggestions on how to improve the manuscript were also extremely important.

I wish to thank Charmaine Dean, whose excellent graduate courses provided me with the strong statistical background that was critical both to this work, and to the set of skills necessary for any statistician.

I wish to thank Gavin Steininger and John Bentley, fellow students of statistics, for their suggestions on graphing techniques as well as potential applications of VWUO-MD. These were very helpful in this thesis.

Finally, I wish to thank my supervisor of nine years at the Arthritis Research Centre of Canada (ARC), Jacek Kopec. Jacek provided ample opportunity for me to pursue higher degrees, write abstracts and papers, and participate in a stream of challenging projects. The rich research environment I was immersed in at ARC was a big contributing factor to this degree.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Scientific research begins with the formulation of hypotheses. Data are collected and analyzed in order to test those hypotheses. Data mining, defined as the process of discovering patterns in data, can aid in generating hypotheses for testing.[1] Philosophically, it is not a very big step to perform data mining compared to the traditional approach of coming up with hypotheses on one's own. For consider what it means to come up with a hypothesis "on one's own". The research scientist must draw upon his or her personal and professional knowledge, or *internal data set*. This data set lives in the scientist's brain and is the result of years of observation and study of *other data sets*, but fundamentally, it is data. What makes the traditional approach to scientific research statistically sound is that the data on which hypotheses are tested are not of the internal data set from which the ideas arose. The same principle can and should be applied to data mining; it is widely held that one should not test hypotheses with the same data that were used to formulate them.[2,3,4] Doing so might be termed "data dredging", rather than data mining, and p-values from such analyses would not be valid estimates of Type I error probability. However, as long as researchers abide by this basic tenet, data mining can be a powerful resource for accelerating the growth of knowledge. Indeed it has already been widely used for this purpose, in such diverse areas as epidemiology, genomics, biomedical research, credit card fraud detection, and many more.[5,6,7,8]

Cluster analysis (CA) is a common data mining methodology, and is commonly used for the purpose of hypothesis generation.[2,3,4,9,10,11] For example, Gilman et al (1995)[2] used space-time cluster analysis to generate hypotheses about childhood cancers in Britain, then tested those hypotheses on an independent segment of their data. Bredel et al (2004)[9] in an article on genomics-based hypothesis generation stated that "... the most common approach to organisation of [DNA] microarray data is hierarchal clustering." Stegmann et al (2003)[11] performed co-word clustering of existing scientific literature to generate new hypotheses, as well as confirm a known relationship between Reynaud's disease, fish oil, migraines and magnesium.

In CA, objects are grouped into homogeneous "clusters" with the goal of maximizing similarity between objects within the same cluster while minimizing similarity between objects in different clusters. "Similarity" (or dissimilarity) can be defined with a symmetric, *n* by *n*, one-way proximity matrix. In the cluster analysis of two-way (objects by variables) data, a one-way proximity matrix can be created on which to perform CA. In a representative sample (in which one did not go out of one's way to collect data that appeared to be clustered), clusters are generally motivated by the relationship between variables in the population. By "representative", we mean that every subject represents a known proportion of the target population, and has a sample weight that reflects that (usually based on the inverse probability of selection) except in the case of a simple random sample where no sample weight is required as such a weight would be identically 1. If variables X and Y are normally distributed and independent of each other, a

two-dimensional plot of X versus Y will show one mass (bivariate normal). If X and Y are positively correlated, that mass will be diagonally oriented. If *g* is a latent group variable affecting X and Y by adding constants to their means depending on group, then (whether or not *g* is measured) the 2 by 2 plot of X versus Y will contain multiple clusters diagonally distributed. A CA that revealed these clusters might suggest testing some hypothesis involving X and Y. (Note that in such a case, X and Y would not be marginally independent, and there would be a hypothesis to be generated). If X and Y were the only two variables available, this would be a rather roundabout way of doing things; why not just develop a hypothesis involving X and Y to begin with? However, if there were 20 variables in the data set and the only clear clusters were defined according to X and Y, then CA would have suggested the most promising hypothesis for testing. Perhaps two other variables, U and V, are also well separated by the optimal cluster solution. Then the researcher could consider the definitions of X, Y, U and V, and formulate an appropriate hypothesis based on the most sensible "dependent" variable from those, treating the others as independent variables (for example). For this the researcher would need to draw on his or her knowledge of subject matter. The point is that 16 other variables would have been eliminated from consideration, and the researcher would have generated a concise hypothesis using CA. Testing of the hypothesis would be done on a different data set, or an independent segment of data that was not used to generate it.

Hypothesis generation (HG) via traditional CA as discussed above is useful, but has some limitations. First, in the most traditional application of CA,

data consist only of continuous variables, and similarity is measured by Euclidean distance. This definition precludes the analysis of other types of variables, such as binary, ordinal or nominal. A second limitation of HG via traditional CA is that cluster analysis is traditionally performed on unweighted variables, that is, all variables are treated equally in the CA procedure. Clusters that are well defined only on a small subset of the variables may not be easily recovered with the additional “noise” variables, and therefore promising hypotheses may not be easily generated. Third, CA solutions are non-unique in the sense that there are many possible solutions that an analyst could arrive at from the same data set, depending in part on linkage method (single, complete, centroid, etc.) and the number of clusters determined to be the optimal solution. Fourth, only the most obvious candidate variables might be made visible by a cursory examination of the clusters (for example with k-way plots). More sophisticated HG from a CA solution generally requires additional statistical analysis (e.g., ANOVA, or the calculation of some index of observed versus expected proportions in small intervals) in order to determine what variables are associated most strongly with the clusters.

The first limitation of HG via CA has been addressed to a certain extent. Formulas for distance between objects measured on sets of binary variables have been suggested. For example, Johnson and Wichern (2002)[12] suggest a variety of formulas, generally in the form of number of matches divided by number of mismatches. Differences between the formulas are driven for example by a variety of treatments for 0-0 matches versus 1-1 matches. Distance

formulas for binary and other variable types (ordinal and nominal) have been suggested for example by Dr. Stephen Kwek of the Human Genome Laboratory in the Department of Computer Science at the University of Texas at San Antonio.[13] His distance measures for binary variables are consistent with those of Johnson and Wichern. While these formulas can be useful, a remaining limitation is that data sets may contain a variety of variable types; Johnson and Wichern suggest converting all variables into binary representations so that all the variables can be analyzed simultaneously using the distance measure proposed for binary variables. It would be preferable to retain the full information in the continuous, ordinal and nominal variables however. This is easily overcome. Kwek suggests an alternative weighted distance formula, but we will combine distances for different type-specific subspaces (subsets of variables corresponding to single variable types) using the square root of the sum of squared type-specific distances, similar in structure to the formula for Euclidean distance.

The latter three limitations of HG via CA can be overcome with *variable weighting for cluster analysis* (VWCA) techniques that do not rely on *a priori* knowledge of the clusters. The idea of VWCA predates 1970, however early efforts required CA to be performed either in advance or as part of the estimation of weights, for example, Hogeweg (1976),[14] Art et al (1982)[15] and DeSarbo et al (1984).[16] For the purposes of HG, we would prefer to retain the advantages of the variable weights without the disadvantages of performing CA per se (such as the subjectivity associated with choice of linkage and the number of clusters).

Among the earliest efforts in this direction were two seminal papers by Geert De Soete (1985, 1986).[17,18] In both papers, De Soete presented an order n^3 VWCA method upon which we shall build. (In the 1986 paper, he also added an order n^4 variant which we do not investigate due to prohibitively high computational requirements.) The variable weights in VWCA correspond to the variables' relative importance in the object groupings in the data. For example, suppose variables X and Y arise from conditional distributions that depend on a latent group indicator *g*, while variable Z comes from a single homogenous distribution that is nearly independent of *g*. Then CA ought to show stronger separation of X and Y values into clusters (different distributions in different clusters), but a similar distribution of Z within all clusters. VWCA ought to assign larger weights to variables X and Y (correctly, since they are related through *g*), and a smaller weight to Z. For the purpose of performing CA per se, VWCA has been shown to reduce the influence of noisy, superfluous variables and thereby *enhance* the groupings in the data. However, we have found that this is mainly evident only in small, artificial data sets. Regardless, for the purpose of HG, the variable weights are enough; De Soete's method (which we extend) does not require that one perform CA (even though the concept of CA is a motivation behind his approach). VWCA can overcome the second limitation of HG via CA discussed above directly; by down weighting the unimportant variables, those that remain with higher weights automatically become the focus. The third limitation is overcome if VWCA has a unique solution, which our extension of De Soete's approach generally does. The concept of a set of solutions depending on choice

of linkage or stopping rules no longer applies—we will not perform CA at all. Finally, the fourth limitation is also overcome with VWCA. There is no secondary statistical analysis required in order to obtain information about ranking variables' importance. This comes directly in the form of the variable weights. The variable weights lie on a continuum which allows the analyst to make fully informed decisions when formulating hypotheses, considering the distribution of the entire set of variables (not just the most obvious ones).

Often for the purpose of CA, how well the groupings have been enhanced by VWCA is measured (somewhat subjectively) with graphical devices such as dendrograms,[12] or more objectively with a numerical function representing for example the degree of *ultrametricity*, a desirable property naturally leading to better clustering as described above. We will take only a cursory look at how well groupings are enhanced by VWCA as measured by dendrograms. For the purpose of HG (our motivation), ultrametricity is an important concept that will drive our VWCA algorithm. In De Soete's method (and ours), an ultrametric loss function (measuring the degree of departure from the desirable property of ultrametricity) is minimized, to arrive at the variable weights solution.

De Soete's approach has been cited many times, often neutrally when describing or performing CA on an applied problem, describing VWCA, or developing alternative approaches to VWCA (usually involving preliminary or simultaneous CA to obtain variable weights).[19,20,24,26,27,28,29,30,31,32,33,34,36,37,38,39,41,42,43,47,48,49,50,51,53,55,5657,59,61,62,63,65,66] At times De Soete's method has been cited positively,[21,35,46,52,54,56,60,64] and at

other times negatively.[22,23,25,40,44,45,58] His work has been expanded on by many of his critics.

Among the positive citations, most of the articles citing De Soete's method (positively or negatively), also cite the study by Milligan (1989),[54] in which De Soete's method was tested and found to successfully recover the true clustering structure in the presence of "masking variables" (those unrelated to the object groupings, and which we will refer to as "noise" variables), in artificial data with a variety of dimensions. Milligan had presented his findings three years earlier at the 21st Numerical Taxonomy Conference.[64] Breckenridge (2000)[21] comments generally on the benefit of VWCA in reducing the masking effect of noise variables. Donoghue (1995)[35] found that De Soete's VWCA yielded significantly higher recovery of known cluster structures than unweighted CA. Jedidi et al (1991)[46] cite De Soete's purported success with first-order estimation (which we will refute to an extent) as a justification for their own use of that approach in their unique estimation of weights for three-way, objects by variables by discrete selection data. Milligan et al (1987)[52] cite De Soete's VWCA as particularly useful after first eliminating obvious non-candidate variables from the analysis. This is the approach we take when applying our method to real-world data, in *Chapter 6: An application of VWUO-MD*. Milligan et al (2003)[56] cited De Soete's VWCA as an effective means of reducing the effect of masking variables. Sokal (1986)[60] pointed out the advantage of De Soete's VWCA in its ability to differentially weight individual characteristics in phenetic taxonomy.

There were some critical reviews of both VWCA in general, and De Soete's method specifically. Carmone et al (1999)[25] and Huang et al (2005, 2007)[44,45] pointed to the order of the algorithm as a problem with De Soete's method. They developed a method of variable selection involving an iterative process of cluster analysis. Huang et al developed variable weights as part of performing k-means CA. Brusco (2001, 2004)[22,23] cited Gnanadesikan et al's (1995)[40] study which found that De Soete's approach performed poorly in its ability to assign objects back to known clusters. Gnanadesikan et al developed an iterative algorithm for estimating within- and between-groups (clusters) sums of squares for the construction of variable weights proportional to between-groups sums of squares and inversely proportional to the within-groups sums of squares. Their approach appears to be touted as a compromise between methods requiring *a priori* knowledge of the clusters, and those (like De Soete's) that do not, but it does involve the subjectivity of stopping rules. Brusco interpreted the aforementioned study as a criticism of VWCA in general, and took it as lending support for his variable selection approach to clustering binary data sets. Schweinberger et al (2003)[58] pointed out the problem of multiple local optima (minima), which we also notice and deal with quite effectively in a strategic random restart procedure.

There are many relatively neutral citations of De Soete's method, including Arabie et al (1992, 1995),[19,20] Leonard et al (2008),[49] Chun (1995),[26] and Corter (1996)[28] who briefly cite De Soete's VWCA approach during an overview of CA. Bull et al (1992)[24] weight their variables according to "genetic variability" over

"phenotypic variability" and cite De Soete's approach to VWCA only in a brief mention of VWCA. Debska et al (2003)[29] cite De Soete's article but perform unweighted CA in their study of the relationship between aromatic properties and molecular structure. Chung et al (2006)[27] analyzed De Soete's 1986 data set, applying three different variable weighted k-means clustering methods, but not De Soete's ultrametric optimization. Chung et al compared their own k-means approach to that of Makarenkov et al (2001)[50] who had extended De Soete's approach in their own k-means clustering algorithm. DeSarbo et al (1988)[30] developed an expectation-maximization algorithm using normal mixture distributions. DeSarbo et al (1989)[31] developed an approach combining piecewise multiple regression with CA. De Soete (1987)[32] developed a method of VWCA with topological constraints on ultrametric (and "additive") trees imposed, for example a constraint whereby all the pairs in a selected subset of objects compared only with objects outside the subset satisfy as well as possible ultrametricity (as opposed to unconstrained, which we consider, where all ordered triples are considered equally). De Soete released software for his 1986 algorithms in 1988.[33] Donoghue (1995)[34] studied a variety of CA methods (mainly varying linkage method) under a variety of within-cluster covariances. While citing De Soete, Donoghue did not report on the performance of VWCA per se. He reported that covariance in the same direction as the separation of clusters helped facilitate cluster recovery—we will actually find mixed results on that question in the investigation of our method. In a section on variable weighting in the 2001 textbook "Cluster Analysis" (fourth edition), Everitt et al[36] cited De

Soete's 1986 paper, as well as the papers described above by Milligan and Gnanadesikan (among others) that investigated the approach. Fovell et al (1993)[37] cited De Soete as a primary example of VWCA, however they were more interested in *lower* weighted variables as they represented less "redundancy". Fowlkes et al (1988)[38] developed a variable selection method involving simultaneous CA. Friedman et al (2004)[39] and Soffritti (2003)[59] developed an approach for identifying different variable subsets to be deemed important for different clusters, based on sequential joining of variables according to Rand indices for comparing partitions. Friedman and Soffritti's approaches, like most alternatives to De Soete's method, involved performing CA within the algorithm. Hand et al (2005)[43] cited Friedman's study of differential variable weighting for clusters, and pointed out the lack of support for this concept as a limitation in De Soete's approach. We acknowledge the possibility of competing cluster structures, or analogously, disjoint relationships between variables. In such cases our method will generally compromise (with some exceptions to be explored later), spreading the weight between those variables that participate in *some* relationship. While this is useful, as Hand suggested, it would also be informative to know which variables are involved in different relationships than others. This can actually be accomplished with De Soete's method (and ours) with multiple parallel analyses using different groups of input variables. Gordon (1990)[41] developed a method of subjective assignment of weights by an analyst studying the training portion of a data set. Green et al (1990)[42] modified the method of DeSarbo et al (1984)[16] to obtain weights while simultaneously

performing k-means CA. Jing et al (2007)[47] developed a k-means CA algorithm that also produced different weights for different clusters. Lapointe et al (1992)[48] developed a method of comparing "additive trees" (structures with the property optimized in De Soete's 1986 order n^4 algorithm upon which we do not build due to prohibitively high order). Meulman et al (1993)[51] cite De Soete's method as having aspects similar to their stratified principal components analysis studying the effect of subjective point of view in forming groups of people. Milligan et al (1988)[53] cite VWCA in passing, when looking at the related topic of variable standardization in CA. Milligan (1996)[55] cites De Soete's approach in a brief section on VWCA in a general chapter on CA. Morris et al (2006)[57] cite De Soete's VWCA papers only in passing, their focus is an applying the results of a CA to the task of identifying consumer "archetypes". Steinley et al (2005)[61] developed a parametric approach to randomly generating overlapping clusters for assessing the performance of CA methods, and cited De Soete when suggesting that possible future extensions of their method could include VWCA. Steinley (2006)[62] cited De Soete among others in a brief section on VWCA in his paper on k-means CA. Steinley et al (2008)[63] cited De Soete among others when mentioning VWCA before making a comparison of several (mainly parametric) variable selection methods for CA. Tsai et al (2008)[65] developed a method of variable weighting incorporated within an iterative k-means CA algorithm. Finally, van Buuren et al (1989)[66] developed an iterative k-means CA approach for mixed-type data that involved subjectivity on two levels, choice of both k and p, the latter being the dimensionality of the solution (≤ the number of variables).

Despite the broad coverage of this topic, it appears that little work has been done on methods of VWCA that do not involve actually performing some variant of CA, and the pitfalls that go along with that. Since "CA" are the last two letters in "VWCA", this might not be surprising. However, for the purposes of HG where at least the visual "enhancement" to the cluster solution is extremely weak (as we shall see is often true), the cluster solution is merely a nuisance parameter and an additional source of variation and subjectivity, and it is the variable weights that provide the most valuable information. For our purposes then, it seems natural to begin from De Soete's 1985/6 approach minimizing an ultrametric loss function, and extend and improve upon that as needed to accomplish our goal of HG for mixed-type data. The idea of utilizing VWCA for HG has not been explicitly considered within the publications we reviewed. However, HG is a natural byproduct of CA, and therefore the idea that HG will prove to be a useful application of the variable weights is promising. This is especially relevant considering that it has been suggested by some (e.g., Gnanadesikan) that De Soete's approach to VWCA provides only a lackluster benefit to known cluster recovery; the order of the variable weights may be more clearly informative, one can hope. In fact that is what we will find. The variable weights will be the focus in this thesis, and we intend to utilize them to generate hypotheses without having to perform cluster analysis at all.

To begin with, let us consider some of the shortcomings of De Soete’s method: a) data must consist only of continuous variables; b) the ultrametricity loss function is not sufficiently penalized for variable weights estimated at 0,

which can lead to degenerate solutions in which one variable is assigned a positive weight and all other variables weights of 0 (not useful for HG), and/or can lead to non-unique solutions involving broad regions exactly tied with the minimum possible loss function; c) only first-order derivatives are used, which generally requires more iterations, and depending on stopping rules can lead to estimates that are not close to a local minimum; and d) the method is order n^3, which is inherently slow and limits the practical application of this methodology.

In this thesis, we will address the first three of the aforementioned problems. We are stuck with the fourth. First, we will develop a variable-weighted, multivariate distance formula resembling Euclidean distance but designed for mixed-type data, data consisting of zero or more variables of types continuous (e.g., body mass index), ordinal (e.g., income quintile), nominal (e.g., ethnicity), binary symmetric (e.g., gender) and binary asymmetric (e.g., a rare disease). (Throughout this thesis, these types may be referred to respectively as types C, O, N, S and A.) Based on this weighted multivariate distance formula, we will develop an ultrametricity optimizer function—appropriately penalized for 0-weight solutions—which, when minimized uniquely, will describe (via the resultant variable weights) which variables participate most strongly in the object groupings in the data. We will utilize second-order derivatives to ensure faster, more accurate estimation of the variable weights. Specifically, we will obtain convergence in relatively few iterations, and ensure that we have located true local minima, as opposed to De Soete's method.

Our extension of the De Soete method is a variable-weighted ultrametric optimization for mixed-type data (VWUO-MD). Other improvements we will make will include support in VWUO-MD for object-weighted data, such as complex survey data in which each object (person) can represent several (not necessarily an integer number of) objects.[67] We will develop two covariance matrix estimators to describe the (co)variability of the weight estimates: we will first describe a central limit theorem (CLT) based estimator,[68] develop a U-statistic-based covariance estimator,[69,70,71] and develop a bootstrap estimator.[72] We will investigate how well VWUO-MD performs on an artificially constructed data set (are the estimates consistent with *a priori* latent clusters and the variables associated with them in the random generation of the data), as well as study the performance of the variance estimators in Monte Carlo simulations. Since we are focusing on the HG potential of VWUO-MD, we will also perform Monte Carlo simulations to study the performance of VWUO-MD on a variety of data sets involving clustered as well as non-clustered (single group) linear and quadratic relationships. In addition, we will perform a VWUO-MD analysis of real-world, sample- and bootstrap-weighted, mixed-type complex survey data: a selected subsample of the Joint Canada/United States Survey of Health (JCUSH). We will investigate sets of variables consisting of all five types (C, O, N, S and A), ranging across subject matter from socio-demographic to disease specific variables, to generate new hypotheses. We will test those hypotheses on a separate subsample of the JCUSH data. To accommodate all of these analyses, we will develop a complete software package to perform VWUO-MD analyses.

It is our hope that in this thesis we will successfully develop a new hypothesis generating methodology that is suitable for analyses of mixed-type, multivariate data from all fields of research, including (for example) medicine, basic biology, genetics, population health, psychology, physics, chemistry, finance, economics and more. Literally any field in which mixed-type data are collected might benefit from hypothesis generation. As the prevalence, power and storage capacity of modern computers continues to grow, so too do the quality and quantity of data that can be analyzed with this new approach to hypothesis generation.

CHAPTER 2: THE ESTIMATOR

2.1 Dendrograms

While we intend to focus on HG, De Soete's VWCA method was originally designed as an improvement to data for the purpose of CA. Therefore we will briefly review a simple tool for assessing hierarchical clustering strength in data: the dendrogram.[12,36] The dendrogram is a tree diagram in which distance between subsequently joined clusters is plotted against object labels. The roots lie at each object on the horizontal axis (0 line). Vertical lines extend upwards from each object to the distances at which the smallest clusters are fused. Horizontal lines from there join objects with their fellow cluster members. Vertical lines extend upwards from the center of each horizontal line to the distance at which those clusters are fused with the next closest clusters or members. This structure is repeated upwards until all objects are eventually fused into one mega cluster. The shape of the dendrogram can depend on a number of factors, most notably any natural grouping that occurs in the data. Evidence of clusters is found when vertical lines to the next vertices up are especially long in a common vertical span across all groups. This suggests a number of clusters equal to the number of vertical lines in that range. Other factors that can influence the graph include the cluster linkage method used to generate the tree. There are many linkage methods available in cluster analysis, some common ones being single linkage (nearest neighbor, where distance between clusters is the distance

between nearest neighbors); complete linkage (farthest neighbor); centroid linkage, and more. In this thesis, any time we explore the clustering strength of data with a dendrogram, we will use single linkage. This will not be our focus but an aside. For a comprehensive review of the other linkage and graphing methods available in cluster analysis, see Cluster Analysis by Everitt et al.[36]

To demonstrate the dendrogram, an example data set with three well-defined clusters on two variables is shown in Figure 1. Variables C1 and C2 were generated from a mixture distribution of multivariate normal (MVN) distributions with three distinct groups. The dendrogram (single linkage) fit to these data is shown in Figure 2. This graph clearly indicates the clustered structure of these data and suggests three groups. An example data set with no natural group structure on two variables is shown in Figure 3. Variables C1 and C2 were generated from independent random Uniform(0,1) distributions. The dendrogram (single linkage) fit to these data is shown in Figure 4. This graph does not suggest any grouping in these data.

Figure 1. Example data set with three well-defined clusters; variables C1 and C2 arise from a mixture distribution of MVN distributions

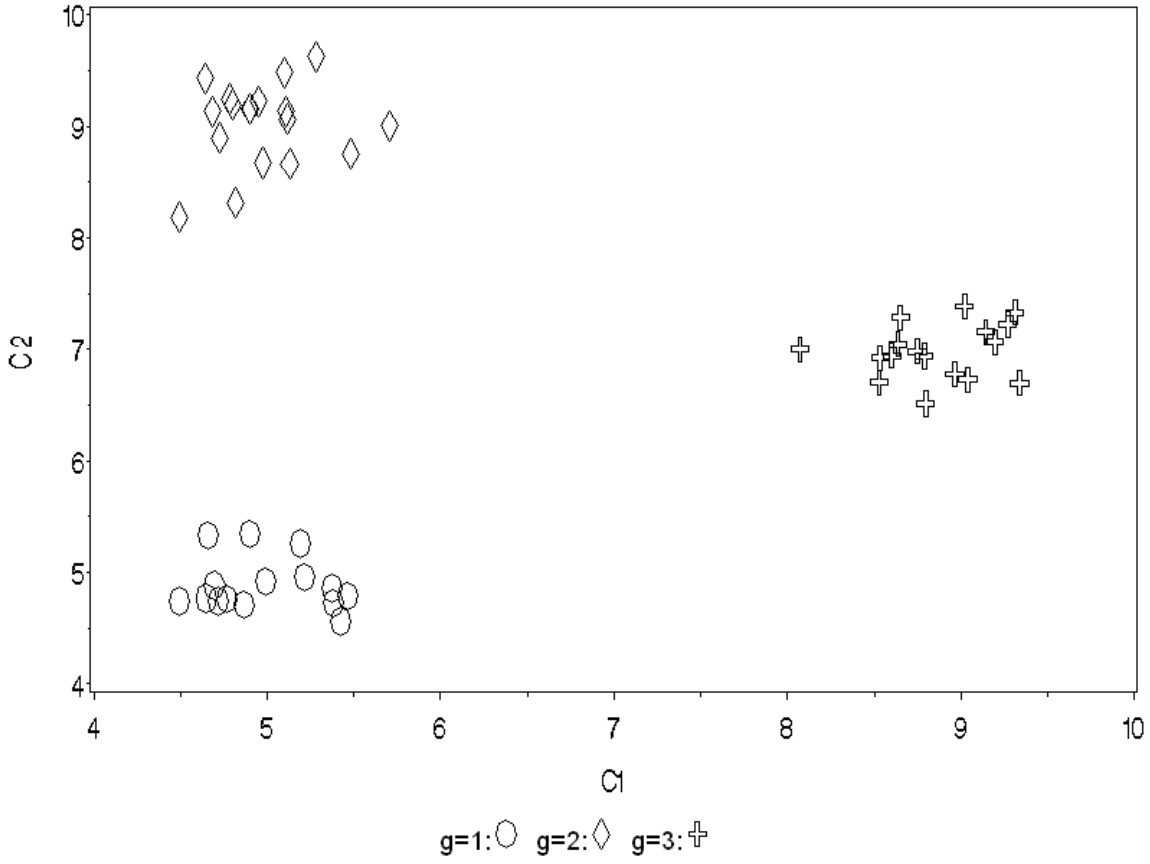


Figure 2. Dendrogram (single linkage) fit to example data set with three well-defined clusters

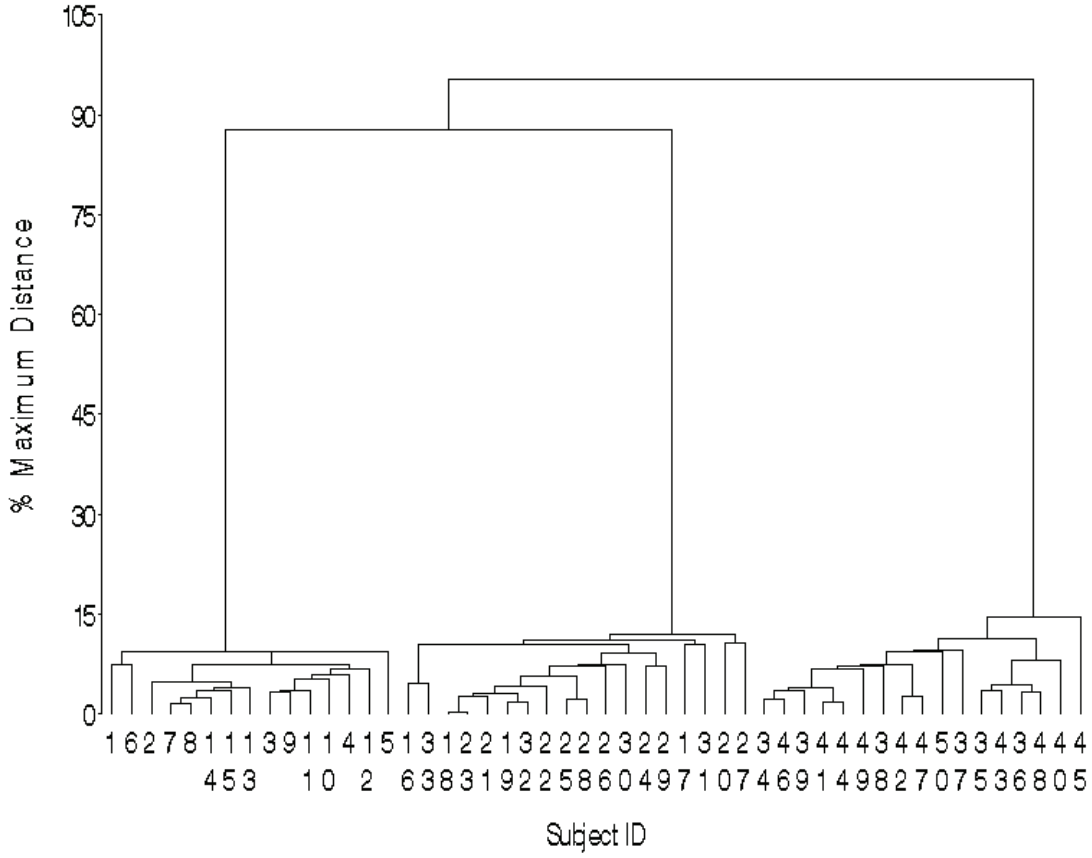


Figure 3. Example data set with no natural group structure; variables C1 and C2 are independent random Uniform(0,1)

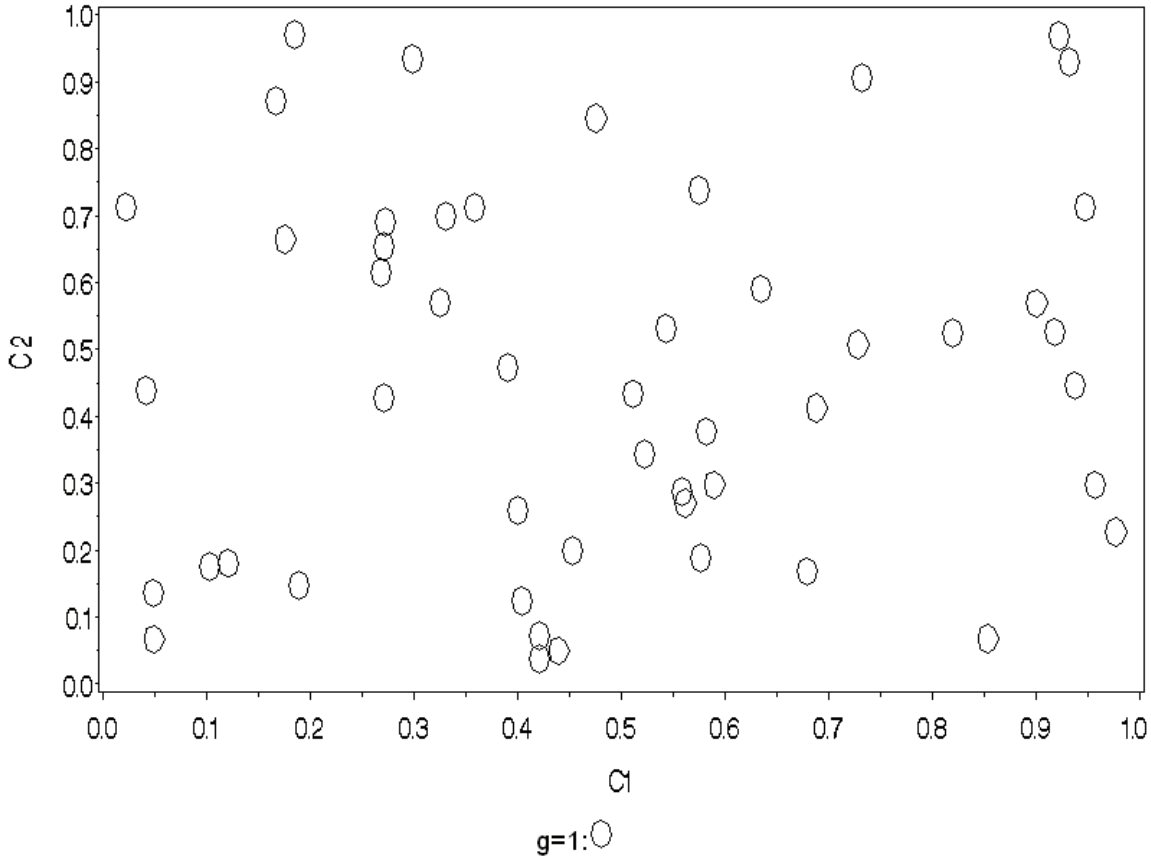
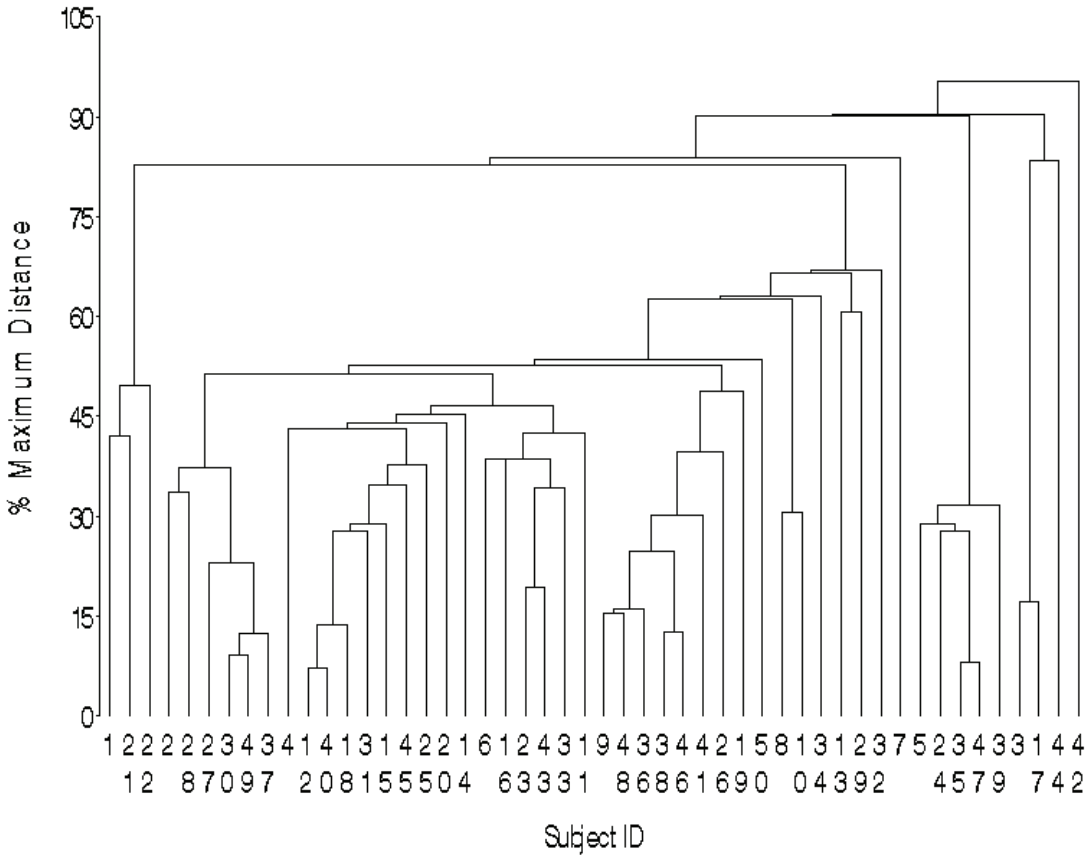


Figure 4. Dendrogram (single linkage) fit to example data set with no natural group structure

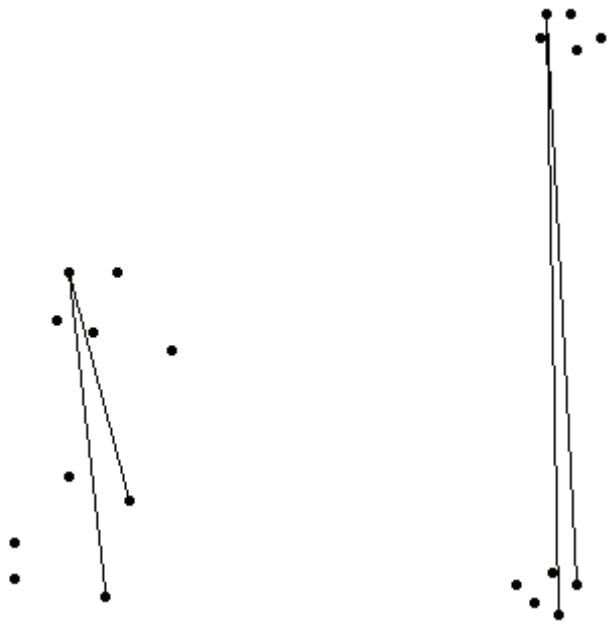


2.2 Ultrametricity

The *ultrametric property* according to Everitt et al[36] asserts that $d_{ij} \leq \max(d_{ik},d_{jk})$ for all *i*, *j* and *k*, where d_{ij} is the distance between objects *i* and *j*. As stated by Everitt et al, "An alternative way of describing this property is that for any three objects, the two largest distances between them are equal." When the ultrametric property is not satisfied, *inversions* can occur in the dendrogram, in which clusters joined at a later stage are fused at a distance that is closer than a fusion that occurred earlier. The principal aspect of the ultrametric property as far as VWUO-MD is concerned however is that greater ultrametricity generally corresponds to greater separation of clusters in the data, which, as described in

the introduction, may lead to a method for HG. To see intuitively how ultrametricity relates to separation of clusters, consider the two data sets depicted in Figure 5. The data set on the right exhibits a higher degree of ultrametricity; that is, triples of objects between clusters have a higher degree of (relative) equality between the two longest distances. This data set also exhibits a clearer separation between clusters.

Figure 5. Ultrametricity and cluster separation; closer relative equality between the two longest distances of a triple of objects implies better cluster separation



However, we ought to describe this more coherently, and also consider how ultrametricity relates to situations with more than two clusters, or only one group of points scattered about some shape (e.g., a linear relationship in two dimensions). We will explore these questions practically later on, in *Chapter 5: Exploratory analyses of distributions for hypothesis generation*. For now, let us consider situations with two and three clusters. With two well separated clusters,

a given triple of points either has two points close together and one far away from those, or has all three close together. In either case, the difference is small between the two longest sides of the triangle joining those three points. With three or more clusters, relative placement of the clusters affects ultrametricity. For example, placing the clusters at the vertices of an isosceles triangle (two equal angles) with the third angle ≤60 degrees is consistent with ultrametricity, since in any given triple of points, either: i) at least two points belong to the same cluster in which case ultrametricity is well satisfied as just described; or ii) the three points all belong to different clusters but the difference between the longest two sides is small due to the shape of the triangular cluster placement. On the other hand, placing three clusters at the vertices of an isosceles triangle with the third angle >60 degrees, or at the vertices of a scalene triangle, is less consistent with ultrametricity, since for triples with points in all three clusters, the difference will be larger between the two longest sides of the triangle joining the points.

Ultrametricity lies at the heart of De Soete's (1985/6) method, and the VWUO-MD method. By finding variable weights that maximize the degree of ultrametricity in a data set, larger weights ought to correspond to dimensions involved in the greatest separation of data, and therefore be useful for generating hypotheses from the candidate variables.

2.3 Variable-weighted, multi-type, multivariate distance

Before we can optimize ultrametricity with variable weights, we require an *n* by *n* proximity matrix that depends in some way on those weights. In order to obtain this from our data, we need a variable-weighted, multi-type, multivariate

distance formula. Our strategy for constructing this distance will be: i) to develop a type-specific multivariate distance for each type-specific subspace (subset of variables corresponding to one variable type; there will be five type-specific multivariate distances); and ii) to combine these type-specific distances into one multi-type, multivariate distance.

For continuous variables, the type-specific (squared) distance between objects *i* and *j* are the (squared) weighted Euclidean distance:

$$d_{ij_C} = \sqrt{\sum_{l=1}^{p_C} \left(x_{il} - x_{jl}\right)^2 w_l / q_l^2}$$

$$d_{ij_C}^2 = \sum_{l=1}^{p_C} \left(x_{il} - x_{jl}\right)^2 w_l / q_l^2$$

where there are p_c continuous variables, and q_l is a normalizing constant for variable x_l to facilitate a scale-free comparison between variables (this is explained further in the next section). It is noteworthy that the variable weights w_l appear in the *squared* distance scale rather than the distance scale. This was done according to the method of De Soete. De Soete's variable weights were constrained to sum to 1; in VWUO-MD, variable weights are constrained to sum to the number of variables. This approach was taken to ease interpretation of weights; weights below or above 1 are below or above average in the set.

The other four type-specific distance formulas were suggested conceptually by Dr. Stephen Kwek of the Human Genome Laboratory in the Department of Computer Science at the University of Texas at San Antonio[13] and are consistent with Johnson and Wichern (2002)'s[12] treatment of binary

variables. We converted these formulas into differentiable algebraic expressions. Where applicable, both the conceptual and algebraic expressions are provided here.

For ordinal variables, the type-specific (squared) distance between objects *i* and *j* is simply:

$$d_{ij_O} = \sqrt{\sum_{l=1}^{p_O} \left(z_{il} - z_{jl}\right)^2 w_l / q_l^2}$$

$$d_{ij_O}^2 = \sum_{l=1}^{p_O} \left(z_{il} - z_{jl}\right)^2 w_l / q_l^2$$

where variable z_l is the rank of variable x_l.

For nominal variables, the type-specific distance between objects *i* and *j* is based on:

$$d_{ij_N} = \frac{p - m}{p} \quad \text{(Kwek)}$$

where there are *m* matching variables from *p* nominal variables. Our squared (squared) distance motivated by this is:

$$d_{ij_N} = \frac{1}{p_N}\left(\sum_{l=1}^{p_N} \sqrt{w_l} / q_l - \sum_{l=1}^{p_N} 1\left(x_{il} = x_{jl}\right)\sqrt{w_l} / q_l\right)$$

$$d_{ij_N}^2 = \frac{1}{p_N^2}\sum_{l=1}^{p_N}\sum_{m=1}^{p_N}\left[1 - 2 * 1\left(x_{im} = x_{jm}\right) + 1\left(x_{il} = x_{jl}\right)1\left(x_{im} = x_{jm}\right)\right]\sqrt{w_l w_m} / \left(q_l q_m\right)$$

For binary symmetric variables, the type-specific distance between objects *i* and *j* is based on:

$$d_{ij_N} = \frac{b+c}{a+b+c+d} \quad \text{(Kwek)}$$

where there are *a* matching positive (=1) variables, *d* matching negative (=0) variables, and *b*+*c* mismatching variables, from a total of *a*+*b*+*c*+*d* binary symmetric variables. Binary symmetric variables should usually involve a "common" outcome (e.g., >20%). Our (squared) distance is:

$$d_{ij_S} = \frac{1}{p_S} \sum_{l=1}^{p_S} 1\left(x_{il} \neq x_{jl}\right) \sqrt{w_l} / q_l$$

$$d_{ij_S}^2 = \frac{1}{p_S^2} \sum_{l=1}^{p_S} \sum_{m=1}^{p_S} 1\left(x_{il} \neq x_{jl}\right) 1\left(x_{im} \neq x_{jm}\right) \sqrt{w_l w_m} / \left(q_l q_m\right)$$

However, for consistency, VWUO-MD treats binary symmetric variables as nominal, which is easily shown to be equivalent.

For binary asymmetric variables, the type-specific distance between objects *i* and *j* is based on:

$$d_{ij_N} = \frac{b+c}{a+b+c} \quad \text{(Kwek)}$$

where there are *a* matching positive (=1) variables, *d* matching negative (=0) variables, and *b*+*c* mismatching variables, from a total of *a*+*b*+*c*+*d* binary asymmetric variables. *d* was dropped from the denominator in the binary asymmetric formula under the notion that two objects matching on the basis of negatives (=0) is not informative. For example, two objects whom do not have a given rare condition are not necessarily “similar” because of that fact, but nor are they dissimilar. Binary asymmetric variables should usually involve an

"uncommon" outcome (e.g., ≤20%), but conceptual considerations should also be made, i.e., if an outcome is uncommon but matching 1's is conceptually no different than matching 0's, the variable should be treated as binary symmetric. Our (squared) distance is:

$$d_{ij_A} = \frac{\sum_{l=1}^{p_A} 1\left(x_{il} \neq x_{jl}\right)\sqrt{w_l} / q_l}{p_A - \sum_{l=1}^{p_A} 1\left(x_{il} = 0, x_{jl} = 0\right)\sqrt{w_l} / q_l}$$

$$d_{ij_A}^2 = \frac{\sum_{l=1}^{p_A}\sum_{m=1}^{p_A} 1\left(x_{il} \neq x_{jl}\right)1\left(x_{im} \neq x_{jm}\right)\sqrt{w_l w_m} /\left(q_l q_m\right)}{p_A^2 - 2p_A\sum_{l=1}^{p_A} 1\left(x_{il} = 0, x_{jl} = 0\right)\sqrt{w_l} / q_l + \sum_{l=1}^{p_A}\sum_{m=1}^{p_A} 1\left(x_{il} = 0, x_{jl} = 0\right)1\left(x_{im} = 0, x_{jm} = 0\right)\sqrt{w_l w_m} /\left(q_l q_m\right)}$$

The derivations above are easy to follow by assuming all weights and normalizing constants equal 1. Importantly, for even arbitrary weights and normalizing constants, when all variables match perfectly between two objects, every type-specific distance above equals 0.

Finally, the four type-specific distances are combined as the square root of the sum of squared type-specific distances, similar in structure to the formula for Euclidean distance:

$$d_{ij} = \sqrt{d_{ij_C}^2 + d_{ij_O}^2 + d_{ij_N}^2 + d_{ij_A}^2}$$

2.4 Transformation of variable weights

Because the variable weights w_l are constrained to sum to the number of variables p, it is more convenient to differentiate distance functions with respect

to *p*-1 unconstrained v_l related to the variable weights in the following manner (De Soete, 1985/6):

$$w_l = \frac{pv_l^2}{1+\sum_{i=1}^{p-1} v_i^2}, \; l=1,\ldots,p\text{-}1$$

$$w_p = p - \sum_{i=1}^{P-1} w_i$$

Note that De Soete did not constrain his weights to sum to *p*, but rather to 1. We have chosen to constrain our weights to sum to *p* for easier interpretation; regardless of the number of variables, a weight >1 is above average in relative importance. Like De Soete, the VWUO-MD method will obtain the optimal set of v_l and then transform those into constrained w_l.

2.5 The ultrametric loss function

The ultrametric loss function that is minimized with respect to the v_l to produce optimal and informative variable weights differs in an important way between VWUO-MD and De Soete:

$$L_{DS}\left(w_1,\ldots,w_p\right) = \frac{\sum_{\Omega}\left(d_{ik}-d_{jk}\right)^2}{\sum\sum_{i<j} d_{ij}^2} \quad \text{(De Soete)}$$

$$L_{U}\left(w_1,\ldots,w_p\right) = \frac{\sum_{\Omega}\left(d_{ik}-d_{jk}\right)^2}{\left(\prod_{l=1}^{p} w_l\right)^{2/3}} \quad \text{(VWUO-MD)}$$

In L_{DS}, Ω is the set of all triples of objects *i*, *j* and *k* that fail the ultrametric property, and d_{ik} and d_{jk} are, without loss of generality, the two longest sides of the triangle of distances between objects *i*, *j* and *k*. However, it can easily be seen that defining Ω as the set of all triples of objects *regardless* of ultrametricity is equivalent, because the contribution to L_{DS} of the "difference" between two equal distances is 0. For L_U then, Ω is defined as the set of all triples of objects regardless of ultrametricity.

As reported in De Soete, "The denominator in [L_{DS}] is necessary to prevent degenerate solutions where one weight is [*p*] and the others zero."[18] For the purposes of HG, it is clear that such solutions can be termed "degenerate", in that they certainly are not very informative. Ideally, every variable would receive a positive weight, so that one could rank the variables according to involvement in the clustering. In fact it does not appear the denominator in L_{DS} sufficiently penalizes the loss function for zero weights, or even for the more extreme situation described by De Soete above where all but one variable is weighted 0. To see this, simply consider that even with all but one dimension weighted 0, the denominator is still a non-zero quantity. If the numerator can be made to equal 0, which we discover in *4.1 The improved penalty for degenerate solutions* occurs in De Soete's own 1986 data set, then such a degenerate solution arises. Degenerate solutions as described by De Soete put all objects onto the same axis in *p*-space ($\mathbf{R}^p$ in the case of all continuous variables). De Soete obtained non-degenerate solutions when he tested his method on those data, however that may be the result of his first-order estimation (conjugate gradient) method

which apparently (evidenced by his results) can lead to solutions that are far from any local minima. This is discussed in *4.1 The improved penalty for degenerate solutions*, along with an illustrative example.

Despite the problems with De Soete's penalty, it is clear that such a penalty is an important concept, because what use (how informative) is a solution of all w_l=0 except for one w_s=p for HG? This idea motivated the penalty placed in the denominator of the VWUO-MD loss function L_U, the product of variable weights (raised to the power of 2/3). This operates in a trivial manner: if any weight equals 0, the loss function is infinity if the numerator is non-zero, or else it is undefined (0/0). In either case a solution with a weight of exactly 0 should not form the minimum on the loss function surface. The root was taken to increase the differential between small and large weights in estimated variable weight vectors, which in turn better enhances the differences between variables. Other roots were attempted including the p^{th} root, which has the nice theoretical property of being in the unit (single-variable) scale. However, numerical instability was encountered with high-dimensional problems (≥20 dimensions) with anything smaller than a power of 2/3.

2.6 Differentiability of the ultrametric loss function

Before we proceed to develop derivatives for estimating variable weights, a discussion about differentiability is warranted. Differentiation of L_U is not necessarily a straightforward matter. The concern is the order statistics in the numerator. As the variable weights are varied throughout the p-dimensional parameter space (p-1 free parameters), the order of the three distances between

any given triple in Ω can change many times. This raises the question about whether this produces a non-differentiable crease in L_U at such boundaries. If so, one might expect to encounter difficulties in first-order estimation methods, and possibly even worse problems with second-order methods. We should not be concerned about creases where the order of the largest two distances change places with each other, because the squared difference doesn't care about the order of the two largest distances. Our only potential concern is boundaries at which the identity of the smallest of three distances changes. To consider this potential problem, first note that the derivatives of L_U are functions of the derivatives of the distances between triples of points. In the simplest case where n=3, there are only three distances (one triple), two of which are labeled the largest and contribute to the loss function, while the third (smallest) does not.

For each type, we created two- and three-variable data sets with one triple of points (n=3) set up so that the identity of the smallest distance changed as **w** (the vector of w_l) was varied through *p*-space. We solved and plotted L_U and its derivative function with respect to each w_l, where we define w_p as *p* minus the sum of the other w_l. The results showed that the first-order derivative acts as a step function at such creases, that is, it approaches a different value from either side of the crease. Since the derivative depends at such boundaries on which of the three distances is labeled "smallest", yet "smallest" is not uniquely defined on a crease, the loss function is not (first- or second-order) differentiable on such creases.

The situation is not hopeless, however, as far as numerical estimation in real data is concerned where n>3. There are *n* choose 3 triples in Ω which gets big very fast (for example, 10 choose 3 is 120, and 20 choose 3 is 1140), and it seems extremely likely that at most one unique triple of points will hit a crease of equality for any given **w** in *p*-space. This leaves *n* choose 3 minus 1 well behaved triples in Ω contributing the bulk of the derivatives. We can thus hope that the error associated with the derivative step functions for at most one given triple in Ω at any **w** in *p*-space is negligible, and amounts to numeric error. To help answer this question when applying our methodology, the software we developed for VWUO-MD has (as we will describe in *Chapter 3: VWUO-MD software: VWUO.exe*) detailed reporting of current gradient vectors and Hessian matrices during estimation and at the final solution. Fortunately, with the data we analyzed during development of the software (including real and simulated data of sample sizes from a handful >3 to >100), practically speaking the second-order estimation algorithm performed very well, as we will see. The only indication of practical shortcomings was observed in two-variable, categorical analyses. Fortunately, no practical application of VWUO-MD ought to involve only two variables.

2.7 Derivatives for the estimation of variable weights

As mentioned earlier, De Soete utilized a first-order conjugate gradient method to estimate the weights. The primary advantage of this approach is, according to De Soete, that "... this method requires only the first order derivatives...".[18] It is however possible that it was in part this method that led to

De Soete's oversight about the penalty for degenerate solutions. We will explore this idea in *4.1 The improved penalty for degenerate solutions*. In our experience during the development of VWUO-MD, first-order methods which purport to converge when the gradient becomes "sufficiently small", may in fact "converge" in the middle of a gradual slope nowhere near a local minimum. It would seem that such methods are at least in this sense inferior to methods that utilize second-order derivatives for guiding how far to move against the gradient vector at each step. There are other important disadvantages to first order methods, for example, "optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower."[73] With an order n^3 algorithm, fewer iterations is an important consideration. Now that being said, we did not perform an exhaustive exploration of first-order methods. Rather, we looked at De Soete's 1986 failed estimation on his own data set using a conjugate gradient approach (see *4.1 The improved penalty for degenerate solutions*), attempted some variants of steepest descent on VWUO-MD, then proceeded to Newton-Raphson involving the second-order derivatives.

The loss function L_U, and its gradient and Hessian with respect to **v** (the vector of v_l), are all in the form of a sum over Ω, where (recall) Ω is defined as the set of all triples of objects. L_U is a function of distances between objects and the variable weights w_l. The distances between objects are (recall) functions of type-specific distances, which are functions of the weights w_l. The variable weights are functions of the unconstrained v_l. We will perform Newton-Raphson estimation of

the optimal **v** (that which minimizes L_U) by differentiating L_U with respect to the v_l. Multiple applications of the chain rule on the individual terms in Ω is the easiest way to approach this, and this will require first- and second-order derivatives of the type-specific distances between two arbitrary objects, and the variable weights in **w**, with respect to the v_l. Here we present formulas in matrix form for type-specific distances as functions of **v**, as well as their gradients and Hessians with respect to **v**.

For continuous variables (type C), these quantities are:

$$d^2_{ij_C} = \frac{1}{c}\left(\mathbf{x}_i - \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_C^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_C^{\#\#2}\right)\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)$$

$$\begin{aligned}\nabla d^2_{ij_C} = &-\frac{2}{c^2}\left(\mathbf{x}_i - \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_C^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_C^{2}\right)\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)\#\mathbf{v} \\ &+\frac{2}{c}\left(diag\left(\mathbf{x}_i^{(p-1)} - \mathbf{x}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}\right)\right)\left(\mathbf{1}_C^{(p-1)}\right)\end{aligned}$$

$$\begin{aligned}\mathbf{H}d^2_{ij_C} = &\frac{2}{c}\left(diag\left(\mathbf{x}_i^{(p-1)} - \mathbf{x}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1} \\ &-\frac{4}{c^2}\left(diag\left(\mathbf{x}_i^{(p-1)} - \mathbf{x}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}\mathbf{v}\mathbf{v}' \\ &-\frac{4}{c^2}\mathbf{v}\mathbf{v}'\left(diag\left(\mathbf{x}_i^{(p-1)} - \mathbf{x}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1} \\ &-\frac{2}{c^2}\left(\mathbf{x}_i - \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_C^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_C^{\#\#2}\right)\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)\#\mathbf{I}_{(p-1)} \\ &+\frac{8}{c^3}\left(\mathbf{x}_i - \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_C^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_C^{\#\#2}\right)\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)\#\mathbf{v}\mathbf{v}'\end{aligned}$$

where $p = p_C + p_O + p_N + p_S + p_A$ and $c = \sum_{l=1}^{p-1} v_l^2$. The # operator is element-wise multiplication. The ## operator is element-wise exponentiation. $\mathbf{1}_C^{(p-1)}$ is the (p-1) by 1 vector of indicators for whether the i^{th} variable of the first p-1 variables is

type C. $\mathbf{v}_C$ is the p_C by 1 type C sub-vector of $\mathbf{v}$. $\mathbf{q}$ is the vector of normalizing constants q_l. $\mathbf{q}_C$ is the p_C by 1 type C sub-vector of $\mathbf{q}$. $\mathbf{q}_{(p-1)}$ contains the first p-1 elements of $\mathbf{q}$. $\mathbf{x}_i$ is the p_C by 1 vector of type C variables recorded on object i. If the data contain variables of type O, N or A, then $\mathbf{x}_i^{(p-1)}$ is a (p-1) by 1 super-vector of $\mathbf{x}_i$, with non-type C elements filled with 0s and the p^{th} element removed. Otherwise $\mathbf{x}_i^{(p-1)}$ is a (p-1) by 1 sub-vector of $\mathbf{x}_i$ with the p^{th} element removed.

For ordinal variables (type O), these quantities are:

$$d_{ij_O}^2 = \frac{1}{c}\left(\mathbf{z}_i - \mathbf{z}_j\right)'\left(diag\left(\mathbf{q}_O^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_O^{\#\#2}\right)\right)\left(\mathbf{z}_i - \mathbf{z}_j\right)$$

$$\nabla d_{ij_O}^2 = -\frac{2}{c^2}\left(\mathbf{z}_i - \mathbf{z}_j\right)'\left(diag\left(\mathbf{q}_O^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_O^{2}\right)\right)\left(\mathbf{z}_i - \mathbf{z}_j\right)\#\,\mathbf{v}$$
$$+\frac{2}{c}\left(diag\left(\mathbf{z}_i^{(p-1)} - \mathbf{z}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}\right)\right)\left(\mathbf{1}_O^{(p-1)}\right)$$

$$\mathbf{H}d_{ij_O}^2 = \frac{2}{c}\left(diag\left(\mathbf{z}_i^{(p-1)} - \mathbf{z}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}$$
$$-\frac{4}{c^2}\left(diag\left(\mathbf{z}_i^{(p-1)} - \mathbf{z}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}\mathbf{v}\mathbf{v}'$$
$$-\frac{4}{c^2}\mathbf{v}\mathbf{v}'\left(diag\left(\mathbf{z}_i^{(p-1)} - \mathbf{z}_j^{(p-1)}\right)\right)^{\#\#2}\left(diag\left(\mathbf{q}_{(p-1)}^{\#\#2}\right)\right)^{-1}$$
$$-\frac{2}{c^2}\left(\mathbf{z}_i - \mathbf{z}_j\right)'\left(diag\left(\mathbf{q}_O^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_O^{\#\#2}\right)\right)\left(\mathbf{z}_i - \mathbf{z}_j\right)\#\,\mathbf{I}_{(p-1)}$$
$$+\frac{8}{c^3}\left(\mathbf{z}_i - \mathbf{z}_j\right)'\left(diag\left(\mathbf{q}_O^{\#\#2}\right)\right)^{-1}\left(diag\left(\mathbf{v}_O^{\#\#2}\right)\right)\left(\mathbf{z}_i - \mathbf{z}_j\right)\#\,\mathbf{v}\mathbf{v}'$$

$\mathbf{1}_O^{(p-1)}$ is the (p-1) by 1 vector of indicators for whether the i^{th} variable of the first p-1 variables is type O. $\mathbf{v}_O$ is the p_O by 1 type O sub-vector of $\mathbf{v}$. $\mathbf{q}_O$ is the p_O by 1 type O sub-vector of $\mathbf{q}$. $\mathbf{z}_i$ is the p_O by 1 vector of type O variables recorded on object i as ordinal ranks of the categories (e.g., <unable, very difficult, somewhat

difficult, a little difficult, no problem> would be coded <1, 2, 3, 4, 5>). If the data contain variables of type N or A, then $\mathbf{z}_i^{(p-1)}$ is a (p-1) by 1 super-vector of $\mathbf{z}_i$, with non-type O elements filled with 0s and the p^{th} element removed. Otherwise $\mathbf{z}_i^{(p-1)}$ is a (p-1) by 1 vector made from $\mathbf{z}_i$ with its last element removed, left padded with the number of non-type O elements filled with 0s.

For nominal variables (type N), including binary symmetric variables, these quantities are:

$$
\begin{aligned}
d_{ij_N}^2 = & \frac{1}{cp_N^2}\left(\mathbf{1}_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2} \\
& -\frac{2}{cp_N^2}\left(\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\left(\mathbf{1}_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right) \\
& +\frac{1}{cp_N^2}\left(\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}
\end{aligned}
$$

$$
\begin{aligned}
\nabla d_{ij_N}^2 = & \frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\#\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N \\
& -\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\#\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N \\
& -\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{x}_j^{(p-1)}\right)\#\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N \\
& +\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{x}_j^{(p-1)}\right)\#\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N \\
& -\frac{2}{c^2p_N^2}\left(\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{v} \\
& +\frac{4}{c^2p_N^2}\left(\left(\mathbf{x}_i = \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\mathbf{v} \\
& -\frac{2}{c^2p_N^2}\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{v}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{H}d^2_{ij_N} = & \frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\left(\mathbf{1}_N^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& -\frac{4}{c^2p_N^2}\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\mathbf{v}' \\
& -\frac{4}{c^2p_N^2}\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\mathbf{v}\left(\mathbf{1}_N^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& -\frac{2}{c^2p_N^2}\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{I}_{(p-1)} \\
& +\frac{8}{c^3p_N^2}\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{v}\mathbf{v}' \\
& -\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& -\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)\left(\mathbf{1}_N^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& +\frac{4}{c^2p_N^2}\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_N^{(p-1)}\mathbf{v}' \\
& +\frac{4}{c^2p_N^2}\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)\mathbf{v}' \\
& +\frac{4}{c^2p_N^2}\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\mathbf{v}\left(\mathbf{1}_N^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& +\frac{4}{c^2p_N^2}\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\mathbf{v}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& +\frac{4}{c^2p_N^2}\left(\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\mathbf{I}_{(p-1)} \\
& -\frac{16}{c^3p_N^2}\left(\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\left(\mathbf{1}'_N\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)\#\mathbf{v}\mathbf{v}' \\
& +\frac{2}{cp_N^2}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& -\frac{4}{c^2p_N^2}\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)\mathbf{v}' \\
& -\frac{4}{c^2p_N^2}\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\#\mathbf{v}\left(\mathbf{x}_i^{(p-1)}=\mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& -\frac{2}{c^2p_N^2}\left(\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{I}_{(p-1)} \\
& +\frac{8}{c^3p_N^2}\left(\left(\mathbf{x}_i=\mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_N\right)\right)^{-1}\mathbf{v}_N\right)^{\#\#2}\#\mathbf{v}\mathbf{v}'
\end{aligned}
$$

$\mathbf{1}_N^{(p-1)}$ is the (p-1) by 1 vector of indicators for whether the i^{th} variable of the first p-1 variables is type N. $\mathbf{v}_N$ is the p_N by 1 type N sub-vector of $\mathbf{v}$. $\mathbf{q}_N$ is the p_N by 1 type N sub-vector of $\mathbf{q}$. $\mathbf{x}_i$ is the p_N by 1 vector of type N variables recorded on object i, and $\mathbf{x}_i = \mathbf{x}_j$ is the element-wise vector of 0/1 indicators signifying equality between objects i and j on the type N variables. If the data contain variables of type A, then $\mathbf{x}_i^{(p-1)} = \mathbf{x}_j^{(p-1)}$ is a (p-1) by 1 super-vector of $\mathbf{x}_i = \mathbf{x}_j$, with non-type N elements filled with 0s and the p^{th} element removed. Otherwise $\mathbf{x}_i^{(p-1)} = \mathbf{x}_j^{(p-1)}$ is a (p-1) by 1 vector made from $\mathbf{x}_i = \mathbf{x}_j$ with its last element removed, left padded with the number of non-type N elements filled with 0s.

For binary asymmetric variables (type A), these quantities are:

$$d_{ij_A}^2 = F_{ij} / G_{ij}$$

$$\nabla d_{ij_A}^2 = \nabla F \# \left(1/G_{ij}\right) - \nabla G_{ij} \# \left(F_{ij}/G_{ij}^2\right)$$

$$\begin{aligned}\mathbf{H} d_{ij_A}^2 = {} & \mathbf{H}F_{ij} \# \left(1/G_{ij}\right) - \nabla F_{ij} \nabla' G_{ij} \# \left(1/G_{ij}^2\right) - \nabla G_{ij} \nabla' F_{ij} \# \left(1/G_{ij}^2\right) \\ & - \mathbf{H}G_{ij} \# \left(F_{ij}/G_{ij}^2\right) + \nabla G_{ij} \nabla' G_{ij} \# \left(2F_{ij}/G_{ij}^3\right)\end{aligned}$$

where:

$$F_{ij} = \frac{1}{c}\left(\left(\mathbf{x}_i \neq \mathbf{x}_j\right)' \left(diag\left(\mathbf{q}_A\right)\right)^{-1} \mathbf{v}_A\right)^{\#\#2}$$

$$G_{ij} = p_A^2 - \frac{2p_A}{\sqrt{c}}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)' \left(diag\left(\mathbf{q}_A\right)\right)^{-1} \mathbf{v}_A + \frac{1}{c}\left(\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)' \left(diag\left(\mathbf{q}_A\right)\right)^{-1} \mathbf{v}_A\right)^{\#\#2}$$

$$\nabla F_{ij} = \frac{2}{c}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}\right)\#\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A$$
$$-\frac{2}{c^2}\left(\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2}\#\mathbf{v}$$

$$\nabla G_{ij} = -\frac{2p_A}{\sqrt{c}}\left(diag\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)\right)\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\mathbf{1}_A^{(p-1)}$$
$$+\frac{2p_A}{c^{3/2}}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\#\mathbf{v}$$
$$+\frac{2}{c}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)\#\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A$$
$$-\frac{2}{c^2}\left(\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2}\#\mathbf{v}$$

$$\mathbf{H}F_{ij} = \frac{2}{c}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}\right)\left(\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}$$
$$-\frac{4}{c^2}\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}\right)\mathbf{v}'$$
$$-\frac{4}{c^2}\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\#\mathbf{v}\left(\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}$$
$$-\frac{2}{c^2}\left(\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2}\#\mathbf{I}_{(p-1)}$$
$$+\frac{8}{c^3}\left(\left(\mathbf{x}_i \neq \mathbf{x}_j\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2}\#\mathbf{v}\mathbf{v}'$$

$$
\begin{aligned}
\mathbf{H}G_{ij} = & \frac{2p_A}{c^{3/2}}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)\mathbf{v}' \\
& + \frac{2p_A}{c^{3/2}}\mathbf{v}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& + \frac{2p_A}{c^{3/2}}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A \# \mathbf{I}_{(p-1)} \\
& - \frac{6p_A}{c^{5/2}}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A \# \mathbf{v}\mathbf{v}' \\
& + \frac{2}{c}\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& - \frac{4}{c^2}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A \#\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)\mathbf{v}' \\
& - \frac{4}{c^2}\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A \# \mathbf{v}\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_{(p-1)}\right)\right)^{-1} \\
& - \frac{2}{c^2}\left(\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2} \# \mathbf{I}_{(p-1)} \\
& + \frac{8}{c^3}\left(\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)'\left(diag\left(\mathbf{q}_A\right)\right)^{-1}\mathbf{v}_A\right)^{\#\#2} \# \mathbf{v}\mathbf{v}'
\end{aligned}
$$

$\mathbf{v}_A$ is the p_A by 1 type A sub-vector of $\mathbf{v}$. $\mathbf{q}_A$ is the p_A by 1 type A sub-vector of $\mathbf{q}$. $\mathbf{x}_i$ is the p_A by 1 vector of type A variables recorded on object i, $\mathbf{x}_i \neq \mathbf{x}_j$ is the element-wise vector of 0/1 indicators signifying inequality between objects i and j on the type A variables, and $\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)$ is the element-wise vector of 0/1 indicators signifying equality to 0 on both objects i and j on the type A variables. $\mathbf{x}_i^{(p-1)} \neq \mathbf{x}_j^{(p-1)}$ and $\left(\mathbf{x}_i^{(p-1)} = \mathbf{0}, \mathbf{x}_j^{(p-1)} = \mathbf{0}\right)$ are (p-1) by 1 vectors made from $\mathbf{x}_i \neq \mathbf{x}_j$ and $\left(\mathbf{x}_i = \mathbf{0}, \mathbf{x}_j = \mathbf{0}\right)$ respectively with their last elements removed, left padded with the number of non-type A elements filled with 0s.

2.8 Newton-Raphson estimation of variable weights, and use of sample weights

The quantities obtained in the previous section are utilized in Newton-Raphson to find the **v** that minimizes L_U. Each Newton-Raphson iteration is updated from the derivatives obtained on the last iteration as follows:

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \mathbf{H}^{-1} L_U(\mathbf{v}_t) \nabla L_U(\mathbf{v}_t)$$

The VWUO-MD loss function, gradient and Hessian (with respect to **v**) can be expressed as (possibly weighted) sums over Ω:

$$L_U(v_1,...,v_{p-1}) = \sum_{\Omega} t_i t_j t_k U_{ijk}$$

$$\nabla L_U = \sum_{\Omega} t_i t_j t_k \nabla_{U_{ijk}}$$

$$\mathbf{H} L_U = \sum_{\Omega} t_i t_j t_k \mathbf{H}_{U_{ijk}}$$

where:

$$U_{ijk} = F_{ijk} / G_{ijk}$$

$$\nabla U_{ijk} = \nabla F_{ijk} \# (1/G_{ijk}) - \nabla G_{ijk} \# (F_{ijk} / G_{ijk}^2)$$

$$\begin{aligned} \mathbf{H} U_{ijk} = {} & \mathbf{H} F_{ijk} \# (1/G_{ijk}) - \nabla F_{ijk} \nabla' G_{ijk} \# (1/G_{ijk}^2) - \nabla G_{ijk} \nabla' F_{ijk} \# (1/G_{ijk}^2) \\ & - \mathbf{H} G_{ijk} \# (F_{ijk} / G_{ijk}^2) + \nabla G_{ijk} \nabla' G_{ijk} \# (2F_{ijk} / G_{ijk}^3) \end{aligned}$$

with:

$$F_{ijk} = (d_{ik} - d_{jk})^2$$

$$\nabla F_{ijk} = \nabla d_{ik}^2 - \sqrt{\left(d_{jk}^2 / d_{ik}^2\right)} \# \nabla d_{ik}^2 - \sqrt{\left(d_{ik}^2 / d_{jk}^2\right)} \# \nabla d_{jk}^2 + \nabla d_{jk}^2$$

$$\begin{aligned}\mathbf{H}F_{ijk} = {} & \mathbf{H}d_{ik}^2 - \frac{1}{2}\left(d_{jk}^2\right)^{-1/2}\left(d_{ik}^2\right)^{-1/2} \# \nabla d_{ik}^2 \nabla' d_{jk}^2 + \frac{1}{2}\left(d_{jk}^2\right)^{1/2}\left(d_{ik}^2\right)^{-3/2} \# \nabla d_{ik}^2 \nabla' d_{ik}^2 - \left(d_{jk}^2\right)^{1/2}\left(d_{ik}^2\right)^{-1/2} \# \mathbf{H}d_{ik}^2 \\ & + \mathbf{H}d_{jk}^2 - \frac{1}{2}\left(d_{ik}^2\right)^{-1/2}\left(d_{jk}^2\right)^{-1/2} \# \nabla d_{jk}^2 \nabla' d_{ik}^2 + \frac{1}{2}\left(d_{ik}^2\right)^{1/2}\left(d_{jk}^2\right)^{-3/2} \# \nabla d_{jk}^2 \nabla' d_{jk}^2 - \left(d_{ik}^2\right)^{1/2}\left(d_{jk}^2\right)^{-1/2} \# \mathbf{H}d_{jk}^2\end{aligned}$$

$$G_{ijk} = \left(\prod_{l=1}^{p} w_l\right)^{2/3}$$

$$\nabla G_{ijk} = \mathbf{v} \# \left(\frac{-4pG_{ijk}}{3c}\right) + \mathbf{v}^{\#\#-1} \# \left(\frac{4G_{ijk}}{3}\right)$$

$$\begin{aligned}\mathbf{H}G_{ijk} = {} & \mathbf{v}\nabla' G_{ijk} \# \left(\frac{-4p}{3c}\right) + \mathbf{I}_{(p-1)} \# \left(\frac{-4pG_{ijk}}{3c}\right) + \mathbf{v}\mathbf{v}' \# \left(\frac{8pG_{ijk}}{3c^2}\right) + \mathbf{v}^{\#\#-1}\nabla' G_{ijk} \# \left(\frac{4}{3}\right) \\ & + diag\left(\mathbf{v}^{\#\#-2}\right) \# \left(\frac{-4G_{ijk}}{3}\right)\end{aligned}$$

where, without loss of generality, d_{ik} and d_{jk} are the two longest distances between objects *i*, *j* and *k*. t_i is the sample weight for the i^{th} data record if the data have sample weights, or 1 otherwise.

The sample weight terms t_i were added to the loss function to facilitate variance estimation by bootstrapping. However, they can also facilitate situations of a simple random sample (SRS) with multiple instances of data vectors; use of the weight terms is computationally efficient especially with an order n^3 method. We will discuss these topics in *2.10 Covariance estimation and* $\hat{\mathbf{w}}$ *versus w*.

Newton-Raphson is performed starting with **w=1**, and the procedure is iterated until the largest change in the elements of **w** is smaller than a pre-specified convergence criterion. The final solution can be denoted $\hat{\mathbf{w}}$.

2.9 Normalizing multipliers and the calibration data set

The normalizing constants in the previous section are first, for each variable, set to the range. For a continuous variable, the range is the maximum value minus the minimum value in the data. Conceptually, larger differences contributing to the ultrametric loss function L_U should have less importance if the range of the corresponding continuous variable is also very large. Specifically, for continuous variables, the solution should be invariant to scale (affine transformations). For an ordinal or nominal variable, the range is the maximum integer label minus the minimum integer label. Conceptually, larger differences (less equality) contributing to the ultrametric loss function should have less importance if the corresponding ordinal or nominal variable has many categories, since it is generally harder to achieve equality on multinomial variables with many categories. Treating binary asymmetric variables in the same way, the normalizing constant is initially set to 1 for those variables.

The above-described initial normalizing constants provide a fair comparison between variables of the same type, but do not address comparisons between variables of different types. For example, a type C variable might show very clear clustering properties with respect to a latent group structure while a type N variable is independent of any latent grouping in the data, yet without appropriate normalizing constants the type N variable might receive a higher

weight due to making a smaller contribution to L_U solely due to the difference in type-specific distance formulas. We address this issue in a *calibration phase*, where normalizing multipliers are developed to apply to the initial normalizing constants for a fairer comparison between variables of different types. To this end, artificial data are constructed (the calibration data set) containing four variables of each type (16 in total), two that are strongly clustered according to a grouping variable *g* and two that are independent of *g*. In a "fair" comparison, the eight clustering (by *g*) variables' weights (regardless of type) should all receive weights greater than the eight independent (of *g*) variables' weights. In addition, the average variable weight of each type should equal 1 on this calibration data set (a fair comparison between types). To achieve this, the calibration data set is fed into the VWUO-MD procedure with normalizing multipliers initially set to 1. Upon convergence, the average variable weight for each type is calculated. For each type X (where X=C, O or N), the normalizing multiplier for type X is multiplied by the ratio of average type A variable weight over average type X variable weight. The idea is to increase the impact of under-weighted types and decrease the impact of over-weighted types. The normalizing multiplier for type A variables is always 1 (the anchor type). The variable weights are then re-estimated using the new normalizing multipliers and the process is continued until the normalizing multipliers converge to within a pre-specified convergence criterion. This procedure can be lengthy, potentially requiring several hundred iterations in total, even if estimation of variable weights under each set of normalizing multipliers requires only a handful of iterations to converge. Upon

convergence of the normalizing multipliers, the four variables weights of each type when obtained on the calibration data set will average 1.

2.10 Covariance estimation and $\hat{\mathbf{w}}$ versus w

The variable weights in $\hat{\mathbf{w}}$ are supposed to be informative about each variable's participation in the object groupings in the data, and consequently and/or additionally about related variables promising for HG. However, if one estimated variable weight is bigger than another, how can one know whether that represents a legitimate difference in grouping participation, or is merely due to chance? For this purpose, a covariance matrix for $\hat{\mathbf{w}}$ is required. Inferential statistical testing using $\hat{\mathbf{w}}$ would be for differences between the elements of **w**. Before we can consider this, it is necessary to define **w**. The data set on which $\hat{\mathbf{w}}$ was obtained is usually a random sample from a bigger population, e.g., an SRS of all women aged 18 to 24 in Canada on July 1st, 2008. Further, the population can be seen as a random sample from the conceptual "super population" (SP) of all subjects that *might have been* given the characteristics of the subjects and country. This conceptual SP is a relatively stable entity that will not change with every new entry into or exit out of the population. Simply put, **w** is the set of minimizing weights for the SP. That is, if one were to gather the entire SP into a (generally infinite) data set, then **w** would minimize L_U as calculated on that data set. Equivalently, **w** minimizes the expectation of the loss function for a randomly selected triple of points. $\hat{\mathbf{w}}$ estimates **w**.

To estimate the covariance matrix of $\hat{\mathbf{w}}$, we will investigate three approaches. The first approach is an asymptotic method based on the central limit theorem (CLT),[68] the second is a U-statistic-based variance estimator,[69,70,71] and finally we will develop a standard bootstrap variance estimator.[72] For complex survey samples, bootstrap variance estimation can account for the complex survey design, and this can be accomplished with the sample weight terms in the loss function. For other asymptotic variance estimation methods that we consider, use of the sample weight terms must be restricted to the case of multiple instances of data vectors appearing in a simple random sample, where it is computationally more efficient to use the weights to represent multiplicity. Importantly, an SRS is assumed in such methods. For complex survey designs such as stratified or clustered designs where the sample weights represent for example the inverse probability of selection, the bootstrap covariance matrix estimator should be used.

2.10.1 Central limit theorem-based covariance matrix estimators

We first consider an asymptotic method based on the central limit theorem.[68] This is most easily applied in obtaining $\hat{Var}(\hat{\mathbf{v}})$, after which, if we find that $\hat{Var}(\hat{\mathbf{v}})$ is a good estimator, the multivariate delta method can be employed to obtain $\hat{Var}_{(p-1)}(\hat{\mathbf{w}})$. $Var_{(p-1)}(\hat{\mathbf{w}})$ is the (*p*-1) by (*p*-1) submatrix of $Var(\hat{\mathbf{w}})$ created by dropping the last row and column. This should be done at least during theoretical development because $Var(\hat{\mathbf{w}})$ based on all *p* variables will be singular since w_p is *p* minus the sum of the other weights. Later, we will point out that use of the *p* by

p matrix $\hat{Var}(\hat{\mathbf{w}})$ for estimating the variance of individual weight estimates (or in fact any contrast of weights not involving all of them at once), is asymptotically equivalent, more convenient, and possibly more stable than calculating the variance of the last variable weight using all the entries in $\hat{Var}_{(p-1)}(\hat{\mathbf{w}})$. We begin with the estimating equation for $\hat{\mathbf{v}}$:

$$\nabla_{\hat{\mathbf{v}}} = \mathbf{0}$$

The primary assumption to be made about $\nabla_{\hat{\mathbf{v}}}$ is that it is a sum of approximately independent and identically distributed (iid) random variables. (For VWUO-MD we assume that the $\nabla_{U_{ijk}}$ terms in the sum over Ω in the previous section are at least "sufficiently" iid.) We perform a first-order Taylor expansion about **v:**

$$\nabla_{\mathbf{v}} + \mathbf{H}_{\mathbf{v}}(\hat{\mathbf{v}} - \mathbf{v}) \doteq \mathbf{0} \rightarrow \hat{\mathbf{v}} - \mathbf{v} \doteq -\mathbf{H}_{\mathbf{v}}^{-1}\nabla_{\mathbf{v}}$$

Now suppose there are sequences of constants a_n and b_n such that:

$$-b_n\mathbf{H}_{\mathbf{v}} \doteq \mathbf{M}^{-1} \text{ and } a_n\nabla_{\mathbf{v}} \dot{\sim} N(\mathbf{0}, \Sigma)$$

Then:

$$\frac{a_n}{b_n}(\hat{\mathbf{v}} - \mathbf{v}) \doteq (-b_n\mathbf{H}_{\mathbf{v}})^{-1} a_n\nabla_{\mathbf{v}} \dot{\sim} N(\mathbf{0}, \mathbf{M}\Sigma\mathbf{M}')$$

$$\rightarrow Var(\hat{\mathbf{v}}) \doteq Var\left(\frac{b_n}{a_n}N(\mathbf{0}, \mathbf{M}\Sigma\mathbf{M}')\right) = \left(\frac{b_n}{a_n}\right)^2 \mathbf{M}\Sigma\mathbf{M}'$$

For VWUO-MD:

$$b_n = 1/|\Omega|$$

$$a_n = 1/\sqrt{|\Omega|}$$

$$-\hat{\mathbf{M}}^{-1} = \frac{1}{|\Omega|}\sum_{ijk\in\Omega}\mathbf{H}_{U_{ijk}}$$

$$\hat{\Sigma} = \frac{1}{|\Omega|}\sum_{ijk\in\Omega}\nabla_{U_{ijk}}\nabla'_{U_{ijk}}$$

and:

$$\hat{Var}_{CLT}(\hat{\mathbf{v}}) = \left(\frac{b_n}{a_n}\right)^2 \hat{\mathbf{M}}\hat{\Sigma}\hat{\mathbf{M}}'$$

Next, the multivariate delta method[68] is applied to obtain $\hat{Var}(\hat{\mathbf{w}})$ from $\hat{Var}(\hat{\mathbf{v}})$. Each entry w_{ij} in $\hat{Var}(\hat{\mathbf{w}})$ is calculated as follows, derived from the first-order Taylor approximation of the transformation from $\mathbf{v}$ to each w_i:

$$w(\hat{\mathbf{v}}) \doteq w(\mathbf{v}) + \nabla' w(\mathbf{v})(\hat{\mathbf{v}} - \mathbf{v})$$

$$\rightarrow Cov(w_r, w_s) \doteq \nabla' w_r(\mathbf{v}) Var(\hat{\mathbf{v}}) \nabla w_s(\mathbf{v})$$

$$\rightarrow \hat{Cov}(\hat{w}_r, \hat{w}_s) = \nabla' w_r(\hat{\mathbf{v}}) \hat{Var}(\hat{\mathbf{v}}) \nabla w_s(\hat{\mathbf{v}})$$

where

$$\nabla_{\mathbf{v}} w_i(\mathbf{v}) = \frac{2p}{1+\sum_{k=1}^{p-1} v_k^2} \#\mathbf{v}_{(i)} - \frac{2pv_i^2}{\left(1+\sum_{k=1}^{p-1} v_k^2\right)^2} \#\mathbf{v}, \; i \neq p$$

$$\nabla_{\mathbf{v}} w_p(\mathbf{v}) = \frac{-2p}{\left(1+\sum_{k=1}^{p-1} v_k^2\right)^2} \#\mathbf{v}$$

and where $\mathbf{v}_{(i)}$ is the **0** vector except for the i^{th} element which contains v_i. Finally, $\hat{Var}_{(p-1)}(\hat{\mathbf{w}})$ is the (p-1) by (p-1) submatrix of $\hat{Var}(\hat{\mathbf{w}})$ created by dropping the last row and column.

Unfortunately, it is clear that this estimator will badly underestimate the variance (and some practical analyses confirmed this), because the assumption of iid terms in $\nabla_{L_U} = \sum_{\Omega} t_i t_j t_k \nabla_{U_{ijk}}$ is too strongly violated. To see this, consider the following. Ω is the set of all triples of objects in the data, and it is not possible to partition Ω into any sized pieces that are iid. This can be shown by the following reasoning. Suppose that there were a partition of Ω into k>1 parts such that those parts were iid. Then the partition containing a given triple would need to contain at least those other triples with common elements to it. Because all objects are grouped together with all other objects in the three-tuples in Ω, applying similar reasoning to each of the additional members implies by induction that the entire set Ω would have to be contained in the one partition. Therefore, Ω cannot be split into any number of partitions k>1 such that the members are iid.

The CLT-based approach to variance estimation may still be useful to us, however, with some modifications that we will cover next.

2.10.2 The U-statistic-based covariance matrix estimator

Here we develop an asymptotic method based on U-statistics. U-statistics are a class of statistic based on adding terms involving overlapping subsets of a data set. As such, the terms within a U-statistic are not generally sufficiently iid

for usual asymptotic theory for iid samples to hold. Retaining the notation of Serfling (1980)[69], the general form of a U-statistic is:

$$U_n = \frac{1}{\binom{n}{m}} \sum_c h\left(X_{i_1},...,X_{i_m}\right)$$

where $h(\bullet)$ is called the kernel of the U-statistic, the subscript *c* represents "combinations", or "choices" of *m* objects from the sample of size *n*, and the subscript *i* indicates that the kernel is symmetric (invariant to the order of its *m* arguments). Our loss function is proportional to a U-statistic:

$$L_U\left(w_1,...,w_p\right) = \frac{\sum_{\Omega}\left(d_{ik}-d_{jk}\right)^2}{\left(\prod_{l=1}^{p} w_l\right)^{2/3}} = \frac{1}{\binom{n}{3}} \sum_{\Omega} \binom{n}{3} \frac{\left(d_{ik}-d_{jk}\right)^2}{\left(\prod_{l=1}^{p} w_l\right)^{2/3}} \propto \frac{1}{\binom{n}{3}} \sum_c h\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\right)$$

Importantly, the kernel should not be a function of *n*. Had we divided by *n* choose 3, it would have been a U-statistic, but we will see that we can still apply this theory to our method. In our case, the arguments of the kernel are the three selected vectors of the *p* variables.

Essentially we will apply the asymptotic approach of the previous section but with a corrected covariance matrix estimate for the gradient vector, and corrected sequences of constants as needed. First, we make use of Theorem A in Section 5.5.1 in Serfling: If θ=E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)), E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)^2)<∞ and ξ_1=Var(E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)| $\mathbf{x_1}$))>**0** then $U_n \dot{\sim} N(\theta, m^2\xi_1/n)$, or $n^{1/2}(U_n-\theta) \dot{\sim} N(\mathbf{0}, m^2\xi_1)$.

In our case, ξ_1 is a *p-1* by *p-1* matrix, and since θ is the vector **0**, the criterion $E(h(\mathbf{x_1},...,\mathbf{x_k})^2)<\infty$ amounts to a finite covariance matrix for the gradient. U_n is the U-statistic obtained on a sample of size *n*, in our case the gradient function. As mentioned above, *m* is the number of arguments in the kernel, for us *m*=3. Recall in the previous section that we needed a sequence of constants a_n such that $a_n\nabla_v \rightsquigarrow N(\mathbf{0},\Sigma)$. Under U-statistic theory, $a_n=n^{1/2}$ and $\Sigma=m^2\xi_1$. So all that remains is to estimate ξ_1. This might present a serious challenge to solve analytically, thanks to a loss function involving order statistics and multi-type, multivariate distances, made more convoluted by differentiating to obtain the gradient. Instead, we will estimate ξ_1=Var(E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)| $\mathbf{x_1}$)) numerically. The algorithm requires processing every triple in the data, which means for every $\mathbf{x_1}$ to be conditioned on, we must evaluate the kernel at *n-1* choose 2 pairs ($\mathbf{x_2}$,$\mathbf{x_3}$). Averaging these vectors will provide us with an estimate of E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)| $\mathbf{x_1}$) for every $\mathbf{x_1}$ in the data set. Finally, from these *n* estimates we can calculate the sample covariance matrix to estimate Var(E(h($\mathbf{x_1}$,...,$\mathbf{x_k}$)| $\mathbf{x_1}$)).

If we do not first divide the loss function by *n* choose 3 to turn it into a U-statistic, we can still make use of this theory. To do so we need to consider the gradient to be its would-be U-statistic *multiplied* by *n* choose 3. Therefore, in such a case variance estimates as obtained above should be multiplied by *n* choose 3 squared. Another way to see this is to realize that the sequence of constants a_n needed to make $a_n\nabla_v \rightsquigarrow N(\mathbf{0},\Sigma)$ would have to be divided by *n* choose 3 when the loss function is not first divided by *n* choose 3 (or else the

gradient blows up because it is a sum rather than an average), and $\hat{Var}_{CLT}(\hat{\mathbf{v}})$ is proportional to $1/a_n^2$. We will refer to the U-statistic-based covariance matrix estimators as $\hat{Var}_U(\hat{\mathbf{v}})$ and $\hat{Var}_U(\hat{\mathbf{w}})$.

Others (e.g., Lee, 1990) have also suggested bootstrap variance estimation for U-statistics, and that is where we will go next.[71]

2.10.3 The bootstrap covariance matrix estimator

Next we investigate a standard bootstrap variance estimator.[72] This is most easily applied directly to obtain $\hat{Var}_{BS(p-1)}(\hat{\mathbf{w}})$.

The bootstrap variance estimator is similar to a sample covariance matrix of replicated $\hat{\mathbf{w}}_i$, where each $\hat{\mathbf{w}}_i$ in the set is obtained on the i^{th} bootstrap replicate sample, except that the deviations are between each $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{w}}$. If the original sample is an SRS, a bootstrap replicate sample can be obtained by randomly sampling (with replacement) *n*-1 subjects from the original sample of size *n*. If the original sample is a stratified sample, a bootstrap replicate sample can be obtained by randomly sampling (with replacement) n_h-1 primary sampling units (PSUs) in each h^{th} stratum of size n_h in the original sample. The bootstrap weight for a record (≥0) is the number of times it appears in the replicate sample, adjusted so the weight sums to the original sample size. Other post-stratification steps may be performed, depending on what was done to produce the original weight (in a non-SRS).

Each bootstrap weight defined on all subjects represents one replicate sample taken with replacement from the original full sample. On each replicate sample, **w** can be estimated by minimizing the sample weighted loss function using the bootstrap weight that represents the given replicate sample. The variability of these replicate estimates of **w** estimates the variance of $\hat{\mathbf{w}}$. With N_B bootstrap replicate samples, the bootstrap covariance matrix estimator for $\hat{\mathbf{w}}$ is:

$$\hat{Var}_{BS}(\hat{\mathbf{w}}) = \frac{1}{N_B}\sum_{i=1}^{N_B}(\hat{\mathbf{w}}_i - \hat{\mathbf{w}})(\hat{\mathbf{w}}_i - \hat{\mathbf{w}})'$$

and $\hat{Var}_{BS(p-1)}(\hat{\mathbf{w}})$ is the (p-1) by (p-1) submatrix of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ created by dropping the last row and column.

CHAPTER 3: VWUO-MD SOFTWARE: VWUO.EXE

The VWUO-MD approach compares the three distances between objects in every triple in the data. This is necessarily an order n^3 method, which is inherently very slow. For this reason, it was particularly important to develop fast software for performing VWUO-MD. The program VWUO.exe was developed in Microsoft C++ .NET, along with several custom object-oriented matrix classes to ensure fast execution. Threading priority control was added to free up computer resources for other uses even as lengthy analyses run (e.g., Monte Carlo simulations).

3.1 Input data set

The input data set for VWUO.exe should be saved in a tab-delimited text file. The first record should list the variable names, including if applicable sample and replicate weights. Variable names must begin with a ‘C’, ‘O’, ‘N’, ‘A’ or ‘W’ and are not case-sensitive. The first four letters listed correspond to the variable types (e.g., a variable starting with an ‘O’ will be treated as ordinal), while variables beginning with a ‘W’ are treated as sample or replicate weights.

The remaining records in the input data set are data records. Type C variables can contain any decimal or integer numbers, positive or negative. Type O and N variables must contain only nonnegative integers (0 or higher; the category labels or ranks). Type A variables must contain only 0s or 1s. Type W

variables (sample or replicate weights) must contain nonnegative (≥0) decimal or integer numbers. Values of 0 for a weight signify that when that weight is applied to an analysis, the records with 0 weight will not be utilized. This is relevant when a data set contains bootstrap weights, since bootstrap weights are always equal to 0 on one or more records. Table 1 lists the contents of an example input data set which contains 10 bootstrap weights plus a full sample weight. In this example the full sample weight is identically 1 which is appropriate for an SRS.

Table 1. Example input data set for VWUO.exe

C1	C2	CRAND1	O1	O2	ORAND1	N1	N2	NRAND1	A1	A2	ARAND1
4.605775	4.37909	38.71097	1	1	2	1	1	2	0	0	1
9.028632	9.54648	43.30493	4	3	2	4	1	3	1	1	1
8.648056	4.850329	38.35449	3	3	1	4	3	3	1	1	1
8.809451	5.851655	29.98476	4	3	2	4	3	2	1	1	1
9.479575	8.573832	36.50593	4	3	1	4	2	1	0	1	1
8.939512	6.093002	31.83314	4	3	3	4	3	3	1	1	1
9.38209	9.31931	34.80929	4	3	3	4	3	3	1	1	0
7.687192	9.001745	38.21132	4	3	2	4	3	2	1	1	1
4.632366	5.361925	34.35547	1	2	1	1	1	1	0	0	1
5.081679	4.588927	37.66782	1	1	1	1	1	3	0	0	1

W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
1	1.020408	4.081633	2.040816	2.040816	1.020408	2.040816	3.061224	0	0	1.020408
1	0	0	0	1.020408	3.061224	1.020408	1.020408	2.040816	1.020408	1.020408
1	2.040816	0	2.040816	0	1.020408	0	2.040816	0	0	1.020408
1	3.061224	0	1.020408	2.040816	1.020408	0	0	0	0	0
1	0	0	0	0	0	2.040816	0	3.061224	1.020408	2.040816
1	0	0	0	3.061224	1.020408	1.020408	0	1.020408	3.061224	2.040816
1	2.040816	1.020408	2.040816	2.040816	1.020408	1.020408	1.020408	1.020408	0	0
1	0	0	0	0	0	1.020408	1.020408	0	1.020408	0
1	0	0	1.020408	2.040816	0	0	0	1.020408	3.061224	0
1	4.081633	0	1.020408	1.020408	1.020408	2.040816	1.020408	1.020408	1.020408	3.061224

3.2 VWUO.ini configuration file

When VWUO.exe is first opened, the program looks for a file named VWUO.ini in the same folder in which the program was executed. This file can contain several options controlling various aspects of the software. If VWUO.ini is not found, or is found but does not contain a given option, then that option is set to the default for the option. Table 2 lists the available options in VWUO.ini, their defaults, and describes their usage. References to surface maps, graphs and replays of saved analyses will become clearer over the following sections.

Table 2. Available options in VWUO.ini

Option	Description and allowable values	Default
bCalib	Indicates whether the current analysis is for the calibration of normalizing multipliers; 0 or 1	bCalib=0
dEqFactMultC	(Initial, if bCalib=1) normalizing multiplier for type C; decimal >0	dEqFactMultC=1
dEqFactMultO	(Initial, if bCalib=1) normalizing multiplier for type O; decimal >0	dEqFactMultO=1
dEqFactMultN	(Initial, if bCalib=1) normalizing multiplier for type N; decimal >0	dEqFactMultN=1
asInitW	Initial **w** vector, with elements separated by spaces; set to 1 to indicate **w=1**	asInitW=1
iMaxIter	Maximum number of iterations (for each estimating phase if bCalib=1), or 0 to generate solution files at the initial vector; integer ≥0	iMaxIter=100
iThreadPriority12345	Integer from 1 to 5 controlling Windows thread priority, where: 1=THREAD_PRIORITY_HIGHEST 2=THREAD_PRIORITY_ABOVE_NORMAL 3=THREAD_PRIORITY_NORMAL 4=THREAD_PRIORITY_BELOW_NORMAL 5=THREAD_PRIORITY_LOWEST	iThreadPriority12345=3
bAutoMinimize	Indicates whether the application should start minimized (only applies when VWUO.exe is run with command line parameters); 0 or 1	bAutoMinimize=0
dConvCrit	Convergence criterion for estimating variable weights and calibrating the normalizing multipliers (a setting ≤0 will cause the program to run indefinitely which can be useful in certain diagnostic situations); decimal number	dConvCrit=.000001
iNumRandRestart	The number of random restarts, each occurring after convergence of the current process (setting to 0 means no random restarts)	iNumRandRestart=0
dMaxRandRestartDist	The maximum distance to **w** in *p*-space (in a random direction) upon restarting the estimation post-convergence of the current process	dMaxRandRestartDist=0.5
dMinValidW	Minimum allowable weight, below which, for numerical stability, the procedure is not allowed to go during estimation; decimal ≥0	dMinValidW=0.000001

bDeSoeteSurface	Indicates whether the application should read (if available) or create (if directed) a surface map based on L_{DS} (De Soete), otherwise maps will be based on L_U (VWUO-MD); 0 or 1	bDeSoeteSurface=0
dSurfaceByW2Vars	When 1D surface maps are created on two-variable data sets, this controls grid spacing; decimal >1	dSurfaceByW2Vars=0.001
dSurfaceByW3Vars	When 2D surface maps are created on three-variable data sets, this controls grid spacing; decimal >1	dSurfaceByW3Vars=0.02
dSurfaceByW4Vars	When 3D surface maps are created on four-variable data sets, this controls grid spacing; decimal >1	dSurfaceByW4Vars=0.05
dSurfaceMinW1	When surface maps are created, this controls the minimum weight for the first variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no minimum; decimal ≥0	dSurfaceMinW1=0
dSurfaceMaxW1	When surface maps are created, this controls the maximum weight for the first variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no maximum; decimal ≥0	dSurfaceMaxW1=0
dSurfaceMinW2	When surface maps are created, this controls the minimum weight for the second variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no minimum; decimal ≥0	dSurfaceMinW2=0
dSurfaceMaxW2	When surface maps are created, this controls the maximum weight for the second variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no maximum; decimal ≥0	dSurfaceMaxW2=0
dSurfaceMinW3	When surface maps are created, this controls the minimum weight for the third variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no minimum; decimal ≥0	dSurfaceMinW3=0
dSurfaceMaxW3	When surface maps are created, this controls the maximum weight for the third variable at which to estimate the loss function (useful for high resolution, localized maps), 0 indicates no maximum; decimal ≥0	dSurfaceMaxW3=0
dMultAxes	Controls the apparent length of the gradient vectors on graphs of three- or four-variable data sets; decimal >0	dMultAxes=1

dSurfacePow	The power to which the loss function should be raised when a power transformation of the surface map is requested; decimal	dSurfacePow=-1
dPercSurfMap	The (lower) proportion of the range of the plotted surface that should contribute to the color gradations, useful when surface maps contain very deep wells; decimal >0 and ≤1	dPercSurfMap=1
iOriginX	Controls the starting horizontal screen position of the graph in three- or four-variable analyses; integer	iOriginX=1150
iOriginY	Controls the starting vertical position of the graph in three- or four-variable analyses; integer	iOriginY=-125
d3dScale	Controls the starting size of the 3D graph in four-variable analyses; integer >0	d3dScale=850
iSleepMilli	The number of milliseconds to pause between iterations during the replay of a saved analysis; integer ≥0	iSleepMilli=25
bHighRes	Set to 0 to display 2D and 3D surface maps in lower resolution (showing only every 25th grid point), or set to 1 to show all grid points	bHighRes=1
bAlwaysUpdateAll	Set to 1 to clear the display and draw the output entirely every refresh (may produce clearer output), or to 0 to selectively erase and draw only what was updated (may produce smoother animations)	bAlwaysUpdateAll=0

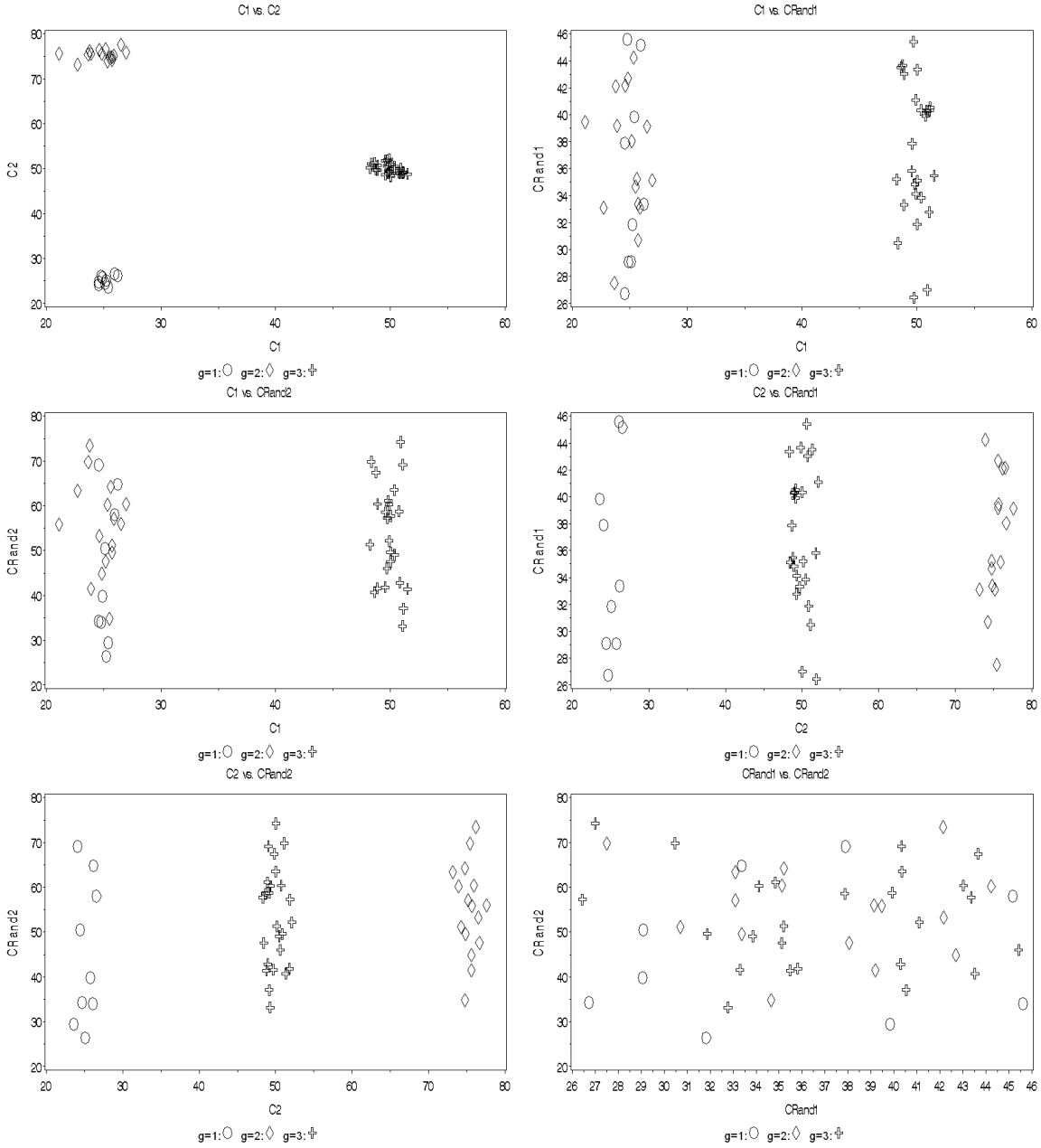
3.3 Calibration of normalizing multipliers

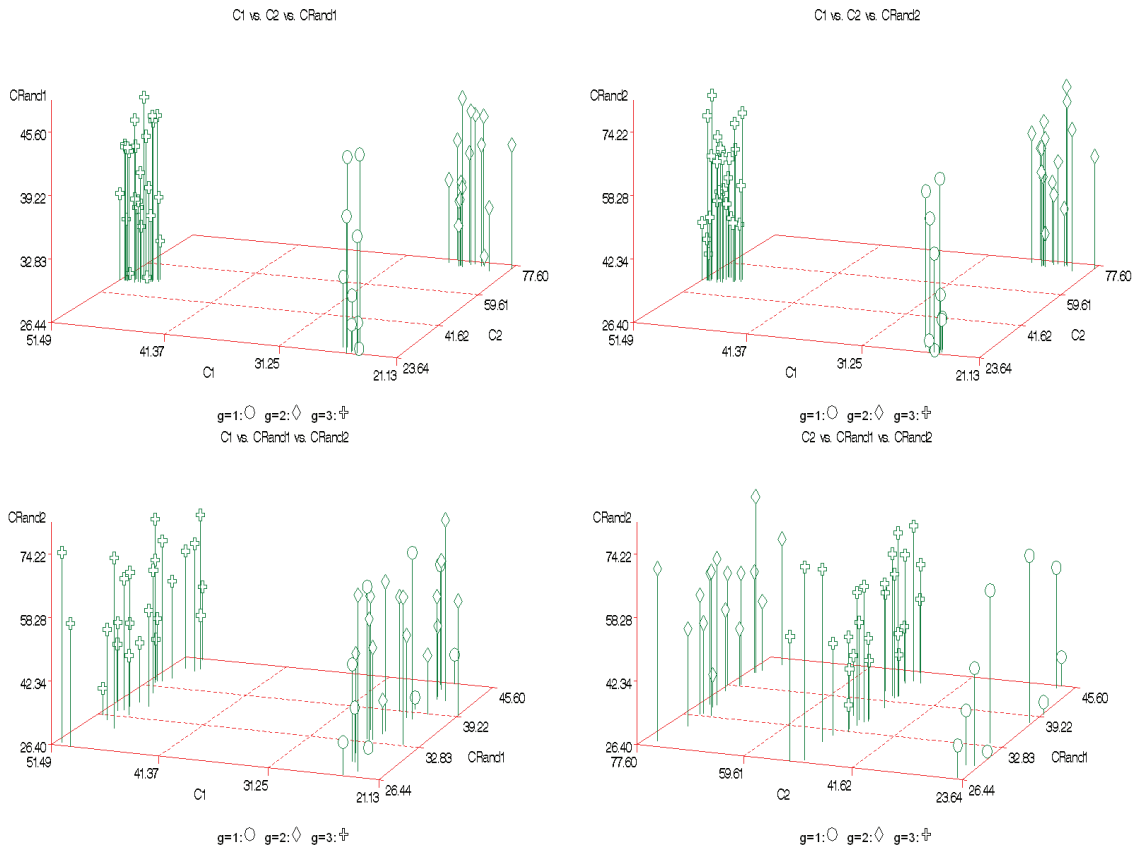
Earlier the concept of normalizing multipliers was discussed. This is the first step in using VWUO.exe, so that subsequent analyses will treat variables of different types fairly. The normalizing multipliers obtained in this section can be used in subsequent analyses; calibration is not required to be performed by all users of the software. However, later on we shall explore the idea of calibration under other scenarios that might depend on a specific target analysis or set of analyses, so this might be a step some users will choose to perform. In this section we illustrate the calibration procedure.

Our calibration data are constructed with three *clearly distinct* clusters according to a pre-assigned three-level group variable *g*, nine records with *g*=1, 16 records with *g*=2 and 25 records with *g*=3. These proportions were chosen to approximately coincide with population proportions of 1/6, 2/6 and 3/6 respectively. The clusters produced in this data set are designed to be extremely clear with little to no variance, so that the normalizing constants developed from it are virtually free of random noise. While it is true that the relative locations of the clusters will impact the multipliers, random noise should ideally have little effect.

Continuous variables C1 and C2 were created from (conditionally on *g*) independent normal distributions depending on *g*. $\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 25 \\ 25 \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$ when *g*=1, $\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 25 \\ 75 \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$ when *g*=2, and $\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 50 \\ 50 \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$ when *g*=3. CRand1 and CRand2 were created from truncated independent normal distributions independent of *g*. $\begin{pmatrix} CRand1 \\ CRand2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 37.5 \\ 50 \end{pmatrix}, \begin{bmatrix} 6.75^2 & 0 \\ 0 & 12.5^2 \end{bmatrix}\right)$, resampled until 25≤CRand1≤ 50 and 25≤CRand2≤75. The four continuous variables are plotted against each other in Figure 6, with symbols indicating each point's value of *g*.

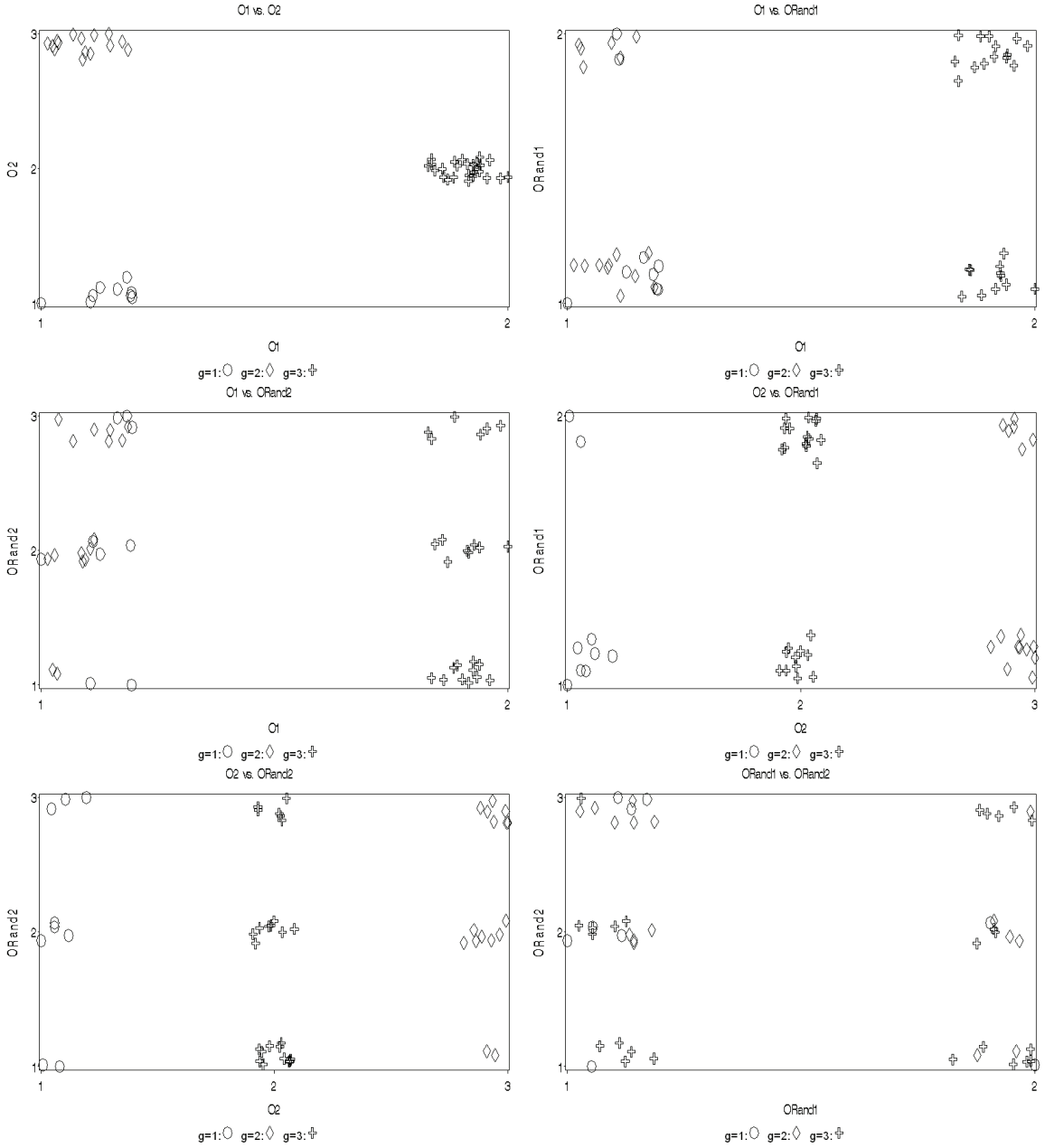
Figure 6. Continuous variables in the calibration data set

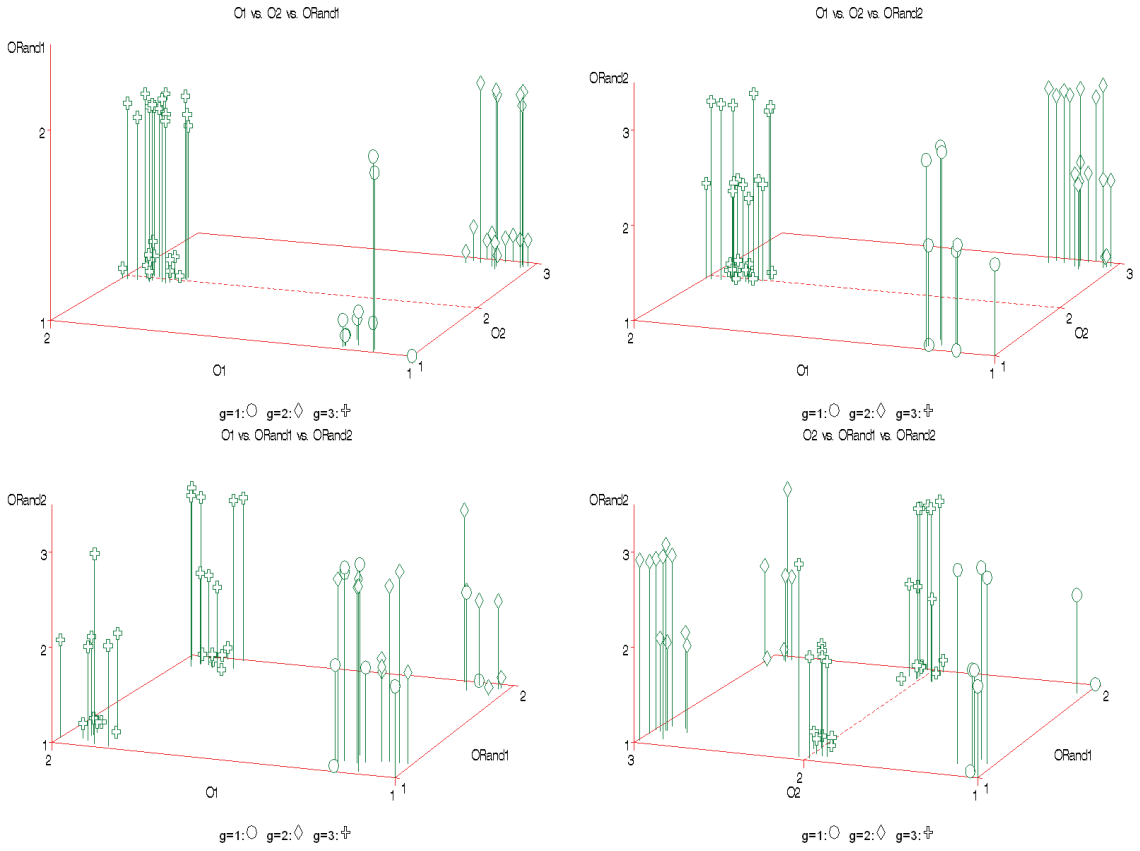




Ordinal variables O1 and O2 were created from (conditionally on *g*) independent multinomial distributions depending on *g*. O1 was selected from a two-level multinomial distribution with probability vector <1, 0> when *g*=1 or *g*=2, and <0, 1> when *g*=3. O2 was selected from a three-level multinomial distribution with probability vector <1,0,0> when *g*=1, <0, 0, 1> when *g*=2, and <0, 1, 0> when *g*=3. ORand1 was created from a two-level multinomial distribution independent of *g*, with probability vector <.5, .5>. ORand2 was created from a three-level multinomial distribution independent of *g*, with probability vector <.333, .333, .334>. The four ordinal variables are plotted against each other in Figure 7, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

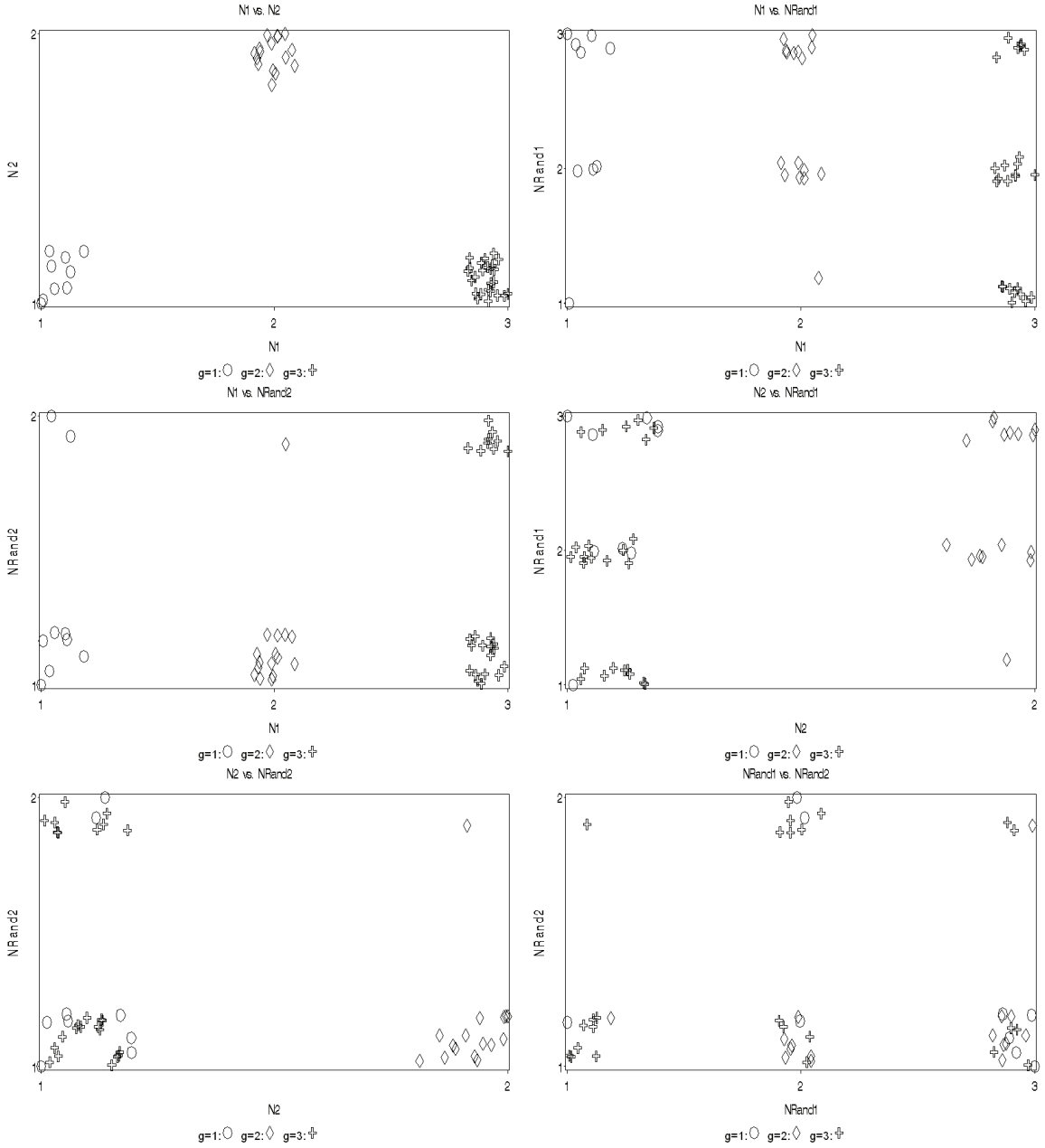
Figure 7. Ordinal variables in the calibration data set; data are jittered

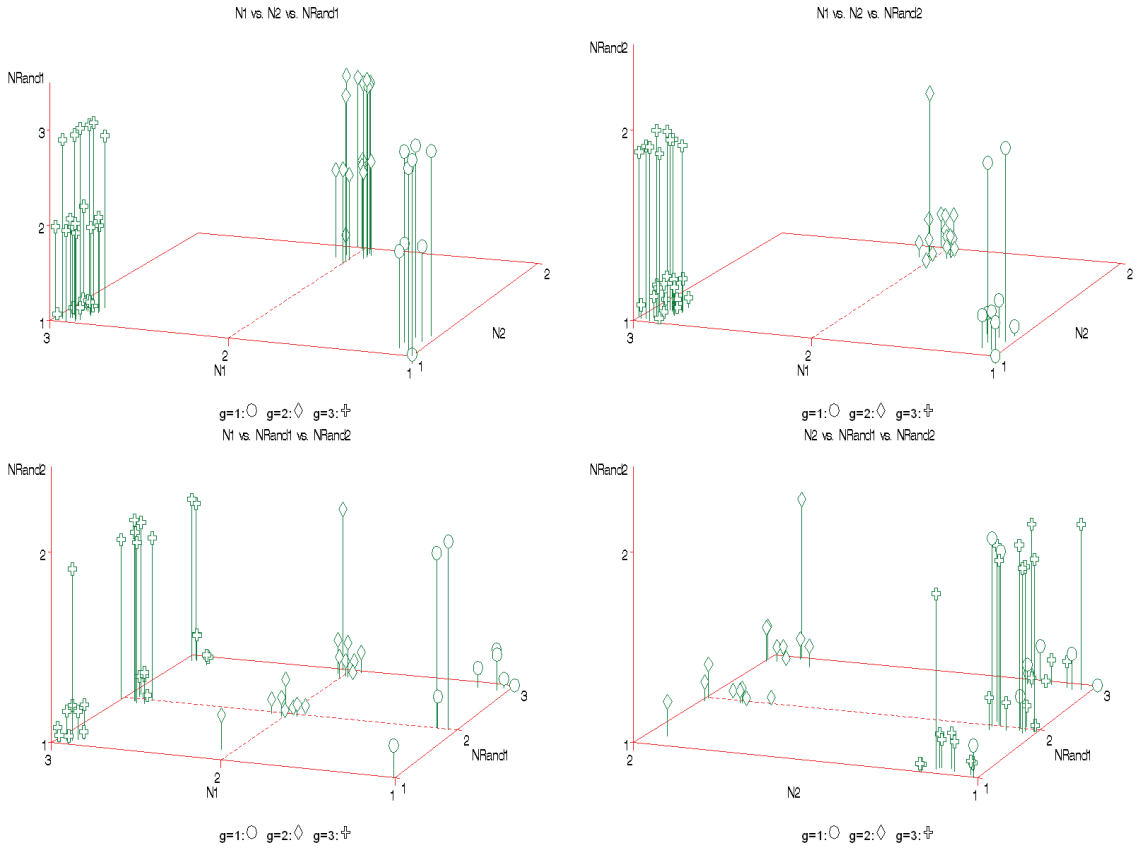




Nominal variables N1 and N2 were created from (conditionally on *g*) independent multinomial distributions depending on *g*. N1 was selected from a three-level multinomial distribution with probability vector <1, 0, 0> when *g*=1, <0, 1, 0> when *g*=2, and <0, 0, 1> when *g*=3. N2 was selected from a two-level multinomial distribution (binary symmetric) with probability vector <1, 0> when *g*=1 or *g*=3, and <0, 1> when *g*=2. NRand1 was created from a three-level multinomial distribution independent of *g*, with probability vector <.333, .333, .334>. NRand2 was created from a two-level multinomial distribution independent of *g*, with probability vector <.667, .333>. The four nominal variables are plotted against each other in Figure 8, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

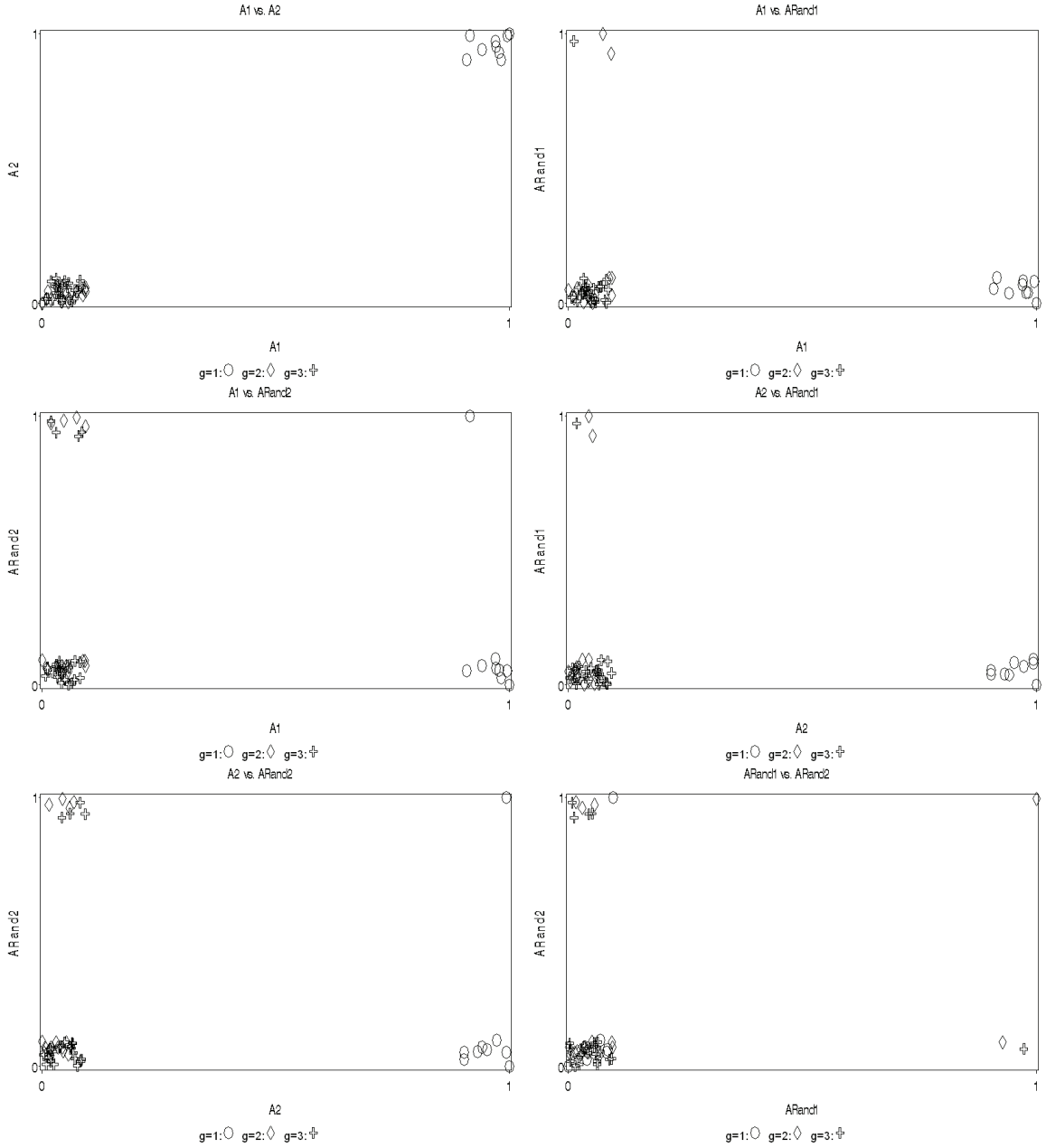
Figure 8. Nominal variables in the calibration data set; data are jittered

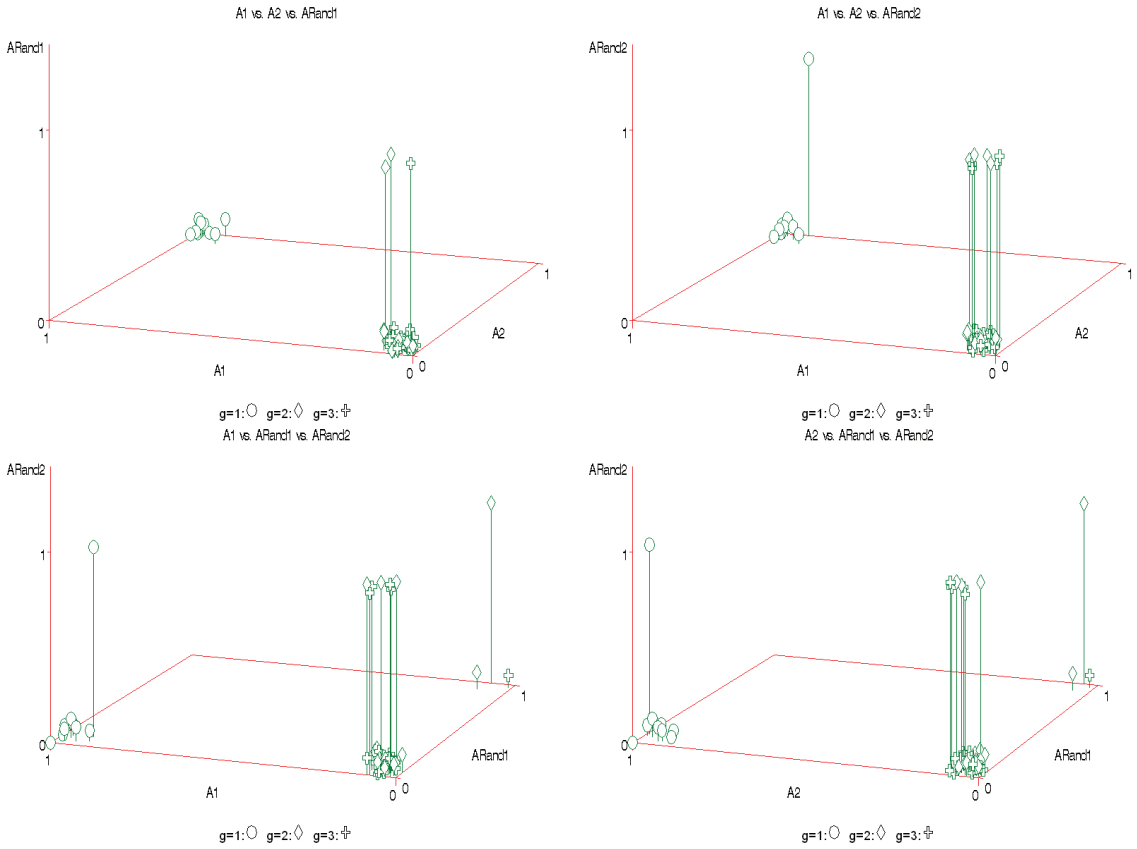




Binary asymmetric variables A1 and A2 were created from (conditionally on *g*) independent Bernoulli distributions depending on *g*. A1 and A2 were selected from a Bernoulli distribution with P(A1=1)=1 when *g*=1, and P(A1=1)=0 when *g*=2 or *g*=3. ARand1 and ARand2 were created from a Bernoulli distribution independent of *g*, with P(ARand2=1)=.167. The four binary asymmetric variables are plotted against each other in Figure 9, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

Figure 9. Binary asymmetric variables in the calibration data set; data are jittered



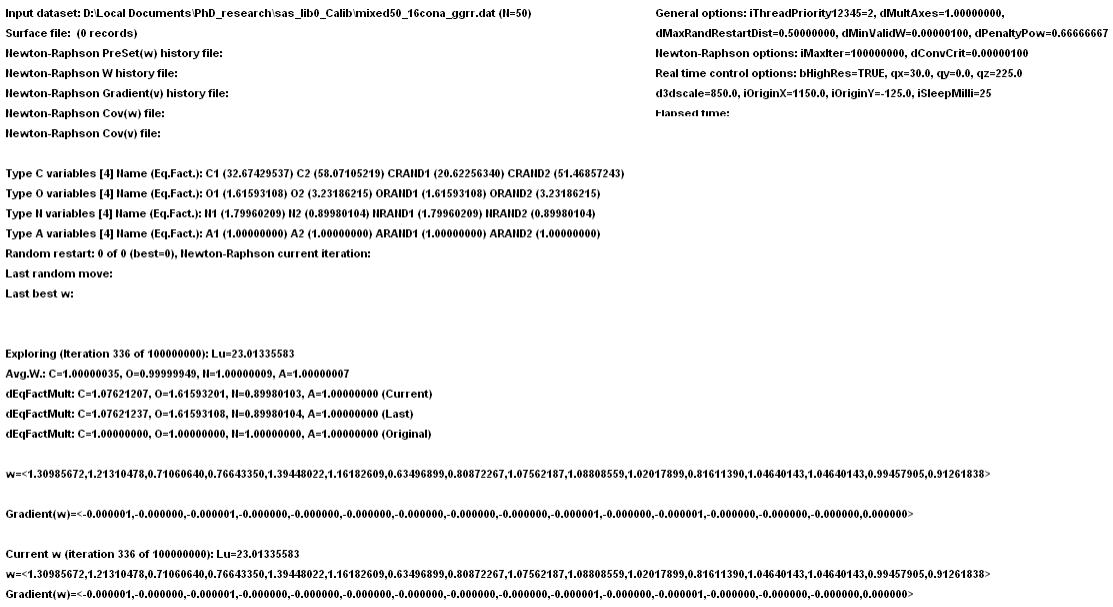


To calibrate the normalizing multipliers, the following lines are included in VWUO.ini:

```
bCalib=1
dEqFactMultC=1
dEqFactMultO=1
dEqFactMultN=1
iMaxIter=100000000
```

The iMaxIter=100000000 setting is important because potentially hundreds of iterations will be required in order to calibrate the normalizing multipliers. VWUO.exe is then run, and the calibration data file, which we have named Mixed50_16cona_ggrr.dat, is opened. The Analyze->Newton-Raphson menu command is selected, and the calibration of the normalizing multipliers will run until complete. Figure 10 shows the results upon convergence.

Figure 10. Calibration of the normalizing multipliers



Upon convergence, several output files are written to the data folder. These will be explained in more detail later. For now, we note that the output file named Mixed50_16cona_ggrr.dat.NRT.Whist.out contains, amongst other information, the total number of iterations, the final variable weight vector $\hat{\mathbf{w}}$, the calculated average weight per variable type, and the estimated normalizing multipliers.

```
Data: D:\Local Documents\PhD_research\sas_lib0_Calib\mixed50_16cona_ggrr.dat
Method: Newton-Raphson T

Results
  Iterations  Convergence          LUT
336.00000000   1.00000000  23.01335583

Avg.W.
          C            O            N            A
  1.00000035   0.99999949   1.00000009   1.00000007

dOrgEqFactMult
          C            O            N            A
  1.00000000   1.00000000   1.00000000   1.00000000

dEqFactMult
          C            O            N            A
  1.07621207   1.61593201   0.89980103   1.00000000

v^2
          C1           C2       CRAND1       CRAND2           O1           O2
ORAND1       ORAND2           N1           N2       NRAND1       NRAND2           A1
A2       ARAND1       ARAND2
  1.43527323   1.32925745   0.77864573   0.83981818   1.52799928   1.27306892
0.69576616   0.88615647   1.17861080   1.19226789   1.11785935   0.89425539
1.14659254   1.14659254   1.08980826   1.00000000
```

```
W history
          Lu            C1            C2        CRAND1        CRAND2            O1
O2        ORAND1        ORAND2           N1           N2        NRAND1        NRAND2
A1            A2        ARAND1        ARAND2
 23.01335583   1.30985672   1.21310478   0.71060640   0.76643350   1.39448022
1.16182609   0.63496899   0.80872267   1.07562187   1.08808559   1.02017899
0.81611390   1.04640143   1.04640143   0.99457905   0.91261838
...snip...
 58.99562964   1.00000000   1.00000000   1.00000000   1.00000000   1.00000000
1.00000000   1.00000000   1.00000000   1.00000000   1.00000000   1.00000000
1.00000000   1.00000000   1.00000000   1.00000000   1.00000000
```

The normalizing multipliers obtained from this run are saved in the following lines we add to VWUO.ini:

```
bCalib=0
dEqFactMultC=1.07621207
dEqFactMultO=1.61593201
dEqFactMultN=0.89980103
iMaxIter=100
```

The bCalib=0 setting is important so that subsequent runs will not try to recalibrate the normalizing multipliers.

With the normalizing multipliers obtained in this step, we are now ready to perform VWUO-MD analyses.

3.4 Exploratory analyses of type C clustered data with 2, 3 and 4 variables

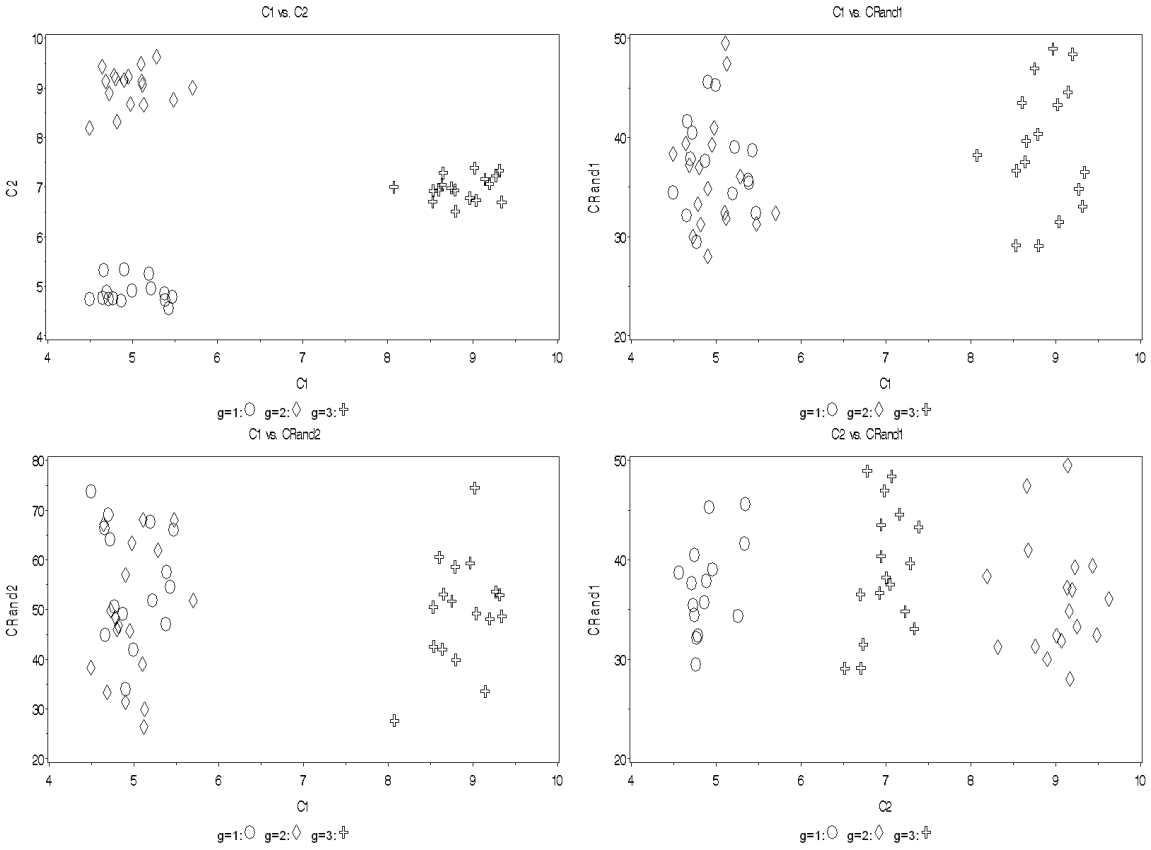
In this section we will create and analyze type C clustered data sets with 2, 3 and 4 variables respectively, in order to illustrate most of the commands, settings and output in VWUO.exe. Our data set has n=50 records, 15 records with *g*=1, 18 records with *g*=2 and 17 records with *g*=3. C1 and C2 were created from (conditionally on *g*) independent normal distributions depending on *g*.

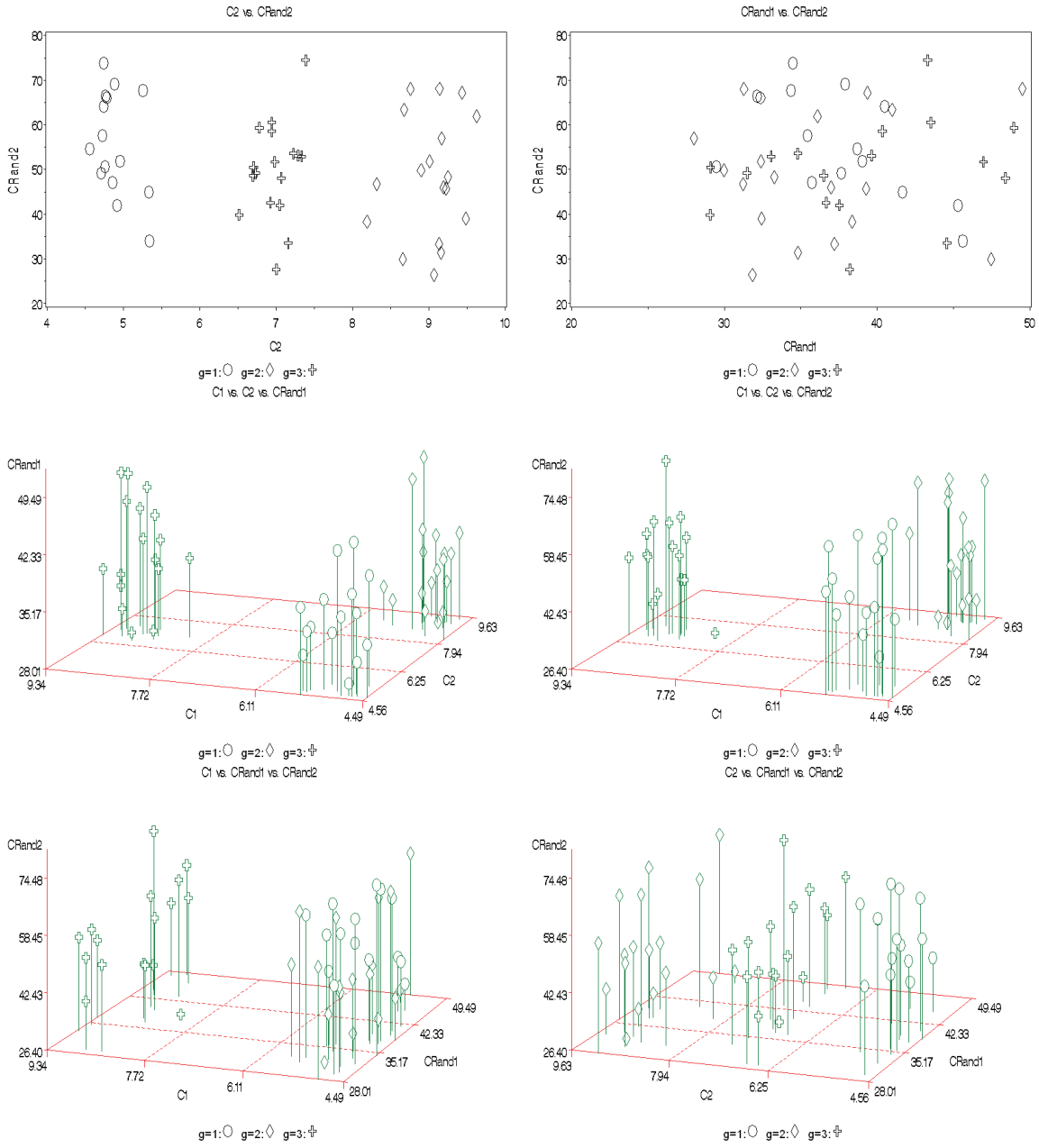
$\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{bmatrix} .1 & 0 \\ 0 & .1 \end{bmatrix}\right)$ when *g*=1, $\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 5 \\ 9 \end{pmatrix}, \begin{bmatrix} .1 & 0 \\ 0 & .1 \end{bmatrix}\right)$ when *g*=2, and

$\begin{pmatrix} C1 \\ C2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 9 \\ 7 \end{pmatrix}, \begin{bmatrix} .1 & 0 \\ 0 & .1 \end{bmatrix} \right)$ when *g*=3. CRand1 and CRand2 were created from truncated independent normal distributions independent of *g*.

$\begin{pmatrix} CRand1 \\ CRand2 \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} 37.5 \\ 50 \end{pmatrix}, \begin{bmatrix} 6.75^2 & 0 \\ 0 & 12.5^2 \end{bmatrix} \right)$, resampled until 25≤CRand1≤ 50 and 25≤CRand2≤75. The four continuous variables in the example data set are plotted against each other in Figure 11, with symbols indicating each point's value of *g*.

Figure 11. Continuous variables in the type C example data set





3.4.1 Opening the two-variable type C example data set

The graphical user interface (GUI) shows an empty screen when VWUO.exe is first loaded. The menu option File->Open or the open icon are used to open the data file as shown in Figure 12.

Figure 12. Opening the two-variable type C example data set

The top left section of text contains the names and paths of the input file and any saved output files. The first line displays the data file name with the number of data records in parentheses. The second line displays the loss function surface map file name (if one has been generated for the data set), which contains a set of loss function values on a grid of variable weights. The third line displays the saved preset history file name (if one has been generated for the data set), which contains, for each iteration of the last VWUO-MD analysis, the starting vector **w**. Typically, unless the analyst manipulates where the analysis should begin, the vectors saved in this file are all equal to **1**. The fourth line displays the saved **w** history file name (if one has been generated for the data set), which contains, for each iteration of the last VWUO-MD analysis, the **w** vector. The fifth line displays the saved $\nabla_{\hat{\mathbf{v}}}$ history file name (if one has been generated for the data set), which contains, for each iteration of the last VWUO-MD analysis, the $\nabla_{\hat{\mathbf{v}}}$ vector. The sixth and seventh lines display the file names containing the saved U-statistic-based asymptotic covariance matrices of **w** and **v** respectively (if they have been generated for the data set), or $\hat{Var}_{CLT}(\hat{\mathbf{w}})$

and $\hat{Var}_{CLT}(\hat{\mathbf{v}})$. VWUO.exe produces these matrices via the methods described earlier.

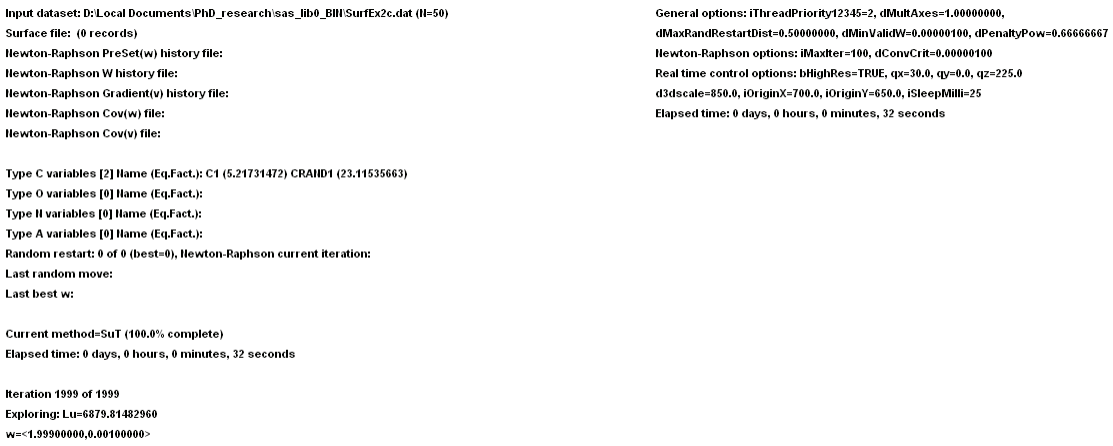
The bottom left section of text contains lists of each variable type, as well as their normalizing constants in parentheses. If VWUO-MD is currently running, it also displays the current iteration number and progress of that iteration.

The upper right section of text displays several options and settings, the relevant ones described earlier.

3.4.2 Preparing 1D L_U and L_{DS} loss function surface maps of the two-variable type C example data set

As we will see shortly, there is a graphical interface that illustrates in real time various estimation factors for three- and four-variable data sets, including (optionally) an intensity surface map of the ultrametric loss function L_U, or for comparison with De Soete, L_{DS} if the analyst requests it. For two-variable data sets, a surface map can also be plotted. Continuing with our example data set, here we will prepare 1D loss function surface maps based on both L_U and L_{DS}. We had run VWUO.exe above with the bDeSoeteSurface=0 setting, thus we will first create a surface map file of L_U. The grid of variable weights will be spaced at dSurfaceByW2Vars=0.005, with no weight falling below dMinValidW=0.005 (or above 2.995). The surface file is created using the Analyze->Surface menu command. The screen shown after the process is complete is illustrated in Figure 13.

Figure 13. Preparing a 1D L_U surface map of the two-variable type C example data set



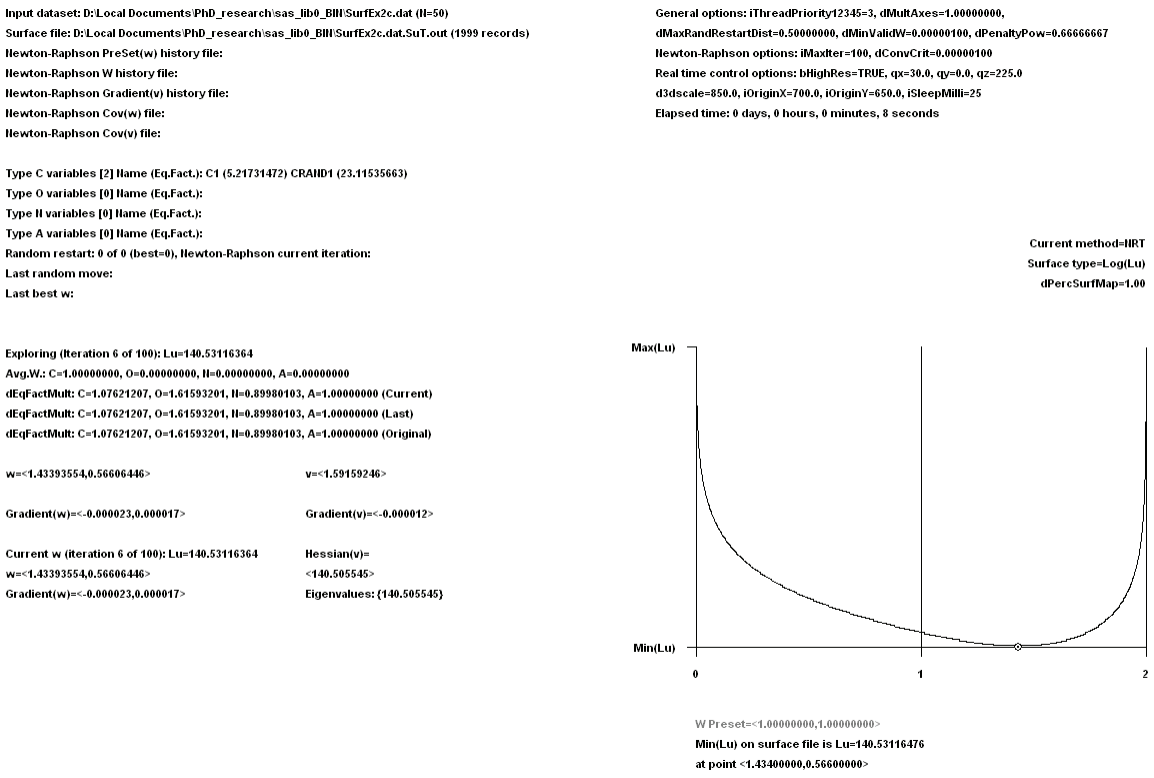
The surface file is saved under the same name as the input data file, but with the suffix ".SuT.out" appended.

Next we close VWUO.exe, implement the bDeSoeteSurface=1 setting, then rerun this process to produce a loss function surface map of L_{DS}. This file is also saved under the same name as the input data file, but with the suffix ".DSSuT.out" appended.

3.4.3 VWUO-MD analysis of the two-variable type C example data set

Having (optionally) created the surface map files, we are now ready to perform VWUO-MD analyses of the two-variable data file, containing variables C1 and CRand1. In this example, we use the bDeSoeteSurface=0 setting, and do not manipulate the starting vector, but just start at **w=1** and allow the program to run. We use the Analyze->Newton-Raphson menu command. This is illustrated in Figure 14.

Figure 14. VWUO-MD analysis of the two-variable type C example data set



1D surface maps are plotted as functions with the first variable weight on the horizontal axis and L_U on the vertical axis. Note in the options shown above the graph that the default function mapped to the screen is the log loss function. This was done to reduce the following problem: it may be hard to discern the shape of the L_U function because of the large range of the loss function obtained on this grid compared to the range near the minimum. To further remedy this, the "<" or ">" commands can be invoked to decrease or increase the proportion of the range that is mapped to the graph (grid points with surface values *above* this range are shown as uniformly maximum level). Another remedy to this issue is the "T" command, which toggles through various transformations of the loss function mapped on the screen. These commands do not affect the estimation, but serve to enhance the visualization of the surface. The plots above, and most

of the other 1D plots in this manuscript will map the log loss function with 100% of the range mapped to the vertical axis.

In VWUO-MD analyses with two variables, the set of possible variable weights falls on a 1D line; the sum of the two variable weights must equal 2. The vector reported in gray text below the plot indicates the starting point, or **w** preset. The black dot at the bottom of the loss function, to which the solution converges, is the observed point of minimum loss function in the plotted surface map. Because the surface map was formed on a grid spaced apart by 0.001, the true solution does not fall exactly onto that spot. With extremely high resolution surface maps (shown later) we will see that the VWUO-MD solution always corresponds exactly with an observed local minimum to within an arbitrary precision.

The text to the left of the graph includes the following results related to the current estimate: $\hat{\mathbf{v}}$, $\hat{\mathbf{w}}$, $\nabla_{\hat{\mathbf{v}}}$, $\nabla_{\hat{\mathbf{w}}}$, Hessian(**v**) and its eigenvalue(s), and the average weight of each type. Also provided for use during calibration of equalizing multipliers, is the original, last and current set of equalizing multipliers. If the bCalib=0 setting is used these sets are always the same, defined in VWUO.ini. The eigenvalues are provided for diagnostic purposes; a well-behaved function at a local minimum should have a nonnegative definite Hessian (nonnegative eigenvalues only).

In this example, the solution is (within the convergence criterion 0.000001) $\mathbf{w}=\langle w_{C1}, w_{CRand1}\rangle=\langle 1.433936, 0.566064\rangle$. C1 receives a bigger weight than CRand1, which is sensible considering the definition of the variables, i.e., that C1

defined clusters in the data, while CRand1 did not. (This is a function of VWUO-MD's optimization of the ultrametric property. For HG, a solution involving a single large weight is probably not as useful as one involving at least two large weights.) The surface map shows that this is the only local minimum. At the solution, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues. Convergence was achieved in six iterations.

3.4.4 Output files

On completion of the VWUO-MD analysis, several output files describing the analysis are written to the folder containing the data file. All have the base name of the data file with suffixes added. The file with suffix ".NRT.Cov.out" contains $\hat{Var}_U(\hat{\mathbf{v}})$. The file with suffix ".NRT.CovW.out" contains $\hat{Var}_U(\hat{\mathbf{w}})$. The file with suffix ".NRT.Dist.out" contains the *n* by *n* variable-weighted distance matrix utilizing the weights in $\hat{\mathbf{w}}$. The file with suffix ".NRT.Phist.out" contains the preset history. The file with suffix ".NRT.Whist.out" contains the **w** history. The file with suffix ".NRT.Ghist.out" contains the $\nabla_{\hat{\mathbf{v}}}$ history.

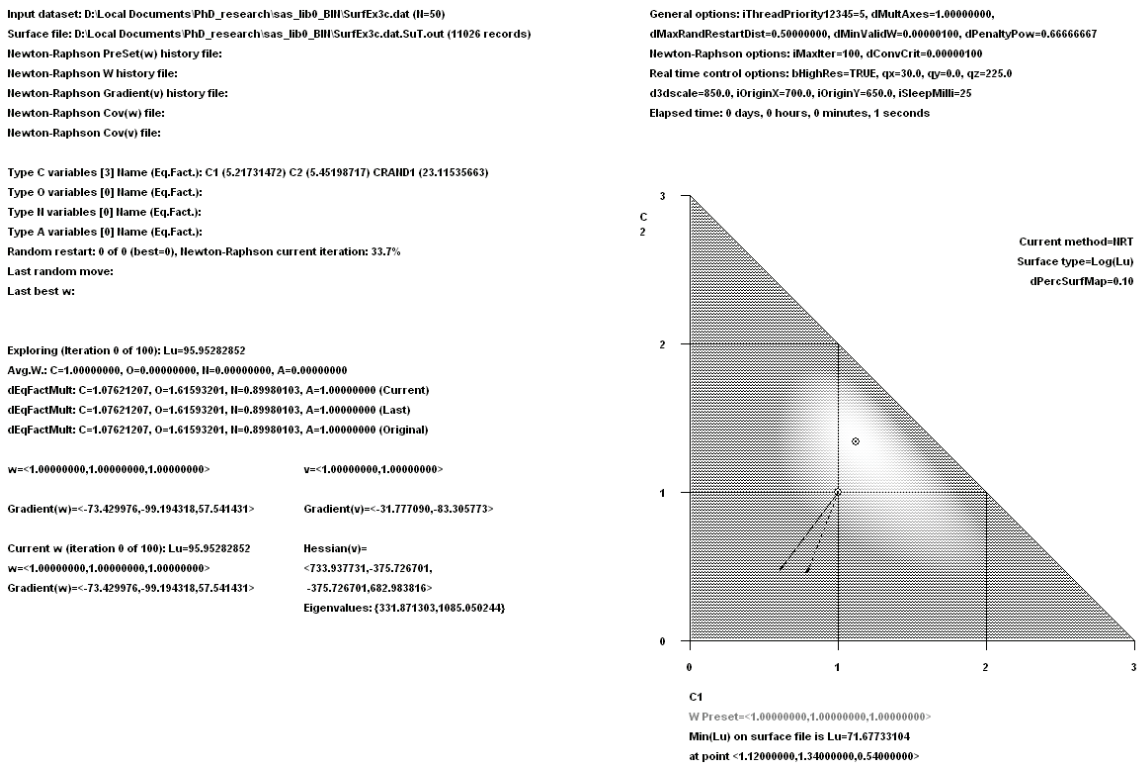
3.4.5 Replaying the VWUO-MD analysis

With the above output files saved, when the data file is reloaded, we will be able to replay the analysis if desired. We can do this with the menu option Analyze->Newton-Raphson Replay. This is a more informative feature when working with three- and four-variable data sets.

3.4.6 VWUO-MD analysis of the three-variable type C example data set

Having (optionally) created the surface map files, we are now ready to perform VWUO-MD analyses of the three-variable data file, containing variables C1, C2 and CRand1. In this example, we use the bDeSoeteSurface=0 setting, and do not manipulate the starting vector, but just start at **w=1** and allow the program to run. We use the Analyze->Newton-Raphson menu command. This is illustrated in Figure 15.

Figure 15. VWUO-MD analysis of the three-variable type C example data set



Input dataset: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx3c.dat (II=50)
Surface file: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx3c.dat.SuT.out (11026 records)
Newton-Raphson PreSet(w) history file:
Newton-Raphson W history file:
Newton-Raphson Gradient(v) history file:
Newton-Raphson Cov(w) file:
Newton-Raphson Cov(v) file:

Type C variables [3] Name (Eq.Fact.): C1 (5.21731472) C2 (5.45198717) CRAND1 (23.11535663)
Type O variables [0] Name (Eq.Fact.):
Type N variables [0] Name (Eq.Fact.):
Type A variables [0] Name (Eq.Fact.):
Random restart: 0 of 0 (best=0), Newton-Raphson current iteration: 15.3%
Last random move:
Last best w:

Exploring (Iteration 2 of 100): Lu=73.32777251
Avg.W.: C=1.00000000, O=0.00000000, N=0.00000000, A=0.00000000
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Current)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Last)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Original)

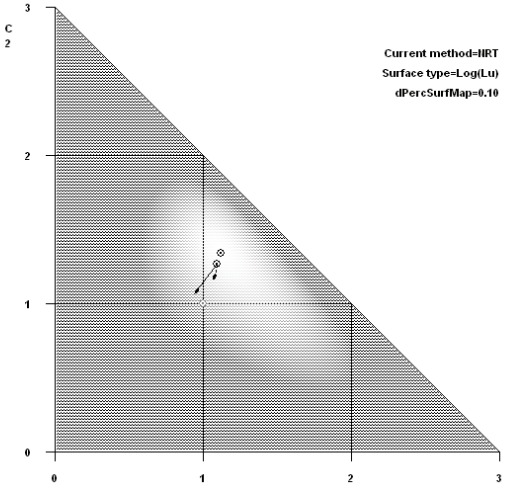
w=<1.09642354,1.26189967,0.64167679>
v=<1.30716680,1.40234299>

Gradient(w)=<-23.498986,-30.670593,21.489351>
Gradient(v)=<-3.371278,-16.523499>

Current w (iteration 2 of 100): Lu=73.32777251
w=<1.09642354,1.26189967,0.64167679>
Gradient(w)=<-23.498986,-30.670593,21.489351>

Hessian(v)=
<510.992324,-389.012519,
-389.012519,437.954369>
Eigenvalues: {83.750458,865.196235}

General options: iThreadPriority12345=5, dMultAxes=1.00000000,
dMaxRandRestartDist=0.50000000, dMinValidW=0.00000100, dPenaltyPow=0.66666667
Newton-Raphson options: iMaxIter=100, dConvCrit=0.00000100
Real time control options: bHighRes=TRUE, qx=30.0, qy=0.0, qz=225.0
d3dscale=850.0, iOriginX=700.0, iOriginY=650.0, iSleepMilli=25
Elapsed time: 0 days, 0 hours, 0 minutes, 4 seconds



C1
W Preset=<1.00000000,1.00000000,1.00000000>
Min(Lu) on surface file is Lu=71.67733104
at point <1.12000000,1.34000000,0.54000000>

Input dataset: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx3c.dat (II=50)
Surface file: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx3c.dat.SuT.out (11026 records)
Newton-Raphson PreSet(w) history file:
Newton-Raphson W history file:
Newton-Raphson Gradient(v) history file:
Newton-Raphson Cov(w) file:
Newton-Raphson Cov(v) file:

Type C variables [3] Name (Eq.Fact.): C1 (5.21731472) C2 (5.45198717) CRAND1 (23.11535663)
Type O variables [0] Name (Eq.Fact.):
Type N variables [0] Name (Eq.Fact.):
Type A variables [0] Name (Eq.Fact.):
Random restart: 0 of 0 (best=0), Newton-Raphson current iteration:
Last random move:
Last best w:

Exploring (Iteration 8 of 100): Lu=71.66689036
Avg.W.: C=1.00000000, O=0.00000000, N=0.00000000, A=0.00000000
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Current)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Last)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Original)

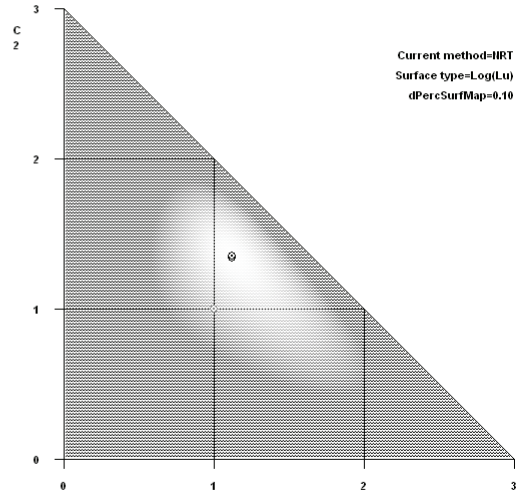
w=<1.12083290,1.34736669,0.53180041>
v=<1.45176427,1.59172695>

Gradient(w)=<-0.000000,-0.000000,0.000000>
Gradient(v)=<0.000000,-0.000000>

Current w (iteration 8 of 100): Lu=71.66689036
w=<1.12083290,1.34736669,0.53180041>
Gradient(w)=<-0.000000,-0.000000,0.000000>

Hessian(v)=
<476.826211,-384.898277,
-384.898277,398.269915>
Eigenvalues: {50.650846,824.445280}

General options: iThreadPriority12345=5, dMultAxes=1.00000000,
dMaxRandRestartDist=0.50000000, dMinValidW=0.00000100, dPenaltyPow=0.66666667
Newton-Raphson options: iMaxIter=100, dConvCrit=0.00000100
Real time control options: bHighRes=TRUE, qx=30.0, qy=0.0, qz=225.0
d3dscale=850.0, iOriginX=700.0, iOriginY=650.0, iSleepMilli=25
Elapsed time: 0 days, 0 hours, 0 minutes, 12 seconds



C1
W Preset=<1.00000000,1.00000000,1.00000000>
Min(Lu) on surface file is Lu=71.67733104
at point <1.12000000,1.34000000,0.54000000>

2D surface maps are plotted as intensity maps with the first variable weight on the horizontal axis and the second variable weight on the vertical axis. L_U determines the intensity of each grid point. Lighter regions in the intensity surface map correspond to lower values of L_U. Note in the options shown to the right of the graph that the default function intensity mapped to the screen is the log loss function. This was done for the same reasons as before. The plots above, and most of the other 2D plots in this manuscript will map the log loss function with the bottom 10% of the range mapped to the grayscale spectrum.

In VWUO-MD analyses with three variables, the set of possible variable weights falls within a 2D triangle; the sum of the first two variable weights cannot exceed 3, and the third variable weight (not plotted) equals 3 minus the sum of the other two. The gray dot centered in the above plots indicates the starting point, or **w** preset (also listed below the graph). The black dot within the light region, to which the solution converges, is the observed point of minimum loss function in the plotted surface map. Because the surface map was formed on a grid spaced apart by 0.02, the true solution does not fall exactly onto that spot. The dotted arrow is the $\nabla_{\hat{\mathbf{v}}}$ vector, while the solid arrow is the $\nabla_{\hat{\mathbf{w}}}$ vector. The former is shown because that is the vector that is directly used for estimation; the latter is shown because it matches the scale of the surface map (**w**). The length of the plotted arrows is proportional to the length of the vectors when they get short enough, providing visual clues about the current proximity to the solution.

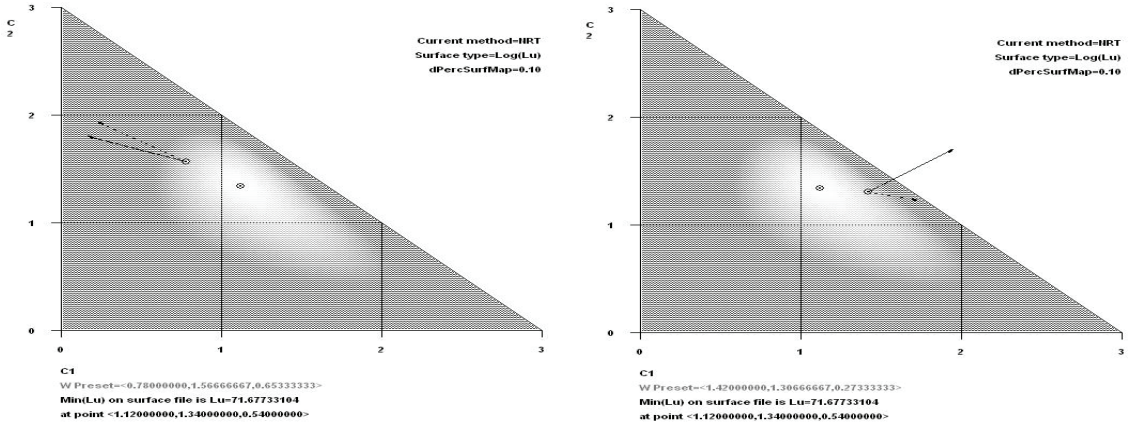
In this example, the solution is (within the convergence criterion 0.000001) $\mathbf{w}=\langle w_{C1}, w_{C2}, w_{CRand1}\rangle=\langle 1.120833, 1.347367, 0.531800\rangle$. C1 and C2 receive

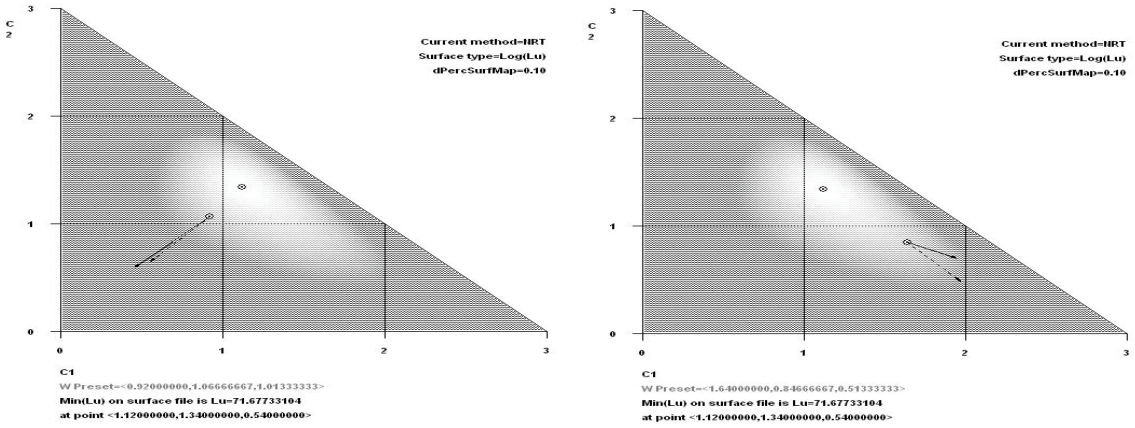
bigger weights than CRand1, which is sensible and informative considering the definition of the variables, i.e., that C1 and C2 together defined clusters in the data, while CRand1 did not. For HG, this solution is more clearly useful than the one obtained in the two-variable example. The surface map graphs in the following section will shows that this is the only local minimum. At the solution, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues. Convergence was achieved in eight iterations.

3.4.7 Exploring the L_U surface of the three-variable type C example data set

Rerunning the analysis with dConvCrit≤0, or (less conveniently) running with dConvCrit>0 and continually resetting the vector before the solution can converge, allows us to explore the 2D surface more interactively. Holding down the right mouse button and moving the pointer around the surface or left clicking on the surface allows us to check that the gradient is pointing in the expected direction relative to the lighter (lower) region of the surface (the gradient should always point uphill). This is illustrated in Figure 16.

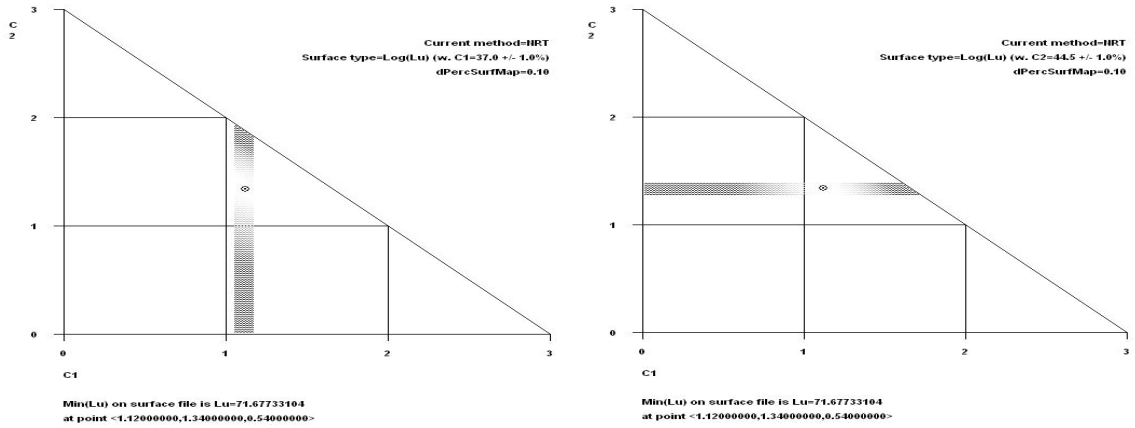
Figure 16. Exploring the L_U surface of the three-variable type C example data set

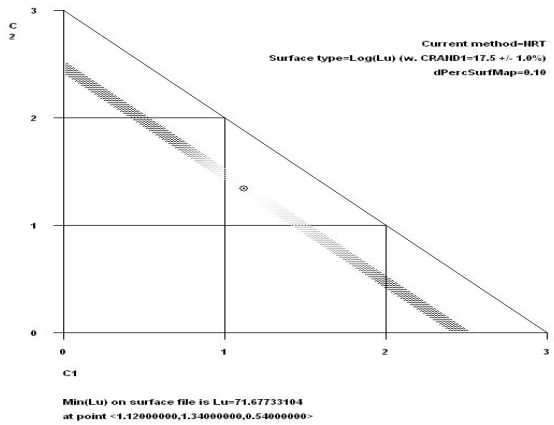




We can use the left or right arrows to toggle through various other views of the surface, useful for exploring the surface. One additional available view is slices in any of three directions that can be moved across the triangle. This is illustrated in Figure 17. We will see that this is actually much more important for visualizing 3D surface maps.

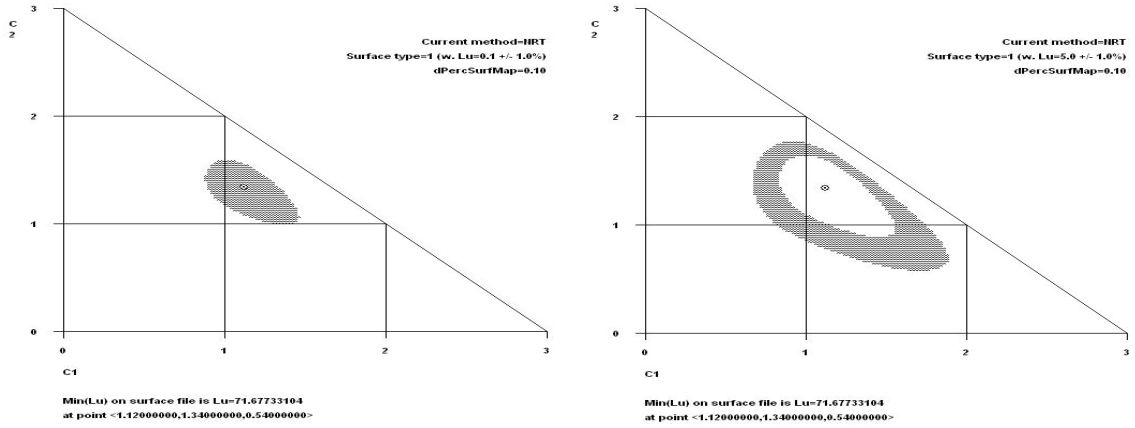
Figure 17. Slices of the 2D surface map of the three-variable type C example data set in three directions

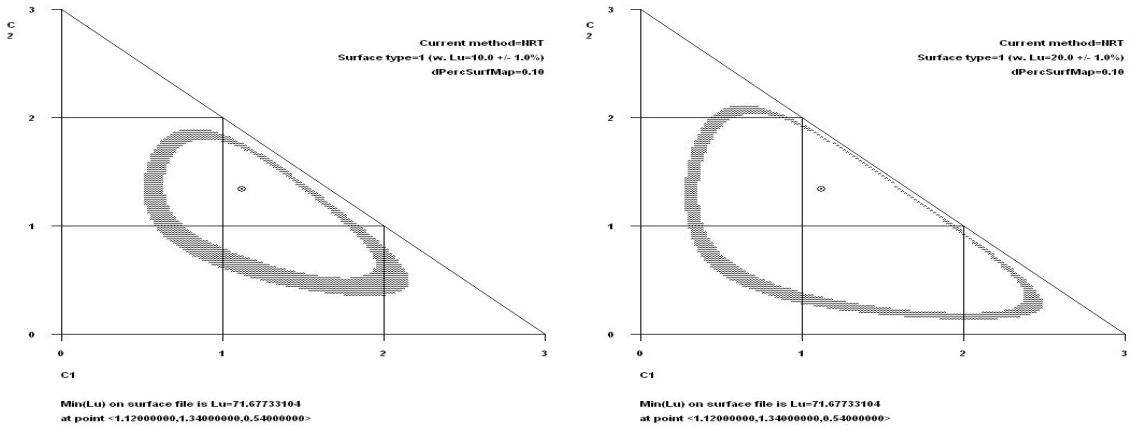




Another available view is a restricted uniform intensity surface map in which only log loss function values at or close to (within 1% of the log loss function range) a specified value are plotted, and that specified value can be changed. This allows one to view the surface map as a series of concentric shells. This is illustrated in Figure 18. Again, while useful for 2D maps, we will see that this is more important for visualizing 3D surface maps.

Figure 18. Restricted uniform intensity 2D surface maps of the three-variable type C example data set





3.4.8 VWUO-MD analysis of the four-variable type C example data set

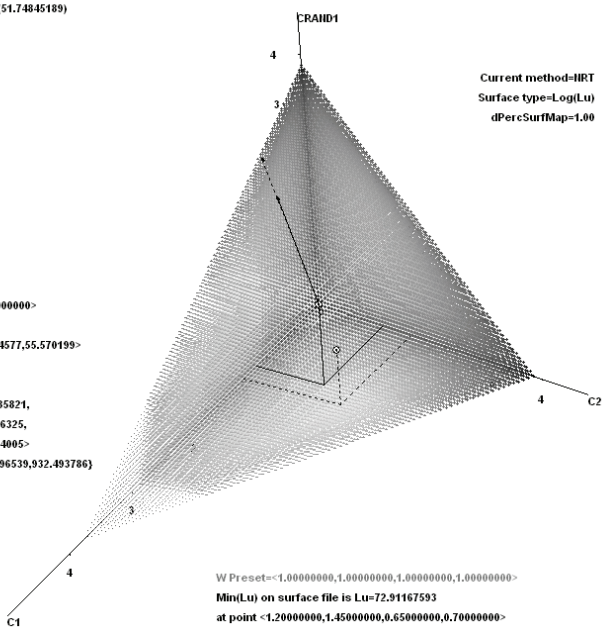
Having (optionally) created the surface map files, we are now ready to perform VWUO-MD analyses of the four-variable data file, containing variables C1, C2, CRand1 and CRand2. In this example, we use the bDeSoeteSurface=0 setting, and do not manipulate the starting vector, but just start at **w=1** and allow the program to run. We use the Analyze->Newton-Raphson menu command. This is illustrated in Figure 19.

Figure 19. VWUO-MD analysis of the four-variable type C example data set

Input dataset: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx4c.dat (II=50)
Surface file: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx4c.dat.SuT.out (79079 records)
Newton-Raphson PreSet(w) history file:
Newton-Raphson W history file:
Newton-Raphson Gradient(v) history file:
Newton-Raphson Cov(w) file:
Newton-Raphson Cov(v) file:

General options: iThreadPriority12345=5, dMultAxes=1.00000000,
dMaxRandRestartDist=0.50000000, dMinValidW=0.00000100, dPenaltyPow=0.66666667
Newton-Raphson options: iMaxIter=100, dConvCrit=0.00000100
Real time control options: bHighRes=TRUE, qx=30.0, qy=0.0, qz=225.0
d3dscale=850.0, iOriginX=1150.0, iOriginY=-125.0, iSleepMilli=25
Elapsed time: 0 days, 0 hours, 0 minutes, 1 seconds

Type C variables [4] Name (Eq.Fact.): C1 (5.21731472) C2 (5.45198717) CRAND1 (23.11535663) CRAND2 (51.74845189)
Type O variables [0] Name (Eq.Fact.):
Type N variables [0] Name (Eq.Fact.):
Type A variables [0] Name (Eq.Fact.):
Random restart: 0 of 0 (best=0), Newton-Raphson current iteration: 39.1%
Last random move:
Last best w:



Exploring (Iteration 0 of 100): Lu=92.47696855
Avg.W.: C=1.00000000, O=0.00000000, N=0.00000000, A=0.00000000
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Current)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Last)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Original)

w=<1.00000000,1.00000000,1.00000000,1.00000000>
v=<1.00000000,1.00000000,1.00000000>

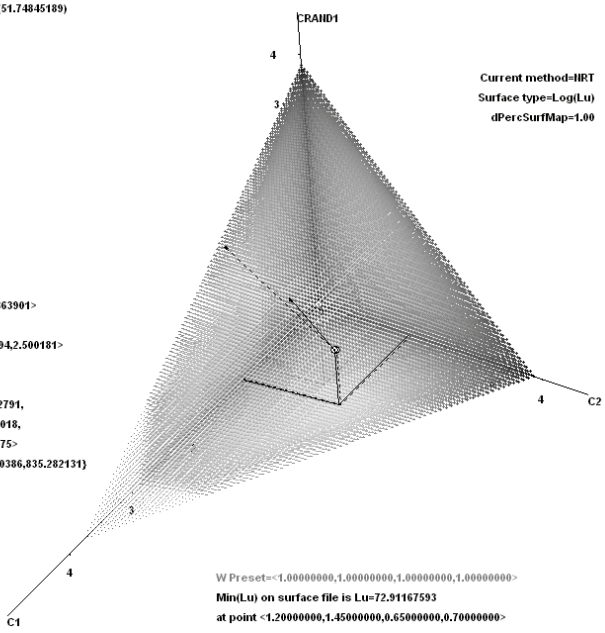
Gradient(w)=<-39.467677,-59.229722,4.147666,23.637433>
Gradient(v)=<-31.660488,-71.184577,55.570199>

Current w (iteration 0 of 100): Lu=92.47696855
w=<1.00000000,1.00000000,1.00000000,1.00000000>
Gradient(w)=<-39.467677,-59.229722,4.147666,23.637433>
Hessian(v)=
<670.787915,-290.484844,-176.335821,
-290.484844,598.622790,-129.596325,
-176.335821,-129.596325,431.484005>
Eigenvalues: {167.104384,601.296539,932.493786}

Input dataset: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx4c.dat (II=50)
Surface file: D:\Local Documents\PhD_research\sas_lib0_BIII\SurfEx4c.dat.SuT.out (79079 records)
Newton-Raphson PreSet(w) history file:
Newton-Raphson W history file:
Newton-Raphson Gradient(v) history file:
Newton-Raphson Cov(w) file:
Newton-Raphson Cov(v) file:

General options: iThreadPriority12345=5, dMultAxes=1.00000000,
dMaxRandRestartDist=0.50000000, dMinValidW=0.00000100, dPenaltyPow=0.66666667
Newton-Raphson options: iMaxIter=100, dConvCrit=0.00000100
Real time control options: bHighRes=TRUE, qx=30.0, qy=0.0, qz=225.0
d3dscale=850.0, iOriginX=1150.0, iOriginY=-125.0, iSleepMilli=25
Elapsed time: 0 days, 0 hours, 0 minutes, 7 seconds

Type C variables [4] Name (Eq.Fact.): C1 (5.21731472) C2 (5.45198717) CRAND1 (23.11535663) CRAND2 (51.74845189)
Type O variables [0] Name (Eq.Fact.):
Type N variables [0] Name (Eq.Fact.):
Type A variables [0] Name (Eq.Fact.):
Random restart: 0 of 0 (best=0), Newton-Raphson current iteration: 37.8%
Last random move:
Last best w:

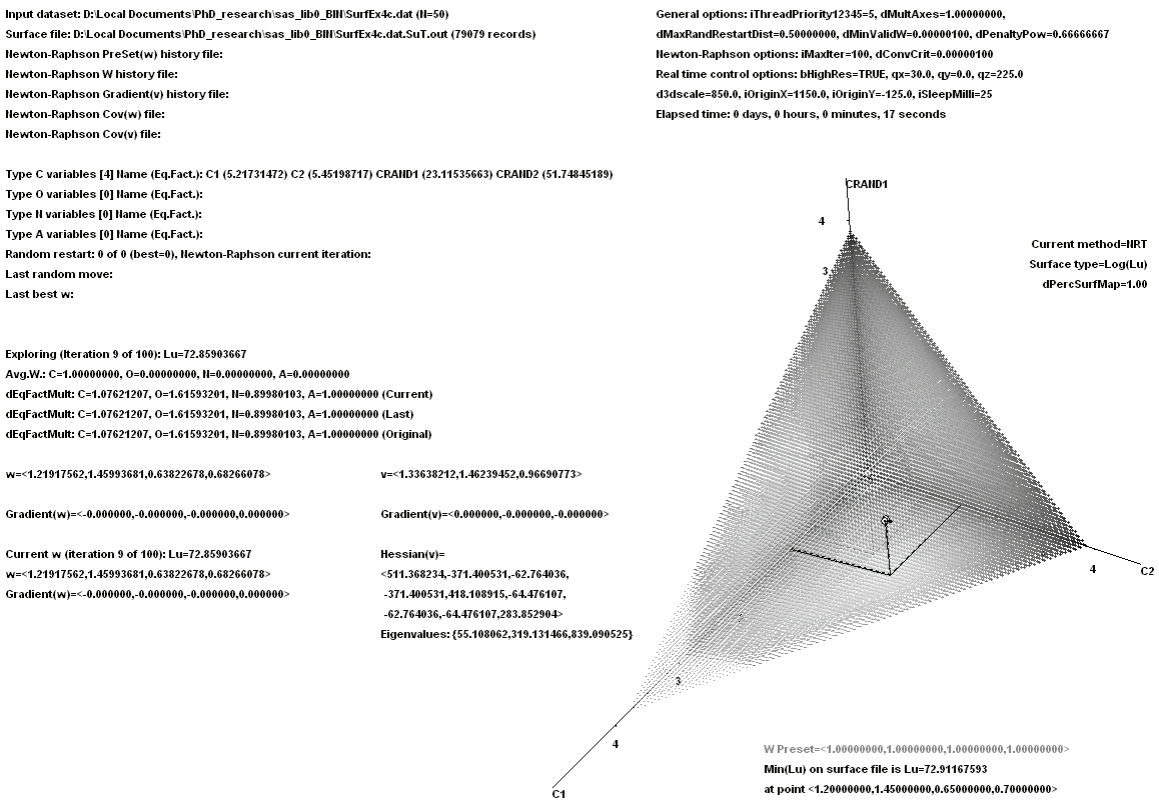


Exploring (Iteration 3 of 100): Lu=72.97510193
Avg.W.: C=1.00000000, O=0.00000000, N=0.00000000, A=0.00000000
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Current)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Last)
dEqFactMult: C=1.07621207, O=1.61593201, N=0.89980103, A=1.00000000 (Original)

w=<1.20981118,1.42450126,0.65401713,0.71167044>
v=<1.30382509,1.41478992,0.95863901>

Gradient(w)=<-3.707647,-6.194602,-1.787334,3.619680>
Gradient(v)=<-0.163249,-5.185194,2.500181>

Current w (iteration 3 of 100): Lu=72.97510193
w=<1.20981118,1.42450126,0.65401713,0.71167044>
Gradient(w)=<-3.707647,-6.194602,-1.787334,3.619680>
Hessian(v)=
<513.427912,-365.239910,-68.132791,
-365.239910,420.439023,-63.339018,
-68.132791,-63.339018,291.575075>
Eigenvalues: {61.489494,328.670386,835.282131}



3D surface maps are plotted as intensity maps with the first variable weight on the near axis (labeled at the bottom center of the screen), the second variable weight on the other horizontal axis (labeled on the right of the screen), and the third variable weight on the vertical axis. L_U determines the intensity of each grid point. Lighter regions in the intensity surface map correspond to lower values of L_U. Note in the options shown to the right of the graph that the default function intensity mapped to the screen is the log loss function. This was done for the same reasons as before. The plots above map the log loss function with 100% of the range mapped to the grayscale spectrum.

In 3D plots, the plotting symbols optionally depend on the points' near axis values (the axis labeled at the bottom center of the screen), to improve the visualization of complex surface maps, especially when the depth of the points

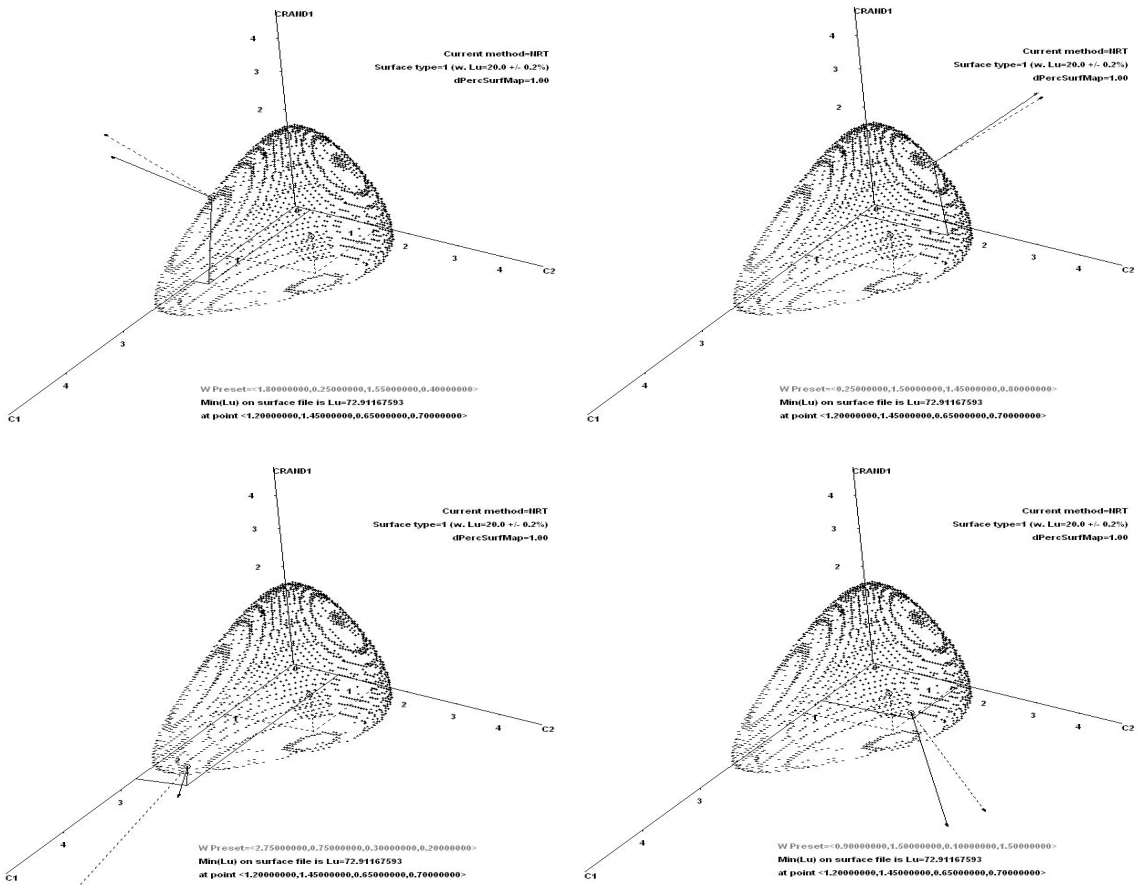
varies widely within the mass. Plotting symbols of points within 1, 2, 3 and 4 units of the origin on the near axis are displayed as '+', '^', '-' and '.' respectively. The idea is to give the impression of the heaviest objects being on the back side of the mass with successively lighter objects closer to the foreground. We will see that this is especially effective when viewing a surface as a 3D contour map, but when viewing slices of a 3D map, visualization is better without symbol coding.

In this example, the solution is (within the convergence criterion 0.000001) $\mathbf{w}=<w_{C1}, w_{C2}, w_{CRand1}, w_{CRand2}>=<1.219176, 1.459937, 0.638227, 0.682661>$. C1 and C2 receive bigger weights than CRand1 and CRand2, which is sensible and informative considering the definition of the variables, i.e., that C1 and C2 together defined clusters in the data, while CRand1 and CRand2 did not. This is an informative result for HG, since C1 and C2 are related (through *g*). The surface map graphs in the following section will shows that this is the only local minimum. At the solution, it was confirmed that $\nabla_{\hat{\mathbf{v}}}=\mathbf{0}$ and Hessian(**v**) had positive eigenvalues. Convergence was achieved in nine iterations.

3.4.9 Exploring the L_U surface of the four-variable type C example data set

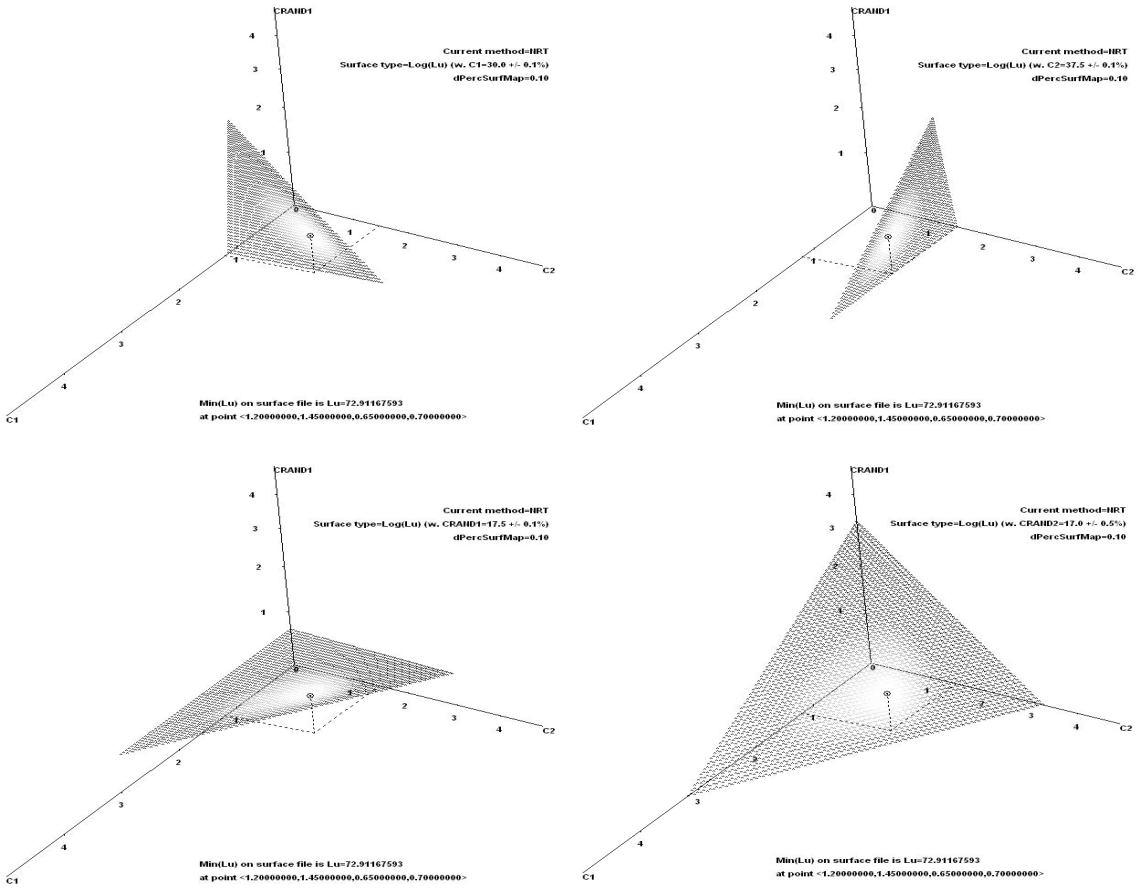
As with the 2D graphs, rerunning the analysis with the dConvCrit≤0 setting allows us to explore the surface more interactively and confirm that the gradient points "uphill" (away from the central region of smallest L_U). This is illustrated on the four-variable type C example data set in Figure 20. The points in this figure are restricted to those falling within +/- 0.2% of 20% of the log loss function range. This type of graph is best viewed as shown, with the symbol coding described earlier.

Figure 20. Exploring the L_U surface of the four-variable type C example data set



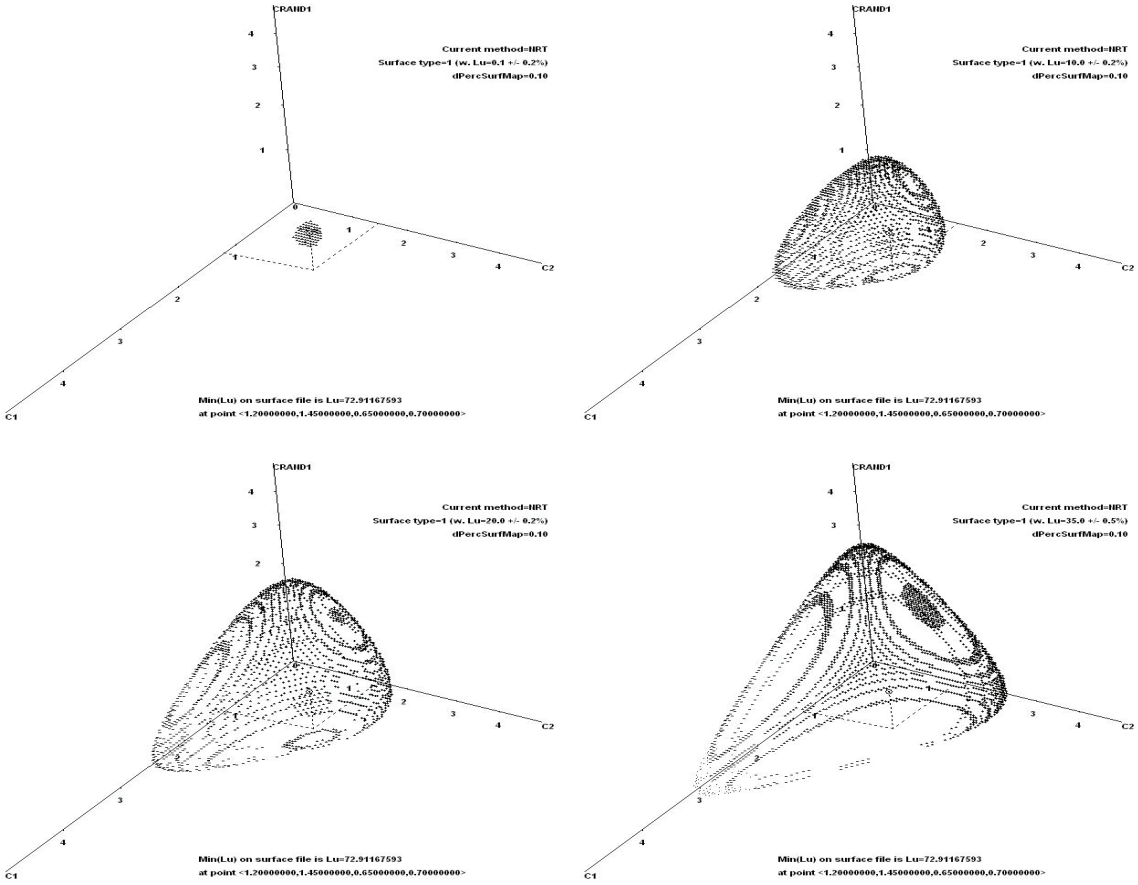
As with the 2D graphs, we can use the left or right arrows to toggle through various other views of the surface. One additional available view for 3D graphs is slices in any of four planes that can be moved through the region. This is illustrated in Figure 21. Here we prefer to turn off symbol coding, since the points fall predictably on a plane and hence the different symbols add no additional information, and may detract from the intensity mapping by L_U. Depending on the viewpoint, thinner or thicker slices may be more revealing of the surface. In the first three graphs the slices are 0.2% of the variable's range thick. In the last graph, the slice is 1% of the variable's range thick.

Figure 21. Slices of the 3D surface map of the four-variable type C example data set in four planes



As with the 2D graphs, another available view is a restricted uniform intensity surface map in which only log loss function values at or close to a specified value are plotted, and that specified value can be changed. This allows one to view the surface map as a series of concentric shells. This is illustrated on the 3D graphs in Figure 22. For optimal visualization, the first three plots are restricted to their smaller specified values +/- 0.2% while the fourth is restricted to a larger specified value +/- 0.5%, in order to provide a heavier mass of points to better fill in the larger space.

Figure 22. Restricted uniform intensity 3D surface maps of the four-variable type C example data set



CHAPTER 4: ADDITIONAL EXPLORATORY ANALYSES OF ARTIFICIAL, CLUSTERED DATA

With our theoretical framework and software in place, we can perform some additional exploratory analyses of the VWUO-MD method. In this chapter we will compare the ultrametric loss function surfaces based on VWUO-MD (L_U) versus De Soete (L_{DS}), and illustrate how the penalty for degenerate solutions has been improved in VWUO-MD. We will examine the L_U surfaces in 2, 3 and 4 variables of each type, as well as mixed-type data. In so doing, we will analyze several artificial, clustered data sets with a variety of clustering patterns, as well as an artificial type C data set that De Soete analyzed in his 1986 paper.[18] Finally, we will perform a Monte Carlo simulation to assess the performance of the U-statistic-based and bootstrap covariance matrix estimators.

4.1 The improved penalty for degenerate solutions

Recall the difference between L_U and L_{DS}:

$$L_{DS}\left(w_1,...,w_p\right)=\frac{\sum_{\Omega}\left(d_{ik}-d_{jk}\right)^2}{\sum\sum_{i<j}d_{ij}^2} \quad \text{(De Soete)}$$

$$L_U\left(w_1,...,w_p\right)=\frac{\sum_{\Omega}\left(d_{ik}-d_{jk}\right)^2}{\left(\prod_{l=1}^{p}w_l\right)^{2/3}} \quad \text{(VWUO-MD)}$$

Recall that according to De Soete, "The denominator in [L_{DS}] is necessary to prevent degenerate solutions where one weight is [*p*] and the others zero."[18] We explained earlier why this does not work, and how the denominator in L_U better accomplishes this goal. Here we will demonstrate this. Table 3 contains the example data set analyzed by De Soete. We have reversed the order of the columns for reasons that will be made clear momentarily.

Table 3. De Soete's example type C data set with column order reversed

C4	C3	C2	C1
-0.0188	0.0564	0.0000	0.4082
0.8879	0.7104	0.0000	0.4082
0.4931	-0.5435	0.0000	0.4082
-0.6123	-0.0227	0.0000	0.4082
0.9475	0.6128	0.3536	-0.2041
-0.7604	-0.7937	0.3536	-0.2041
-0.0368	-0.2072	0.3536	-0.2041
0.1197	0.3818	0.3536	-0.2041
0.3362	0.9152	-0.3536	-0.2041
-0.9367	-0.6031	-0.3536	-0.2041
0.2143	0.4861	-0.3536	-0.2041
-0.0060	-0.3770	-0.3536	-0.2041

Recall that **w** is not estimated directly, but rather **v** is, with:

$$w_l = \frac{p v_l^2}{1 + \sum_{i=1}^{p-1} v_i^2}, \; l=1,\ldots,p\text{-}1$$

$$w_p = p - \sum_{i=1}^{P-1} w_i$$

This transformation actually precludes a 0 weight in the last column, because:

$$w_p = 0 \rightarrow p = \sum_{i=1}^{P-1} w_i \rightarrow p = \frac{\sum_{i=1}^{P-1} pv_i^2}{1 + \sum_{i=1}^{P-1} v_i^2} \rightarrow 1 = \frac{\sum_{i=1}^{P-1} v_i^2}{1 + \sum_{i=1}^{P-1} v_i^2} \rightarrow 1 + \sum_{i=1}^{P-1} pv_i^2 = \sum_{i=1}^{P-1} pv_i^2 \rightarrow 1 = 0$$

This can also be seen by noting that w_p=0 requires that the sum of the v_l is infinity, since:

$$w_p = \frac{p}{1 + \sum_{l=1}^{P-1} v_l}$$

That is why we reversed the order of the columns in the De Soete data set, because it will be shown that contrary to De Soete's findings, the true solution minimizing L_{DS} on these data in fact contains as many as three 0s. To see this, recall that Ω is the set of all triples of objects, and that d_{ik} and d_{jk} are, without loss of generality, the two longest sides of the triangle of distances between objects *i*, *j* and *k*. On any data set perfectly satisfying the ultrametric inequality, the numerator in both L_{DS} and L_U is 0, because by the definition of ultrametricity, d_{ik} and d_{jk} are equal in all triples and so their difference is always 0. This renders both VWUO-MD and the method of De Soete useless in the sense that there are no variable weights that can improve ultrametricity when it is already perfectly satisfied in unweighted data. However, what about situations where ultrametricity is perfectly satisfied only on a subspace of one or more variables in a wider data set? In such cases—which it turns out includes De Soete's 1986 data—the true minimum L_{DS} is 0 which occurs with any solution that *only* weights *ultrametric* subspaces of one or more variables with non-zero weights, in such a way as to preserve the ultrametricity of those variables. (Very extreme weights for the

variables in an ultrametric subspace may actually upset that property). VWUO-MD would not produce such a solution however, because while the numerator in L_U would equal 0, so would the denominator.

In the De Soete data, variables C1 and C2 form an ultrametric subspace, as does C1 alone. Ultrametricity is satisfied on a single continuous variable if and only if there are one or two distinct values in the data. This can be seen by realizing that on a 1-dimensional line, for the two longest distances between any triple of points to be equal, at least two of the points must be the same. This only happens for all possible triples of points if there are at most two distinct values in the data on that dimension. The penalty in the denominator of L_{DS} does preclude degenerate solutions that assign positive weight to any lone dimension that has only one value because of division of 0 by 0. But ironically, in the De Soete data, C1 happens to have exactly two distinct values, and so one solution minimizing L_{DS} is the degenerate solution assigning positive weight to C1 only. We calculated L_{DS} on the De Soete data set for all combinations of weights between 0 and 4 by 0.25 (with the last column C1 receiving a positive weight only), and all the records that received 0 weights for both C3 and C4 are listed in Table 4. It happens that all the records producing L_{DS}=0 are found in this table, along with a few others that received 0 weights for C3 and C4 but were weighted too extremely in C1 and C2 to preserve the ultrametricity of the <C1, C2> subspace (L_{DS}>0 on those records). We can see in this table that the degenerate solution with w_{C1}=4 and all other weights 0 is amongst those that produce L_{DS}=0. Other solutions that should be considered degenerate for HG (for reasons we

discussed earlier; with one or more variables weighted 0) also produce L_{DS}=0. In addition, there is no unique solution producing L_{DS}=0.

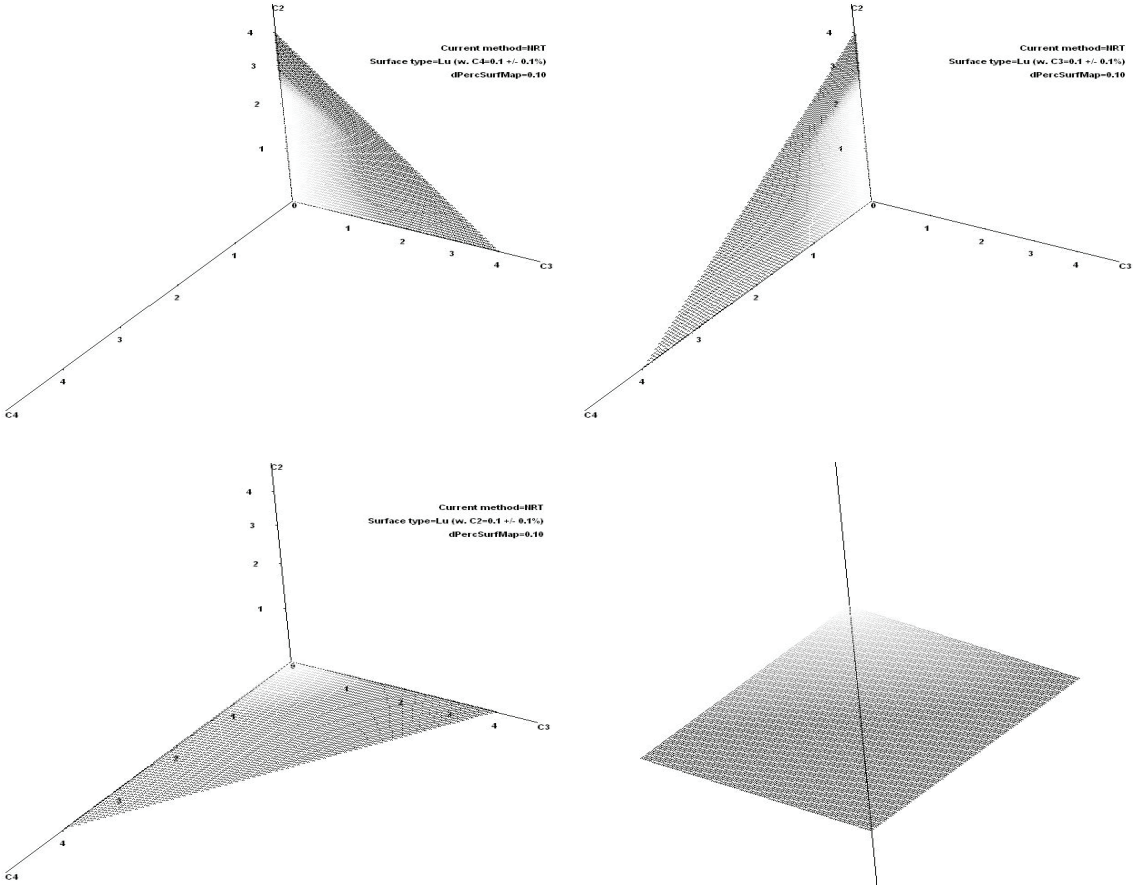
Table 4. Variable weight vectors with w_{C3}=0 and w_{C4}=0 on De Soete's example type C data set with column order reversed

L_{DS}	w_{C4}	w_{C3}	w_{C2}	w_{C1}
0.46825420	0	0	3.75	0.25
0.31201140	0	0	3.50	0.50
0.19172449	0	0	3.25	0.75
0.10303034	0	0	3.00	1.00
0.04283995	0	0	2.75	1.25
0.00902386	0	0	2.50	1.50
0.00000000	0	0	2.25	1.75
0.00000000	0	0	2.00	2.00
0.00000000	0	0	1.75	2.25
0.00000000	0	0	1.50	2.50
0.00000000	0	0	1.25	2.75
0.00000000	0	0	1.00	3.00
0.00000000	0	0	0.75	3.25
0.00000000	0	0	0.50	3.50
0.00000000	0	0	0.25	3.75
0.00000000	0	0	0.00	4.00

De Soete did not identify this problem for two reasons. First, having the columns ordered as he did (C1, C2, C3, C4) precluded solutions with w_{C4}=0 because of the transformation from **v** to **w**. Secondly, he was using a conjugate gradient method of estimation, which we have found (and he has shown us) can stop on a gradual slope not near a local minimum, if that slope becomes shallow enough and the norm of the gradient becomes small. The solution De Soete found on his data (in the sum-to-p scale, p=4) was w_{C1}=2.2300, w_{C2}=1.7576, w_{C3}=0.0008 and w_{C4}=0.0116. This produced a De Soete loss function of L_{DS}=0.00007505. The L_{DS} surface map is shown in Figure 23, displayed without the log transform due to the presence of 0s. The first three graphs fill the

parameter space on a grid spaced at 0.05. They illustrate that the surface tilts downhill all the way to the edges where the degenerate solutions are located and the loss function equals 0. Although there are numbers in Table 4 above that confirm the degenerate solutions, we also confirm by zooming in the surface maps (the fourth graph) near De Soete's solution. The latter graph was created on a grid with C3 and C4 ranging from 0 to 0.001 by 0.00001, and C2 fixed at 1.75. The fourth graph confirms the degenerate solutions.

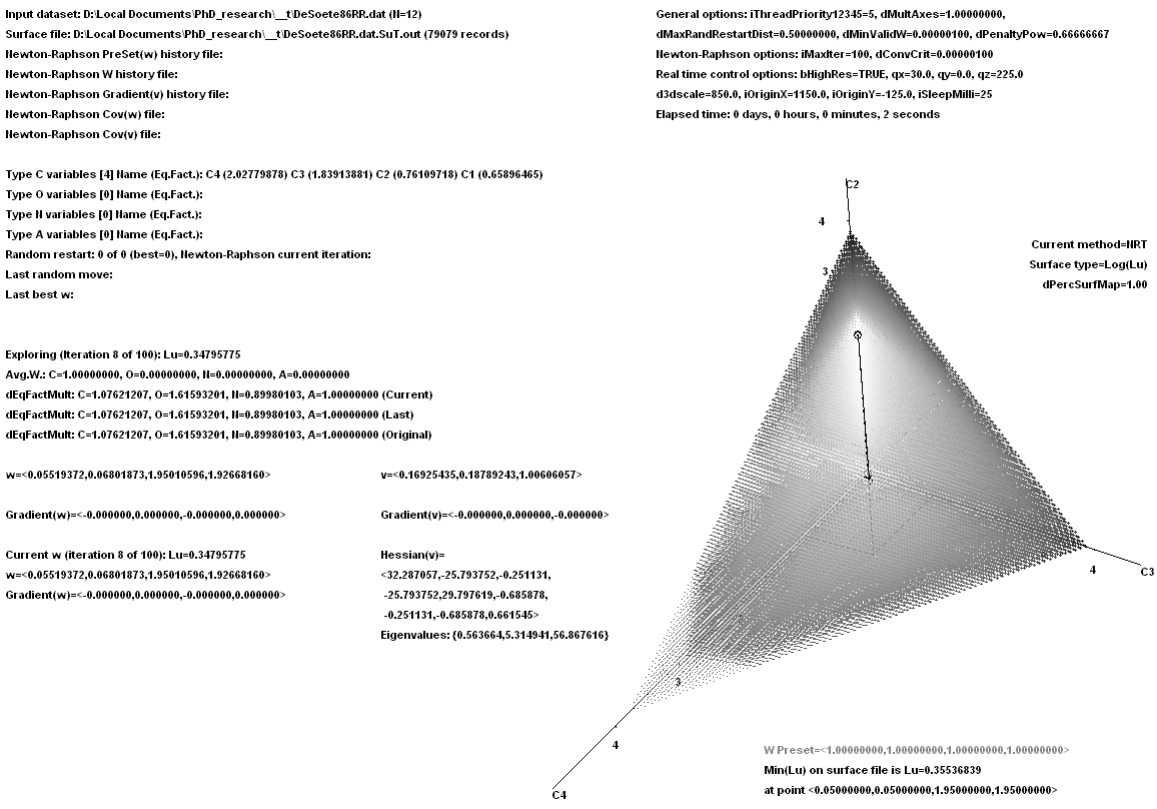
Figure 23. L_{DS} surface map of De Soete's example type C data set with column order reversed



On the other hand, VWUO-MD's penalty in the denominator of L_U, the product of variable weights raised to the power of 2/3, does not allow exactly 0

weights because that would either cause the loss function L_U to be infinity if the numerator were >0, or else render it undefined. To illustrate, we analyze the De Soete data with VWUO-MD. The solution is shown in Figure 24. The VWUO-MD solution is w_{C1}=1.926682, w_{C2}=1.950106, w_{C3}=0.068019 and w_{C4}= 0.055194. The VWUO-MD solution is unique, and captures the relative importance of the variables without resorting to a degenerate solution. At the solution, it was confirmed that $\nabla_{\hat{\mathbf{v}}} = \mathbf{0}$ and Hessian(**v**) had positive eigenvalues.

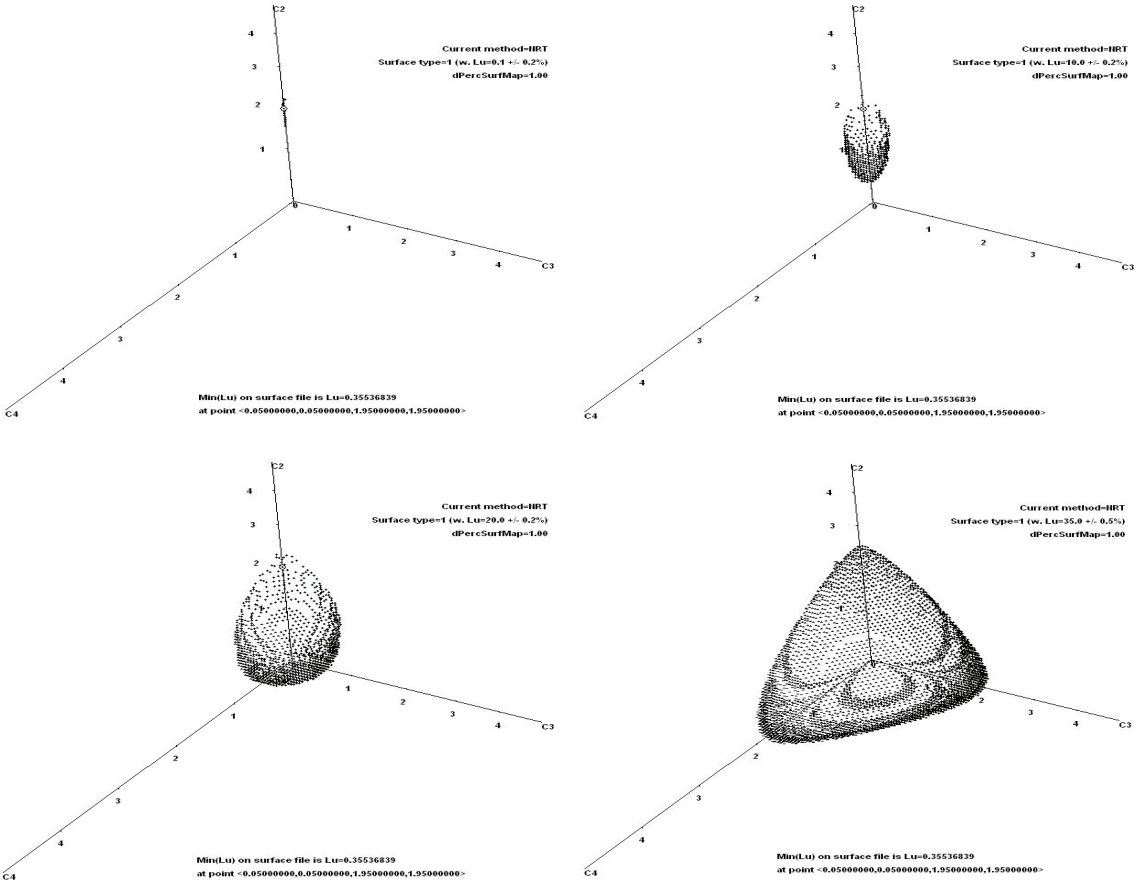
Figure 24. VWUO-MD solution on De Soete's example type C data set with column order reversed

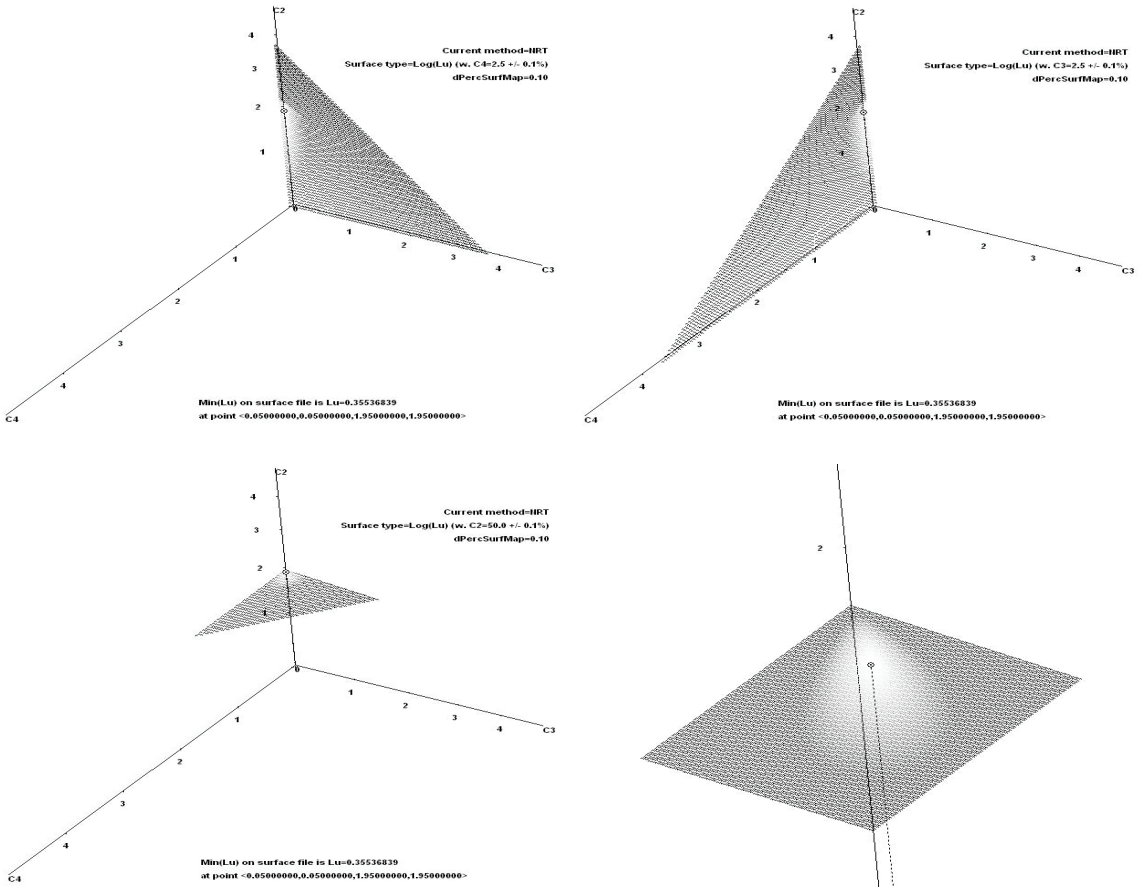


The surface map in the above graph was generated on a 3D grid spaced at 0.05. The observed minimum grid coordinates were w_{C1}=1.95, w_{C2}=1.95, w_{C3}=0.05 and w_{C4}=0.05, consistent with the solution. The L_U surface map of the

De Soete data is shown in Figure 25 as a series of four concentric shells and four slices. As with the De Soete graphs previously, the last slice was zoomed in around the solution, this time to confirm that the true minimum does not occur at a degenerate solution. The last graph was created on a grid with C3 and C4 ranging from 0.01 to 0.2 by 0.002, and C2 fixed at 1.95. These graphs illustrate that the surface tilts downhill away from the edges where the degenerate solutions are located, to bottom out at the unique solution.

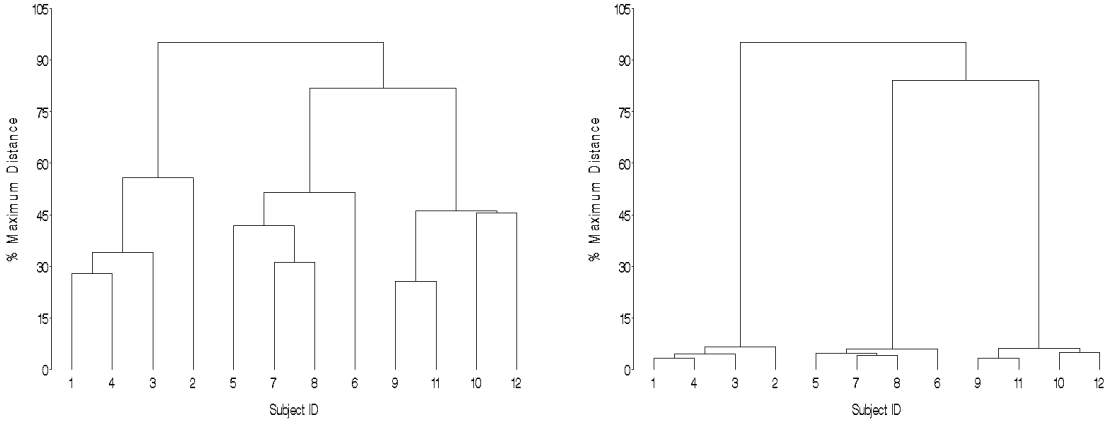
Figure 25. L_U surface map of De Soete's example type C data set with column order reversed





The VWUO-MD solution is informative for the purposes of HG, since variables all related to the clustering are also related to each other. However, do the optimal variable weights also help to enhance the clustering? To find out, we generate dendrograms (single linkage) based on unweighted and variable-weighted distance matrices. These are shown in Figure 26, both graphs showing percentage of maximum distance on the vertical axis, for a scale-free comparison. The three clusters defined by variables C1 and C2 have been dramatically enhanced by down-weighting the random noise variables. Later we will see that this is a much more dramatic result than can be obtained on data with appreciable dispersion about the clustering variables; in De Soete's example type C data set, conditional dispersion about these variables is 0.

Figure 26. Dendrograms (single linkage) on distance matrices from De Soete's example type C data set both unweighted (left) and VWUO-MD variable weighted (right)



4.2 Mixed-type artificial, clustered data

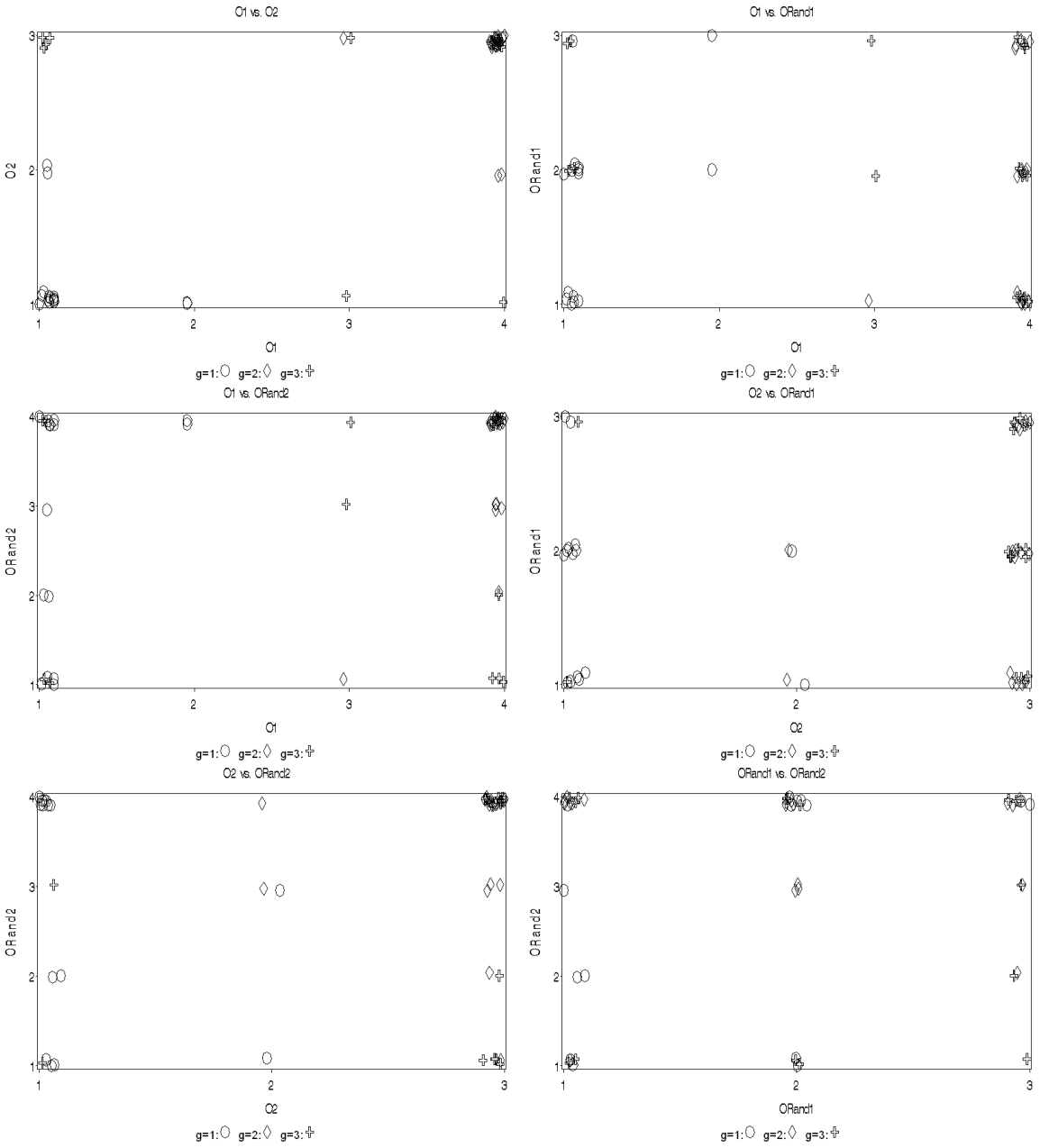
Next we create an artificial clustered data set for purposes of studying the shape of the L_U surface on subspaces of each type, and mixed type. The data were constructed with three clearly distinct clusters according to a pre-assigned three-level group variable *g*. The data set has n=50 records, 15 records with *g*=1, 18 records with *g*=2 and 17 records with *g*=3.

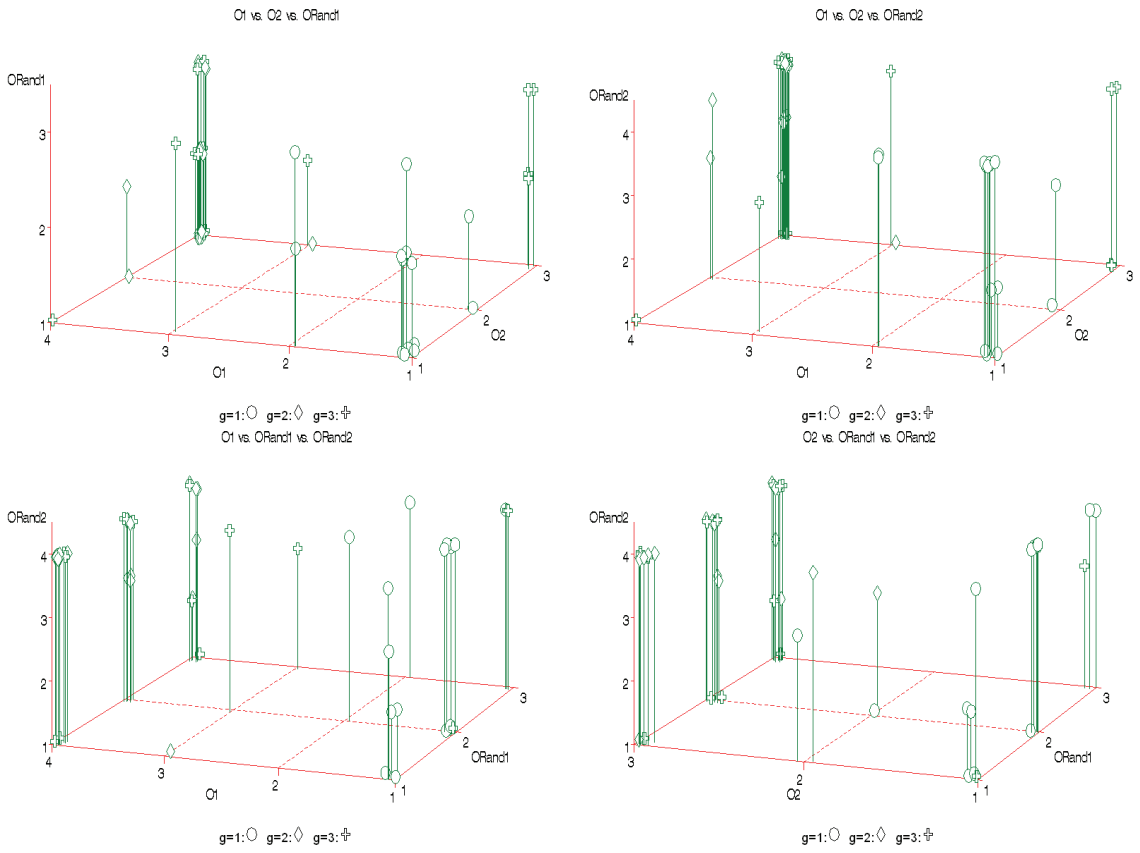
The four type C variables in these data were described in the previous chapter on VWUO.exe.

Ordinal variables O1 and O2 were created from (conditionally on *g*) independent multinomial distributions depending on *g*. O1 was selected from a four-level multinomial distribution with probability vector <.9, .1, 0, 0> when *g*=1 and <.05, .05, .05, .85> when *g*=2 or *g*=3. O2 was selected from a three-level multinomial distribution with probability vector <.9, .1, 0> when *g*=1 and <.05, .05, .9> when *g*=2 or *g*=3. ORand1 was created from a three-level multinomial distribution independent of *g*, with probability vector <.33, .34, .33>. ORand2 was

created from a four-level multinomial distribution independent of *g*, with probability vector <.1, .1, .1, .7>. The four ordinal variables are plotted against each other in Figure 27, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

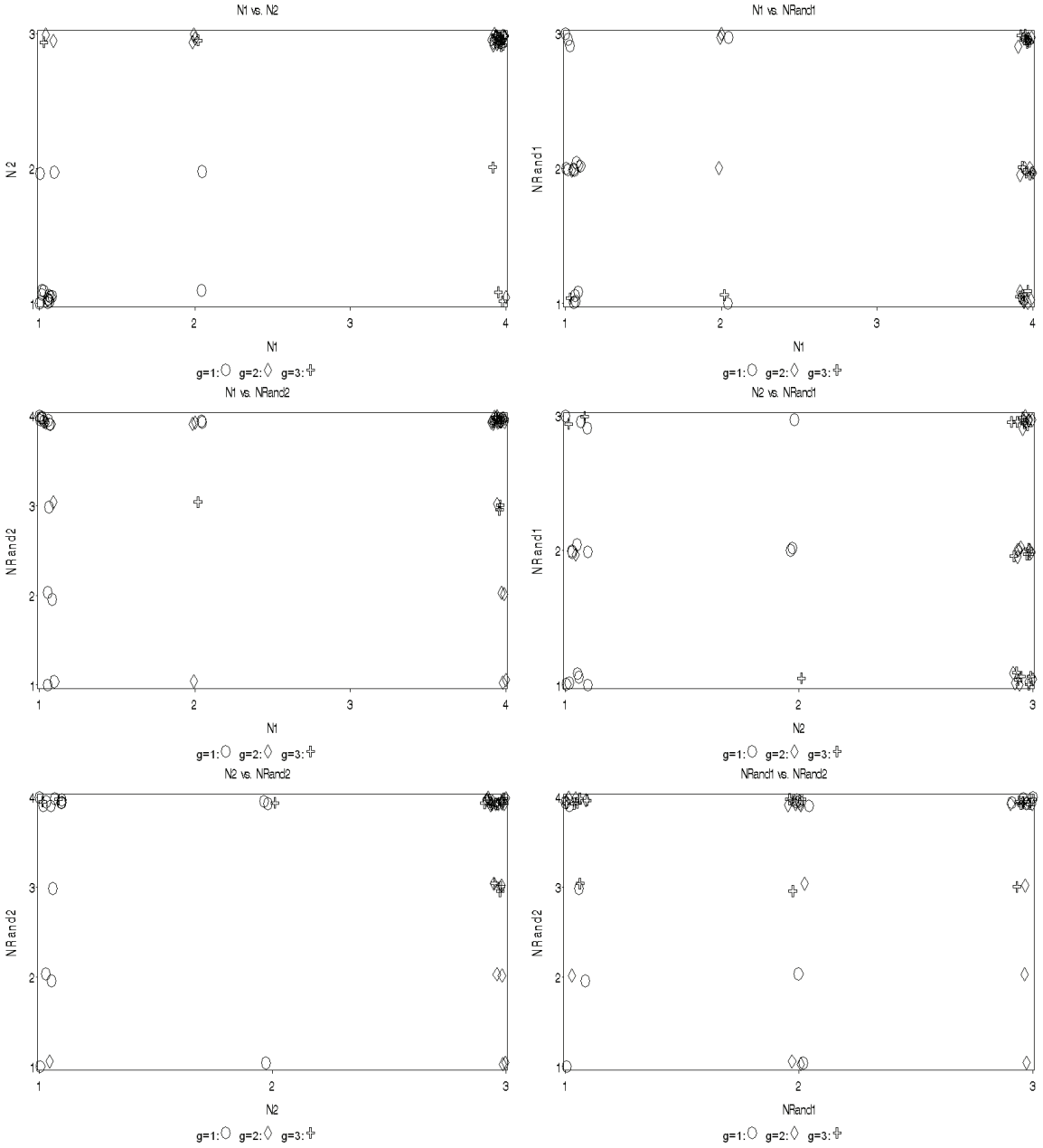
Figure 27. Ordinal variables in the mixed-type artificial data set; data are jittered

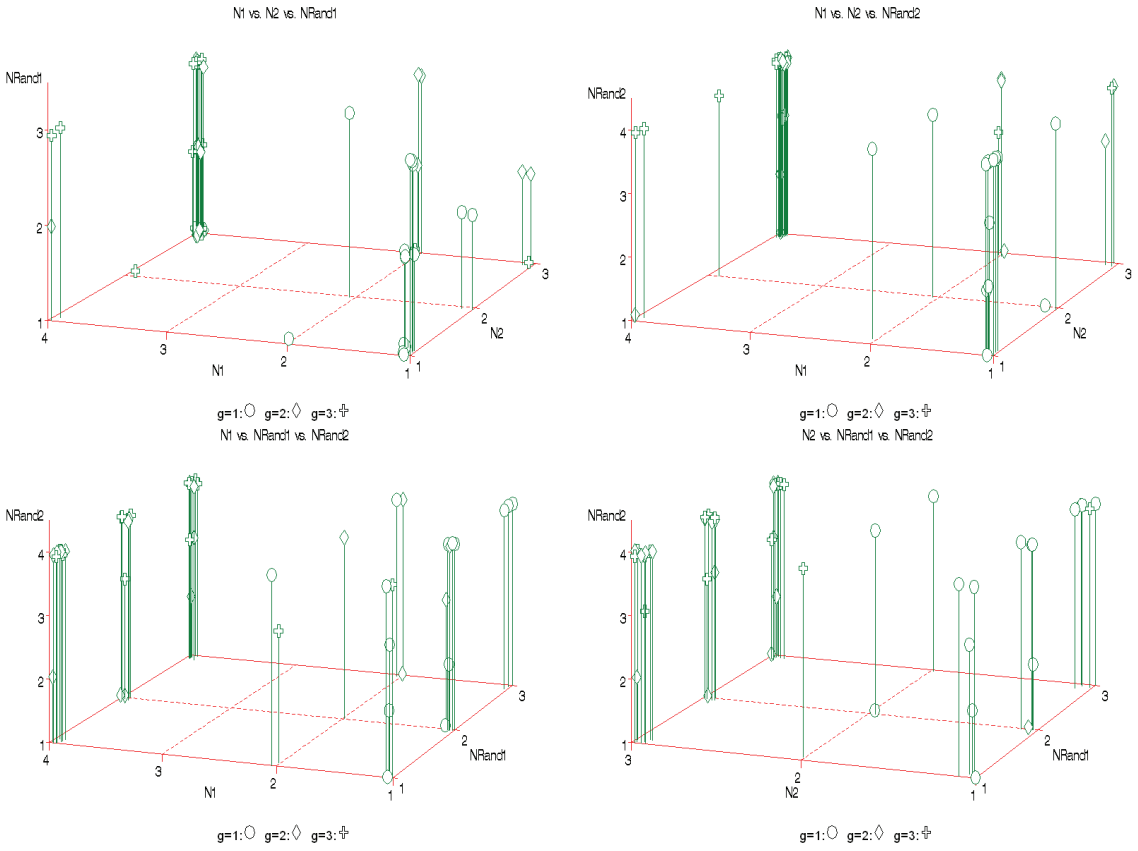




Nominal variables N1 and N2 were created from (conditionally on *g*) independent multinomial distributions depending on *g*. N1 was selected from a four-level multinomial distribution with probability vector <.9, .1, 0, 0> when *g*=1 and <.05, .05, .05, .85> when *g*=2 or *g*=3. N2 was selected from a three-level multinomial distribution with probability vector <.9, .1, 0> when *g*=1 and <.05, .05, .9> when *g*=2 or *g*=3. NRand1 was created from a three-level multinomial distribution independent of *g*, with probability vector <.33, .34, .33>. NRand2 was created from a four-level multinomial distribution independent of *g*, with probability vector <.1, .1, .1, .7>. The four nominal variables are plotted against each other in Figure 28, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

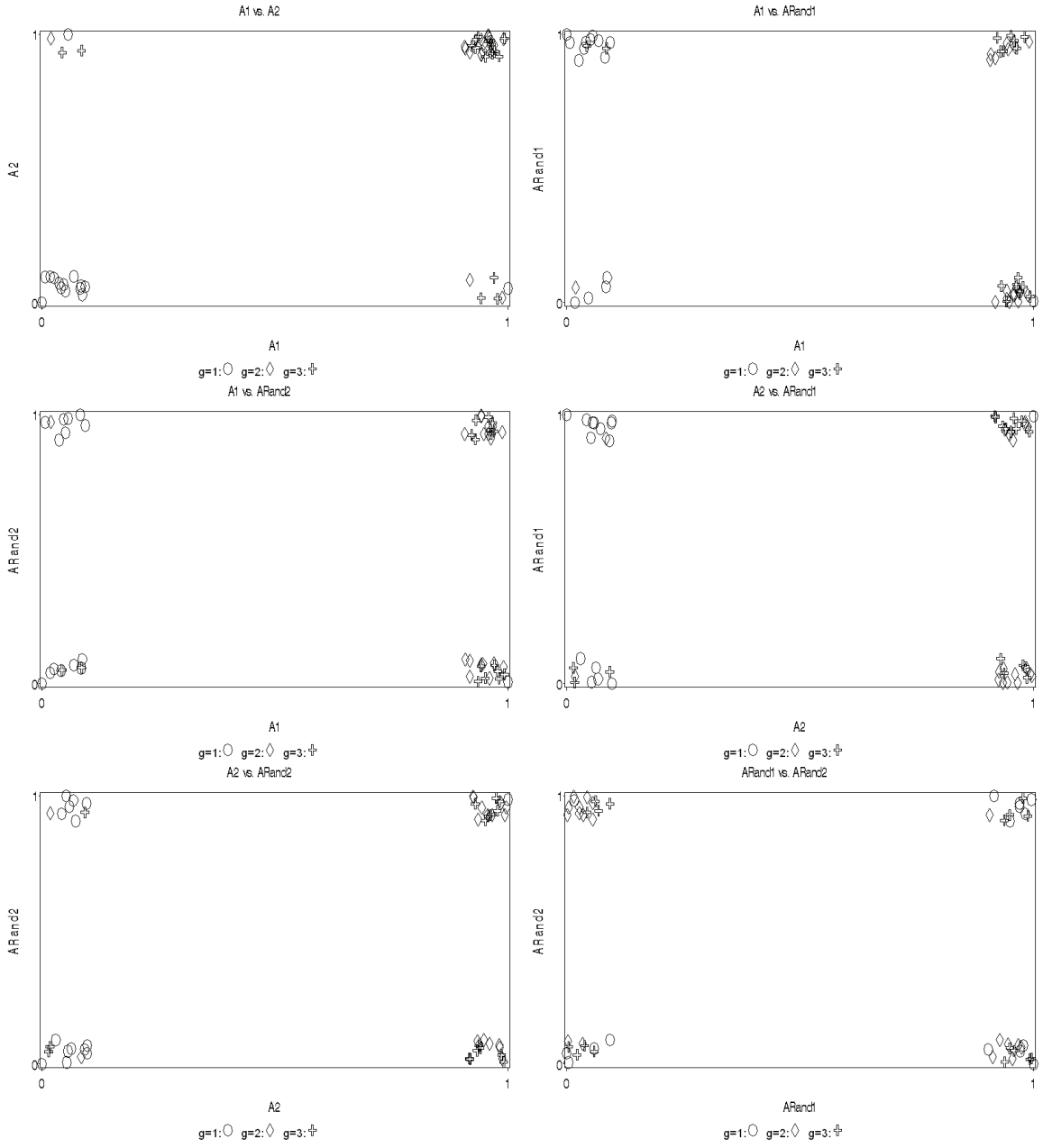
Figure 28. Nominal variables in the mixed-type artificial data set; data are jittered

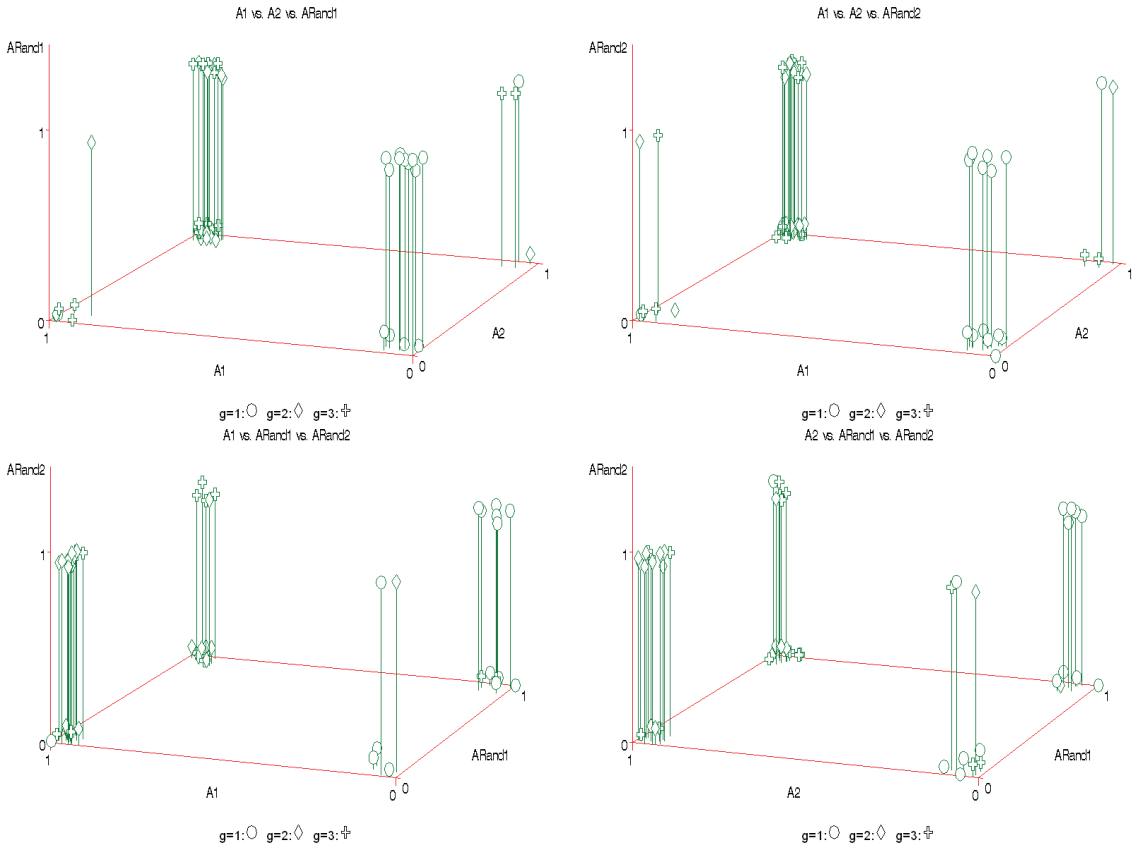




Binary asymmetric variables A1 and A2 were created from (conditionally on *g*) independent Bernoulli distributions depending on *g*. A1 was selected from a Bernoulli distribution with P(A1=1)=.1 when *g*=1, and P(A1=1)=.8 when *g*=2 or *g*=3. A2 was selected from a Bernoulli distribution with P(A2=1)=.1 when *g*=1, and P(A2=1)=.8 when *g*=2 or *g*=3. ARand1 was created from a Bernoulli distribution independent of *g*, with P(ARand1=1)=.5. ARand2 was created from a Bernoulli distribution independent of *g*, with P(ARand2=1)=.5. The four binary asymmetric variables are plotted against each other in Figure 29, with symbols indicating each point's value of *g*. Plots are randomly jittered for improved visualization.

Figure 29. Binary asymmetric variables in the mixed-type artificial data set; data are jittered





We create L_U surface maps, and are ready to analyze these data.

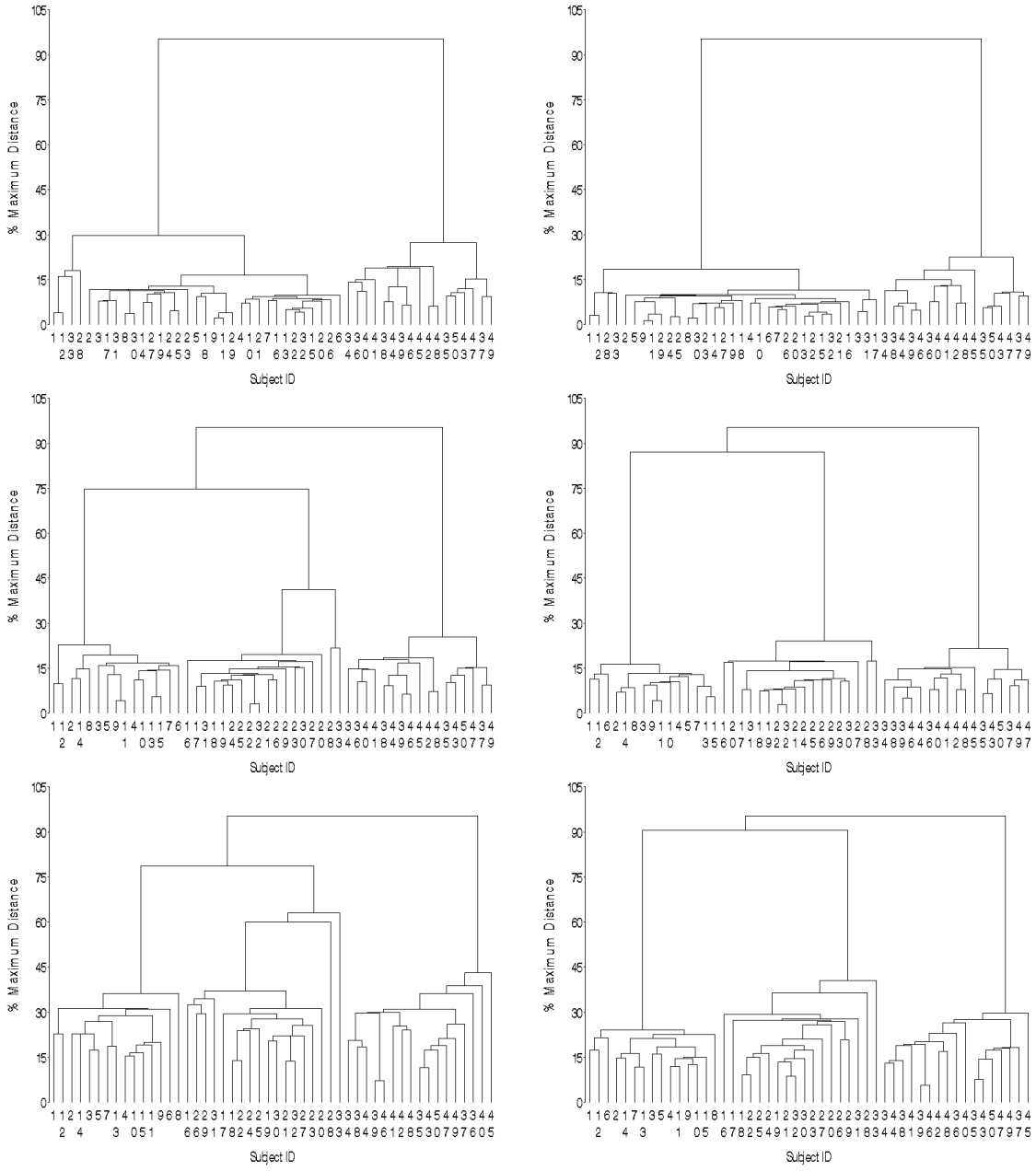
4.3 Point estimation

4.3.1 Analysis of type C variables

We analyzed the type C subspaces in the previous chapter on VWUO.exe. There we found solutions that weighted C1 and C2 more heavily than CRand1 or CRand2, which was sensible and informative considering how the artificial data were constructed, i.e., that C1 and C2 together defined clusters in the data, while CRand1 and CRand2 did not. The 1D, 2D and 3D surface maps showed that there was only one local minimum per map (the grand minimum).

The relative weights are informative about the clustering present in the data. That alone is useful for HG, our primary focus for VWUO-MD. However, do they also help to enhance the clustering? To find out, we create dendrograms (single linkage) on two-, three- and four-variable distance matrices both unweighted and variable weighted with the solutions obtained earlier. Figure 30 contains all six dendrograms. Possibly because at least half the variables in each analysis are related to the clusters, the unweighted dendrograms actually show three clusters fairly clearly in the three- and four-variable examples, and two clusters in the two-variable example (C1 versus CRand1). In addition, we can note that compared to De Soete's data, the clusters are less clearly defined—in the case of De Soete's data, cluster-specific values on variables C1 and C2 had zero dispersion about their conditional means. In our case, the variable weighted dendrograms do appear to enhance the structure somewhat, showing slightly stronger evidence of three clusters as seen in the longer vertical lines in the main vertical span of each graph. It is certainly not as dramatic as we saw with De Soete's data, however.

Figure 30. Dendrograms (single linkage) on two- (top row), three- (middle row) and four-variable (bottom row) type C distance matrices both unweighted (left) and variable weighted (right) with the VWUO-MD solutions obtained earlier

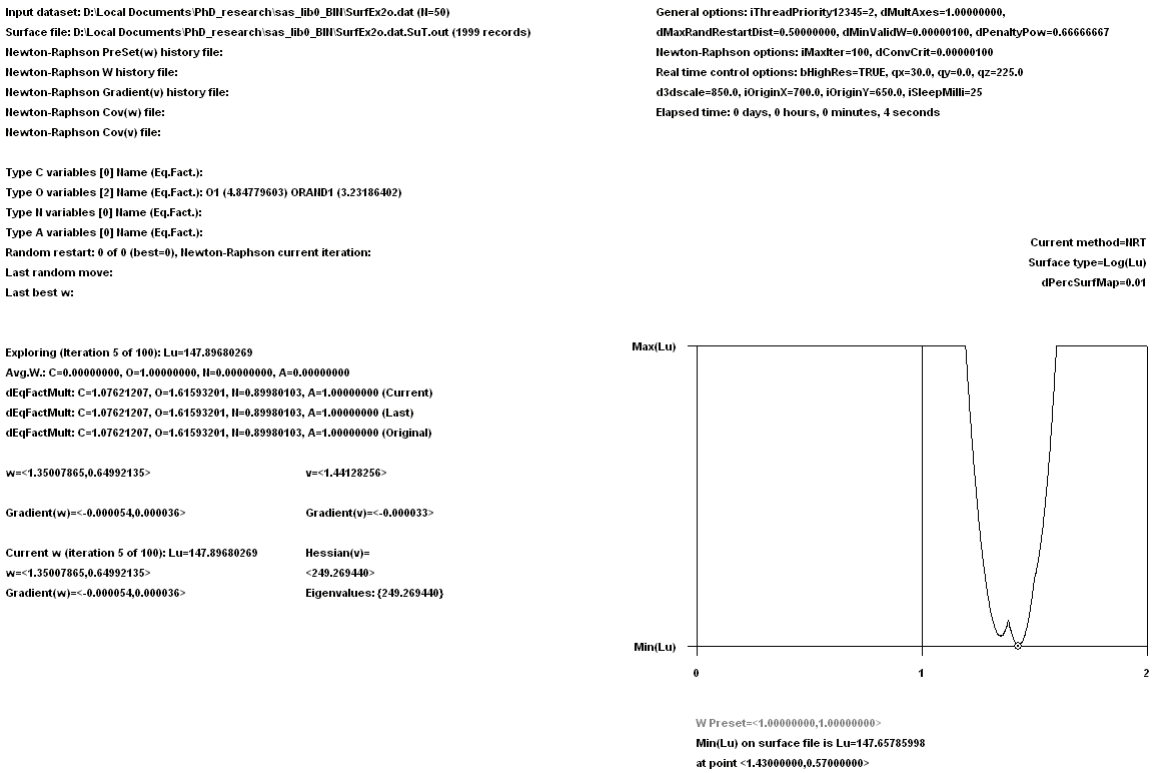


4.3.2 Analysis of type O variables

Next we will perform VWUO-MD analyses of the type O variables. We will analyze the two-variable subspace <O1, ORand1> first. The default solution starting from **w=1** is shown overtop the L_U surface map in Figure 31. The default

solution is **w**=<1.350079, 0.649921> (at which L_U=147.89680269), which is sensible (and informative for HG) considering how the artificial data were constructed, i.e., that O1 and O2 together defined clusters in the data, while ORand1 and ORand2 did not. However this does not correspond with the reported grand minimum on the surface map. We will address this issue later. The surface map was plotted restricting the range to the bottom 1% of the log(L_U) range so that both local minima were visible. To confirm the existence of the second local minimum numerically, we also restarted from **w**=<1.5, 0.5>, and the procedure converged on the grand minimum from the right. The solution is **w**=<1.429902, 0.570098> (at which L_U=147.65785688). At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ =**0** and Hessian(**v**) had positive eigenvalues.

Figure 31. Default VWUO-MD solution to two-variable type O subspace in the mixed-type artificial data set, overtop the L_U surface map



Next we analyze the three-variable subspace **w**=<O1, O2, ORand1>. The default solution starting from **w=1** is shown overtop the L_U surface map in Figure 32. The default solution is **w**=<1.403273, 0.926614, 0.670113>. This does not correspond with the reported grand minimum, and there is an indication of multiple local minima on the surface map. We tried different starting vectors near different apparent depressions on the surface, and thus found four additional local minima. The five solutions are listed in Table 5. There may be more that were not found. All solutions are sensible (and informative for HG) considering how the artificial data were constructed, i.e., that O1 and O2 together defined clusters in the data, while ORand1 and ORand2 did not. Unfortunately, the default solution ranked only fourth out of five ordered by L_U. We will address this

issue later. At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues.

Figure 32. Default VWUO-MD solution to three-variable type O subspace in the mixed-type artificial data set, overtop the L_U surface map

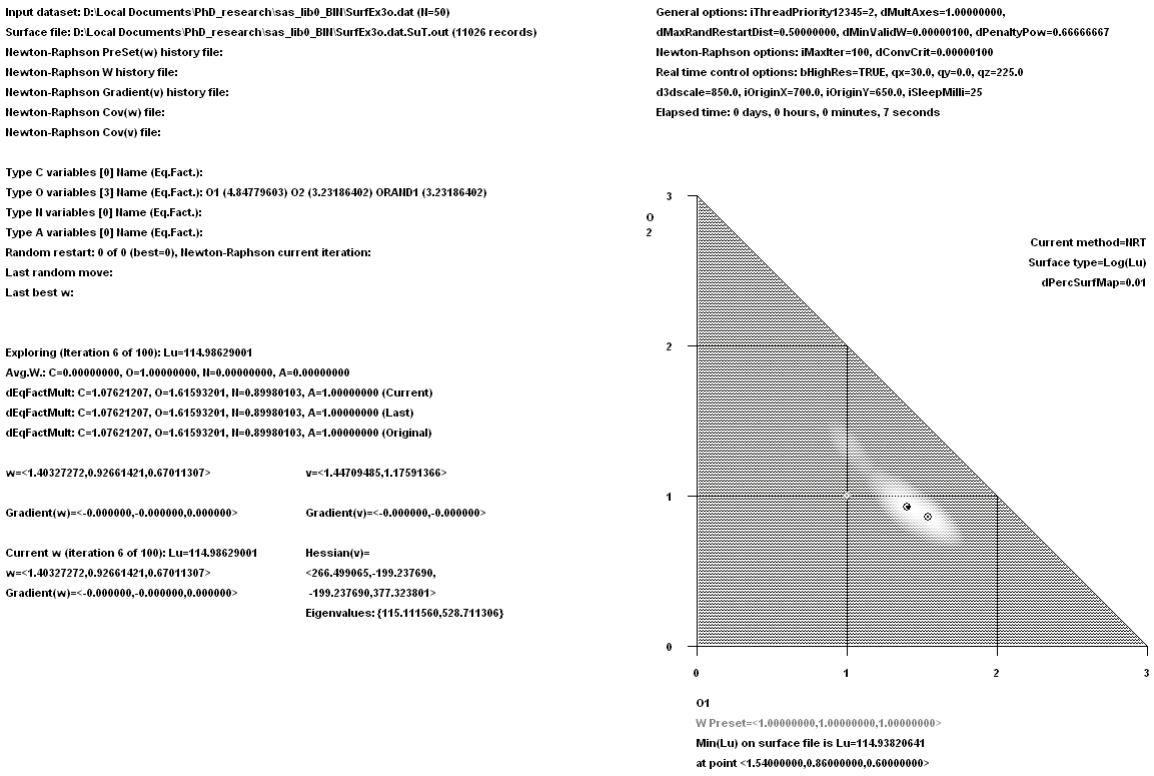


Table 5. Four VWUO-MD solutions to three-variable type O subspace in the mixed-type artificial data set, sorted by L_U; the default solution is in italics

L_U	w_{O1}	w_{O2}	w_{ORand1}
114.93699389	1.441463	0.903499	0.655038
114.93756410	1.462117	0.902288	0.635595
114.93769390	1.539052	0.859347	0.601601
114.98629001	*1.403273*	*0.926614*	*0.670113*
118.88242933	1.012039	1.335304	0.652657

Next we analyze the four-variable subspace **w=**<O1, O2, ORand1, ORand2>. The default solution starting from **w=1** is shown overtop the L_U surface

map in Figure 33. The default solution is **w**=<1.277328, 1.106676, 0.757507, 0.858489>. Investigation with higher resolution surface maps around the first solution (the bottom two graphs) reveals that this does not correspond with the grand minimum. Starting at a variety of locations in this vicinity produces two other local minimum. The three solutions are listed in Table 6. There may be more that were not found. All solutions are sensible (and informative for HG) considering how the artificial data were constructed, i.e., that O1 and O2 together defined clusters in the data, while ORand1 and ORand2 did not. Unfortunately, the default solution ranked last out of the three ordered by L_U. We will address this issue later. At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues.

Figure 33. Selected VWUO-MD solutions to four-variable type O subspace in the mixed-type artificial data set, overtop the L_U surface map

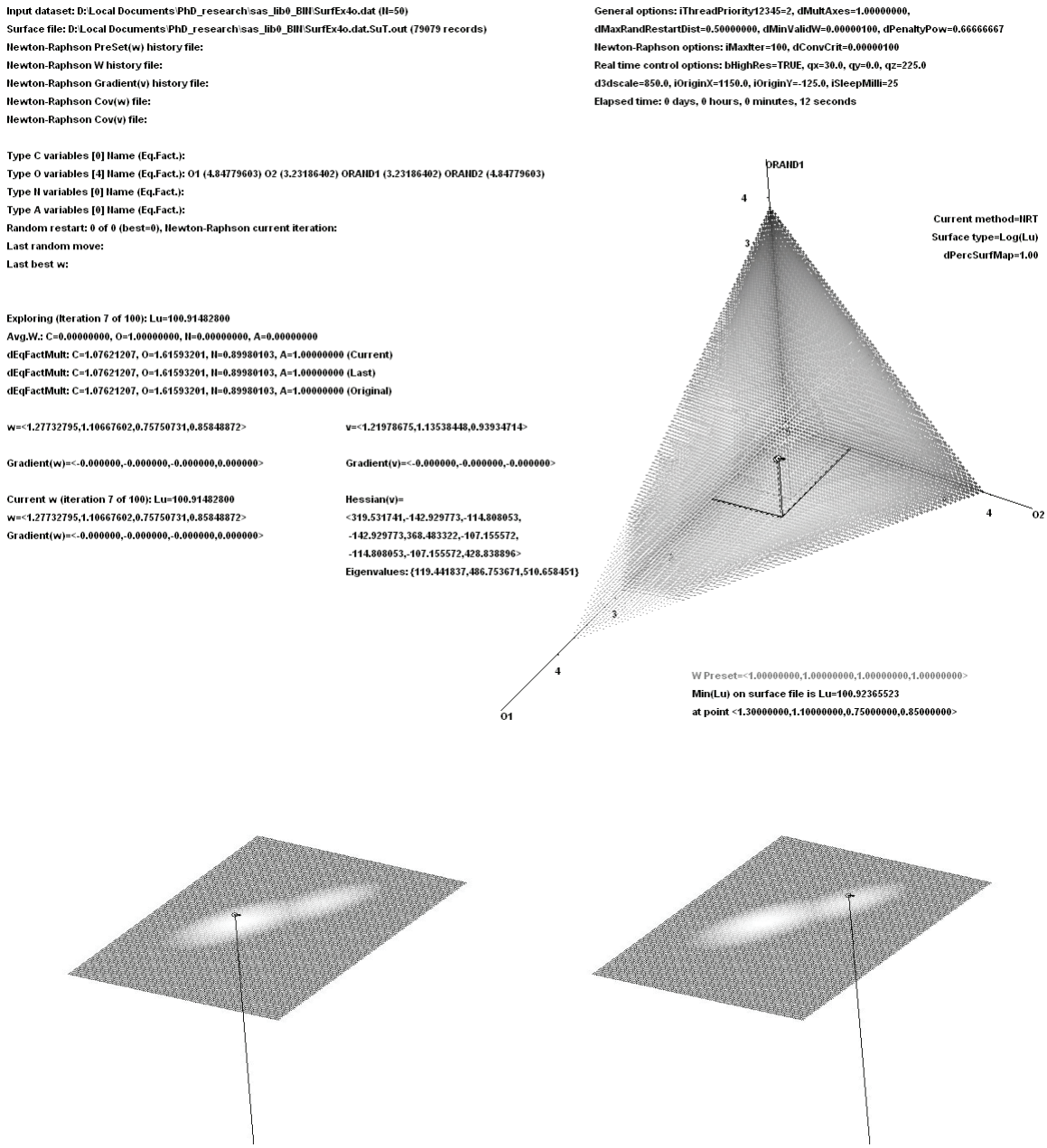
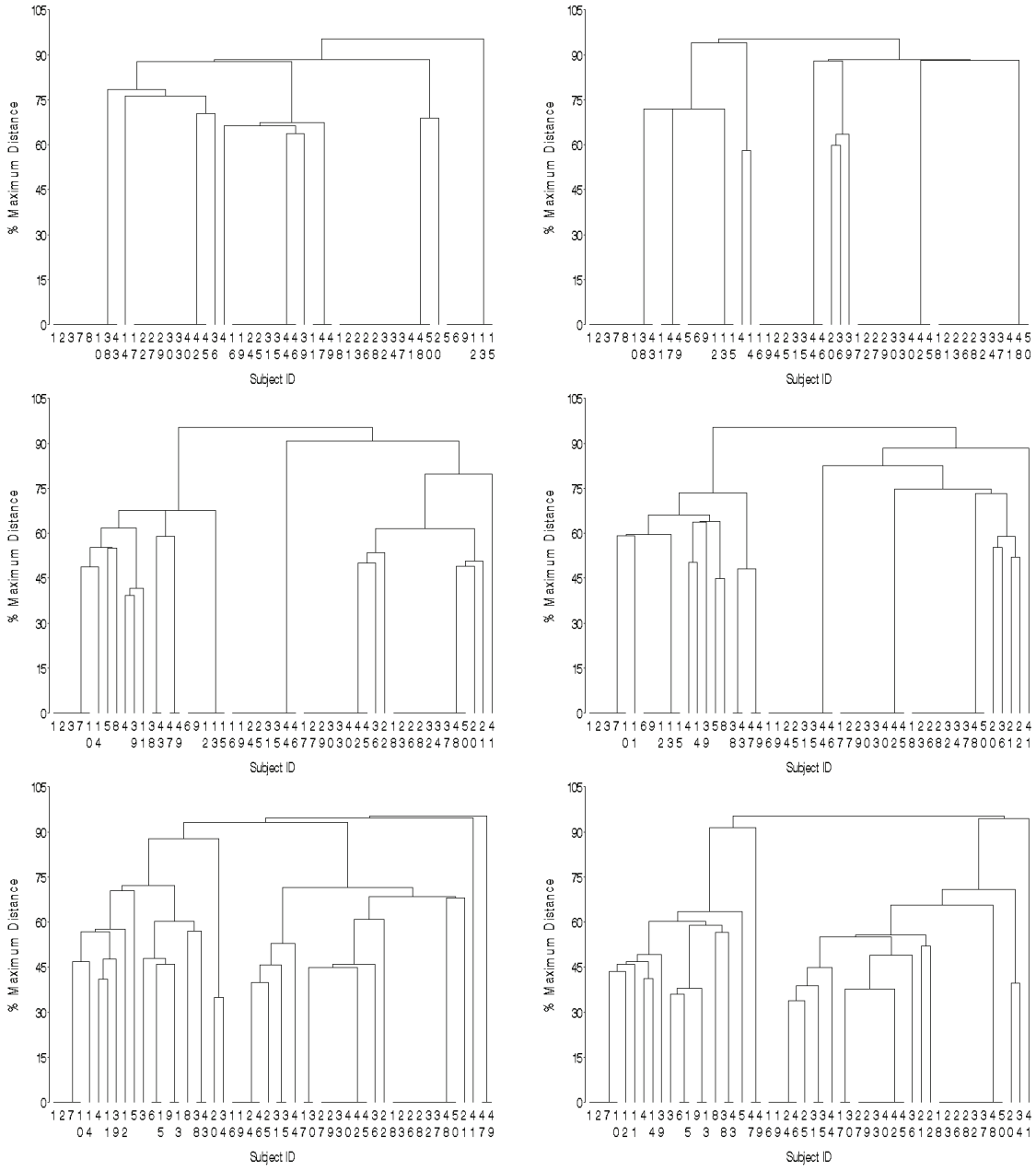


Table 6. Three VWUO-MD solutions to four-variable type O subspace in the mixed-type artificial data set, sorted by L_U; the default solution is in italics

L_U	w_{O1}	w_{O2}	w_{ORand1}	w_{ORand2}
100.90897452	1.299101	1.083801	0.757414	0.859685
100.90924067	1.300458	1.087894	0.754530	0.857118
100.91482800	*1.277328*	*1.106676*	*0.757507*	*0.858489*

The relative weights in the grand minimum solutions are informative about the clustering present in the data. That is useful for HG. However, do they also help to enhance the clustering? To find out, we create dendrograms (single linkage) on two-, three- and four-variable distance matrices both unweighted and variable weighted with the grand minimum solutions obtained earlier. Figure 34 contains all six dendrograms. None appear to be very informative. It seems that in situations with a small number of ordinal variables, the variable weights may form the most informative part of the solution—they reflect the variables that are related to each other through the clustering that we know exists in the data, and can help to generate hypotheses for additional analyses.

Figure 34. Dendrograms (single linkage) on two- (top row), three- (middle row) and four-variable (bottom row) type O distance matrices both unweighted (left) and variable weighted (right) with the VWUO-MD grand minimum solutions

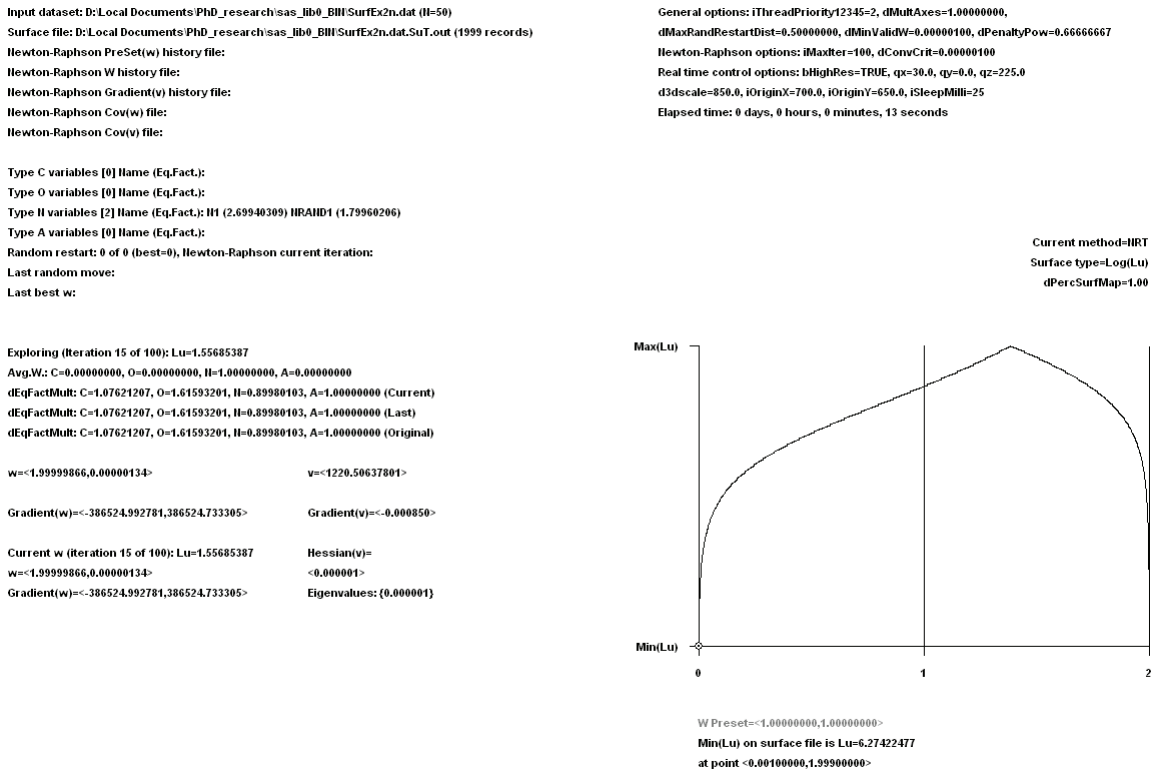


4.3.3 Analysis of type N variables

Next we will perform VWUO-MD analyses of the type N variables. We will analyze the two-variable subspace <N1, NRand1> first. The solution, from the default starting vector **w=1**, is shown overtop the L_U surface map in Figure 35.

The solution is **w**=<1.999999, 0.000001>, a boundary case that only "converged" because of the minimum weight setting of dMinValidW=0.000001. It is clear from the surface map that this function has no local minimum. This indicates a problem with VWUO-MD that can occur when it is used to analyze a set of only two discrete variables (we note however that this did not happen in the analysis of the two-variable type O subspace). We have discovered from this example that it is possible in some (trivial) situations for the numerator in L_U to approach 0 faster than the denominator does, as one or the other variable weight approaches 0. Thus while no variable weight can *equal* 0 per se, in this case there is no local minimum to call the solution. While the penalty function in L_U is an improvement over that in L_{DS}, it is apparently not a panacea. Fortunately, no practical application of VWUO-MD ought to involve only two variables.

Figure 35. VWUO-MD solution to two-variable type N subspace in the mixed-type artificial data set, overtop the L_U surface map



Next we analyze the three-variable subspace **w**=<N1, N2, NRand1>. The default solution starting from **w=1** is shown overtop the L_U surface map in Figure 36. The default solution is **w**=<1.086066, 1.341284, 0.572650>. This does not correspond with the reported grand minimum, and there are multiple local minima visible on the surface map. Trying different starting vectors within the different depressions on the surface produces four additional local minima. The five solutions are listed in Table 7. There may be more that were not found; indeed the graph suggests one or two more, but the algorithm would not converge within the additional apparent depressions. The best four out of five solutions (those with lowest L_U) are sensible (and informative for HG) considering how the artificial data were constructed, i.e., that N1 and N2 together defined clusters in

the data, while NRand1 and NRand2 did not. Unfortunately, the default solution ranked only third out of five ordered by L_U. We will address this issue later. At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues.

Figure 36. Default VWUO-MD solution to three-variable type N subspace in the mixed-type artificial data set, overtop the L_U surface map

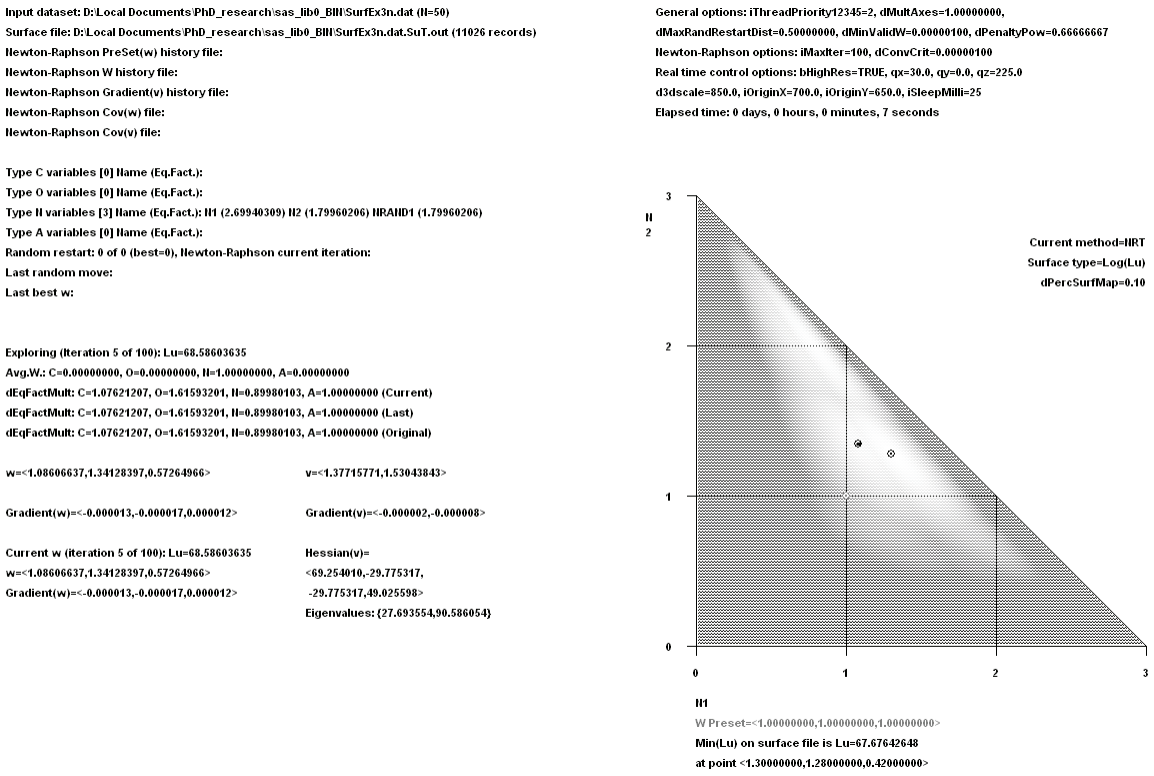
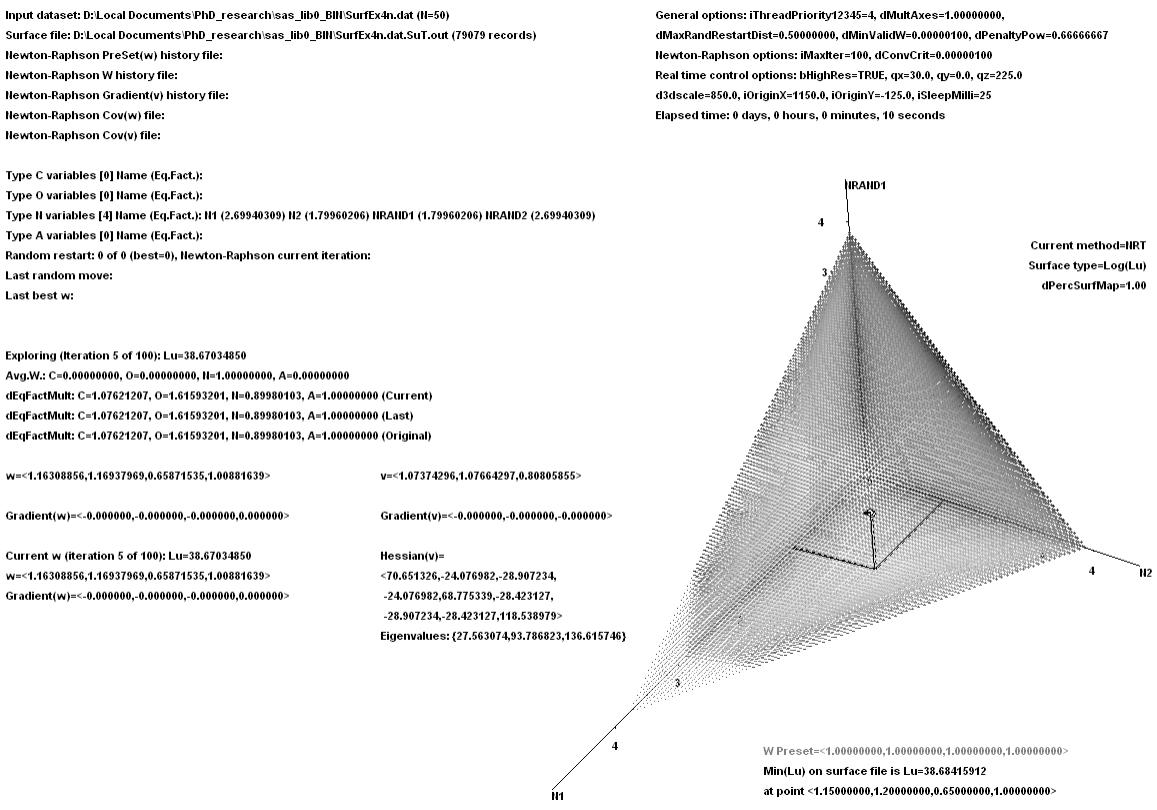


Table 7. Five VWUO-MD solutions to three-variable type N subspace in the mixed-type artificial data set, sorted by L_U; the default solution is in italics

L_U	w_{N1}	w_{N2}	w_{NRand1}
67.67156137	1.307467	1.278827	0.413706
67.95910829	0.778682	2.000000	0.221318
68.58603635	*1.086066*	*1.341284*	*0.572650*
69.89258585	0.676807	2.000000	0.323193
103.57204369	0.613498	0.386502	2.000000

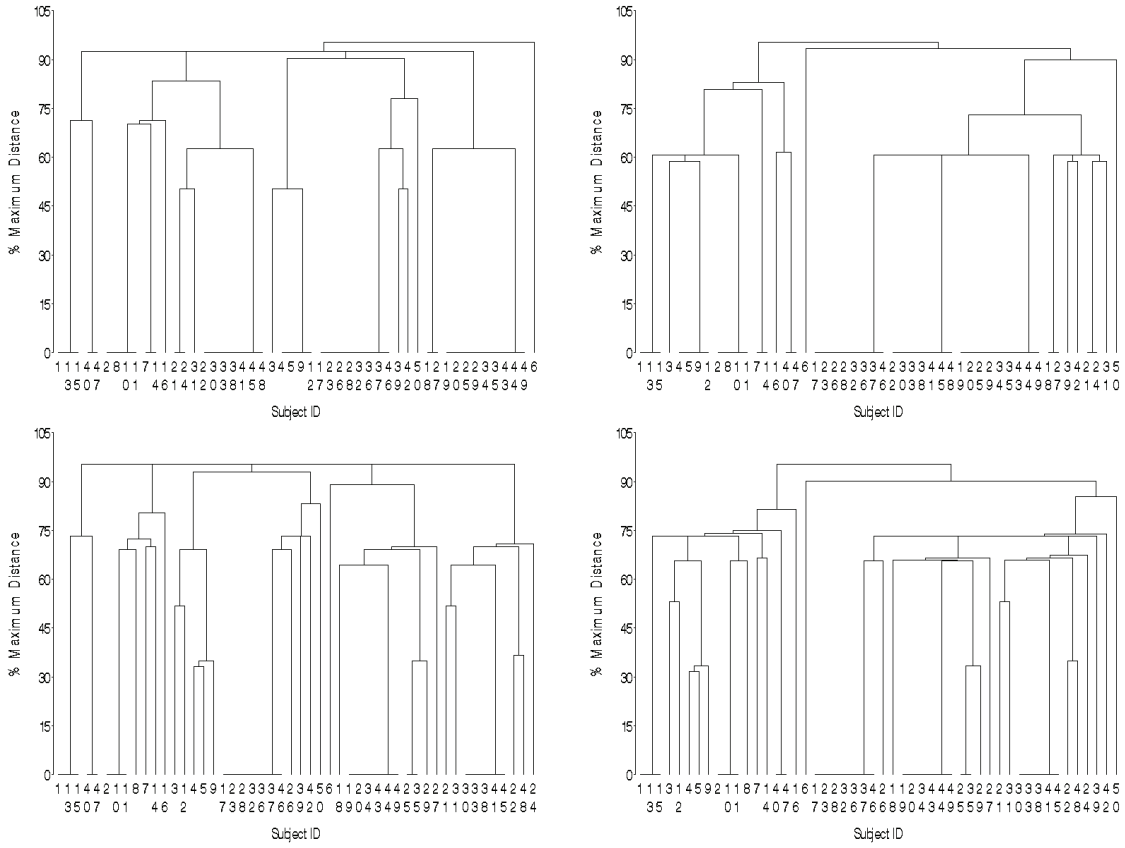
Next we analyze the four-variable subspace **w**=<N1, N2, NRand1, NRand2>. The default solution starting from **w=1** is shown overtop the L_U surface map in Figure 37. The default solution is **w**=<1.163089, 1.169380, 0.658715, 1.008816>. This corresponds with the grand minimum, and investigation with higher resolution surface maps around the first solution confirms that this is the only local minimum. The solution is sensible (and informative for HG) considering how the artificial data were constructed, i.e., that N1 and N2 together defined clusters in the data, while NRand1 and NRand2 did not. At the solution, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ =**0** and Hessian(**v**) had positive eigenvalues.

Figure 37. VWUO-MD solution to four-variable type N subspace in the mixed-type artificial data set, overtop the L_U surface map



The relative weights in the grand minimum solutions are informative about the clustering present in the data. That is useful for HG. However, do they also help to enhance the clustering? To find out, we create dendrograms (single linkage) on three- and four-variable distance matrices both unweighted and variable weighted with the grand minimum solutions obtained earlier. Figure 38 contains all four dendrograms. None appear to be very informative. It seems that in situations with a small number of nominal variables, the variable weights may form the most informative part of the solution—they reflect the clustering that we know exists in the data, and can help to generate hypotheses for additional analyses.

Figure 38. Dendrograms (single linkage) on three- (top row) and four-variable (bottom row) type N distance matrices both unweighted (left) and variable weighted (right) with the VWUO-MD grand minimum solutions

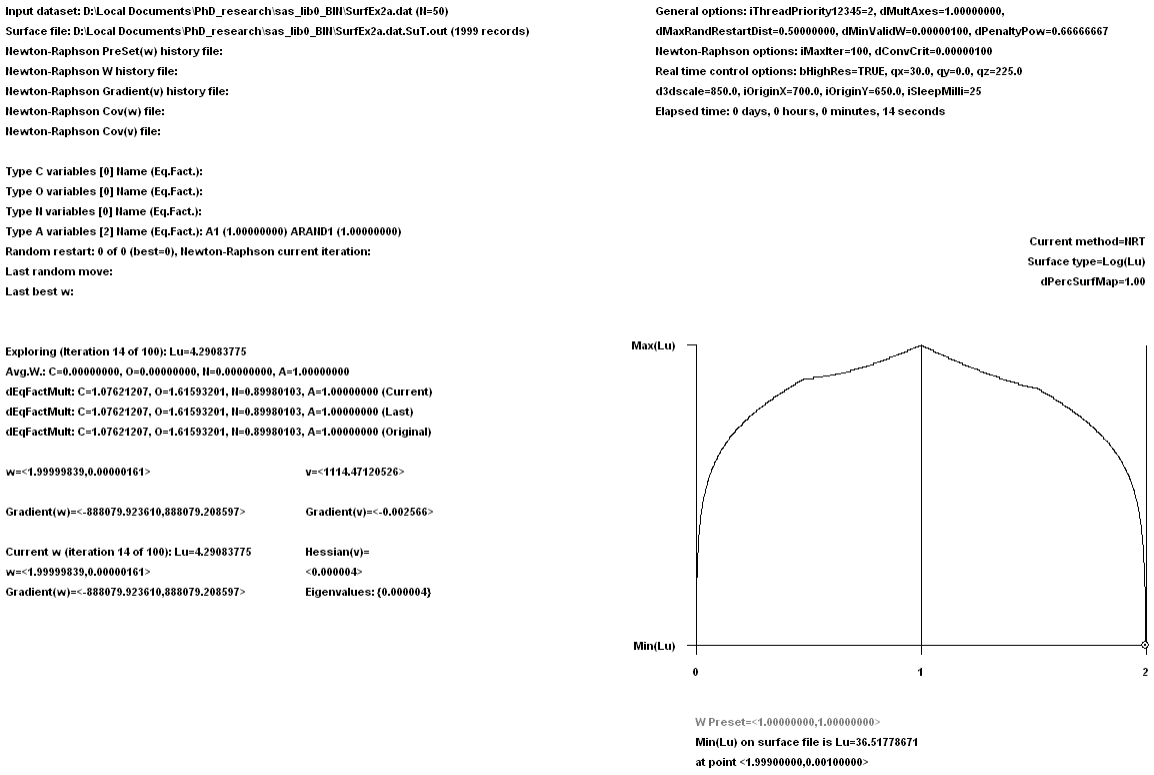


4.3.4 Analysis of type A variables

Next we will perform VWUO-MD analyses of the type A variables. We will analyze the two-variable subspace <A1, ARand1> first. The solution, from the default starting vector **w=1**, is shown overtop the L_U surface map in Figure 39. The solution is **w=**<1.999999, 0.000001>, a boundary case that only "converged" because of the minimum weight setting of dMinValidW=0.000001. It is clear from the surface map that this function has no local minimum, and in this case the two minima on the surface function have equal L_U and are at either boundary case. As with the type N subspace, this indicates a shortcoming of the improved

penalty function in L_U that can occur when VWUO-MD is used to analyze a set of only two discrete variables. Fortunately, no practical application of VWUO-MD ought to involve only two variables.

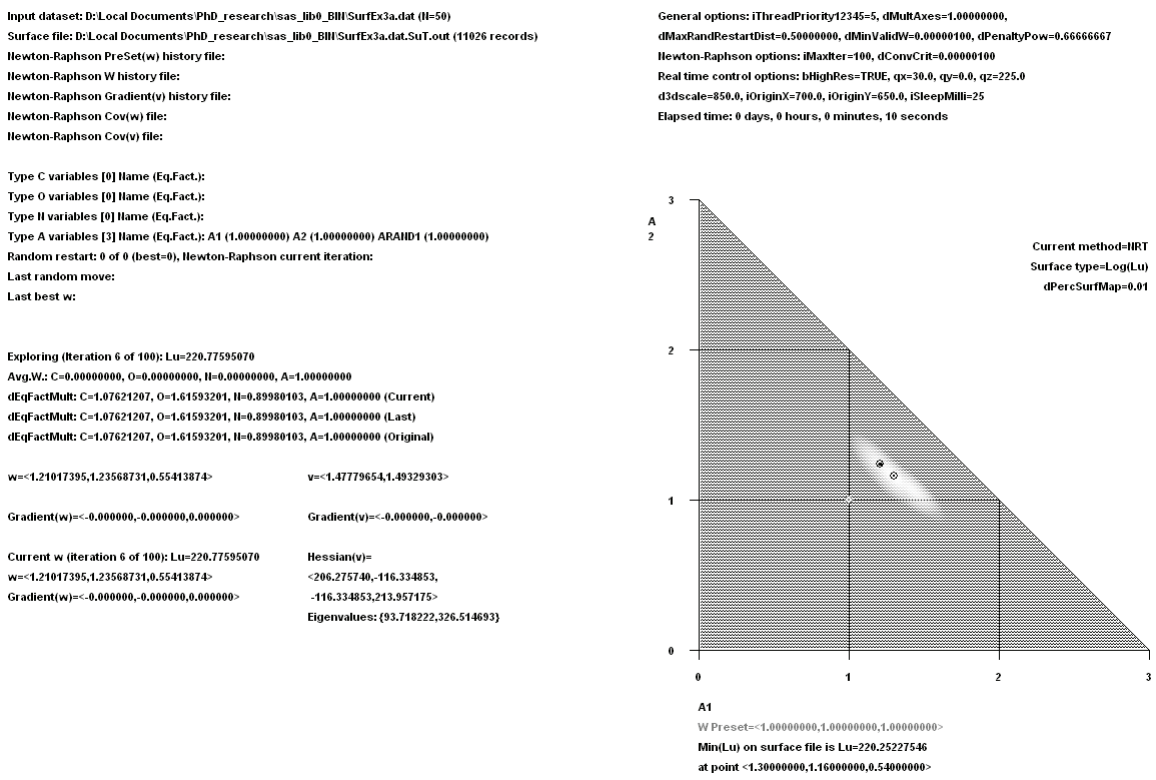
Figure 39. VWUO-MD solution to two-variable type A subspace in the mixed-type artificial data set, overtop the L_U surface map



We will analyze the three-variable subspace <A1, A2, ARand1> next. The solution, from the default starting vector **w=1**, is shown overtop the L_U surface map in Figure 40. The solution is **w=**<1.210174, 1.235687, 0.554139>. This is sensible (and informative for HG) considering how the artificial data were constructed, i.e., that A1 and A2 together defined clusters in the data, while ARand1 and ARand2 did not.

The 2D surface map reveals that there are at least two local minima, one of which is the grand minimum. Unfortunately, the solution attained from the default **w=1** does not equal the grand minimum in this case. To better reveal the additional local minimum, we created a high resolution surface map on a smaller area encompassing the first solution, and restarted VWUO-MD from a point within the depression containing the grand minimum. The second solution and grand minimum is **w**=<1.294313, 1.156153, 0.549534>, which is similar to the other local minimum. At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ =**0** and Hessian(**v**) had positive eigenvalues.

Figure 40. Default VWUO-MD solution to three-variable type A subspace in the mixed-type artificial data set, overtop the L_U surface map



Next we will analyze the four-variable subspace <A1, A2, ARand1, ARand2>. The solution, from the default starting vector **w=1**, is shown overtop the L_U surface map in Figure 41. The solution is **w**=<1.151529, 1.173861, 0.799681, 0.874929>. This is sensible (and informative for HG) considering how the artificial data were constructed, i.e., that A1 and A2 together defined clusters in the data, while ARand1 and ARand2 did not.

The observed minimum on the 3D surface map does not correspond with the default solution. This suggests the existence of multiple local minima, like we saw in the 2D surface map. To reveal the additional local minima, we created a high resolution surface map on a smaller area encompassing the grand minimum from the first output, and other solutions were found by restarting the procedure from the depressions (lighter spots) on different slices of the surface. In a similar manner, restarting the procedure from lighter regions on slices in different planes as well as on slices moved through the subspace, six other local minima were found. At the solutions, it was confirmed that $\nabla_{\hat{\mathbf{v}}}$ **=0** and Hessian(**v**) had positive eigenvalues. There are probably other local minima that we did not find. The seven local minima found are listed in Table 8, sorted by L_U. This table shows that the default solution was only sixth out of seven by L_U. All seven solutions (that we found) were sensible (and informative for HG).

Figure 41. Default VWUO-MD solution to four-variable type A subspace in the mixed-type artificial data set, overtop the L_U surface map

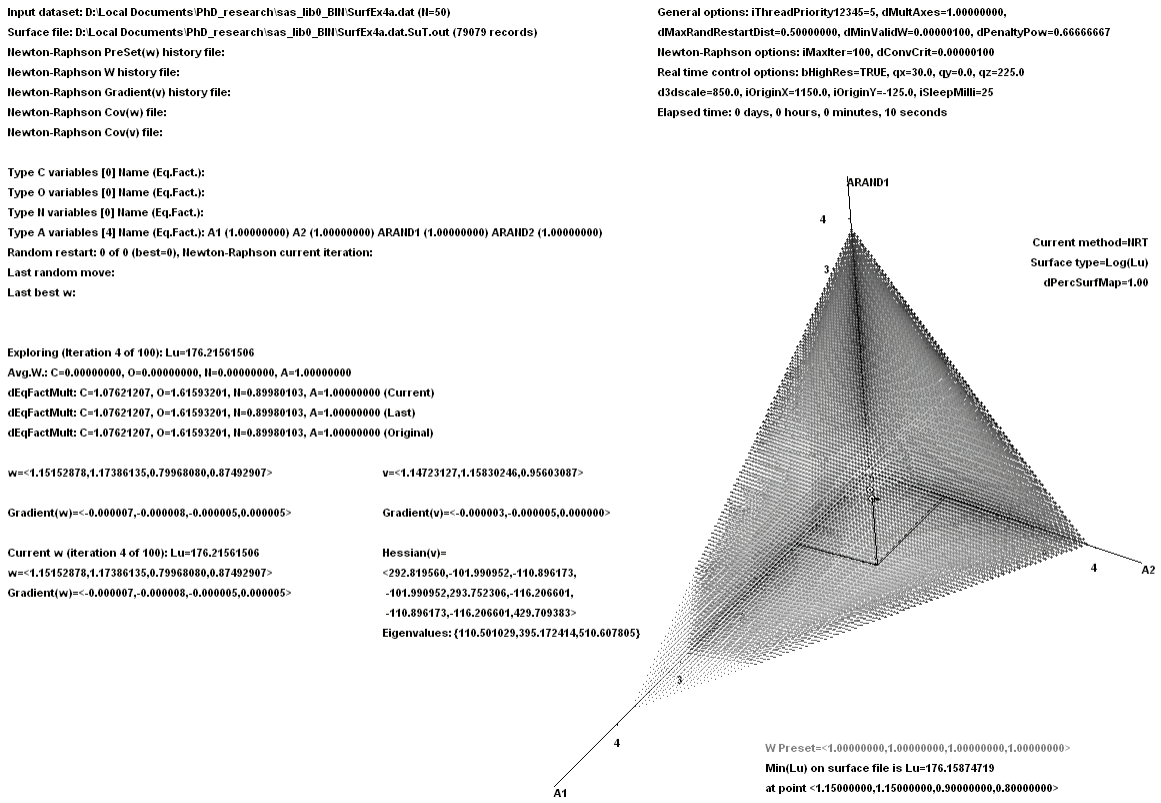


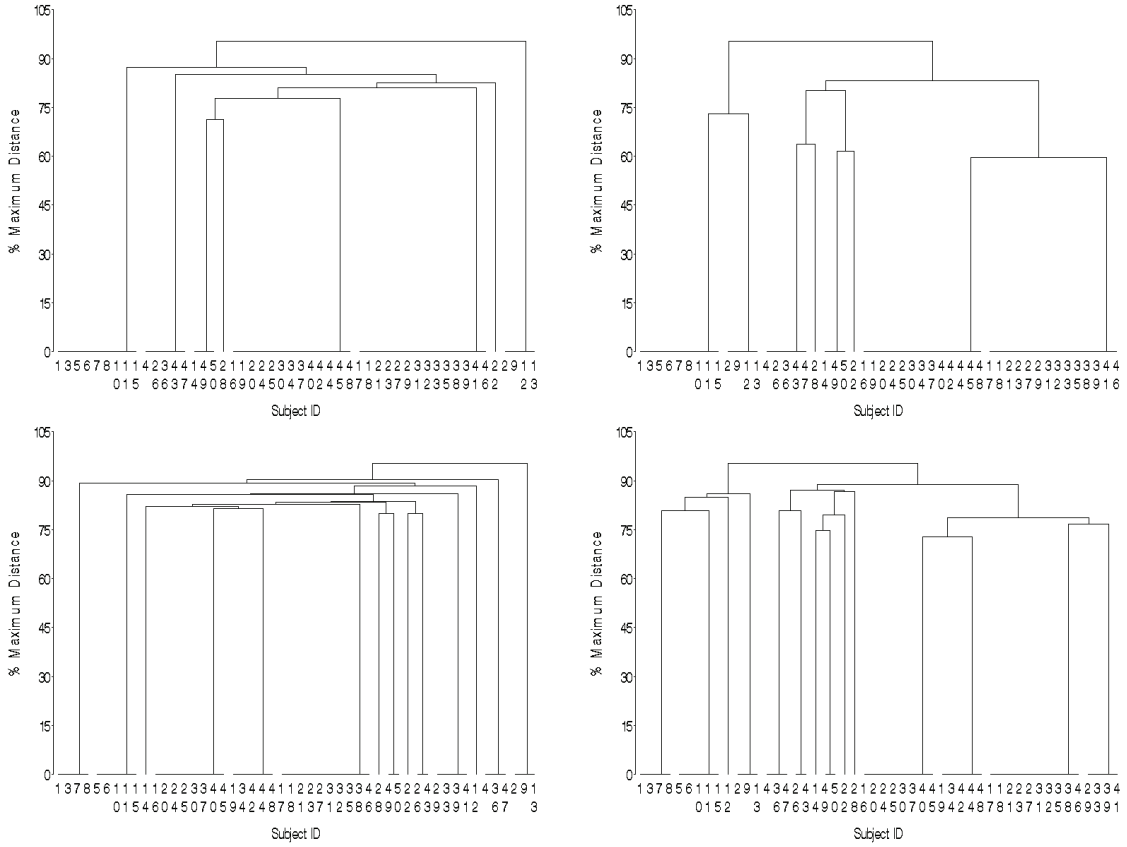
Table 8. Seven VWUO-MD solutions to four-variable type A subspace in the mixed-type artificial data set, sorted by L_U; the default solution is in italics

L_U	w_{A1}	w_{A2}	w_{ARand1}	w_{ARand2}
176.06052889	1.145728	1.186654	0.882716	0.784902
176.09511691	1.173545	1.158007	0.886358	0.782090
176.09602147	1.167399	1.162998	0.890344	0.779259
176.20251582	1.139568	1.187356	0.790741	0.882335
176.20314154	1.139683	1.182622	0.794989	0.882706
176.21561506	*1.151529*	*1.173861*	*0.799681*	*0.874929*
176.22770954	1.163140	1.162324	0.799603	0.874933

The relative weights in the grand minimum solutions are informative about the clustering present in the data. That is useful for HG. However, do they also help to enhance the clustering? To find out, we create dendrograms (single linkage) on three- and four-variable distance matrices both unweighted and

variable weighted with the grand minimum solutions obtained earlier. Figure 42 contains all four dendrograms. None appear to be very informative. It seems that in situations with a small number of binary asymmetric variables, the variable weights may form the most informative part of the solution—they reflect the clustering that we know exists in the data, and can help to generate hypotheses for additional analyses.

Figure 42. Dendrograms (single linkage) on three- (top row) and four-variable (bottom row) type A distance matrices both unweighted (left) and variable weighted (right) with the VWUO-MD grand minimum solutions



4.3.5 Analysis of mixed-type subspaces

Next we will perform VWUO-MD analyses of mixed-type data. We first analyze the six pairs of three-variable subspaces of different types. Table 9 contains the solutions from the default **w=1**. The solutions are all sensible (and informative for HG) within each type considering how the artificial data were constructed, i.e., that variables X1 and X2 (where X=C, O, N or A) defined clusters in the data, while XRand1 did not. Recall that the purpose of the equalizing multipliers was to provide a fair comparison between variables of different types. This means ideally that besides solutions being sensible (and informative for HG) within each type, $w_{XRand1} \leq w_{Y1}$ and $w_{XRand1} \leq w_{Y2}$ for both X and Y. This was true here in four out of six combinations. The exceptions involve w_{NRand1} in the types O and N subspace, and w_{NRand1} in the types N and A subspace.

Table 9. Default VWUO-MD solutions to six-variable type-pair artificial data

TypeX	TypeY	w_{X1}	w_{X2}	w_{XRand1}	w_{Y1}	w_{Y2}	w_{YRand1}
C	O	1.240916	1.362759	0.738439	0.942469	0.954587	0.760830
C	N	1.020536	1.168084	0.549186	1.194739	1.092688	0.974768
C	A	1.309662	1.454200	0.812625	0.897290	0.905580	0.620642
O	N	0.905459	1.005123	0.645914	1.231117	1.162951	1.049436
O	A	1.177644	1.222795	0.895828	1.001425	0.922055	0.780253
N	A	1.264981	1.249465	1.138873	0.960755	0.850283	0.535642

Next we analyze the four triples of three-variable subspaces of different types. Table 10 contains the solutions from the default **w=1**. The solutions are all sensible (and informative for HG) within each type considering how the artificial data were constructed, i.e., that variables X1 and X2 (where X=C, O, N or A) defined clusters in the data, while XRand1 did not. However, on all four

combinations, $w_{XRand1} > w_{Y1}$ or $w_{XRand1} > w_{Y2}$ for some X and Y. The equalizing multipliers were less effective on these data than they were on the six-variable subspaces above. However, presumably solutions on average will be fairer than they would be had we not calibrated the equalizing multipliers.

Table 10. Default VWUO-MD solutions to nine-variable three-types artificial data

TypeX	TypeY	TypeZ	w_{X1}	w_{X2}	w_{XRand1}	w_{Y1}	w_{Y2}	w_{YRand1}	w_{Z1}	w_{Z2}	w_{ZRand1}
C	O	N	1.126165	1.246280	0.719918	0.891936	0.917019	0.744384	1.149486	1.123737	1.081077
C	O	A	1.250057	1.275810	0.885142	0.997847	1.079059	0.904945	0.923272	0.903506	0.780363
C	N	A	1.192012	1.335204	0.784089	1.177362	1.132426	1.094408	0.839768	0.842757	0.601974
O	N	A	1.043310	1.132760	0.877138	1.184978	1.167196	1.132745	0.915173	0.836156	0.710544

Finally, we analyze the 12-variable subspace containing three variables of each type, two clustering variables and one independent of the clusters. The solution from the default **w=1** is <w_{C1}, w_{C2}, w_{CRand1}, w_{O1}, w_{O2}, w_{ORand1}, w_{N1}, w_{N2}, w_{NRand1}, w_{A1}, w_{A2}, w_{ARand1}>=<1.170928, 1.211323, 0.863377, 0.964366, 1.038497, 0.888572, 1.129983, 1.114780, 1.097046, 0.890474, 0.874921, 0.755734>. This is sensible (and informative for HG) within each type considering how the artificial data were constructed, i.e., that variables X1 and X2 (where X=C, O, N or A) defined clusters in the data, while XRand1 did not. However, $w_{XRand1}>w_{Y1}$ or $w_{XRand1}>w_{Y2}$ for some X and Y.

4.4 Strategic random restarts or surface maps for overcoming multiple local minima

The existence of multiple local minima in many of the subspaces we analyzed presents a problem. The local minima with lower L_U appear to be more informative in general, but how can you be reasonably sure you have found one of the lowest, if not the absolute lowest local minimum? One possibility is to restart the analysis multiple times from random locations, as others (e.g., De Soete and Makarenkov) have counseled for continuous variables when using the De Soete method.[18,50] This is a good idea and one adopted in VWUO-MD, but the method of random restarts in VWUO-MD differs from that used by De Soete in two important ways. First, the random restart is not just at any point in the parameter space, but rather at a random point within a (p-1)-dimensional hypersphere surrounding the *last best* solution (the point of lowest L_U to which the algorithm has so far converged). The p-1 dimensions include all but one

(randomly selected) variable weight, which is set to *p* minus the sum of the other weights. The second way that random restarts in VWUO-MD differ from those used by De Soete is that in VWUO-MD, the final solution is taken to be the point of convergence with the smallest L_U, by definition the best solution the algorithm could find. In De Soete's approach, the *average* **w** vector is taken regardless of how large L_{DS} might have been at some of the local minima.

The theoretical reasoning for restricting the random restarts within a hypersphere about the last best solution is for the algorithm to work its way towards the lowest local minimum, and not stray too far away towards regions of higher local minima. The assumption is that the lowest depressions occur in close proximity. That this was the case in the subspaces analyzed above can most easily be seen by noting that the local minima with the lowest few L_U all occurred at "sensible" locations. The radius of the hypersphere is set by the analyst in VWUO-MD, and must be chosen carefully. If the radius is too small, the algorithm will always restart within the last best local depression and arrive at the same solution with every restart. If the radius is too large, the algorithm will not retain the advantage of staying close to the lowest depression yet found. The default value of 0.5 was chosen after experimentation on several of the data sets analyzed above and elsewhere.

To demonstrate the effectiveness of the approach, we will reanalyze the three-variable type N subspace. Figure 43 contains the output from an analysis started from **w=1** with 10 random restarts within hyperspheres of radius 0.5. The grand minimum was found on the 7th random restart, indicated in the output by

"Random restart: 10 of 10 (best=7)". The analysis was repeated nine more times, and the list of 10 solutions is contained in Table 11. The grand minimum solution was found seven out of 10 times, with the remaining three solutions being the second smallest local minimum. All ten solutions are sensible (and informative for HG).

Figure 43. A successful application of random restarts for finding the grand minimum in three-variable type N subspace with multiple local minima; the grand minimum was found on the 7th random restart

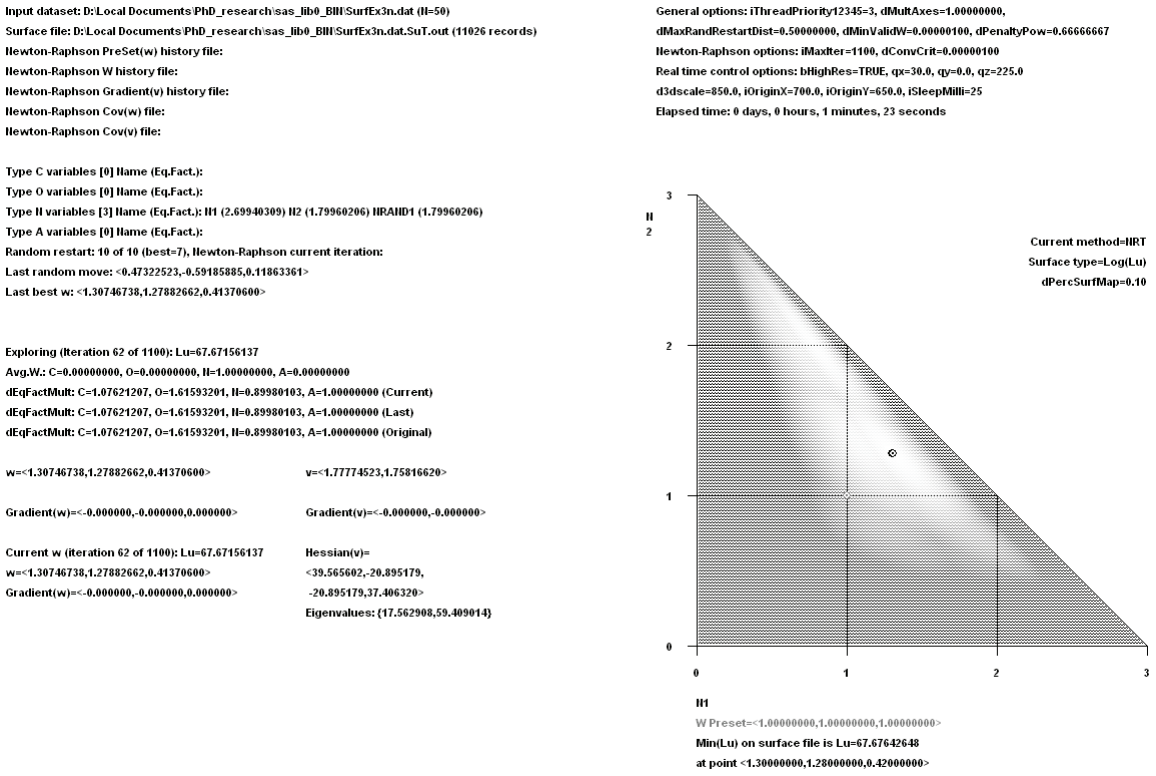


Table 11. Ten VWUO-MD solutions to three-variable type N subspace in the mixed-type artificial data set; sorted by L_U and K=required number of random restarts (maximum 10) to find each solution

K	L_U	w_{S1}	w_{S2}	w_{SRand1}
1	67.67156137	1.307467	1.278827	0.413706
2	67.67156137	1.307467	1.278827	0.413706
2	67.67156137	1.307467	1.278827	0.413706

2	67.67156137	1.307467	1.278827	0.413706
4	67.67156137	1.307467	1.278827	0.413706
7	67.67156137	1.307467	1.278827	0.413706
8	67.67156137	1.307467	1.278827	0.413706
1	67.95910829	0.778683	2.000000	0.221318
1	67.95910829	0.778683	2.000000	0.221318
6	67.95910829	0.778683	2.000000	0.221318

For lower-dimensional data sets, an alternative solution to the problem of multiple local minima could be surface mapping, by serving as a preprocessing step to obtain the approximate location of the grand minimum. This would be followed up with Newton-Raphson to refine the estimate. Combinatorics is the enemy at higher dimensions however, rendering the minimum feasible grid spacing prohibitively wide. For example, four grid lines on each of 10 variables requires L_U to be calculated at >1M points, and considering how close together some of the local minima can be, four grid lines will never be close to sufficient to identify the grand minimum with any confidence. To demonstrate the feasibility of this approach at least conceptually however, consider the three-variable type N subspace: when the estimation is started from within the depression containing the grand minimum reported on the surface map, VWUO-MD converges on the grand minimum every time; the key is to know approximately where to look.

4.5 Invariance of point estimates and covariance matrix estimators to category order, column order, record order and affine transformations

The VWUO-MD solution on a type C subspace is invariant to record order and 1-variable affine transformations. These properties are evident in the sums over Ω in L_U and the type C distance formulas. Estimates are invariant to record

order because the order of the Ω terms in the sum L_U does not affect the total. Estimates are invariant to changes in range because in the type C distance formula, the squared normalizing constant lives in the denominator below the squared difference measured on that variable, and the former being a multiple of the range, range cancels out. Estimates are invariant to changes in the origin because the difference is being taken. The latter two properties applied in series suggest invariance for 1-variable affine transformations. We would expect by the weighted type C distance formula that estimates are invariant to column (variable) order also, since no weight is treated differently than any others in the formula. However, while we find that the solution is invariant to column order excluding the last column, it is found to be only nearly invariant to permutations of column order that change the last column. To demonstrate the first three properties on the four-variable type C artificial data, we scramble the order of the first three columns, randomly rearrange the data records, and apply an affine transformation to each variable as follows: C1=(C1-2)*0.5; C2=(C2-1)*1; CRand1=(CRand1+1)*1.5; and CRand2=(CRand2+2)*2. VWUO-MD analysis of the transformed data produced exactly the same solution as before, iteration by iteration, with the exact same L_U evaluated at the solution. By “exact”, we mean within 0.00000001 (two digits beyond the convergence criterion and all that are recorded in the output files). The solution is, within the convergence criterion, $\mathbf{w}=\langle w_{C1}, w_{C2}, w_{CRand1}, w_{CRand2}\rangle=\langle 1.219176, 1.459937, 0.638227, 0.682661\rangle$. The near invariance to column permutations that switch the last column with another is shown by changing the last column with another in the transformed

data and reanalyzing. This time, the variable weights differed by as much as 0.0001, $\mathbf{w}=\langle w_{C1}, w_{C2}, w_{CRand1}, w_{CRand2}\rangle=\langle 1.219051, 1.459977, 0.638302, 0.682670\rangle$. Conceptually, the transformation from **v** to **w** should not affect the solution regardless of which is the last column, so the lack of invariance to column order involving the last column is probably due to numerical instability combined with the transformation from **v** to **w**, in which the p^{th} variable weight has a slightly different formula than the other p-1 columns. In any case, the effect does not appear to be big enough to substantively affect results.

The VWUO-MD solution on a type O subspace is invariant to record order and column order *including* permutations of column order that change the last column. Type O estimates are not completely invariant to category order since that could change the ranks, except when categories are only *completely* reversed on a variable if permuted at all. Estimates are invariant to record order because the order of the Ω terms in the sum L_U does not affect the total. Estimates are invariant to column (variable) order since no weight is treated differently than any others in the type O distance formula. Estimates are invariant to complete reversals of ordinal variable labels because that can be accomplished with an affine transformation of the variable ranks, and the type O distance formula, having the same structure as the type C distance formula, is invariant to 1-variable affine transformations. To demonstrate invariance on the four-variable type O artificial data, we scramble the order of the columns ensuring that the last column is changed, reverse the category labels of two variables, and randomly rearrange the data records. VWUO-MD analysis of the

transformed data produced the same solution as before to within the convergence criterion, with the same L_U evaluated at the solution. Estimates are not invariant to the range of category scores for the same reason that type C variables *are* invariant to variable scale: each normalizing constant is a multiple of the variable's "range". (Recall that this was to penalize differences between objects on a discrete variable less harshly when there are more categories.)

The VWUO-MD solution on a type N subspace is invariant to record order and column order *including* permutations of column order that change the last column. Type N estimates are invariant to category order. Estimates are invariant to record order because the order of the Ω terms in the sum L_U does not affect the total. Estimates are invariant to column (variable) order since no weight is treated differently than any others in the type N distance formula. Estimates are invariant to category order because the type N distance formula only considers equality between objects, not ranks. To demonstrate these properties on the four-variable type N artificial data, we scramble the order of the columns ensuring that the last column is changed, scramble the category label order of all the variables, and randomly rearrange the data records. VWUO-MD analysis of the transformed data produced the same solution as before to within the convergence criterion, with the same L_U evaluated at the solution.

The VWUO-MD solution on a type A subspace is invariant to record order and column order *including* permutations of column order that change the last column. Estimates are invariant to record order because the order of the Ω terms in the sum L_U does not affect the total. Estimates are invariant to column

(variable) order since no weight is treated differently than any others in the type A distance formula. To demonstrate these properties on the four-variable type A artificial data, we scramble the order of the columns ensuring that the last column is changed, and randomly rearrange the data records sorting by a uniform random variable. This time VWUO-MD analysis of the transformed data produced a different *default* solution, which was equal to the best solution found previously (the top row in Table 8) to within the convergence criterion, with the same L_U evaluated at the solution. Type A variables are not invariant to category labels, since by design equality between objects at value 1 is treated more importantly than equality at value 0.

Finally, we will check the above invariance properties on the 12-variable subspace containing three variables of each type, two clustering variables and one independent of the clusters. Records, columns and labels were randomly rearranged according to the descriptions above for each type. (VWUO.exe requires pre-ordering column types according to C, O, N and A, however we can reorder the columns within each type.) VWUO-MD analysis of the transformed data set produced the same solution as before to within the convergence criterion, with the same L_U evaluated at the solution.

The invariance of the covariance matrix estimators to the above conditions follows from their derivations. With identical point estimates and replicate estimates, the bootstrap covariance matrix estimator must also be invariant to the above conditions. With identical solutions, gradients and Hessians, the U-

statistic-based covariance matrix estimator must also be invariant to the above conditions. This was confirmed practically in a number of test data sets.

VWUO-MD is a reasonably robust procedure, invariant to most transformations that one should consider inconsequential, such as the arbitrary labels attached to nominal variables, or the scaling of a continuous variable. That all three discrete types as well as the mixed-type data set were invariant to column permutations that switched the last column with another offers support to the idea that the slight lack of invariance seen in this regard on type C data was due to numerical instability.

4.6 Monte Carlo simulations comparing bootstrap and U-statistic-based covariance matrix estimators

Earlier we developed covariance matrix estimators for **w** based on U-statistics and the bootstrap approach. In this section, we will perform Monte Carlo simulations to assess the performance of these estimators.

One hundred replicates of four three- and four-variable type-specific data sets and a 12-variable combined-types data set were randomly generated based on the clustered super population behind the artificial, clustered data set developed earlier. Specifically, the same probability distributions described earlier were drawn from, 100 times for each of the five data sets. Recall the importance of the super population: the absolute minimizing weight vector $\hat{\mathbf{w}}$ that would be obtained on the entire SP comprises “true” **w**, or equivalently, **w** minimizes the expectation of the loss function for a randomly selected triple of points. As part of the SP definition above, conditional distributions of variables

were described that depended on a latent group variable *g*. Proportions of the SP at different levels of *g* were described. Since in this SRS example *g* is latent (unmeasured), it would not be realistic to suppose one could selectively sample from each group. Therefore, different replicates have different sample distributions of *g* based on its probability distribution, besides the other variables. It has been found that for simple to complex regression models, use of at least 200 to 400 bootstrap samples is sufficient for stability of p-values from bootstrap-based hypothesis testing.[75] We generated 500 bootstrap weights for each replicate data set in the manner described earlier appropriate for an SRS.

The type N subspace was analyzed in two scenarios, one without random restarts, and one with five random restarts within hyperspheres of radius 0.5. The latter scenario was added to determine how each covariance matrix estimator could handle the additional variation introduced by multiple local minima and random restarts. Type N was chosen for this purpose because it was one of the most demanding data types: its three-variable subspace was tied on the highest number of local minima, with a default solution that ranked third out of five very widespread local minima in the example data set analyzed earlier. Five random restarts were used instead of 10 because five restarts would inflate the additional variance more than 10 restarts. (Recall that with 10 random restarts, most solutions ended up at the grand minimum in our type N analysis, which is of course a good thing.) All scenarios were analyzed starting from the default vector **w=1**. VWUO-MD was executed 501 times (once on the full sample and once on each bootstrap sample) on all 100 replicates of all six scenarios. The sample

covariance matrix of **w** was calculated for each scenario based on the 100 full sample replicates. The bootstrap and U-statistic-based covariance matrices were calculated on each replicate data set, providing a sample of 100 bootstrap or U-statistic-based estimates for comparison with each sample covariance matrix. Finally, means, 5^{th} percentiles and 95^{th} percentiles of each element in the bootstrap and U-statistic-based covariance matrices were compared to the sample covariance matrices.

In the following tables, the results on the three- and 12-variable data sets are explored in detail. At the end of the section, the performance of the U-statistic-based covariance matrix estimator on the four-variable data sets is described, compared to the results on the smaller and larger data sets, and implications are discussed.

Table 12 lists, for type C data, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The variance entries in the mean $\hat{Var}_U(\hat{\mathbf{w}})$ are underestimated by factors of 0.8, 2.3 and 3.2 (compared to the sample covariance matrix of the 100 full sample replicates). Cov(w_{C1},w_{C2}) is not as well estimated, but the other two covariances are well approximated.

Table 12. The performance of $\hat{Var}_U(\hat{\mathbf{w}})$ on type C data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.004681	-0.003434	-0.001247
w_{C2}	-0.003434	0.004962	-0.001529
w_{CRand1}	-0.001247	-0.001529	0.002775

P_5 $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.000916	-0.000758	-0.002506
w_{C2}	-0.000758	0.001216	-0.003491
w_{CRand1}	-0.002506	-0.003491	0.001893

Mean $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.001479	-0.000004	-0.001475
w_{C2}	-0.000004	0.002128	-0.002124
w_{CRand1}	-0.001475	-0.002124	0.003599

P_{95} $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.002308	0.000776	-0.000717
w_{C2}	0.000776	0.003503	-0.000952
w_{CRand1}	-0.000717	-0.000952	0.005635

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 13 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates the entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 1.4 to 1.6, conservative estimates but with the right order of magnitude. The 5th percentiles of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ are slightly liberal. This is a reasonable estimator, and better than the U-statistic-based estimator.

Table 13. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on type C data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.004681	-0.003434	-0.001247
w_{C2}	-0.003434	0.004962	-0.001529
w_{CRand1}	-0.001247	-0.001529	0.002775

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.002513	-0.018757	-0.003956
w_{C2}	-0.018757	0.003152	-0.004285
w_{CRand1}	-0.003956	-0.004285	0.002277

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.007473	-0.005461	-0.002013
w_{C2}	-0.005461	0.007639	-0.002179
w_{CRand1}	-0.002013	-0.002179	0.004191

P_{95} $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{C1}	w_{C2}	w_{CRand1}
w_{C1}	0.020897	-0.000932	-0.000800
w_{C2}	-0.000932	0.019987	-0.000563
w_{CRand1}	-0.000800	-0.000563	0.007058

Table 14 lists, for type O data, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{v}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{v}})$ for comparison. The mean $\hat{Var}_U(\hat{\mathbf{v}})$ underestimates the variance entries in $Var(\hat{\mathbf{v}})$ by factors of 3.4 to 6.2. While Cov(w_{O1},w_{ORand1}) is out by a bigger factor, the comparison is close to 0 and the absolute difference is small.

Table 14. The performance of $\hat{Var}_U(\hat{\mathbf{v}})$ on type O data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.122552	-0.130348	0.007796
w_{O2}	-0.130348	0.160223	-0.029875
w_{ORand1}	0.007796	-0.029875	0.022079

P_5 $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.005948	-0.052603	-0.013822
w_{O2}	-0.052603	0.006033	-0.030343
w_{ORand1}	-0.013822	-0.030343	0.002577

Mean $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.020500	-0.020048	-0.000452
w_{O2}	-0.020048	0.026036	-0.005988
w_{ORand1}	-0.000452	-0.005988	0.006440

P_{95} $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.044132	-0.004576	0.015496
w_{O2}	-0.004576	0.078155	0.005808
w_{ORand1}	0.015496	0.005808	0.014297

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 15 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates most of the entries in $\hat{Var}(\hat{\mathbf{w}})$ by a factor of 1.2, with the remaining covariance entry off by a factor of 0.8, generally conservative estimates but with the right order of magnitude and very close to optimal values. This is a reasonable estimator, and better than the U-statistic-based estimator.

Table 15. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on type O data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.122552	-0.130348	0.007796
w_{O2}	-0.130348	0.160223	-0.029875
w_{ORand1}	0.007796	-0.029875	0.022079

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.038030	-0.378112	-0.054228
w_{O2}	-0.378112	0.035798	-0.126523
w_{ORand1}	-0.054228	-0.126523	0.009040

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.152720	-0.159030	0.006310
w_{O2}	-0.159030	0.192625	-0.033595
w_{ORand1}	0.006310	-0.033595	0.027285

P_{95} $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{O1}	w_{O2}	w_{ORand1}
w_{O1}	0.423632	-0.034654	0.073939
w_{O2}	-0.034654	0.486865	0.025438
w_{ORand1}	0.073939	0.025438	0.058949

Table 16 lists, for type N data analyzed without random restarts, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_U(\hat{\mathbf{w}})$ underestimates the variance entries in $\hat{Var}_U(\hat{\mathbf{w}})$ by factors of 2.8 to 6.8. The covariance entries are off by factors of 0.8 to 11.0.

Table 16. The performance of $\hat{Var}_U(\hat{\mathbf{v}})$ on type N data analyzed without random restarts

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.027997	-0.026712	-0.001284
w_{N2}	-0.026712	0.043258	-0.016546
w_{NRand1}	-0.001284	-0.016546	0.017830

P_5 $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.001840	-0.005614	-0.003214
w_{N2}	-0.005614	0.000000	-0.008146
w_{NRand1}	-0.003214	-0.008146	0.001883

Mean $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.004086	-0.002439	-0.001647
w_{N2}	-0.002439	0.007219	-0.004780
w_{NRand1}	-0.001647	-0.004780	0.006426

P_{95} $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.007089	0.000000	-0.000225
w_{N2}	0.000000	0.012638	0.000000
w_{NRand1}	-0.000225	0.000000	0.009075

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 17 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates most of the entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 1.6 to 2.1, conservative estimates but with the right order of magnitude. The remaining covariance entry is out by a bigger factor and the sign has changed, but the comparison is very close to 0 and the absolute difference is small. This is a reasonable estimator, and better than the U-statistic-based estimator.

Table 17. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on type N data analyzed without random restarts

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.027997	-0.026712	-0.001284
w_{N2}	-0.026712	0.043258	-0.016546
w_{NRand1}	-0.001284	-0.016546	0.017830

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.014927	-0.111620	-0.018532
w_{N2}	-0.111620	0.029280	-0.058048
w_{NRand1}	-0.018532	-0.058048	0.009718

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.056365	-0.056711	0.000346
w_{N2}	-0.056711	0.085173	-0.028462
w_{NRand1}	0.000346	-0.028462	0.028117

P_{95} $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.101132	-0.009735	0.020615
w_{N2}	-0.009735	0.150390	-0.010790
w_{NRand1}	0.020615	-0.010790	0.059074

Table 18 lists, for type N data analyzed with random restarts, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_U(\hat{\mathbf{w}})$ underestimates the variance entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 3.5 to 35.1. Covariances are out by bigger factors and some of the signs have changed, and here not all the absolute differences are small.

Table 18. The performance of $\hat{Var}_U(\hat{\mathbf{v}})$ on type N data analyzed with 10 random restarts

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.046062	-0.061213	0.015151
w_{N2}	-0.061213	0.088908	-0.027695
w_{NRand1}	0.015151	-0.027695	0.012543

P_5 $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.001326	-0.004344	-0.002581
w_{N2}	-0.004344	0.000000	-0.007382
w_{NRand1}	-0.002581	-0.007382	0.001326

Mean $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.002680	-0.000836	-0.001844
w_{N2}	-0.000836	0.002534	-0.001698
w_{NRand1}	-0.001844	-0.001698	0.003542

P_{95} $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.005808	0.000000	-0.000602
w_{N2}	0.000000	0.011887	0.000000
w_{NRand1}	-0.000602	0.000000	0.008502

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 19 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates the entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 1.6 to 1.9, conservative estimates but with the right order of magnitude. The additional variation introduced by multiple local minima and random restarts has been adequately captured by the bootstrap approach. This is a reasonable estimator, and better than the U-statistic-based estimator.

Table 19. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on type N data analyzed with 10 random restarts

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.046062	-0.061213	0.015151
w_{N2}	-0.061213	0.088908	-0.027695
w_{NRand1}	0.015151	-0.027695	0.012543

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.022946	-0.202358	-0.000606
w_{N2}	-0.202358	0.032524	-0.099485
w_{NRand1}	-0.000606	-0.099485	0.003264

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.087636	-0.111891	0.024255
w_{N2}	-0.111891	0.159943	-0.048052
w_{NRand1}	0.024255	-0.048052	0.023797

P_{95} $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{N1}	w_{N2}	w_{NRand1}
w_{N1}	0.168038	-0.025774	0.051969
w_{N2}	-0.025774	0.280333	-0.006950
w_{NRand1}	0.051969	-0.006950	0.067387

Table 20 lists, for type A data, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{v}})$ for comparison. The mean $\hat{Var}_U(\hat{\mathbf{w}})$ underestimates most of the entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 4.6 to 7.6. The remaining covariance entry is out by a bigger factor but the comparison value is close to 0 and the absolute difference is small.

Table 20. The performance of $\hat{Var}_U(\hat{\mathbf{v}})$ on type A data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.012608	-0.000022	-0.012587
w_{A2}	-0.000022	0.015757	-0.015735
w_{ARand1}	-0.012587	-0.015735	0.028322

P_5 $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.000817	-0.002722	-0.004245
w_{A2}	-0.002722	0.000998	-0.004119
w_{ARand1}	-0.004245	-0.004119	0.002213

Mean $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.002732	-0.000679	-0.002053
w_{A2}	-0.000679	0.002746	-0.002067
w_{ARand1}	-0.002053	-0.002067	0.004120

P_{95} $\hat{Var}_U(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.005461	0.000706	-0.000782
w_{A2}	0.000706	0.005674	-0.000622
w_{ARand1}	-0.000782	-0.000622	0.006402

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 21 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates most of the entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 0.9 to 2.0, generally conservative estimates but with the right order of magnitude. The remaining covariance entry is out by a bigger factor, but the comparison value is close to 0 and the absolute difference is small. This is a reasonable estimator, and better than the U-statistic-based estimator.

Table 21. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on type A data

$\hat{Var}(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.012608	-0.000022	-0.012587
w_{A2}	-0.000022	0.015757	-0.015735
w_{ARand1}	-0.012587	-0.015735	0.028322

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.011106	-0.030974	-0.031331
w_{A2}	-0.030974	0.011977	-0.037138
w_{ARand1}	-0.031331	-0.037138	0.012327

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.025107	-0.009943	-0.015164
w_{A2}	-0.009943	0.024351	-0.014408
w_{ARand1}	-0.015164	-0.014408	0.029572

P_{95} $\hat{Var}_{BS}(\hat{\mathbf{w}})$

w	w_{A1}	w_{A2}	w_{ARand1}
w_{A1}	0.046766	0.000719	-0.004422
w_{A2}	0.000719	0.053755	-0.002940
w_{ARand1}	-0.004422	-0.002940	0.058401

Table 22 lists, for mixed-type data, the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_U(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_U(\hat{\mathbf{w}})$ differ from the variance entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 0.7 to 1.8, which are much better than we saw in smaller subspaces. However, most of the diagonal entries remain liberal. Thirty-eight of the 66 covariance entries are out by factors >0.2 and <5, and with the correct sign retained. The remaining entries have comparison values relatively close to 0 and the absolute differences are small.

Table 22. The performance of $\hat{Var}_U(\hat{\mathbf{w}})$ on mixed-type data

$\hat{Var}(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	5.81	-2.96	0.09	-2.00	-1.24	0.91	-0.05	-0.11	0.04	-0.17	-0.48	0.18
w_{C2}	-2.96	6.28	-0.33	0.44	-0.40	-1.17	0.04	0.18	-0.37	-0.37	-0.23	-1.11
w_{CRand1}	0.09	-0.33	2.60	-1.39	-1.27	0.10	0.06	0.01	0.10	0.02	-0.26	0.27
w_{O1}	-2.00	0.44	-1.39	7.30	-1.34	-1.44	-0.09	-0.01	-0.29	-0.65	0.16	-0.70
w_{O2}	-1.24	-0.40	-1.27	-1.34	7.36	-0.54	0.00	-0.29	-0.26	-0.74	-0.67	-0.62
w_{ORand1}	0.91	-1.17	0.10	-1.44	-0.54	1.70	-0.01	-0.02	0.19	-0.02	-0.28	0.58
w_{N1}	-0.05	0.04	0.06	-0.09	0.00	-0.01	0.06	0.06	0.04	-0.04	-0.07	-0.01
w_{N2}	-0.11	0.18	0.01	-0.01	-0.29	-0.02	0.06	0.22	0.08	-0.01	-0.05	-0.05
w_{NRand1}	0.04	-0.37	0.10	-0.29	-0.26	0.19	0.04	0.08	0.20	0.04	-0.01	0.24
w_{A1}	-0.17	-0.37	0.02	-0.65	-0.74	-0.02	-0.04	-0.01	0.04	1.69	0.06	0.18
w_{A2}	-0.48	-0.23	-0.26	0.16	-0.67	-0.28	-0.07	-0.05	-0.01	0.06	1.71	0.12
w_{ARand1}	0.18	-1.11	0.27	-0.70	-0.62	0.58	-0.01	-0.05	0.24	0.18	0.12	0.92

P_5 $\hat{Var}_U(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	1.96	-3.59	-0.82	-1.78	-1.86	-0.59	-0.19	-0.38	-0.30	-0.66	-0.91	-0.57
w_{C2}	-3.59	2.79	-1.24	-2.04	-1.97	-0.99	-0.25	-0.48	-0.41	-1.02	-1.04	-1.05
w_{CRand1}	-0.82	-1.24	1.77	-1.53	-1.65	-0.70	-0.14	-0.29	-0.27	-0.85	-0.82	-0.51
w_{O1}	-1.78	-2.04	-1.53	1.74	-2.90	-1.37	-0.27	-0.30	-0.57	-1.21	-1.25	-1.35
w_{O2}	-1.86	-1.97	-1.65	-2.90	2.15	-1.49	-0.33	-0.50	-0.63	-1.64	-1.22	-1.49
w_{ORand1}	-0.59	-0.99	-0.70	-1.37	-1.49	1.00	-0.10	-0.22	-0.16	-0.64	-0.54	-0.08
w_{N1}	-0.19	-0.25	-0.14	-0.27	-0.33	-0.10	0.04	0.03	0.03	-0.11	-0.17	-0.08
w_{N2}	-0.38	-0.48	-0.29	-0.30	-0.50	-0.22	0.03	0.07	0.05	-0.22	-0.38	-0.22
w_{NRand1}	-0.30	-0.41	-0.27	-0.57	-0.63	-0.16	0.03	0.05	0.19	-0.22	-0.25	0.01
w_{A1}	-0.66	-1.02	-0.85	-1.21	-1.64	-0.64	-0.11	-0.22	-0.22	0.79	-0.37	-0.35
w_{A2}	-0.91	-1.04	-0.82	-1.25	-1.22	-0.54	-0.17	-0.38	-0.25	-0.37	0.79	-0.38
w_{ARand1}	-0.57	-1.05	-0.51	-1.35	-1.49	-0.08	-0.08	-0.22	0.01	-0.35	-0.38	0.58

Mean $\hat{Var}_U(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	3.31	-2.03	-0.11	-0.31	-0.36	-0.03	0.00	-0.02	-0.07	-0.08	-0.18	-0.11
w_{C2}	-2.03	4.44	-0.22	-0.36	-0.43	-0.26	-0.05	-0.06	-0.15	-0.25	-0.32	-0.29
w_{CRand1}	-0.11	-0.22	2.50	-0.67	-0.63	-0.07	-0.02	-0.08	-0.02	-0.31	-0.36	0.01
w_{O1}	-0.31	-0.36	-0.67	4.35	-1.04	-0.57	-0.04	0.05	-0.17	-0.25	-0.37	-0.60
w_{O2}	-0.36	-0.43	-0.63	-1.04	4.69	-0.53	-0.08	-0.16	-0.21	-0.62	-0.12	-0.50
w_{ORand1}	-0.03	-0.26	-0.07	-0.57	-0.53	1.52	-0.01	-0.04	0.05	-0.18	-0.16	0.29
w_{N1}	0.00	-0.05	-0.02	-0.04	-0.08	-0.01	0.09	0.08	0.08	-0.01	-0.04	0.00
w_{N2}	-0.02	-0.06	-0.08	0.05	-0.16	-0.04	0.08	0.22	0.11	0.02	-0.08	-0.04
w_{NRand1}	-0.07	-0.15	-0.02	-0.17	-0.21	0.05	0.08	0.11	0.30	-0.01	-0.04	0.14
w_{A1}	-0.08	-0.25	-0.31	-0.25	-0.62	-0.18	-0.01	0.02	-0.01	1.52	0.08	0.09
w_{A2}	-0.18	-0.32	-0.36	-0.37	-0.12	-0.16	-0.04	-0.08	-0.04	0.08	1.53	0.07
w_{ARand1}	-0.11	-0.29	0.01	-0.60	-0.50	0.29	0.00	-0.04	0.14	0.09	0.07	0.95

$P_{95}\ \hat{Var}_U(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	4.99	-0.92	0.70	0.92	0.73	0.52	0.17	0.22	0.16	0.63	0.38	0.42
w_{C2}	-0.92	6.81	0.91	1.11	0.97	0.42	0.16	0.28	0.16	0.65	0.40	0.39
w_{CRand1}	0.70	0.91	3.34	0.05	0.16	0.52	0.09	0.10	0.18	0.12	0.18	0.46
w_{O1}	0.92	1.11	0.05	7.14	0.33	0.04	0.21	0.64	0.12	0.87	0.54	0.01
w_{O2}	0.73	0.97	0.16	0.33	8.82	0.08	0.13	0.08	0.19	0.33	0.81	0.21
w_{ORand1}	0.52	0.42	0.52	0.04	0.08	2.09	0.10	0.13	0.25	0.21	0.25	0.67
w_{N1}	0.17	0.16	0.09	0.21	0.13	0.10	0.16	0.15	0.15	0.14	0.09	0.09
w_{N2}	0.22	0.28	0.10	0.64	0.08	0.13	0.15	0.44	0.21	0.34	0.09	0.12
w_{NRand1}	0.16	0.16	0.18	0.12	0.19	0.25	0.15	0.21	0.51	0.20	0.15	0.30
w_{A1}	0.63	0.65	0.12	0.87	0.33	0.21	0.14	0.34	0.20	2.54	0.62	0.43
w_{A2}	0.38	0.40	0.18	0.54	0.81	0.25	0.09	0.09	0.15	0.62	2.39	0.52
w_{ARand1}	0.42	0.39	0.46	0.01	0.21	0.67	0.09	0.12	0.30	0.43	0.52	1.43

Next we assess the performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$. Table 23 lists the 5th percentiles, means and 95th percentiles of the entries in $\hat{Var}_{BS}(\hat{\mathbf{w}})$ across the 100 replicates, and the sample covariance matrix $\hat{Var}(\hat{\mathbf{w}})$ for comparison. The mean $\hat{Var}_{BS}(\hat{\mathbf{w}})$ overestimates the variance entries in $\hat{Var}(\hat{\mathbf{w}})$ by factors of 0.9 to 2.3, mostly conservative estimates but with the right order of magnitude. Forty-seven of the 66 covariance entries are out by factors >0.2 and <5, and with the correct sign retained. The remaining entries have comparison values relatively close to 0 and the absolute differences are small. This is a reasonable estimator, and better than the U-statistic-based estimator considering lack of liberal variance entries.

Table 23. The performance of $\hat{Var}_{BS}(\hat{\mathbf{w}})$ on mixed-type data

$\hat{Var}(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	5.81	-2.96	0.09	-2.00	-1.24	0.91	-0.05	-0.11	0.04	-0.17	-0.48	0.18
w_{C2}	-2.96	6.28	-0.33	0.44	-0.40	-1.17	0.04	0.18	-0.37	-0.37	-0.23	-1.11
w_{CRand1}	0.09	-0.33	2.60	-1.39	-1.27	0.10	0.06	0.01	0.10	0.02	-0.26	0.27
w_{O1}	-2.00	0.44	-1.39	7.30	-1.34	-1.44	-0.09	-0.01	-0.29	-0.65	0.16	-0.70
w_{O2}	-1.24	-0.40	-1.27	-1.34	7.36	-0.54	0.00	-0.29	-0.26	-0.74	-0.67	-0.62
w_{ORand1}	0.91	-1.17	0.10	-1.44	-0.54	1.70	-0.01	-0.02	0.19	-0.02	-0.28	0.58
w_{N1}	-0.05	0.04	0.06	-0.09	0.00	-0.01	0.06	0.06	0.04	-0.04	-0.07	-0.01
w_{N2}	-0.11	0.18	0.01	-0.01	-0.29	-0.02	0.06	0.22	0.08	-0.01	-0.05	-0.05
w_{NRand1}	0.04	-0.37	0.10	-0.29	-0.26	0.19	0.04	0.08	0.20	0.04	-0.01	0.24
w_{A1}	-0.17	-0.37	0.02	-0.65	-0.74	-0.02	-0.04	-0.01	0.04	1.69	0.06	0.18
w_{A2}	-0.48	-0.23	-0.26	0.16	-0.67	-0.28	-0.07	-0.05	-0.01	0.06	1.71	0.12
w_{ARand1}	0.18	-1.11	0.27	-0.70	-0.62	0.58	-0.01	-0.05	0.24	0.18	0.12	0.92

P_5 $\hat{Var}_{BS}(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	4.49	-4.78	-1.84	-3.34	-3.62	-1.04	-0.28	-0.68	-0.49	-1.37	-1.42	-0.89
w_{C2}	-4.78	3.85	-1.83	-2.83	-2.77	-1.51	-0.28	-0.54	-0.53	-1.50	-1.58	-1.52
w_{CRand1}	-1.84	-1.83	2.92	-2.68	-2.67	-0.73	-0.21	-0.45	-0.30	-1.23	-1.29	-0.52
w_{O1}	-3.34	-2.83	-2.68	3.05	-3.10	-2.10	-0.33	-0.40	-0.72	-1.57	-1.81	-1.93
w_{O2}	-3.62	-2.77	-2.67	-3.10	3.43	-2.13	-0.43	-0.68	-0.89	-2.48	-1.77	-2.07
w_{ORand1}	-1.04	-1.51	-0.73	-2.10	-2.13	1.59	-0.14	-0.29	-0.22	-0.81	-0.77	-0.13
w_{N1}	-0.28	-0.28	-0.21	-0.33	-0.43	-0.14	0.07	0.05	0.06	-0.15	-0.20	-0.11
w_{N2}	-0.68	-0.54	-0.45	-0.40	-0.68	-0.29	0.05	0.11	0.09	-0.26	-0.48	-0.27
w_{NRand1}	-0.49	-0.53	-0.30	-0.72	-0.89	-0.22	0.06	0.09	0.30	-0.33	-0.32	-0.02
w_{A1}	-1.37	-1.50	-1.23	-1.57	-2.48	-0.81	-0.15	-0.26	-0.33	1.32	-0.30	-0.39
w_{A2}	-1.42	-1.58	-1.29	-1.81	-1.77	-0.77	-0.20	-0.48	-0.32	-0.30	1.26	-0.34
w_{ARand1}	-0.89	-1.52	-0.52	-1.93	-2.07	-0.13	-0.11	-0.27	-0.02	-0.39	-0.34	0.98

Mean $\hat{Var}_{BS}(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	6.53	-2.56	-0.39	-0.97	-1.08	-0.21	-0.07	-0.14	-0.14	-0.31	-0.45	-0.22
w_{C2}	-2.56	6.24	-0.48	-0.45	-0.52	-0.47	-0.01	-0.02	-0.19	-0.44	-0.54	-0.55
w_{CRand1}	-0.39	-0.48	4.36	-1.14	-1.18	-0.05	-0.03	-0.10	-0.03	-0.49	-0.56	0.09
w_{O1}	-0.97	-0.45	-1.14	6.44	-1.35	-0.79	-0.05	0.10	-0.22	-0.26	-0.46	-0.84
w_{O2}	-1.08	-0.52	-1.18	-1.35	7.16	-0.73	-0.09	-0.21	-0.29	-0.85	-0.11	-0.76
w_{ORand1}	-0.21	-0.47	-0.05	-0.79	-0.73	2.31	-0.02	-0.06	0.06	-0.22	-0.21	0.39
w_{N1}	-0.07	-0.01	-0.03	-0.05	-0.09	-0.02	0.12	0.11	0.12	-0.02	-0.05	-0.01
w_{N2}	-0.14	-0.02	-0.10	0.10	-0.21	-0.06	0.11	0.29	0.18	0.02	-0.11	-0.05
w_{NRand1}	-0.14	-0.19	-0.03	-0.22	-0.29	0.06	0.12	0.18	0.46	-0.02	-0.07	0.15
w_{A1}	-0.31	-0.44	-0.49	-0.26	-0.85	-0.22	-0.02	0.02	-0.02	2.14	0.25	0.20
w_{A2}	-0.45	-0.54	-0.56	-0.46	-0.11	-0.21	-0.05	-0.11	-0.07	0.25	2.13	0.17
w_{ARand1}	-0.22	-0.55	0.09	-0.84	-0.76	0.39	-0.01	-0.05	0.15	0.20	0.17	1.42

$P_{95}\ \hat{Var}_{BS}(\hat{\mathbf{w}})*10^3$

w	w_{C1}	w_{C2}	w_{CRand1}	w_{O1}	w_{O2}	w_{ORand1}	w_{N1}	w_{N2}	w_{NRand1}	w_{A1}	w_{A2}	w_{ARand1}
w_{C1}	8.69	-0.93	0.90	0.99	0.69	0.57	0.15	0.22	0.14	0.81	0.49	0.54
w_{C2}	-0.93	9.74	1.05	1.63	1.66	0.45	0.22	0.41	0.18	0.56	0.34	0.33
w_{CRand1}	0.90	1.05	7.52	0.10	-0.09	0.59	0.12	0.12	0.19	0.14	0.16	0.64
w_{O1}	0.99	1.63	0.10	9.64	0.38	0.13	0.27	0.89	0.14	1.12	0.69	0.14
w_{O2}	0.69	1.66	-0.09	0.38	12.98	0.20	0.24	0.17	0.14	0.79	1.20	0.23
w_{ORand1}	0.57	0.45	0.59	0.13	0.20	3.08	0.12	0.14	0.30	0.35	0.37	0.94
w_{N1}	0.15	0.22	0.12	0.27	0.24	0.12	0.20	0.19	0.20	0.15	0.11	0.12
w_{N2}	0.22	0.41	0.12	0.89	0.17	0.14	0.19	0.54	0.27	0.45	0.13	0.15
w_{NRand1}	0.14	0.18	0.19	0.14	0.14	0.30	0.20	0.27	0.68	0.24	0.17	0.34
w_{A1}	0.81	0.56	0.14	1.12	0.79	0.35	0.15	0.45	0.24	3.07	1.00	0.65
w_{A2}	0.49	0.34	0.16	0.69	1.20	0.37	0.11	0.13	0.17	1.00	3.02	0.74
w_{ARand1}	0.54	0.33	0.64	0.14	0.23	0.94	0.12	0.15	0.34	0.65	0.74	2.04

While we do not present additional tables, we also ran simulations to assess the performance of the U-statistic-based covariance matrix estimator on four-variable type-specific data. The performance of $\hat{Var}_U(\hat{\mathbf{w}})$ was substantially improved, assessed as before by similarity to the sample covariance matrix of the 100 replicate variable weight vectors. On four-variable type C data, the diagonal entries were off by factors of 0.5 to 1.2, substantially better than the three-variable results and no longer a clearly liberal estimator. On four-variable type O data, the diagonal entries were overestimates by factors of 2.6 to 6.1, better than the three-variable results but still a liberal estimate. Additional simulations showed that even with as many as nine type O variables, estimates are liberal, as well as in the 12-variable mixed data tabulated above. On four-variable type N data without random restarts, the diagonal entries were off by factors of 0.4 to 1.6, substantially better than the three-variable results and no longer a clearly liberal estimator. On four-variable type A data, the diagonal entries were overestimates by factors of 2.4 to 3.2, better than the three-variable results but still a liberal estimate. Additional simulations showed that even with as many as nine type A variables, estimates are liberal, as well as in the 12-variable mixed data tabulated above. In the above results, when the variance entries were in line, the covariance (off-diagonal) entries were generally a mixture of estimates to either side of the corresponding sample covariance entries.

The fact that type C and N estimates were improved by adding a variable helps to affirm that one does not need mixed-type data to obtain good results from the U-statistic-based covariance matrix estimator, but that the estimator

does not do well (remains clearly liberal) with low-dimensional type C and N data, and with type O and A data of any dimension. On mixtures only involving types C and N, four-variable data is high dimensional enough, while for mixtures involving the other types, estimates remain liberal even with many variables.

In low-dimensional type C and N data, and type O and A data of any dimension, $\hat{Var}_{BS}(\hat{\mathbf{w}})$ outperformed $\hat{Var}_{U}(\hat{\mathbf{v}})$ for estimating $Var(\hat{\mathbf{w}})$. The bootstrap estimator was consistently conservative, overestimating by factors in the low single digits whether the VWUO-MD analysis involved a single type of variable, or mixed-type data. The bootstrap estimator has the added advantage of facilitating variance estimation for VWUO-MD estimates obtained on complex survey data for which bootstrap weights have been developed. However, for SRS data with at least four variables involving only types C and N (or perhaps other SRS data on which approximate, liberal estimates are sufficient), the U-statistic-based estimator can save substantial computing time (several hundred to a thousand times—the number of potential bootstrap weights).

To use either covariance matrix estimator, we will require some knowledge about the asymptotic distribution of $\hat{\mathbf{w}}$ to enable hypothesis testing of contrasts between the elements of **w**.

4.7 The distribution of $\hat{\mathbf{w}}_{(\mathbf{p-1})}$

4.7.1 Non-multivariate normality

For testing with $\hat{Var}(\hat{\mathbf{w}})$ (whichever estimator we use), it would be convenient if the asymptotic distribution of $\hat{\mathbf{w}}_{(\mathbf{p-1})}$ were multivariate normal. To

test this assumption, we performed the Henze-Zirkler T-test for multivariate normality[74] on each set of 100 full sample replicate estimates for $\mathbf{w}_{(p-1)}$, which is based on comparing the distribution of squared Mahalanobis distances to a chi-square distribution on *p*-1 degrees of freedom. Table 24 lists the results. Multivariate normality of the type C replicates is not rejected. However, tests in all other scenarios are statistically significant and we must reject multivariate normality. The least significant of these is the mixed-type scenario, with p-value=0.024, but the other tests are highly significant with p-value<0.001. Chi-square (on *p*-1 degrees of freedom) quantile-quantile plots of the squared Mahalanobis distances are shown in Figure 44. Under multivariate normality, these should be approximately straight lines. Single-type scenarios involving types O and N without random restarts show step function distributions of distances which may be a result of multiple local minima. We might have considered the discrete sample space for these variable types as a potential culprit here, but the type A and mixed-type plots actually look pretty straight (although the Henze-Zirkler test is still significant). The type N scenario with five random restarts has an improved Q-Q plot compared to type N without random restarts. Taking that farther, we tested another type N scenario with 15 random restarts, nearly guaranteeing that the absolute minimum would be found on any given run. The Q-Q plot was not noticeably improved, and the Henze-Zirkler test remained highly significant. It is possible that the lowest local minimum changes position in different replicates relative to the other local minima, which might explain this result.

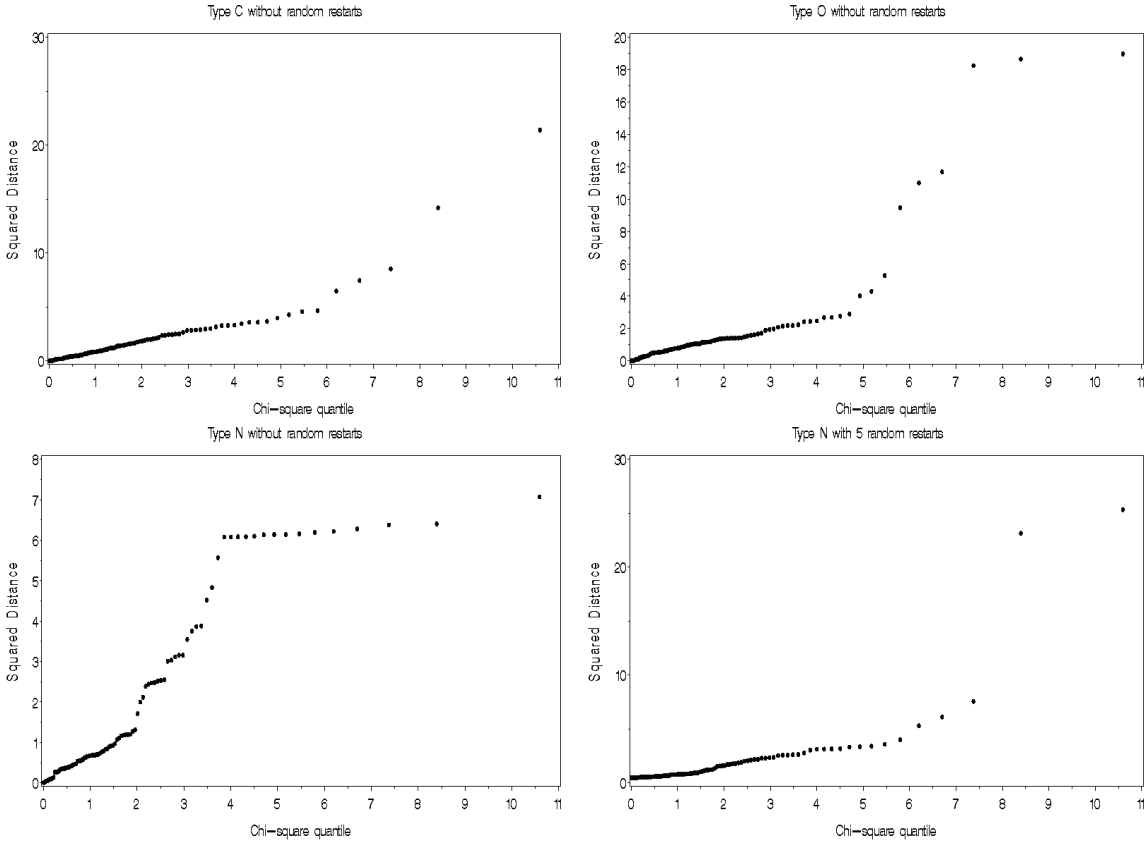
Considering the possibility that analyzing more variables would improve normality (as was possibly indicated on the mixed-type plot), we ran 100 full sample replicates of each single type subspace using four variables instead of three (both the clustering variables and both independent variables), without random restarts and starting from **w=1**. The Q-Q plots were visibly improved (Figure 45), however the Henze-Zirkler test remained highly significant (except for type C).

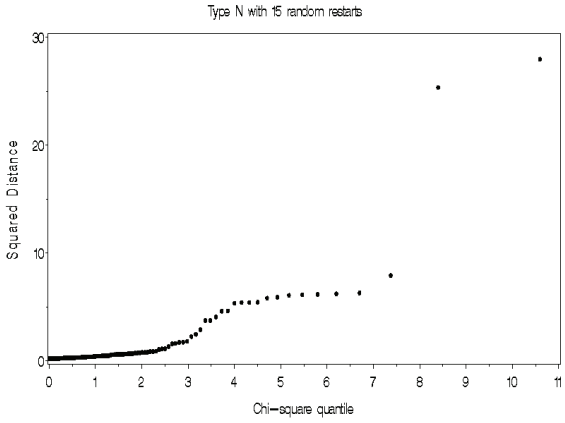
Finally, all these tests were rerun using log transformations of weights, as well as taking contrasts of weights, but the conclusions and plots did not improve.

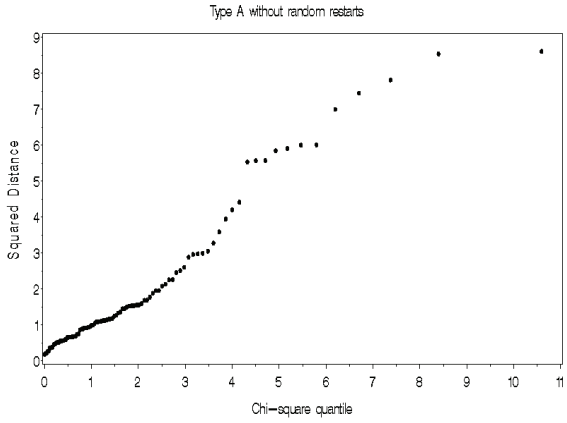
Table 24. Henze-Zirkler T-test for multivariate normality on each set of 100 full sample replicate estimates under each three-variable scenario and the mixed-type scenario

Scenario	Henze-Zirkler p-value
Type C without random restarts	0.948
Type O without random restarts	<0.001
Type N without random restarts	<0.001
Type N with five random restarts	<0.001
Type N with 15 random restarts	<0.001
Type A without random restarts	<0.001
Mixed-type without random restarts	0.024

Figure 44. Chi-square quantile-quantile plots of squared Mahalanobis distances in 100 full sample replicates under each three-variable scenario and the mixed-type scenario




Type N with 15 random restarts
Squared Distance
Chi-square quantile


Type A without random restarts
Squared Distance
Chi-square quantile

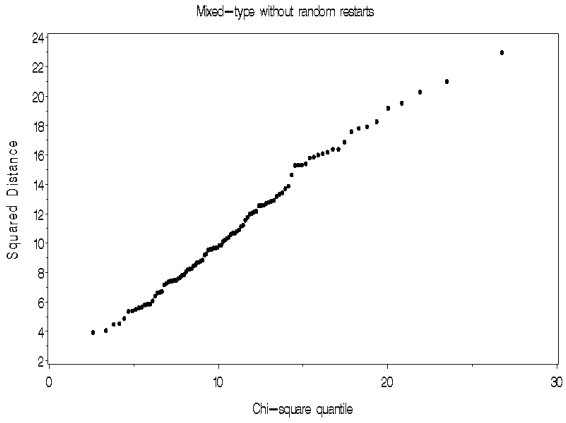
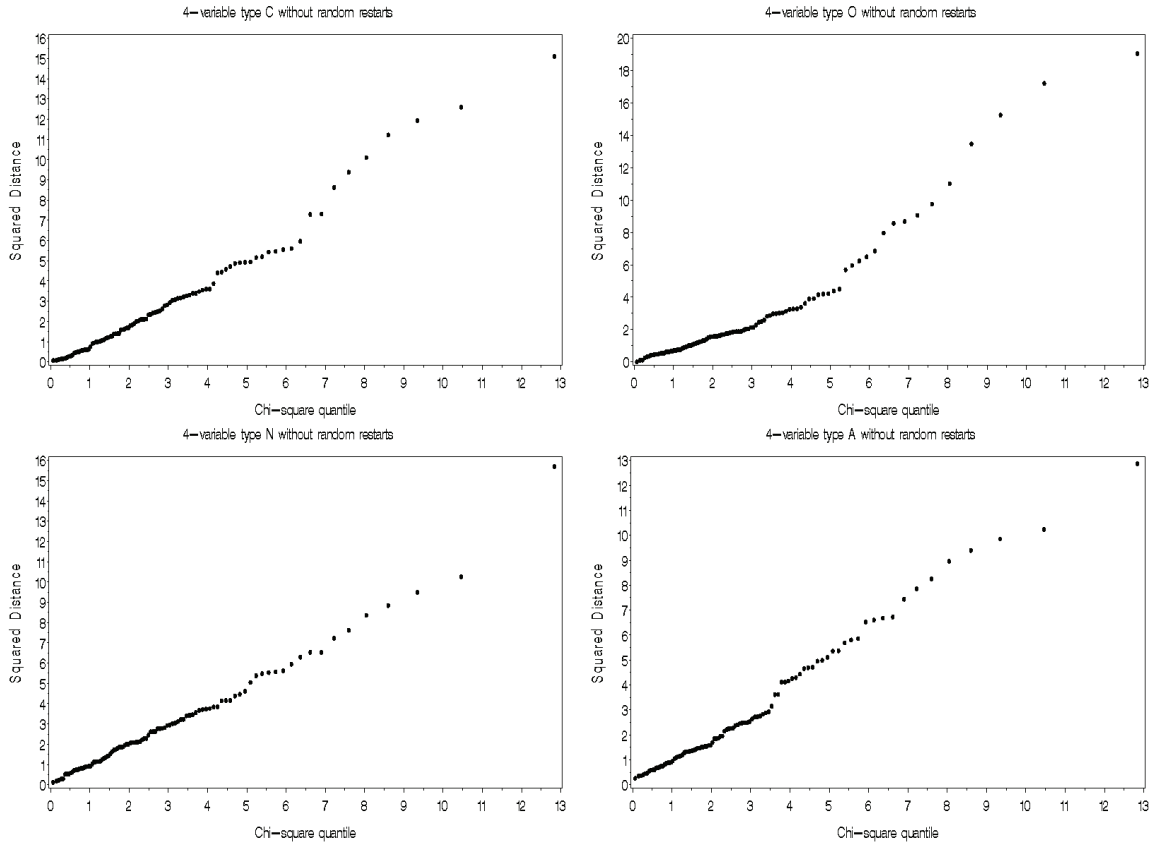

Mixed-type without random restarts
Squared Distance
Chi-square quantile

Figure 45. Chi-square quantile-quantile plots of squared Mahalanobis distances in 100 full sample replicates under each four-variable scenario



4.7.2 Bootstrap percentile confidence intervals

We found that $\hat{Var}_{BS}(\hat{\mathbf{w}})$ provides a reasonable estimate of $Var(\hat{\mathbf{w}})$ under all scenarios tested, and that $\hat{Var}_{U}(\hat{\mathbf{w}})$ is a good estimator for scenarios with more variables of mixed type. Conveniently, for cases with many variables of mixed type, the estimates are also nearly MVN, according to the chi-square quantile-quantile plot of the squared Mahalanobis distances, as are single-type analyses (approximately) in at least four dimensions. While the Henze-Zirkler T-test was statistically significant for the mixed-type and four-variable analyses, the quantile-quantile plots were quite straight, and in fact we will see later that normal-based

intervals perform very well in such scenarios (i.e., the Henze-Zirkler test may be overly sensitive in higher dimensions).

Unfortunately, the finding that the distribution of $\hat{\mathbf{w}}_{(\mathbf{p-1})}$ is not MVN in low-dimensional situations (≤3 variables) means that testing contrasts in **w** using $\hat{Var}_{BS}(\hat{\mathbf{w}})$ may not be straightforward. Therefore, on such data we will recommend utilizing bootstrap replicate samples to calculate confidence intervals (CIs) of contrasts and other statistics directly, rather than imposing distributional assumptions. First we must calculate the statistic of interest $\hat{\theta}_b$ on each bootstrap replicate sample *b*. It has been shown that if $\hat{\theta}$ is an asymptotically unbiased estimate of θ, then the α^{th} and $(1-\alpha)^{\text{th}}$ quantiles of the distribution of bootstrap replicate statistics $\hat{\theta}_b$, called a bootstrap percentile confidence interval, is an asymptotic confidence interval for θ.[76]

CHAPTER 5: EXPLORATORY ANALYSES OF DISTRIBUTIONS FOR HYPOTHESIS GENERATION

In the previous two chapters, we analyzed an artificial, clustered data set with VWUO-MD. We found that while the strength of the clustering was generally not improved (as assessed with dendrograms), the variable weights themselves were informative about which variables were related to the clusters in the data, and therefore to each other. Those data were constructed with MVN mixtures for type C variables and mixtures of multinomial distributions for the other types. Multidimensional clustering was achieved with an "unknown" latent group variable. In this chapter, we will perform Monte Carlo simulations to assess the performance of VWUO-MD on a series of artificial data sets constructed a little differently, and geared more directly towards the purpose of HG. We will assess each data type on its own. Every type T data set (where T=C, O, N, S or A) will have a known group variable *g* plus three variables, T_x, T_y and T_r. T_x and T_y will be related to each other according to a specific relationship (e.g., a quadratic), as well as possibly to group assignment *g*, while T_r will be unrelated to the other two variables and to *g*. Latent group *g* will be split into 1, 2, 3 or 4 levels according to prespecified (approximately equal) proportions, similar to a situation of stratified random sampling with *g* known. In the last example in each section, a situation of two disjoint relationships will be explored, with two pairs of related variables and one noise variable.

Each data set will be replicated 100 times, drawing from the prescribed distribution within each level of *g*. VWUO-MD will be performed on every replicated data set, and summary statistics for each set of variable weights obtained will be calculated and compared. Differences in performance between different shapes of data will be discussed.

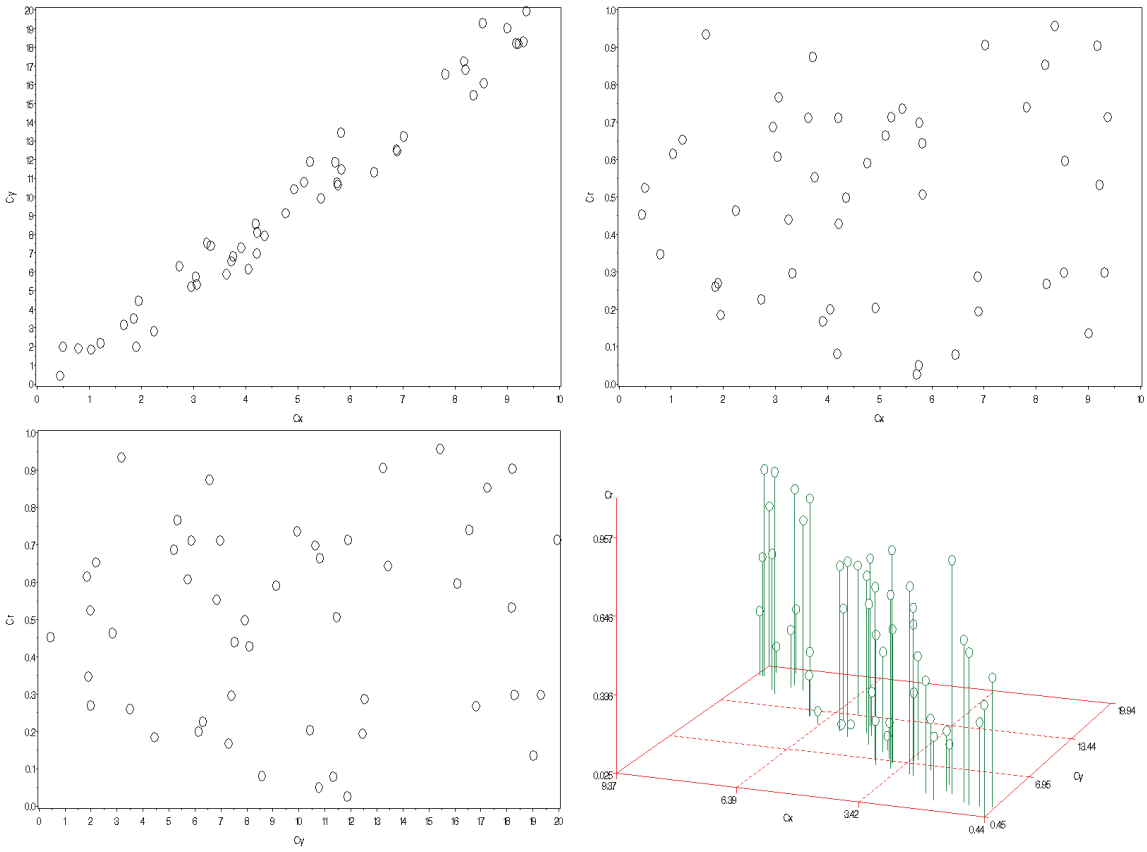
There are a lot of examples considered in this chapter, but most of the discussions below the examples are presented as succinct comparisons with previous examples. While every example except the last one in each section contains four graphs, the most important of these is the first one (or the first and the eighth in the last example of each section), plotting the two related variables against each other. Therefore one should not need to spend too much time on any one example. The goal of this chapter is to develop an overall idea of where VWUO-MD performs well, and where it falls short.

5.1 Type C data

We generated 40 type C data sets, replicated 100 times. Figure 46 to Figure 85 show the multivariate distributions of *g*, C_x, C_y and C_r in these data sets, as well as C_u and C_v in the last example. The captions describe the distributions. The number of groups is the number of levels of *g*. "Linear" describes a linear relationship between C_x and C_y. "Quadratic" describes a full parabolic relationship between C_x and C_y. "Half-quadratic" describes a half-parabolic relationship between C_x and C_y. "Correlated with" means the direction of each C_x versus C_y cluster plotted in two dimensions runs in the parallel direction as the placement of clusters. "Correlated against" means the direction

of each C_x versus C_y cluster plotted in two dimensions runs in the perpendicular direction to the placement of clusters. "Extra wide" means the group means have been moved farther apart relative to the data scale. "Small error" versus "large error" describe relative standard deviations in the normal error term for C_y. Details about each distribution are given in the footnote below its figure, then a very brief discussion of VWUO-MD's performance on that data set is made.

Figure 46. Type C, linear, 1 group, small error



C_x = a random Uniform(0,10)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

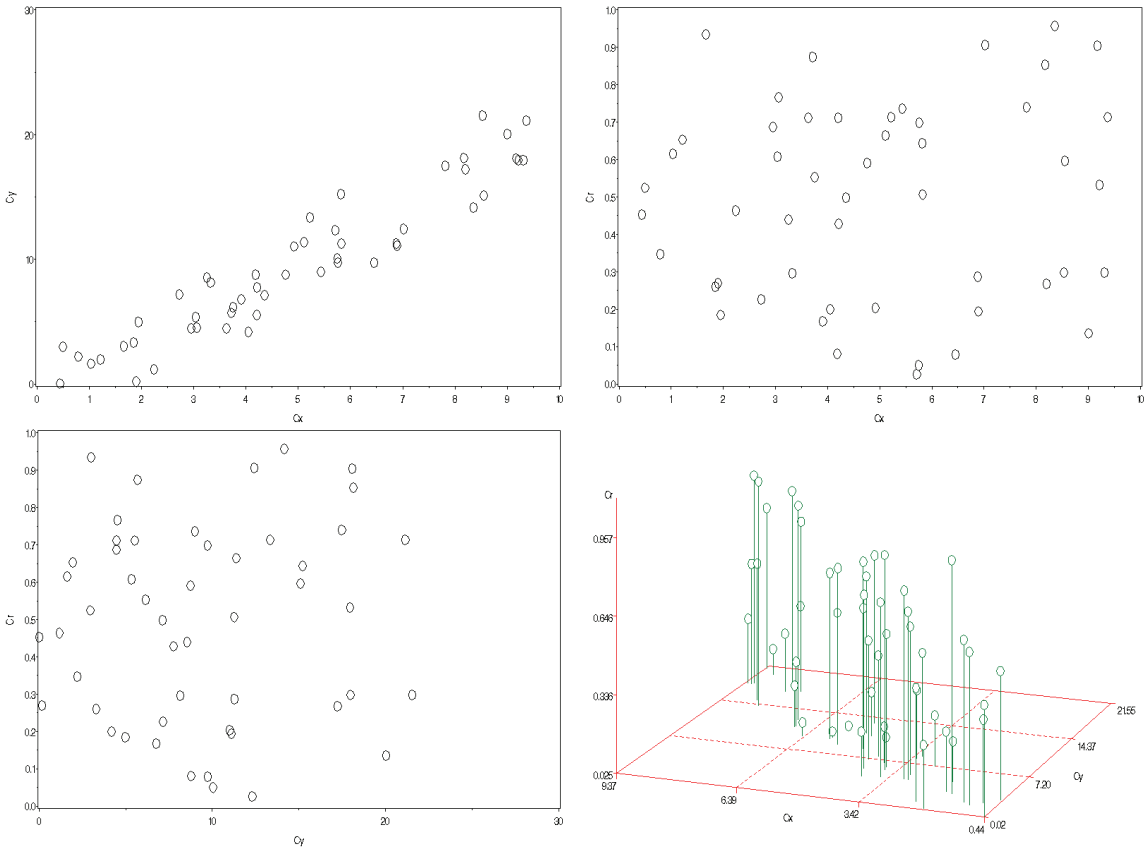
Table 25. Results for type C, linear, 1 group, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.924 (0.917, 0.931)	0.963 (0.955, 0.971)	1.113 (1.101, 1.125)
P5, P50, P95	0.866, 0.926, 0.947	0.904, 0.960, 0.990	1.014, 1.113, 1.152

[1] Confidence intervals are normal-based

VWUO-MD performs poorly for HG on linearly related variables with one group (no clusters). w_{Cr} is the highest weight assigned, which is the opposite of what should be desired. Even its 5th percentile is >1. In 1% of the replicates, w_{Cr} < w_{Cx} and w_{Cy}.

Figure 47. Type C, linear, 1 group, large error



C_x = a random Uniform(0,10)
C_y = 2*C_x + a random N(0,2) error
C_r = a random Uniform(0,1)

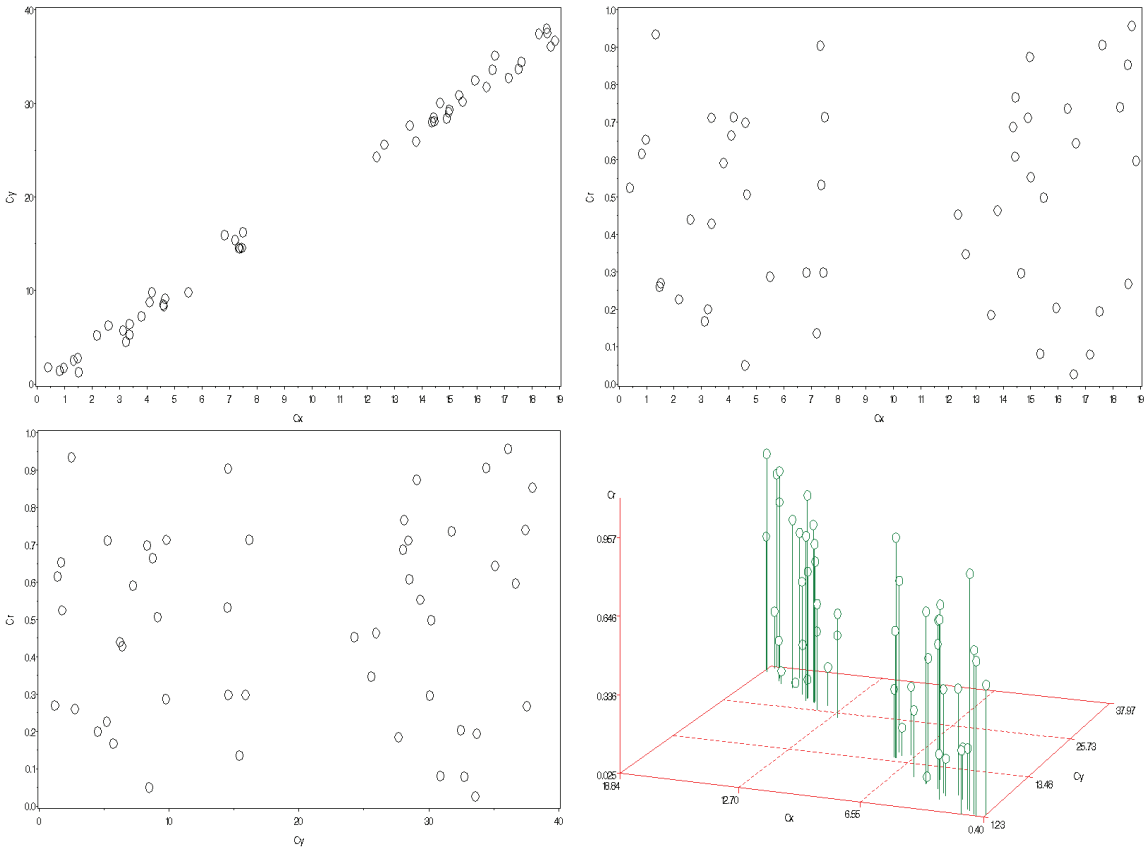
Table 26. Results for type C, linear, 1 group, large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.917 (0.909, 0.925)	1.009 (0.999, 1.019)	1.074 (1.062, 1.086)
P5, P50, P95	0.848, 0.918, 0.950	0.931, 1.004, 1.041	0.984, 1.069, 1.121

[1] Confidence intervals are normal-based

VWUO-MD performs poorly, but less so when the error term is increased, which is sensible since it weakens the linear relationship that was problematic in the previous example. In 4% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 48. Type C, linear, 2 groups (correlated with), small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,8) + 12*I(G2)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

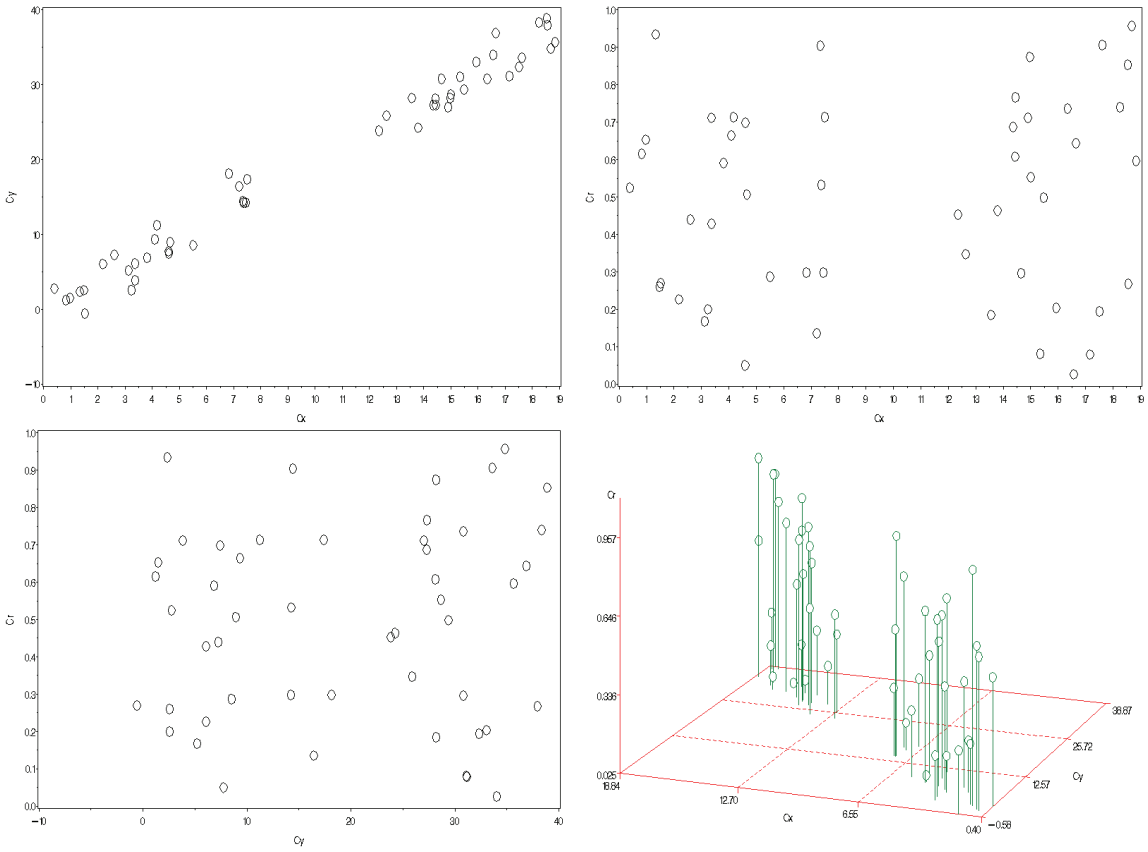
Table 27. Results for type C, linear, 2 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.975 (0.967, 0.983)	0.989 (0.981, 0.997)	1.036 (1.021, 1.050)
P5, P50, P95	0.907, 0.974, 0.999	0.925, 0.988, 1.012	0.919, 1.039, 1.090

[1] Confidence intervals are normal-based

VWUO-MD performs poorly, but less so than the previous two examples (at least considering the magnitude of w_{Cr}), when the related variables are linearly related and correlated parallel to the placement of the two clusters. In 27% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 49. Type C, linear, 2 groups (correlated with), large error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,8) + 12*I(G2)
C_y = 2*C_x + a random N(0,2) error
C_r = a random Uniform(0,1)

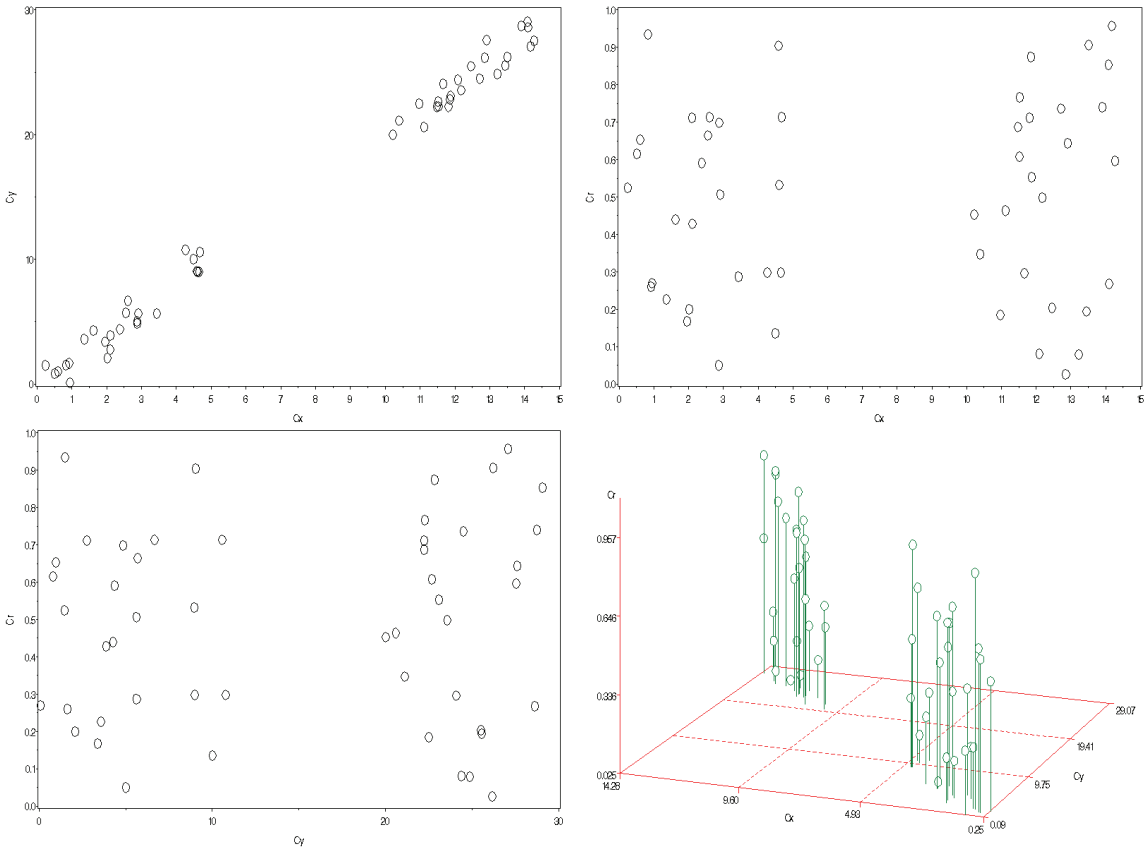
Table 28. Results for type C, linear, 2 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.973 (0.964, 0.982)	1.003 (0.995, 1.012)	1.024 (1.010, 1.038)
P5, P50, P95	0.905, 0.973, 1.001	0.926, 1.001, 1.038	0.918, 1.033, 1.077

[1] Confidence intervals are normal-based

Once again VWUO-MD sensibly performs less poorly when the error is increased. In 30% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 50. Type C, linear, 2 groups (correlated with), extra wide, small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,5) + 10*I(G2)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

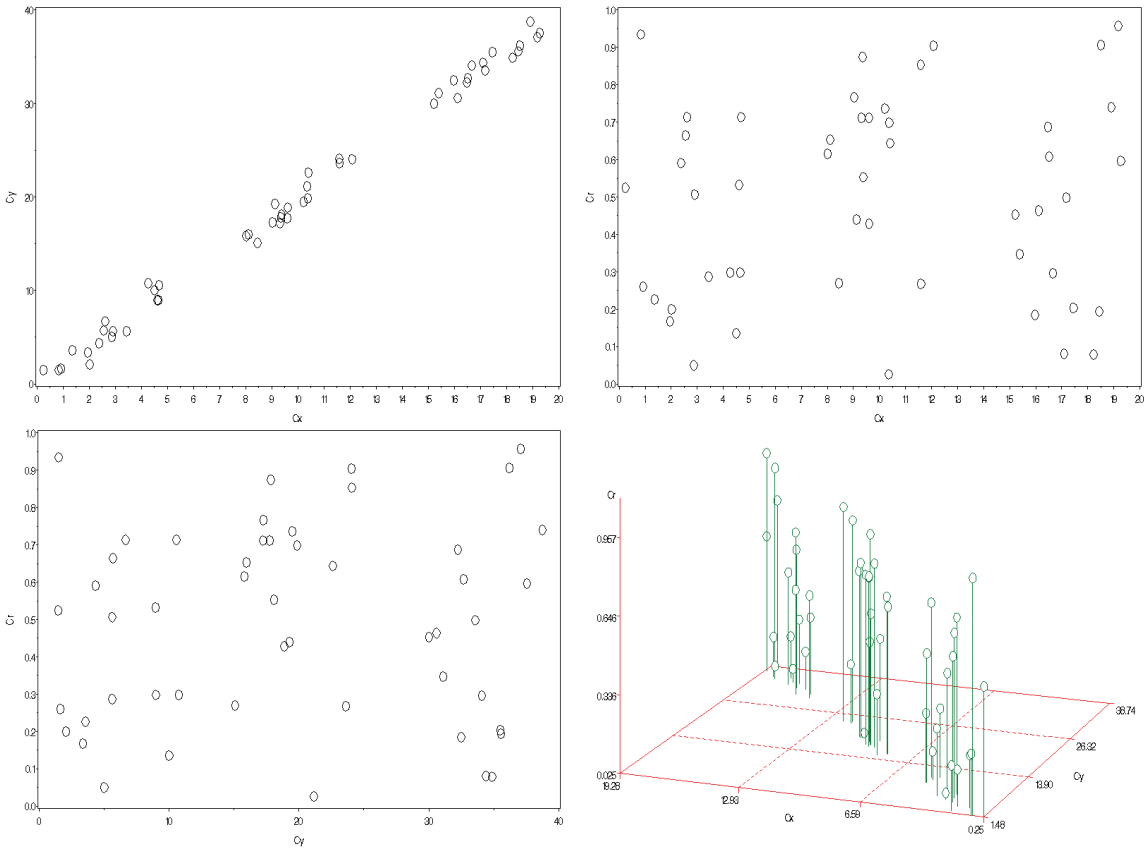
Table 29. Results for type C, linear, 2 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.068 (1.059, 1.077)	1.065 (1.057, 1.074)	0.867 (0.852, 0.881)
P5, P50, P95	0.992, 1.069, 1.096	0.994, 1.065, 1.092	0.747, 0.878, 0.918

[1] Confidence intervals are normal-based

VWUO-MD performs well, once the two clusters are separated enough. In 94% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}. The noise variable receives the lowest average weight, and even the 95th percentile of w_{Cr} is <1. This might indicate that correlation within clusters is the problem, since wider spread effectively reduces the relative strength of the intracluster correlation via compression of the clusters.

Figure 51. Type C, linear, 3 groups (correlated with), small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,5) + 7.5*I(G2) + 15*I(G3)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

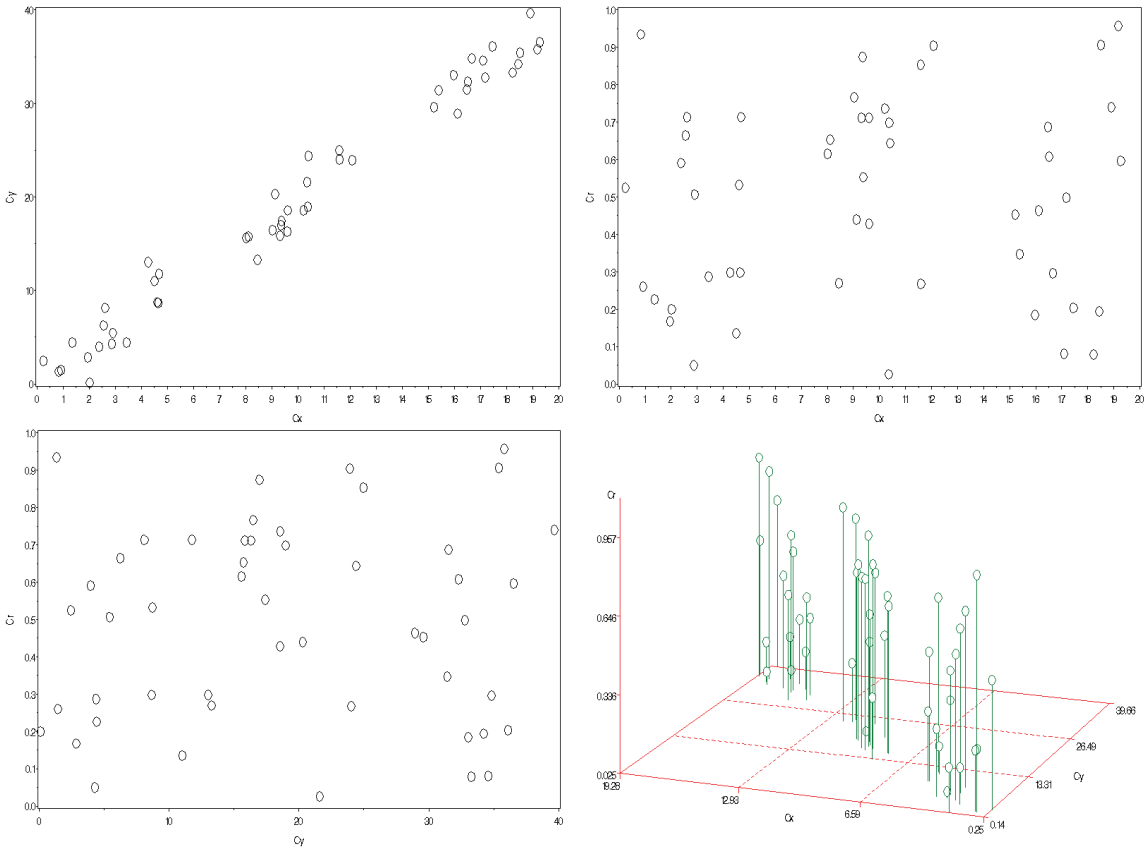
Table 30. Results for type C, linear, 3 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.884 (0.879, 0.890)	0.907 (0.901, 0.913)	1.209 (1.198, 1.220)
P5, P50, P95	0.841, 0.885, 0.899	0.863, 0.906, 0.931	1.127, 1.212, 1.248

[1] Confidence intervals are normal-based

VWUO-MD performs poorly, when three clusters are closely placed, each correlated parallel to cluster placement. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 52. Type C, linear, 3 groups (correlated with), large error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,5) + 7.5*I(G2) + 15*I(G3)
C_y = 2*C_x + a random N(0,2) error
C_r = a random Uniform(0,1)

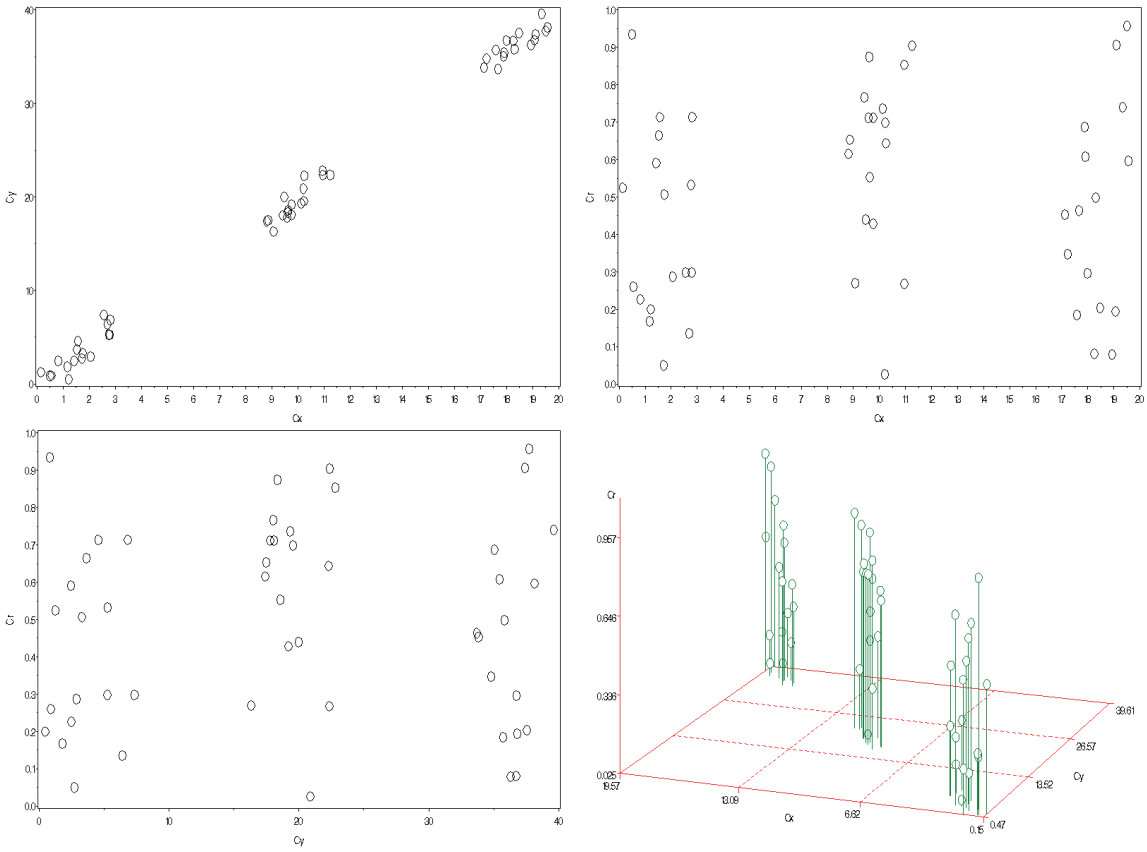
Table 31. Results for type C, linear, 3 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.876 (0.870, 0.882)	0.937 (0.929, 0.945)	1.187 (1.176, 1.198)
P5, P50, P95	0.827, 0.877, 0.892	0.881, 0.939, 0.964	1.093, 1.188, 1.225

[1] Confidence intervals are normal-based

As before, VWUO-MD performs less poorly (considering the magnitude of average weights) when the error term is increased. In 0% of the replicates, w_{Cr} < w_{Cx} and w_{Cy}.

Figure 53. Type C, linear, 3 groups (correlated with), extra wide, small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,3) + 8.5*I(G2) + 17*I(G3)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

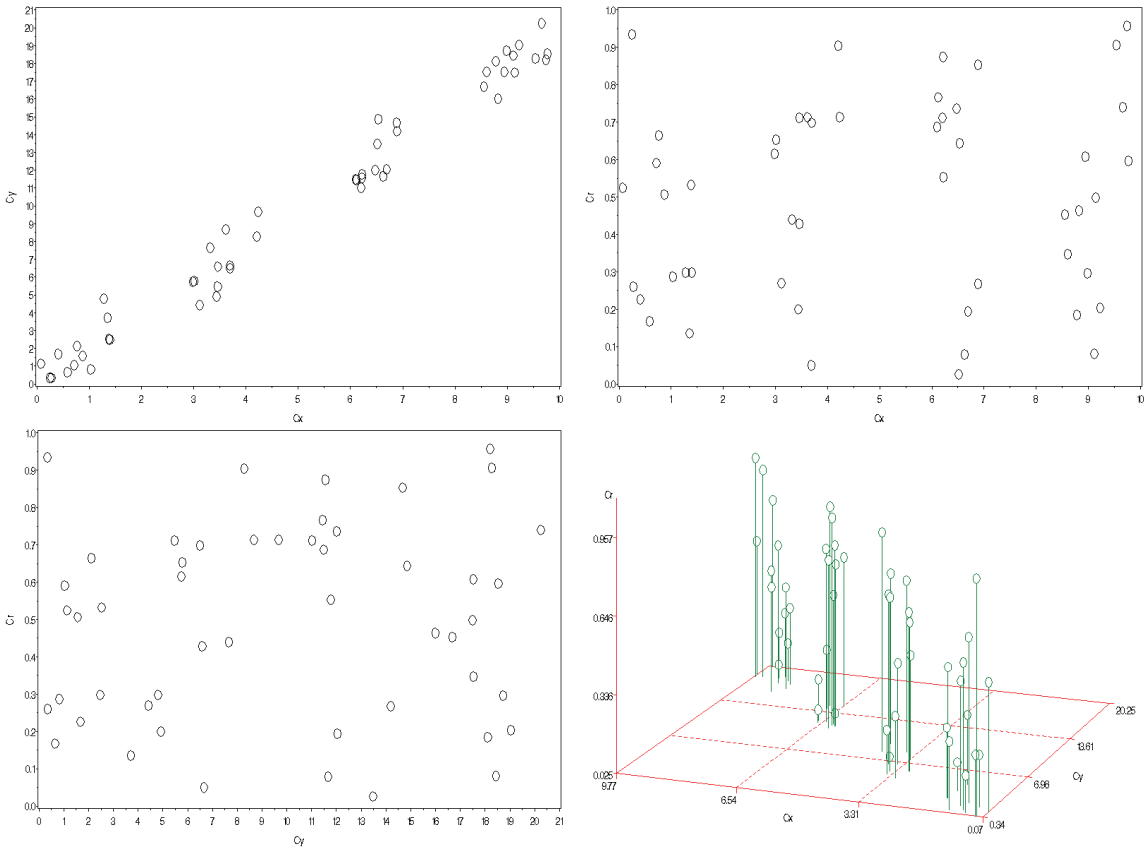
Table 32. Results for type C, linear, 3 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.829 (0.824, 0.835)	0.863 (0.857, 0.870)	1.307 (1.296, 1.319)
P5, P50, P95	0.783, 0.829, 0.849	0.805, 0.861, 0.889	1.216, 1.312, 1.348

[1] Confidence intervals are normal-based

VWUO-MD performs extremely poorly with three clusters linearly placed in the related variables plane. That the performance is worse this time (looking at magnitude) with wider spread may indicate that the correlation is not the culprit with three clusters (as it may have been with two clusters), but rather, the cluster placement itself. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 54. Type C, linear, 4 groups (correlated with), small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1.5) + 2.83*I(G2) + 5.66*I(G3) + 8.49*I(G4)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

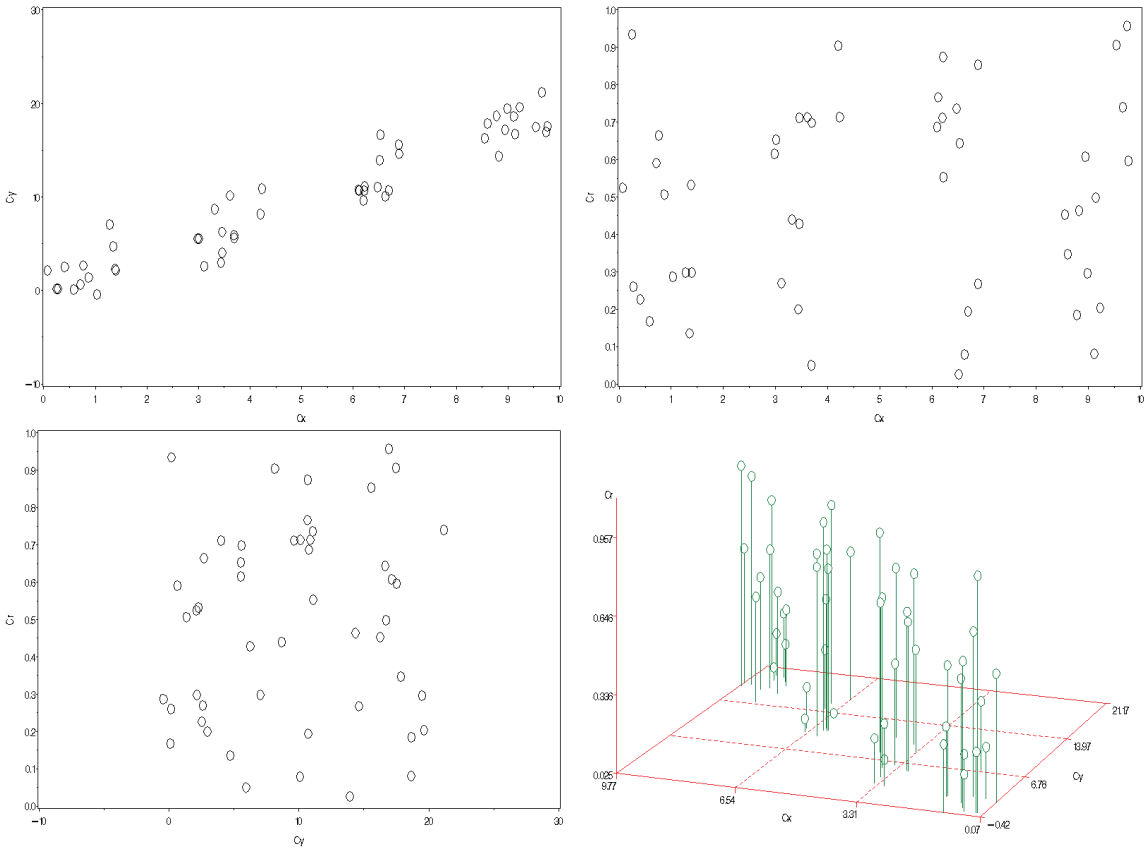
Table 33. Results for type C, linear, 4 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.871 (0.866, 0.877)	0.938 (0.930, 0.946)	1.191 (1.179, 1.202)
P5, P50, P95	0.825, 0.870, 0.891	0.873, 0.934, 0.965	1.088, 1.194, 1.232

[1] Confidence intervals are normal-based

VWUO-MD continues its poor performance when four clusters are linearly placed in the related variables plane. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 55. Type C, linear, 4 groups (correlated with), large error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1.5) + 2.83*I(G2) + 5.66*I(G3) + 8.49*I(G4)
C_y = 2*C_x + a random N(0,2) error
C_r = a random Uniform(0,1)

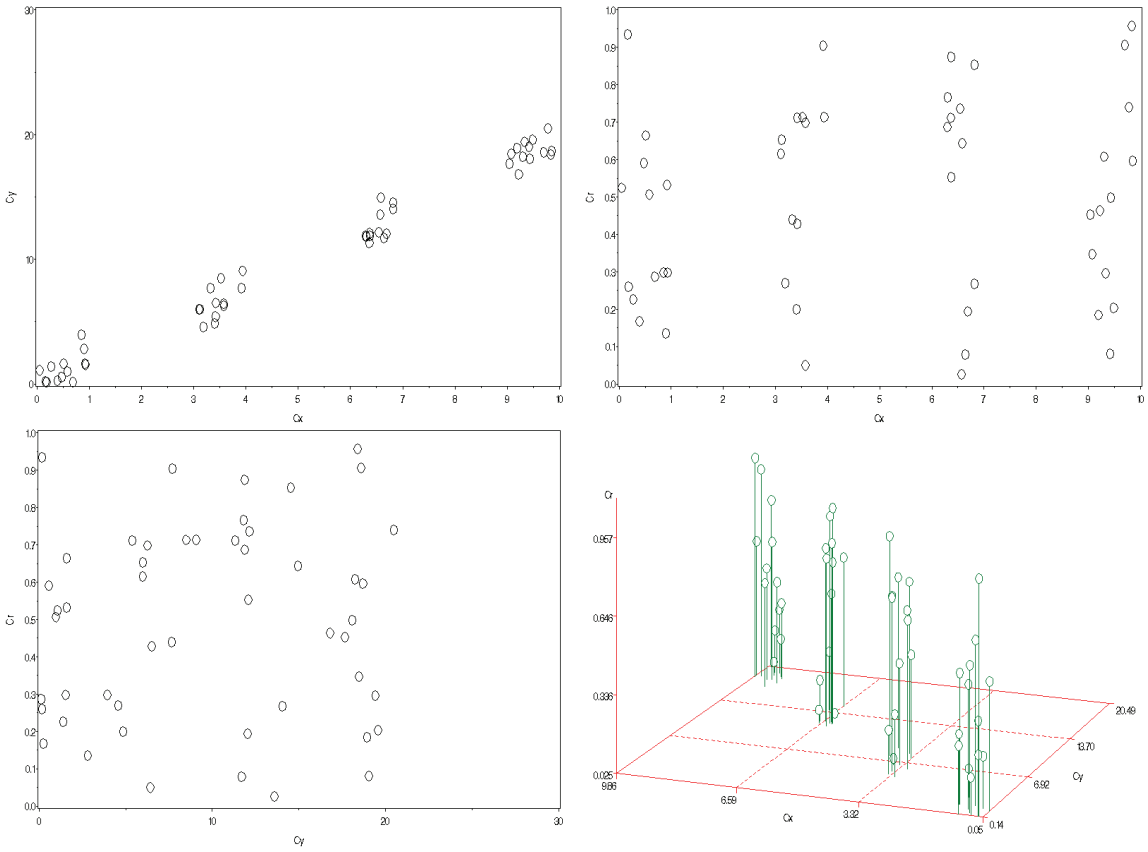
Table 34. Results for type C, linear, 4 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.856 (0.850, 0.862)	1.001 (0.991, 1.012)	1.142 (1.131, 1.154)
P5, P50, P95	0.802, 0.857, 0.877	0.918, 1.000, 1.040	1.041, 1.149, 1.191

[1] Confidence intervals are normal-based

As before, VWUO-MD performs less poorly (looking at magnitude) when the error term is increased. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 56. Type C, linear, 4 groups (correlated with), extra wide, small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1) + 3*I(G2) + 6*I(G3) + 9*I(G4)
C_y = 2*C_x + a random N(0,1) error
C_r = a random Uniform(0,1)

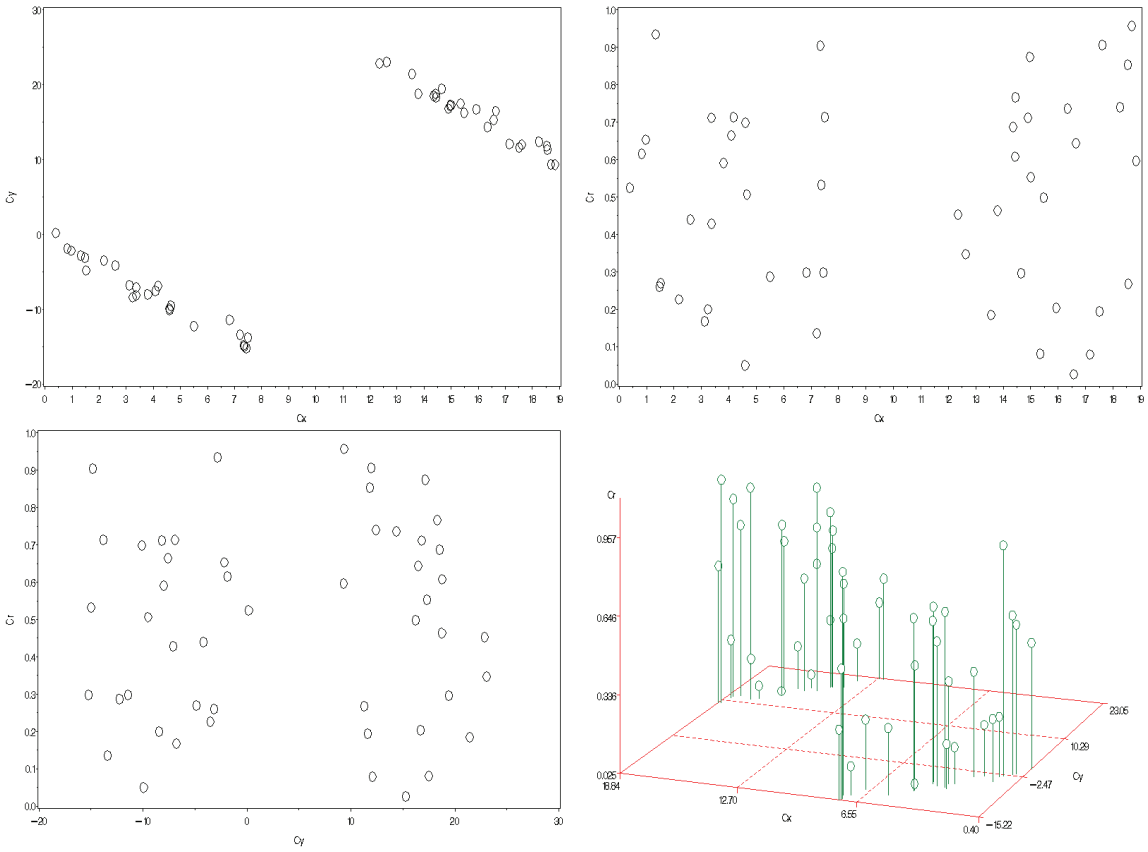
Table 35. Results for type C, linear, 4 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.847 (0.842, 0.852)	0.932 (0.924, 0.939)	1.221 (1.210, 1.231)
P5, P50, P95	0.803, 0.847, 0.868	0.871, 0.932, 0.957	1.125, 1.222, 1.260

[1] Confidence intervals are normal-based

Increasing the cluster spread has only exacerbated (looking at magnitude) the problems VWUO-MD is having with this shape of data. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 57. Type C, linear, 2 groups (correlated against), small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,8) + 12*I(G2)
C_y = -2*C_x + a random N(0,1) error + 48*I(G2)
C_r = a random Uniform(0,1)

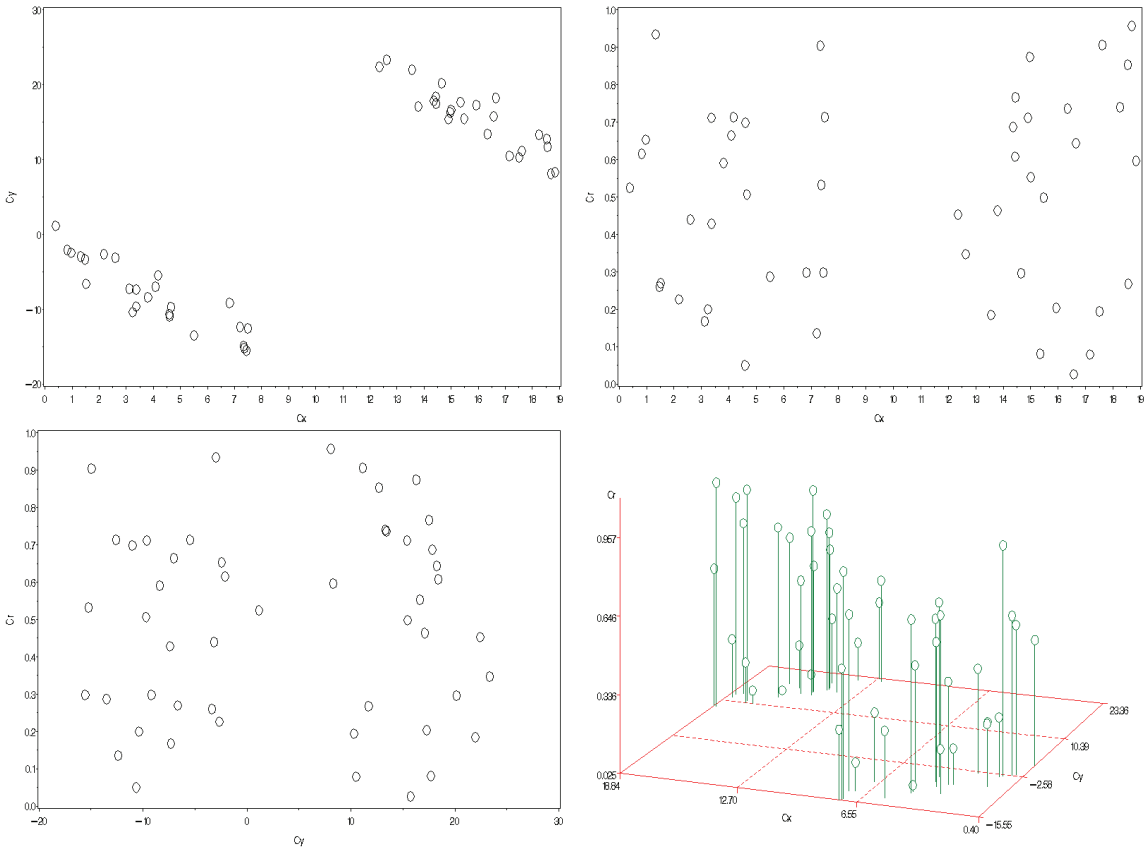
Table 36. Results for type C, linear, 2 groups (correlated against), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.311 (1.301, 1.321)	1.345 (1.336, 1.354)	0.344 (0.337, 0.351)
P5, P50, P95	1.234, 1.306, 1.345	1.275, 1.344, 1.371	0.282, 0.344, 0.361

[1] Confidence intervals are normal-based

Correlating the two clusters perpendicular to the cluster placement has dramatically improved VWUO-MD's ability to detect a relationship between C_x and C_y. Unfortunately, this particular sort of relationship may not be very common in nature. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 58. Type C, linear, 2 groups (correlated against), large error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,8) + 12*I(G2)
C_y = -2*C_x + a random N(0,2) error + 48*I(G2)
C_r = a random Uniform(0,1)

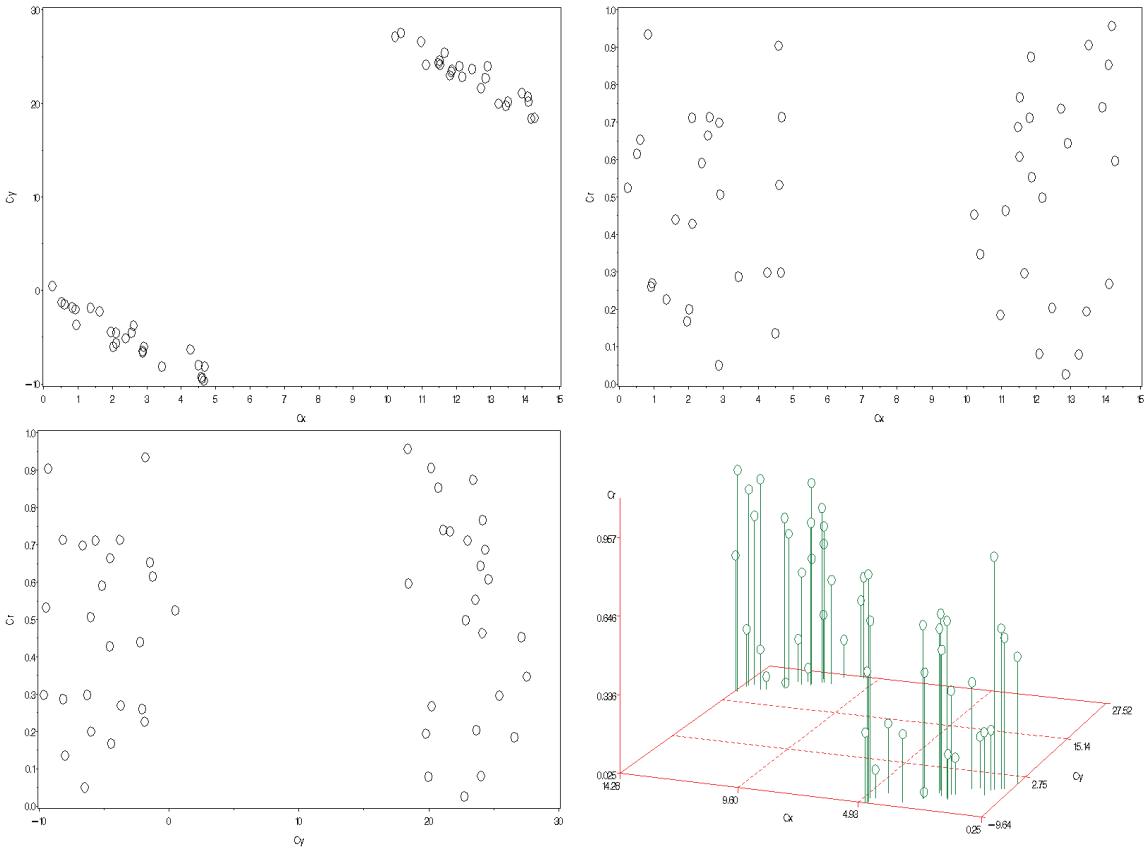
Table 37. Results for type C, linear, 2 groups (correlated against), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.273 (1.262, 1.285)	1.342 (1.330, 1.353)	0.385 (0.378, 0.393)
P5, P50, P95	1.180, 1.270, 1.311	1.241, 1.342, 1.388	0.322, 0.385, 0.406

[1] Confidence intervals are normal-based

In this case the bigger error term has hardly diminished (looking at magnitude) VWUO-MD's ability to detect the relationship between C_x and C_y. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 59. Type C, linear, 2 groups (correlated against), extra wide, small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,5) + 10*I(G2)
C_y = -2*C_x + a random N(0,1) error + 48*I(G2)
C_r = a random Uniform(0,1)

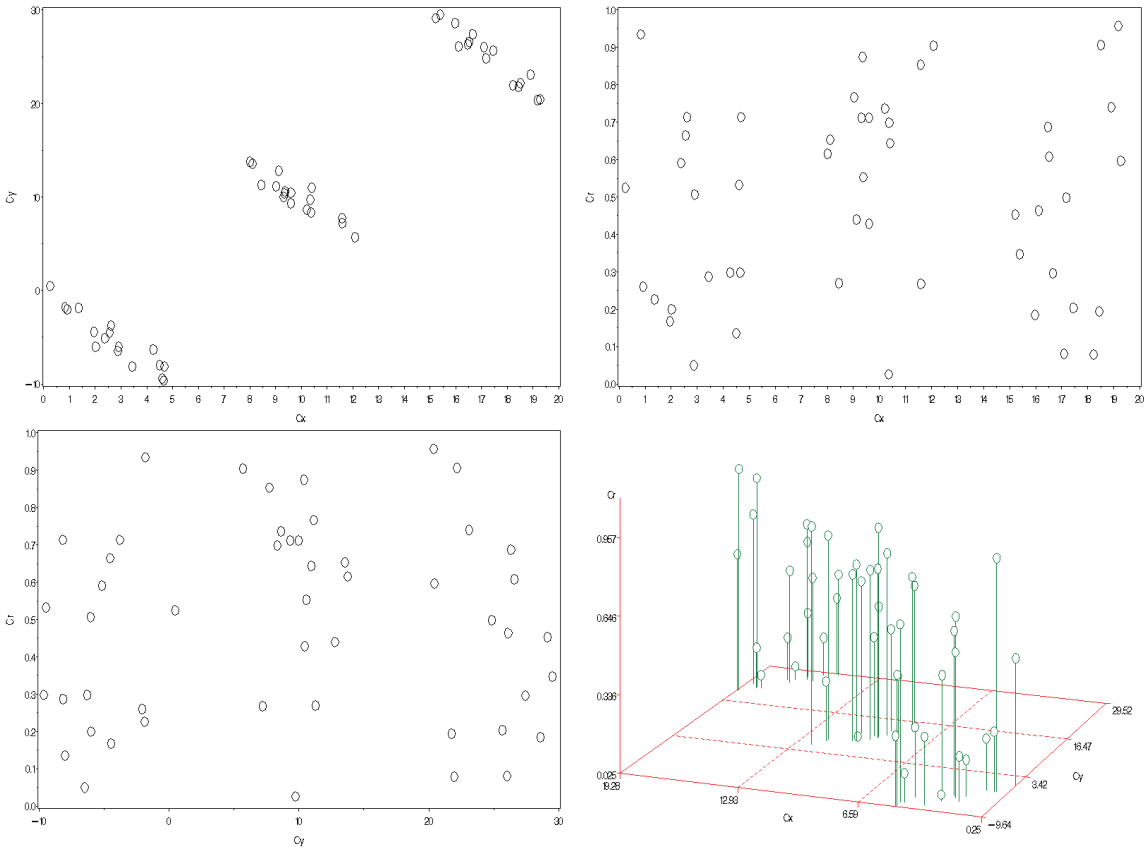
Table 38. Results for type C, linear, 2 groups (correlated against), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.254 (1.246, 1.262)	1.488 (1.480, 1.496)	0.258 (0.253, 0.263)
P5, P50, P95	1.188, 1.249, 1.280	1.421, 1.488, 1.515	0.215, 0.258, 0.273

[1] Confidence intervals are normal-based

Here we have an intuitive result, the wider spread of the clusters has further increased (looking at magnitude) VWUO-MD's ability to detect the relationship. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 60. Type C, linear, 3 groups (correlated against), small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,5) + 7.5*I(G2) + 15*I(G3)
C_y = -2*C_x + a random N(0,1) error + 30*I(G2) + 60*I(G3)
C_r = a random Uniform(0,1)

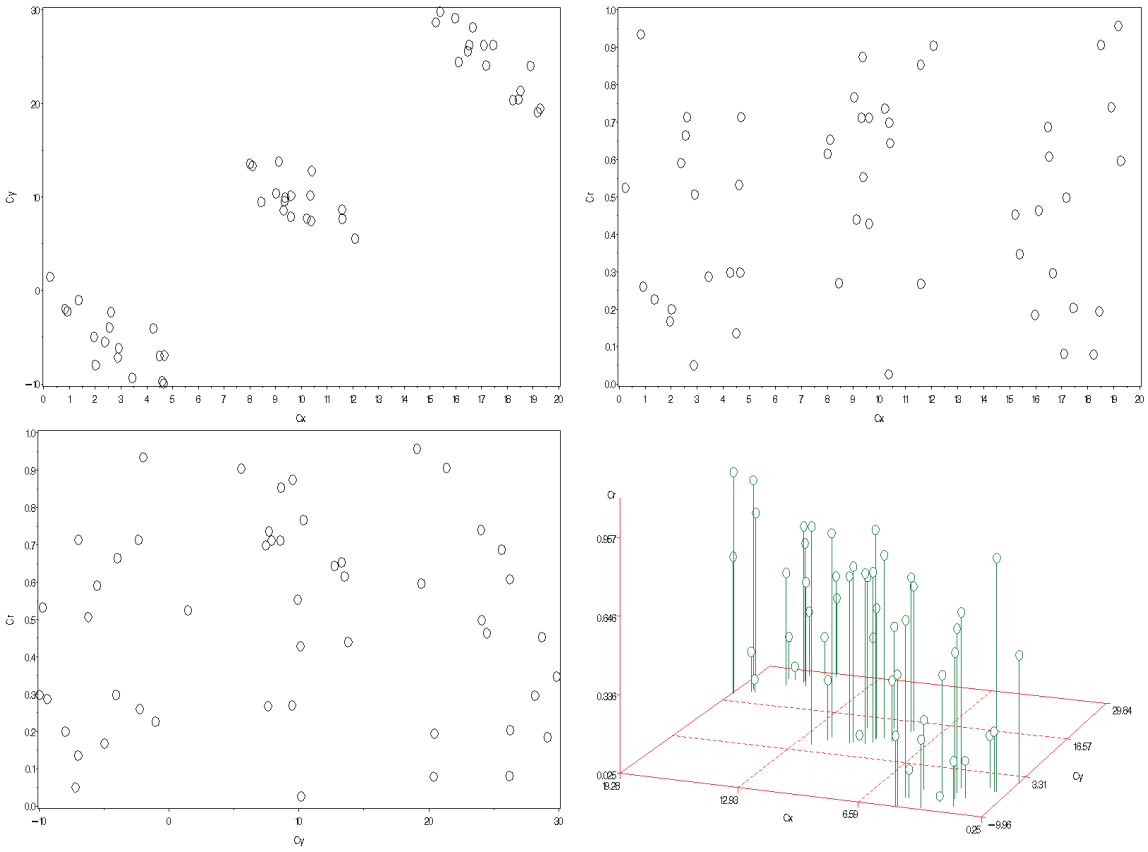
Table 39. Results for type C, linear, 3 groups (correlated against), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.883 (0.875, 0.891)	0.905 (0.897, 0.914)	1.211 (1.200, 1.223)
P5, P50, P95	0.817, 0.885, 0.910	0.839, 0.903, 0.937	1.113, 1.205, 1.249

[1] Confidence intervals are normal-based

This appears to affirm our earlier suspicion that with *three* clusters, cluster placement in a linear fashion is tripping up VWUO-MD's ability to detect the real relationship. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 61. Type C, linear, 3 groups (correlated against), large error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,5) + 7.5*I(G2) + 15*I(G3)
C_y = -2*C_x + a random N(0,2) error + 30*I(G2) + 60*I(G3)
C_r = a random Uniform(0,1)

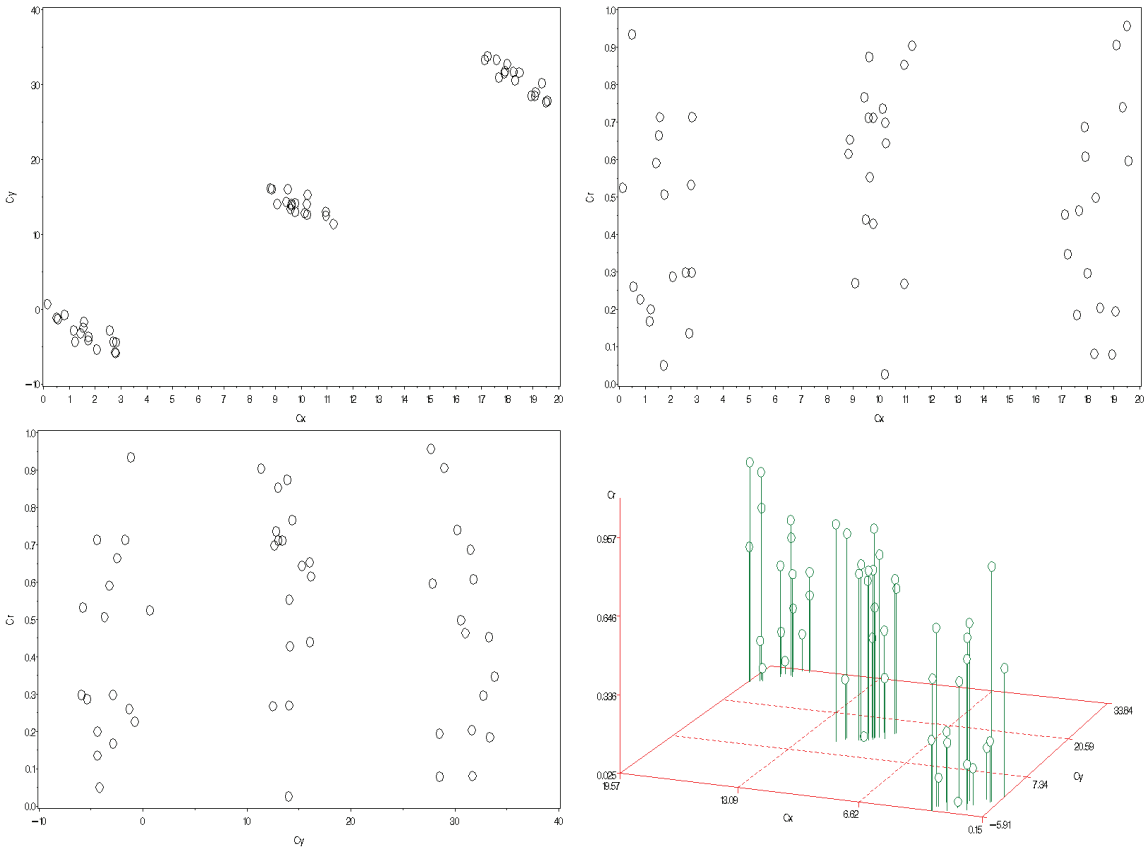
Table 40. Results for type C, linear, 3 groups (correlated against), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.874 (0.867, 0.882)	0.944 (0.934, 0.953)	1.182 (1.170, 1.193)
P5, P50, P95	0.806, 0.873, 0.901	0.870, 0.943, 0.978	1.081, 1.176, 1.222

[1] Confidence intervals are normal-based

As before, VWUO-MD performs less poorly (looking at magnitude) when the error term is increased. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 62. Type C, linear, 3 groups (correlated against), extra wide, small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,3) + 8.5*I(G2) + 17*I(G3)
C_y = -2*C_x + a random N(0,1) error + 34*I(G2) + 68*I(G3)
C_r = a random Uniform(0,1)

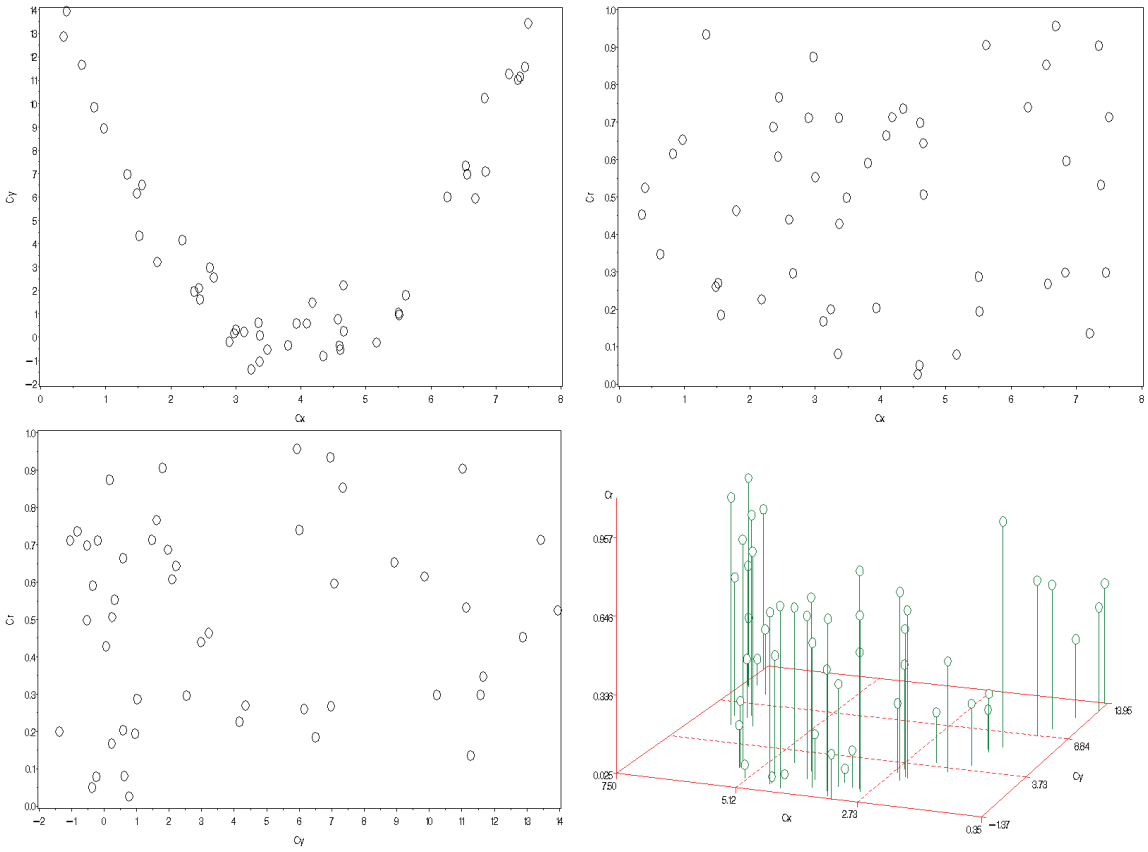
Table 41. Results for type C, linear, 3 groups (correlated against), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.821 (0.815, 0.827)	0.855 (0.848, 0.862)	1.324 (1.313, 1.335)
P5, P50, P95	0.766, 0.821, 0.842	0.800, 0.853, 0.878	1.229, 1.321, 1.362

[1] Confidence intervals are normal-based

The wider spread of clusters only further exacerbates (looking at magnitude) the problem for VWUO-MD, by now an expected result for this shape of data. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 63. Type C, quadratic, 1 group, small error



C_x = a random Uniform(0,8)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

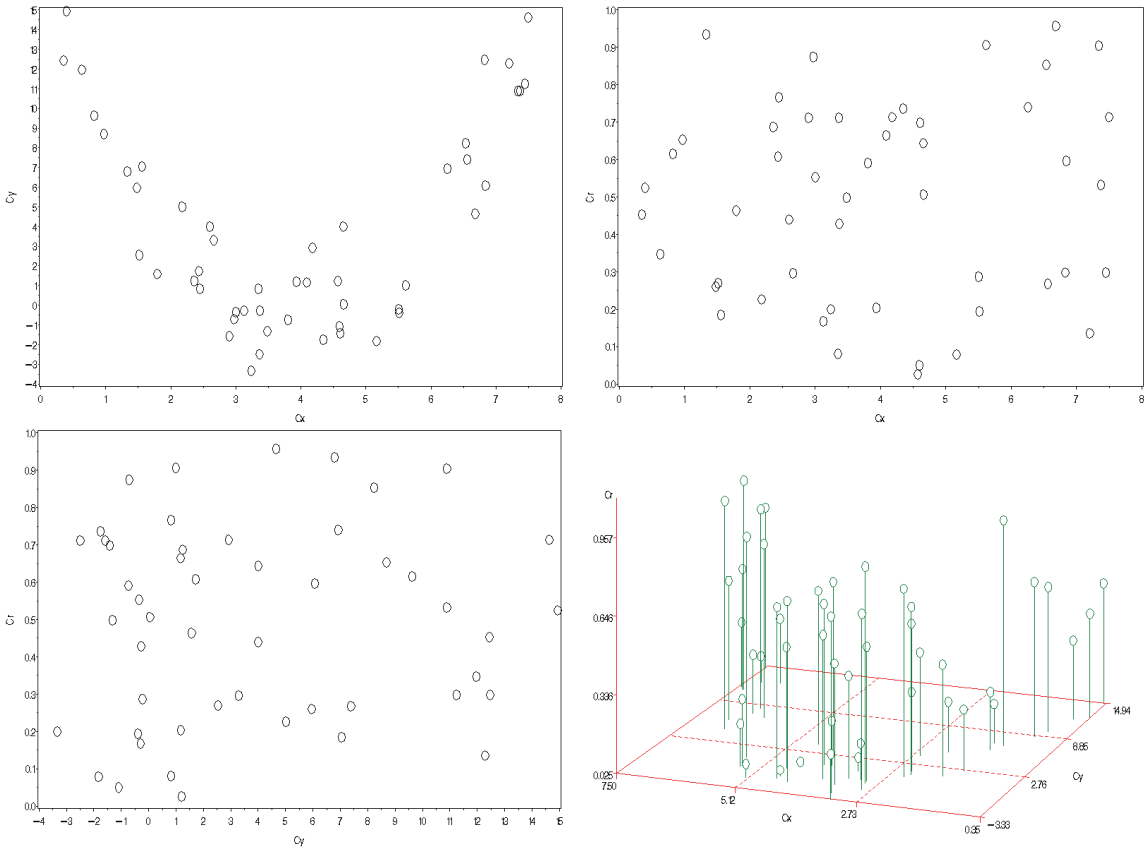
Table 42. Results for type C, quadratic, 1 group, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.053 (1.042, 1.063)	1.152 (1.138, 1.166)	0.795 (0.783, 0.806)
P5, P50, P95	0.955, 1.054, 1.094	1.029, 1.144, 1.193	0.698, 0.793, 0.831

[1] Confidence intervals are normal-based

This result is quite interesting. With no distinct clusters, and a quadratic relationship (full parabola) between C_x and C_y, VWUO-MD successfully detects this relationship and downweights the noise variable C_r. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 64. Type C, quadratic, 1 group, large error



C_x = a random Uniform(0,8)
$C_y = (C_x - 4)^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

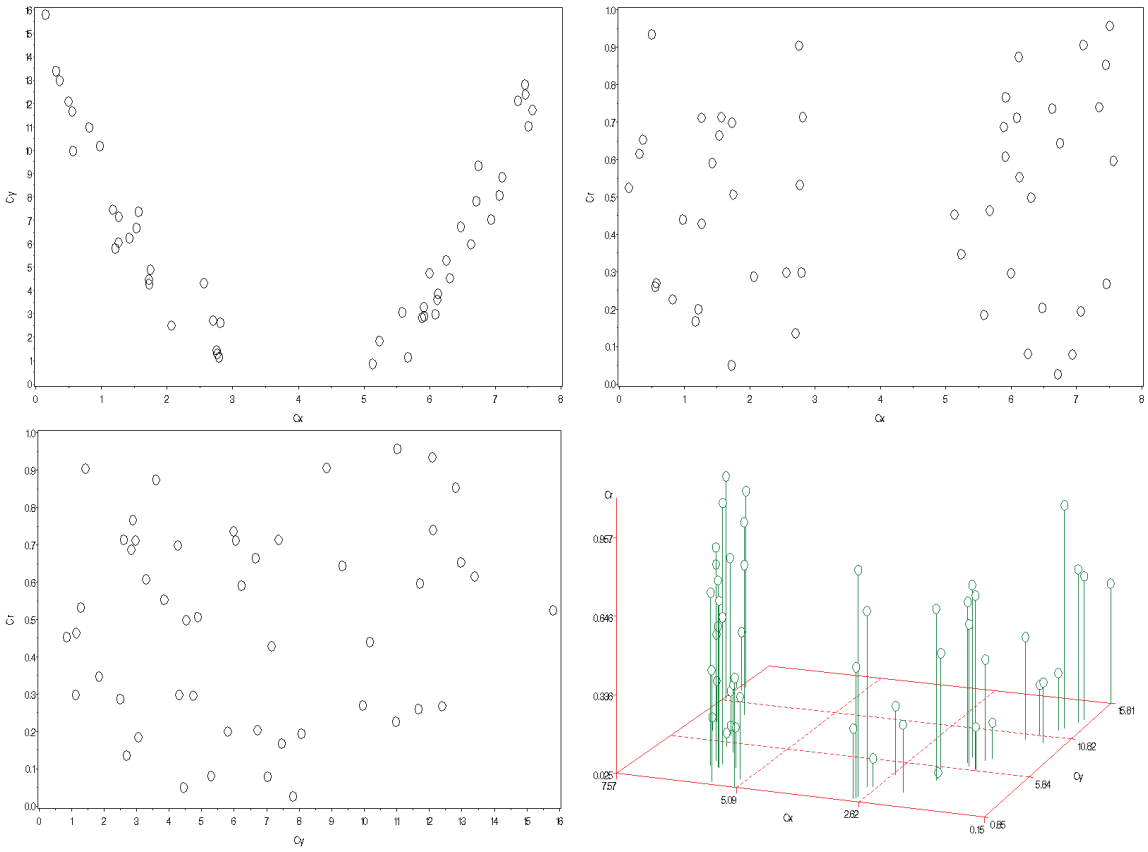
Table 43. Results for type C, quadratic, 1 group, large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.003 (0.992, 1.015)	1.189 (1.174, 1.204)	0.808 (0.796, 0.819)
P5, P50, P95	0.895, 0.993, 1.042	1.072, 1.187, 1.242	0.715, 0.803, 0.845

[1] Confidence intervals are normal-based

Increasing the error term has only slightly decreased the strength (looking at magnitude) of the result. In 99% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 65. Type C, quadratic, 2 groups (correlated with), small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,3) + 5*I(G2)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

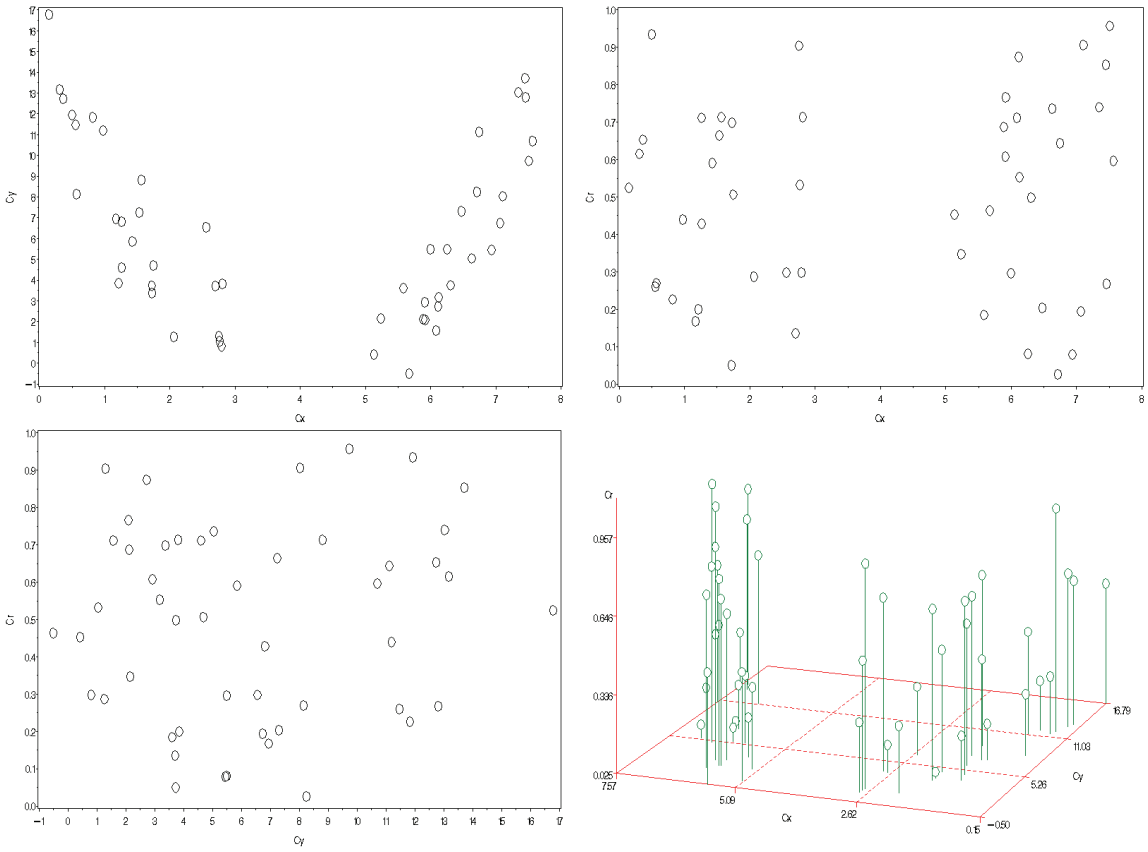
Table 44. Results for type C, quadratic, 2 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.035 (1.024, 1.045)	1.120 (1.107, 1.133)	0.846 (0.834, 0.857)
P5, P50, P95	0.950, 1.041, 1.071	1.002, 1.122, 1.158	0.766, 0.841, 0.893

[1] Confidence intervals are normal-based

With two clusters formed from parts of the previous quadratic shape, VWUO-MD continues to detect the relationship with about the same result. In 97% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 66. Type C, quadratic, 2 groups (correlated with), large error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,3) + 5*I(G2)
$C_y = (C_x - 4)^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

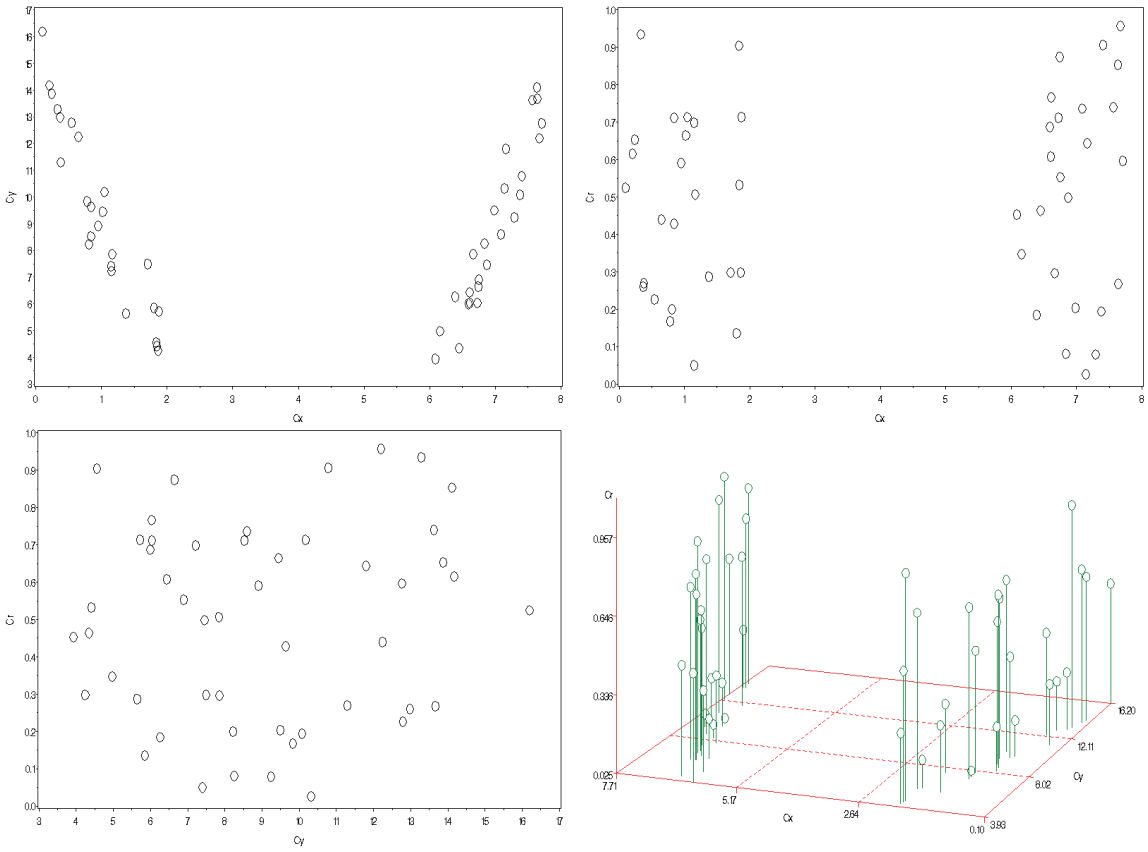
Table 45. Results for type C, quadratic, 2 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.002 (0.990, 1.014)	1.161 (1.146, 1.176)	0.837 (0.826, 0.849)
P5, P50, P95	0.905, 1.000, 1.044	1.028, 1.169, 1.213	0.753, 0.838, 0.883

[1] Confidence intervals are normal-based

This time the increase in error term has no discernable effect, at least in these 100 replicates. In 97% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 67. Type C, quadratic, 2 groups (correlated with), extra wide, small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,2) + 6*I(G2)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

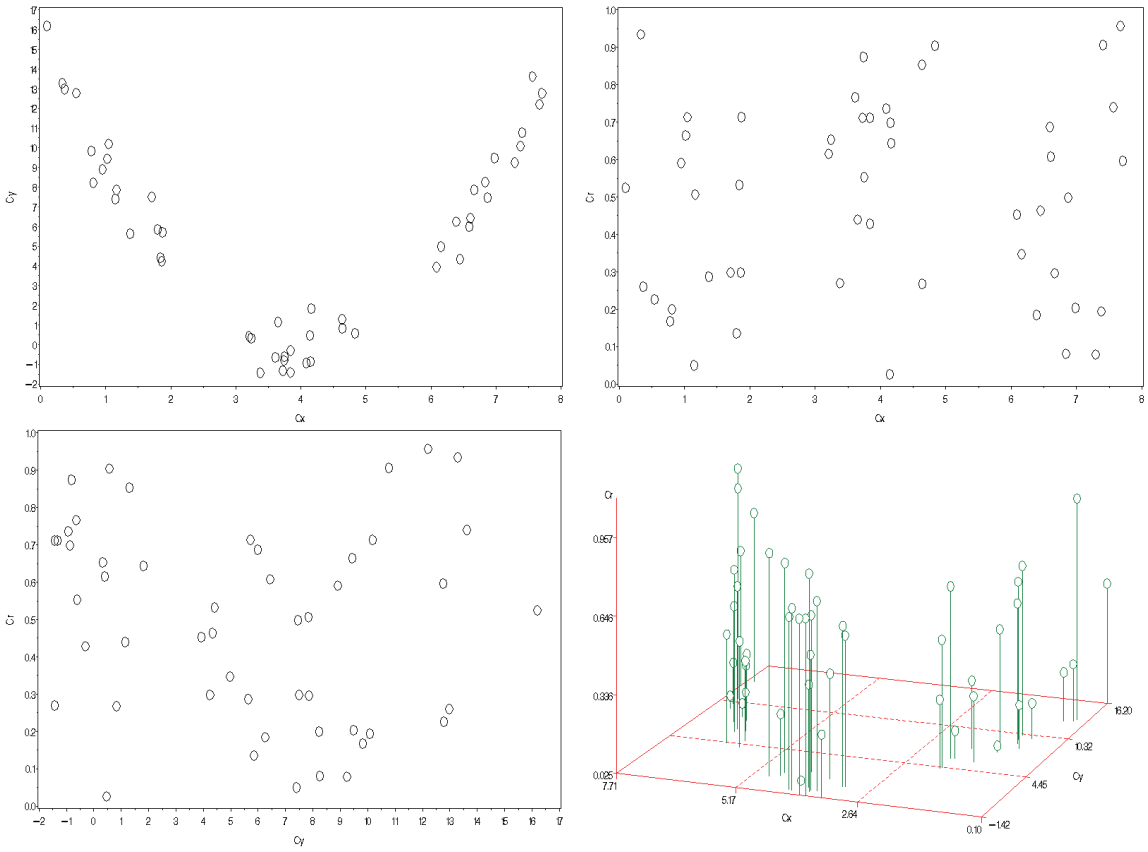
Table 46. Results for type C, quadratic, 2 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.317 (1.298, 1.337)	0.929 (0.915, 0.943)	0.754 (0.739, 0.768)
P5, P50, P95	1.164, 1.323, 1.391	0.802, 0.935, 0.980	0.628, 0.760, 0.798

[1] Confidence intervals are normal-based

Increasing cluster spread improves VWUO-MD's ability to detect the relationship (looking at magnitude). This seems to indicates that the previous result was an anomaly, and of course it is clear that increasing error must eventually catch up with any estimator. In 95% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 68. Type C, quadratic, 3 groups (correlated with), small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,2) + 3*I(G2) + 6*I(G3)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

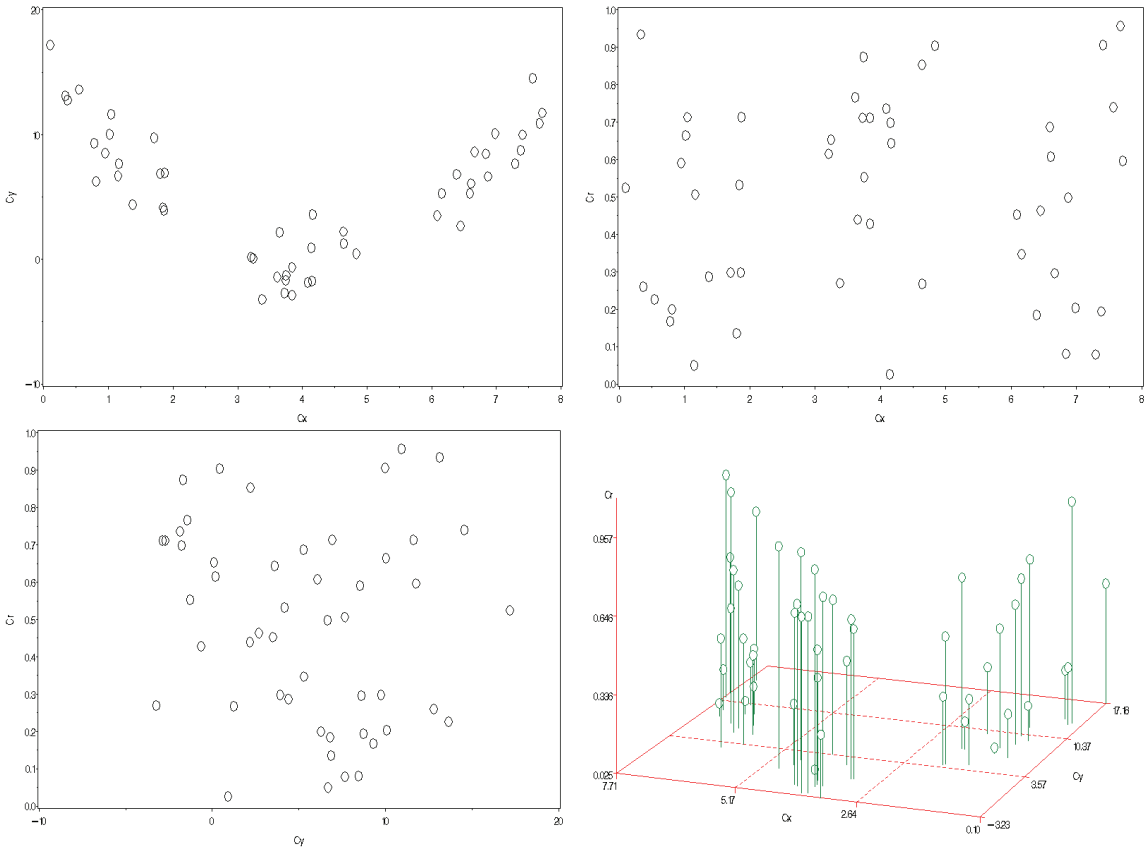
Table 47. Results for type C, quadratic, 3 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.106 (1.097, 1.116)	1.102 (1.090, 1.113)	0.792 (0.781, 0.803)
P5, P50, P95	1.017, 1.112, 1.139	1.001, 1.104, 1.141	0.707, 0.790, 0.833

[1] Confidence intervals are normal-based

Having three clusters in a quadratic spread does not diminish VWUO-MD's ability to detect the relationship, and seems to strengthen it when compared to the example with two clusters (non-widened spread). In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 69. Type C, quadratic, 3 groups (correlated with), large error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,2) + 3*I(G2) + 6*I(G3)
$C_y = (C_x - 4)^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

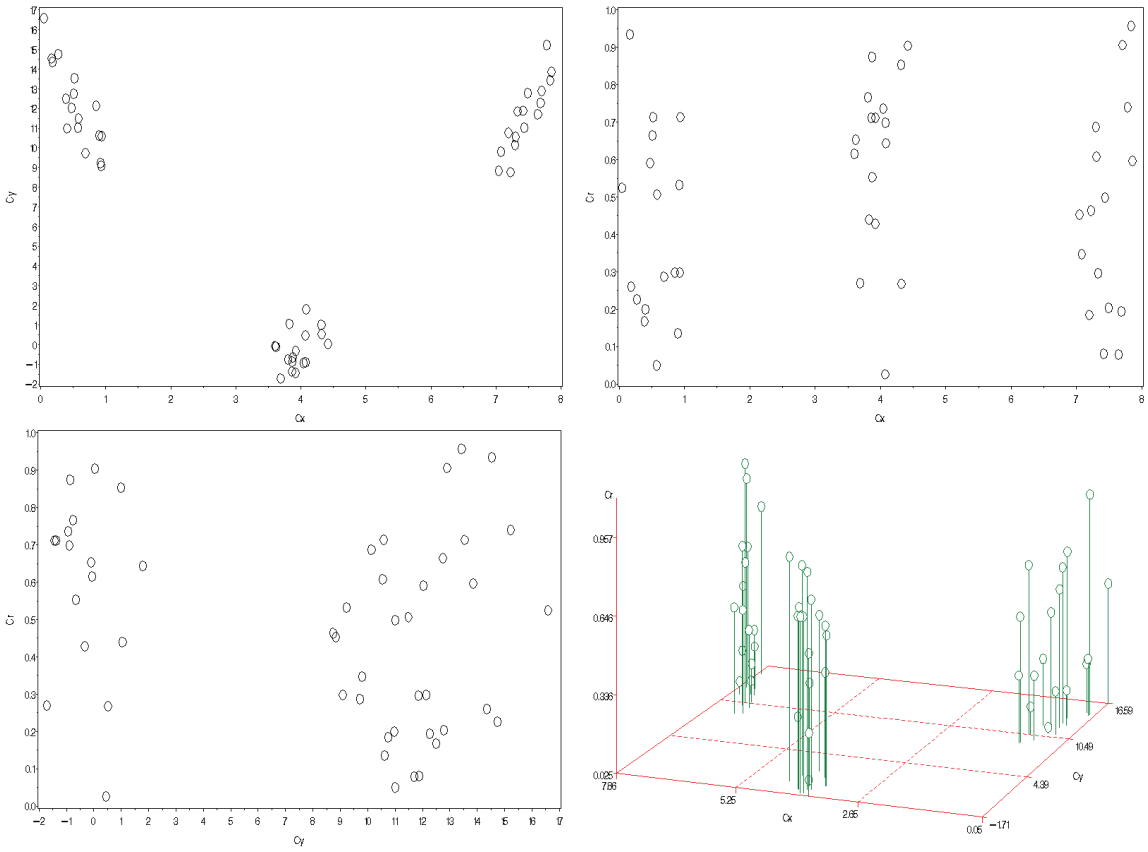
Table 48. Results for type C, quadratic, 3 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.024 (1.013, 1.036)	1.175 (1.160, 1.190)	0.801 (0.790, 0.811)
P5, P50, P95	0.922, 1.027, 1.064	1.061, 1.173, 1.220	0.721, 0.799, 0.844

[1] Confidence intervals are normal-based

Larger error diminishes the strength of the estimates (looking at magnitude), as expected. However, in 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 70. Type C, quadratic, 3 groups (correlated with), extra wide, small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,1) + 3.5*I(G2) + 7*I(G3)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

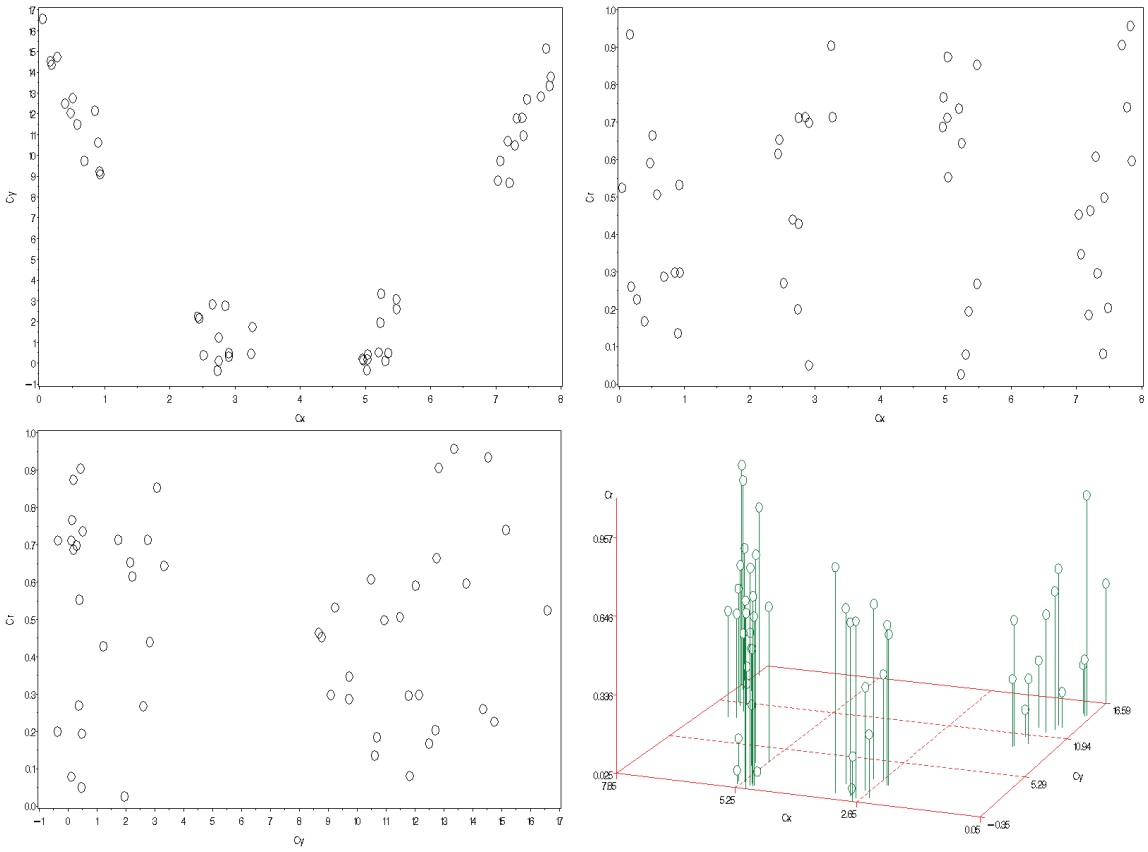
Table 49. Results for type C, quadratic, 3 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.238 (1.228, 1.249)	1.227 (1.216, 1.239)	0.534 (0.524, 0.545)
P5, P50, P95	1.139, 1.242, 1.275	1.140, 1.220, 1.273	0.453, 0.529, 0.568

[1] Confidence intervals are normal-based

Conversely, widening the gap between clusters increases the strength of the estimates (looking at magnitude), again as expected. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 71. Type C, quadratic, 4 groups (correlated with), small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1) + 2.33*I(G2) + 4.66*I(G3) + 6.99*I(G4)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

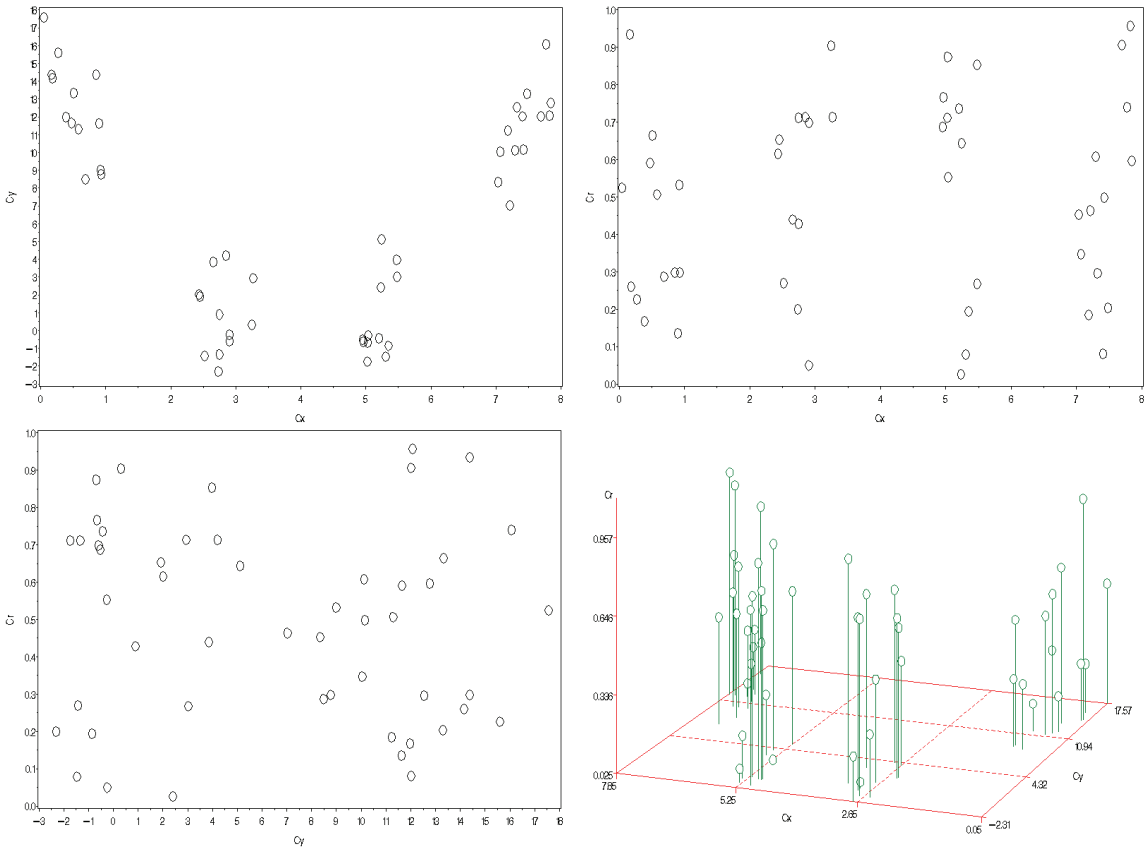
Table 50. Results for type C, quadratic, 4 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.107 (1.098, 1.116)	1.216 (1.205, 1.226)	0.678 (0.667, 0.689)
P5, P50, P95	1.036, 1.112, 1.137	1.136, 1.215, 1.254	0.586, 0.679, 0.719

[1] Confidence intervals are normal-based

Having four clusters in a quadratic spread further strengthens the estimates (looking at magnitude) compared to the examples with fewer clusters (non-widened spread). In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 72. Type C, quadratic, 4 groups (correlated with), large error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1) + 2.33*I(G2) + 4.66*I(G3) + 6.99*I(G4)
$C_y = (C_x - 4)^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

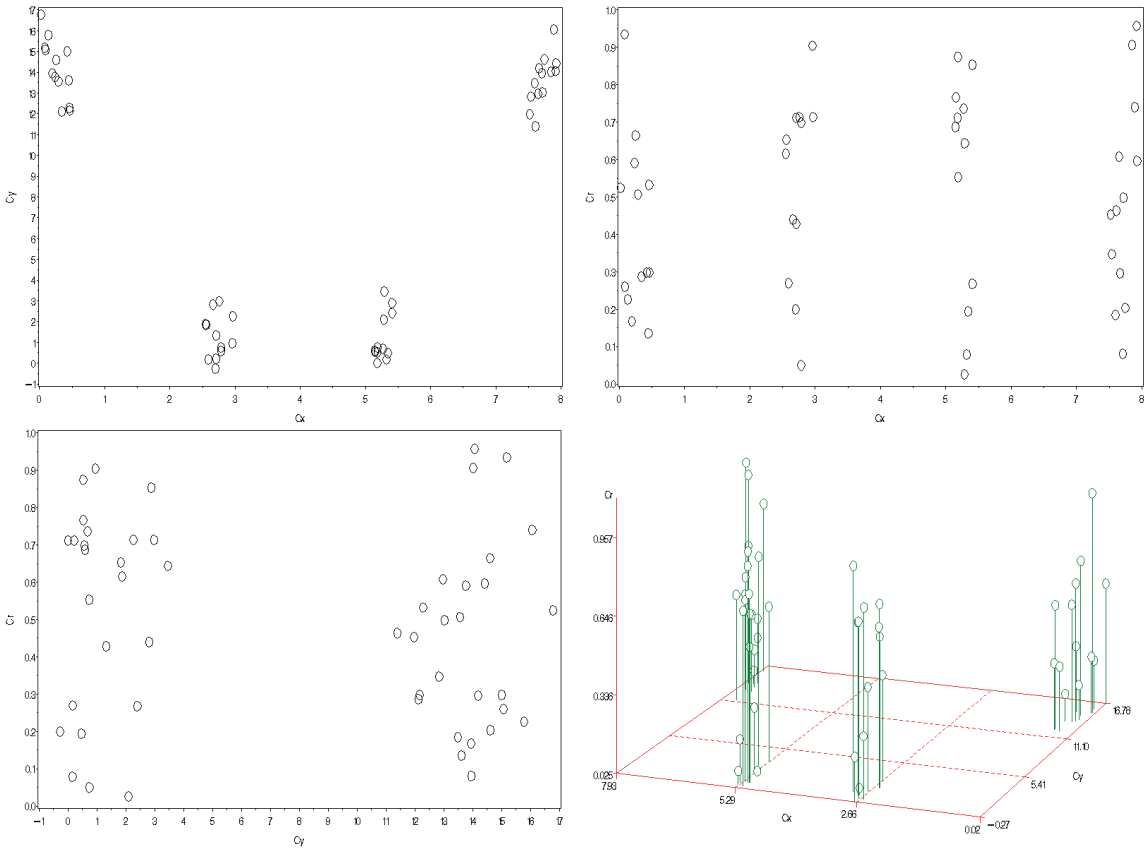
Table 51. Results for type C, quadratic, 4 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.007 (0.997, 1.017)	1.230 (1.217, 1.243)	0.763 (0.752, 0.774)
P5, P50, P95	0.932, 1.007, 1.039	1.132, 1.229, 1.289	0.656, 0.766, 0.803

[1] Confidence intervals are normal-based

Larger error diminishes the strength of the estimates (looking at magnitude), as expected. However, in 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 73. Type C, quadratic, 4 groups (correlated with), extra wide, small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,.5) + 2.5*I(G2) + 5*I(G3) + 7.5*I(G4)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

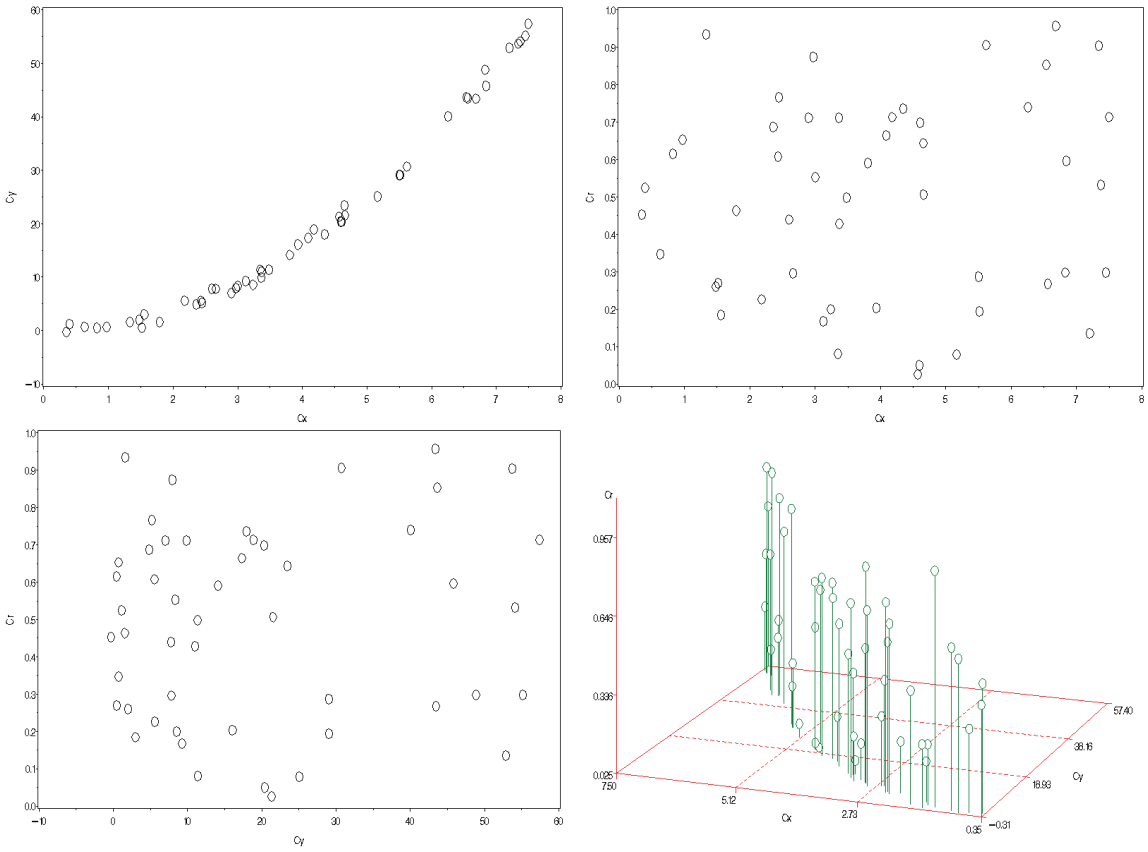
Table 52. Results for type C, quadratic, 4 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.153 (1.144, 1.162)	1.272 (1.262, 1.281)	0.575 (0.566, 0.585)
P5, P50, P95	1.077, 1.156, 1.180	1.197, 1.270, 1.311	0.499, 0.577, 0.610

[1] Confidence intervals are normal-based

Conversely, widening the gap between clusters increases the strength of the estimates (looking at magnitude), again as expected. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 74. Type C, half-quadratic, 1 group, small error



C_x = a random Uniform(0,8)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

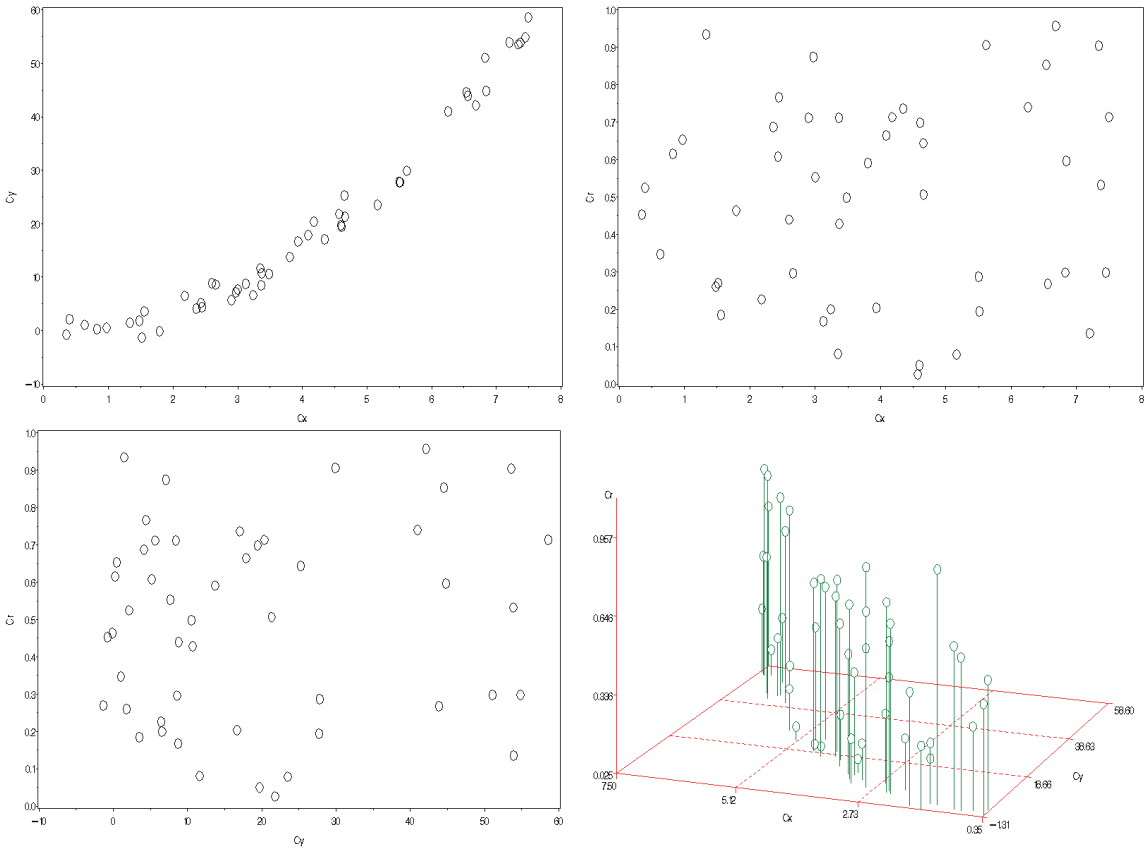
Table 53. Results for type C, half-quadratic, 1 group, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.922 (0.915, 0.929)	0.963 (0.953, 0.973)	1.115 (1.103, 1.127)
P5, P50, P95	0.857, 0.923, 0.947	0.891, 0.957, 0.995	1.014, 1.115, 1.157

[1] Confidence intervals are normal-based

VWUO-MD is having difficulties detecting the half-quadratic shape with no distinct clusters, as it did with the linearly related data. This is not a big surprise, considering how much closer to linear the above shape is, compared to the full parabolic quadratic that worked so well. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 75. Type C, half-quadratic, 1 group, large error



C_x = a random Uniform(0,8)
$C_y = C_x^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

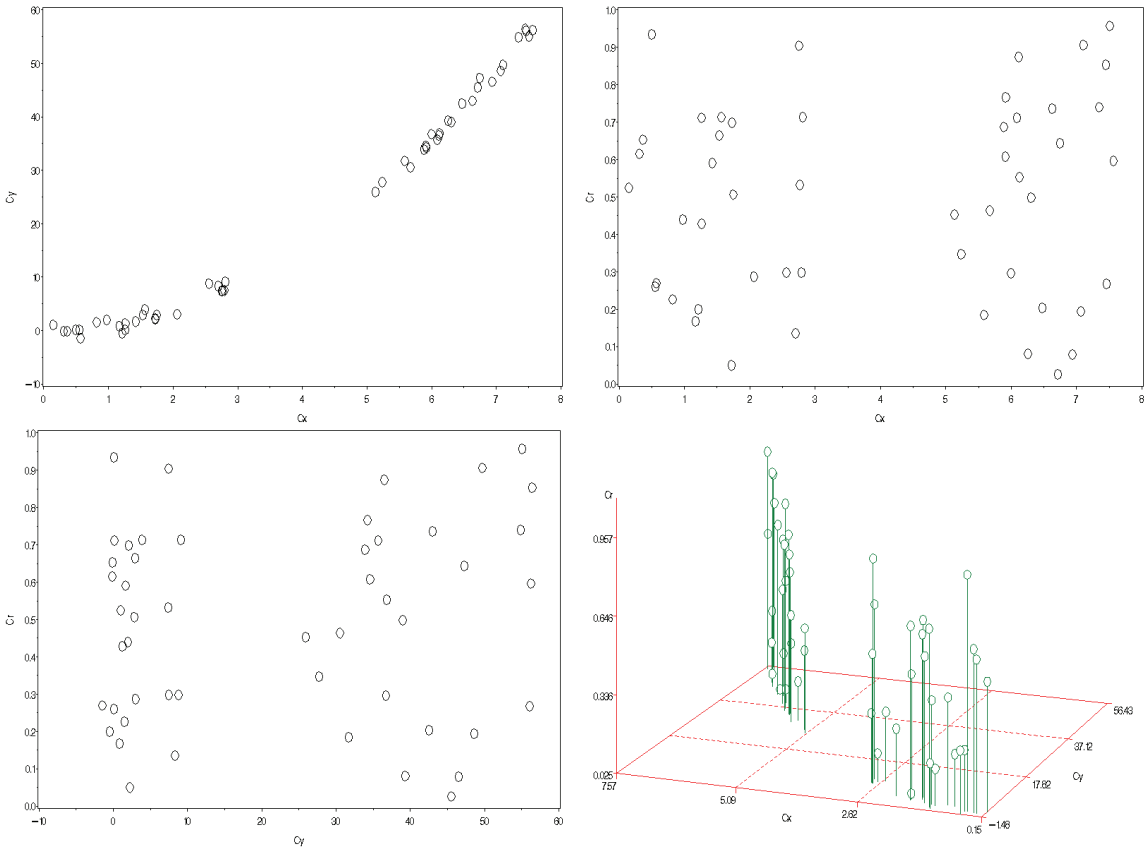
Table 54. Results for type C, half-quadratic, 1 group, large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.918 (0.910, 0.925)	0.979 (0.969, 0.989)	1.103 (1.092, 1.115)
P5, P50, P95	0.853, 0.918, 0.945	0.905, 0.974, 1.011	1.005, 1.103, 1.143

[1] Confidence intervals are normal-based

Larger error diminishes the strength of the (in this case misleading) estimates, as expected. In 1% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 76. Type C, half-quadratic, 2 groups (correlated with), small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,3) + 5*I(G2)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

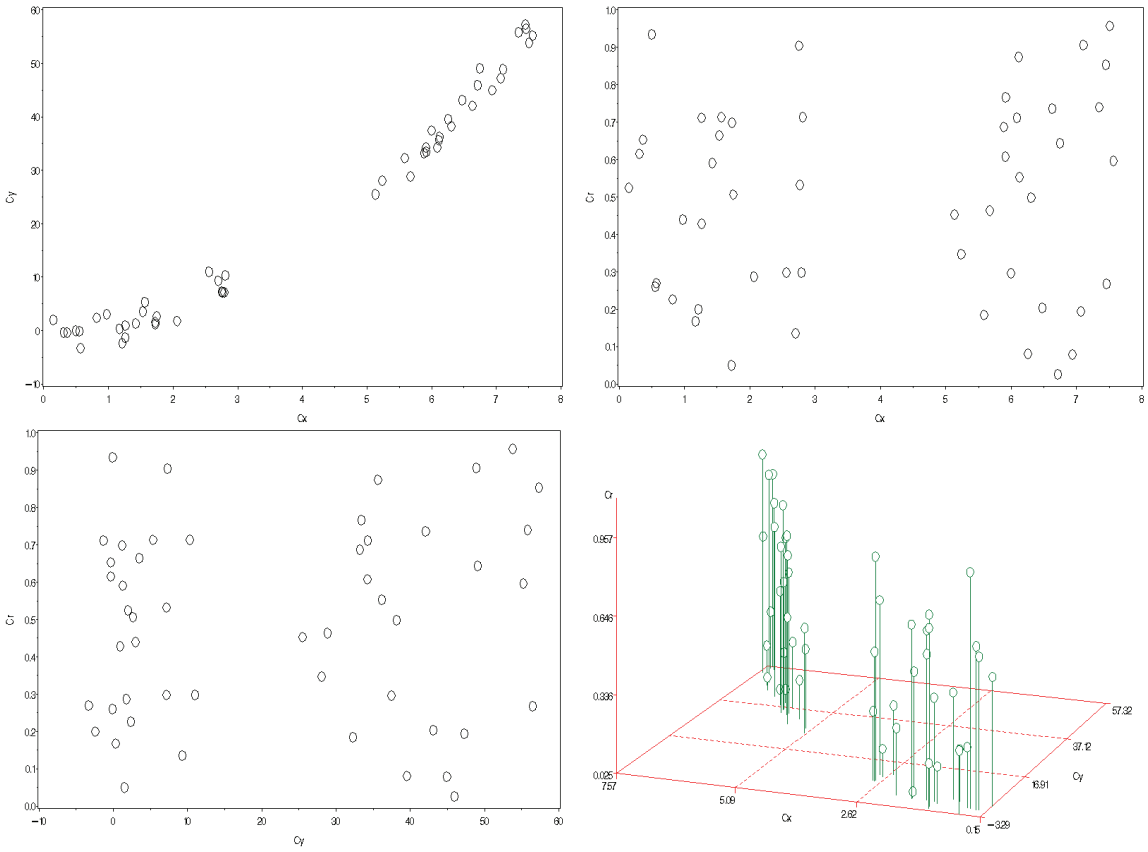
Table 55. Results for type C, half-quadratic, 2 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.033 (1.023, 1.042)	0.952 (0.943, 0.961)	1.015 (1.000, 1.031)
P5, P50, P95	0.959, 1.034, 1.066	0.880, 0.951, 0.976	0.897, 1.016, 1.079

[1] Confidence intervals are normal-based

While the estimates are still not great, separation into two clusters has improved matters. However, recall from our earlier results that two clusters ought not to be correlated in the direction parallel with cluster placement, for optimal performance. Sure enough, the average w_{Cr} remains >1. In 29% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 77. Type C, half-quadratic, 2 groups (correlated with), large error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,3) + 5*I(G2)
$C_y = C_x^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

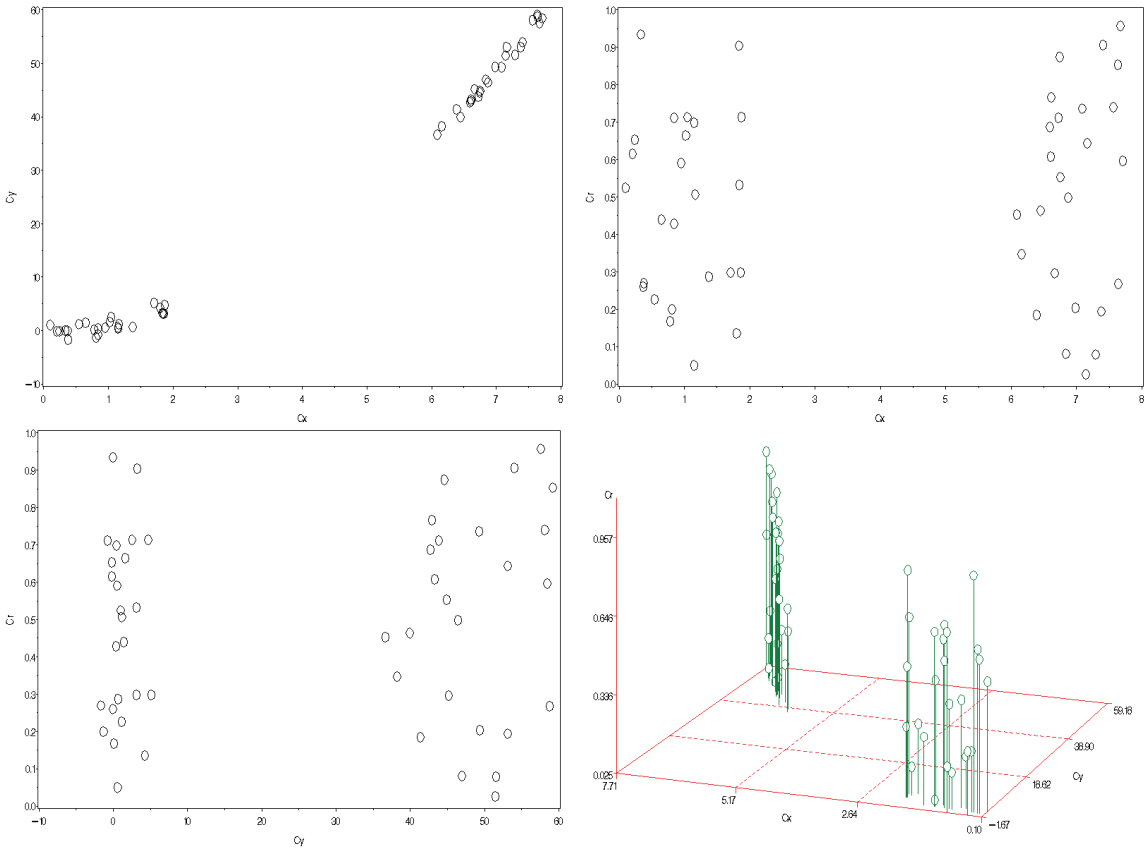
Table 56. Results for type C, half-quadratic, 2 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.027 (1.017, 1.037)	0.970 (0.961, 0.979)	1.003 (0.988, 1.019)
P5, P50, P95	0.952, 1.029, 1.062	0.887, 0.969, 0.998	0.884, 1.006, 1.060

[1] Confidence intervals are normal-based

Again larger error diminishes the strength of the misleading estimates. In 39% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 78. Type C, half-quadratic, 2 groups (correlated with), extra wide, small error



Groups: n_{G1}=25, n_{G2}=25
C_x = a random Uniform(0,2) + 6*I(G2)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

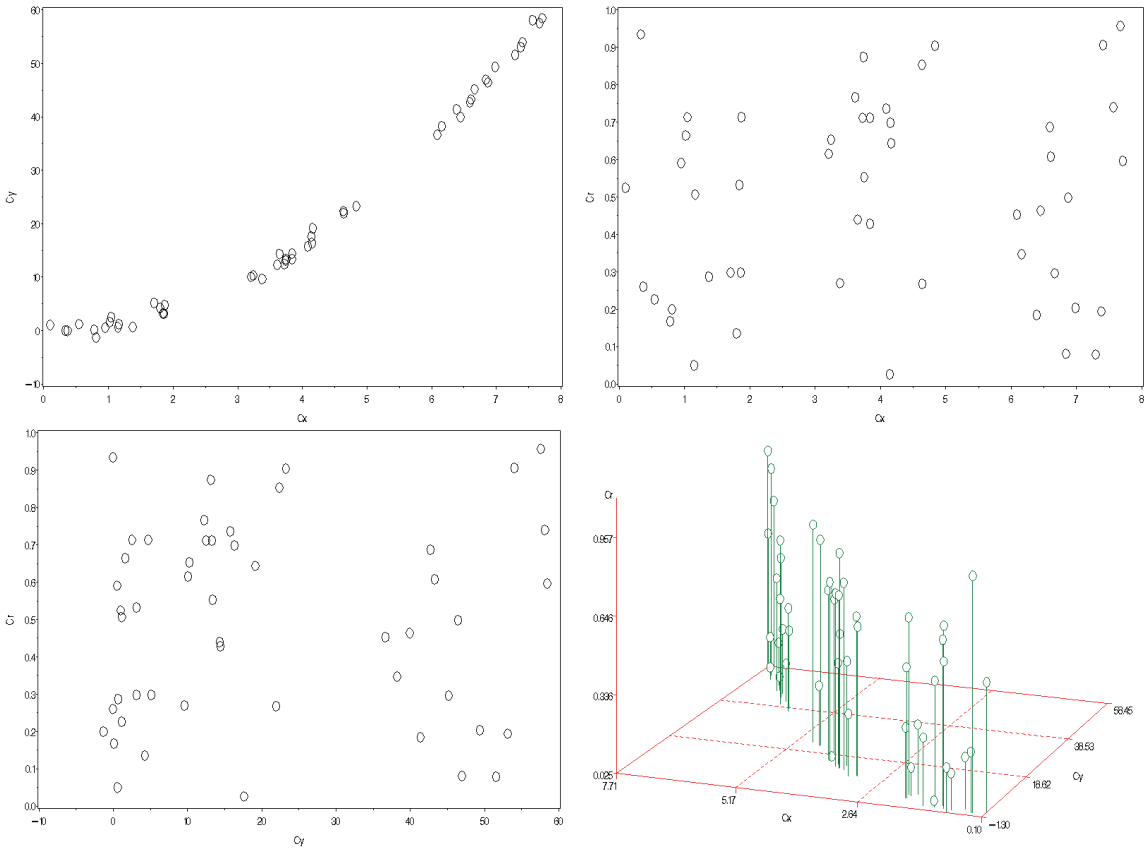
Table 57. Results for type C, half-quadratic, 2 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	1.264 (1.256, 1.273)	1.022 (1.012, 1.032)	0.714 (0.702, 0.725)
P5, P50, P95	1.191, 1.268, 1.294	0.943, 1.022, 1.046	0.608, 0.716, 0.755

[1] Confidence intervals are normal-based

Now, with better separation between clusters and therefore a reduction in correlation relative to scale, the VWUO-MD estimates are successfully detecting the relationship between C_x and C_y. In 100% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 79. Type C, half-quadratic, 3 groups (correlated with), small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,2) + 3*I(G2) + 6*I(G3)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

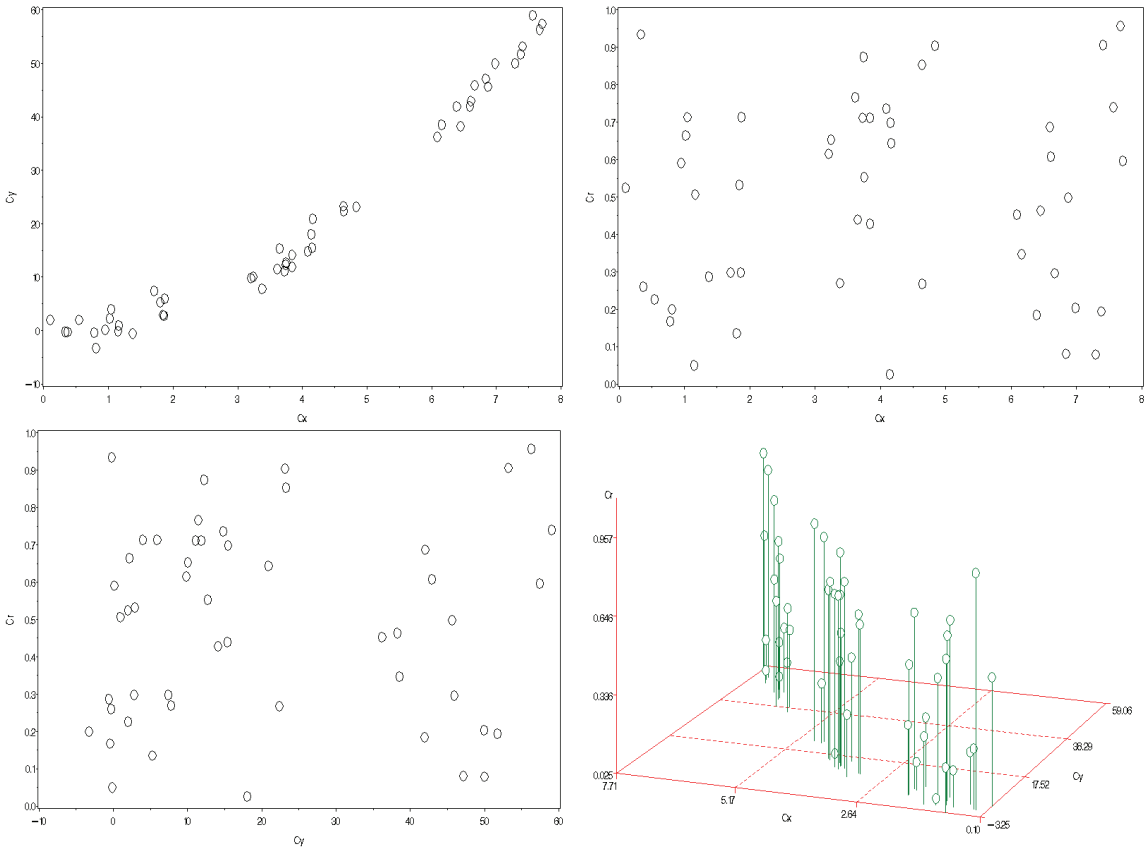
Table 58. Results for type C, half-quadratic, 3 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.858 (0.852, 0.864)	0.991 (0.983, 1.000)	1.151 (1.139, 1.162)
P5, P50, P95	0.807, 0.861, 0.879	0.919, 0.987, 1.024	1.048, 1.147, 1.188

[1] Confidence intervals are normal-based

With three clusters in a half-quadratic placement, as with the three-cluster linear placement before, VWUO-MD weights the noise variable the highest. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 80. Type C, half-quadratic, 3 groups (correlated with), large error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,2) + 3*I(G2) + 6*I(G3)
$C_y = C_x^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

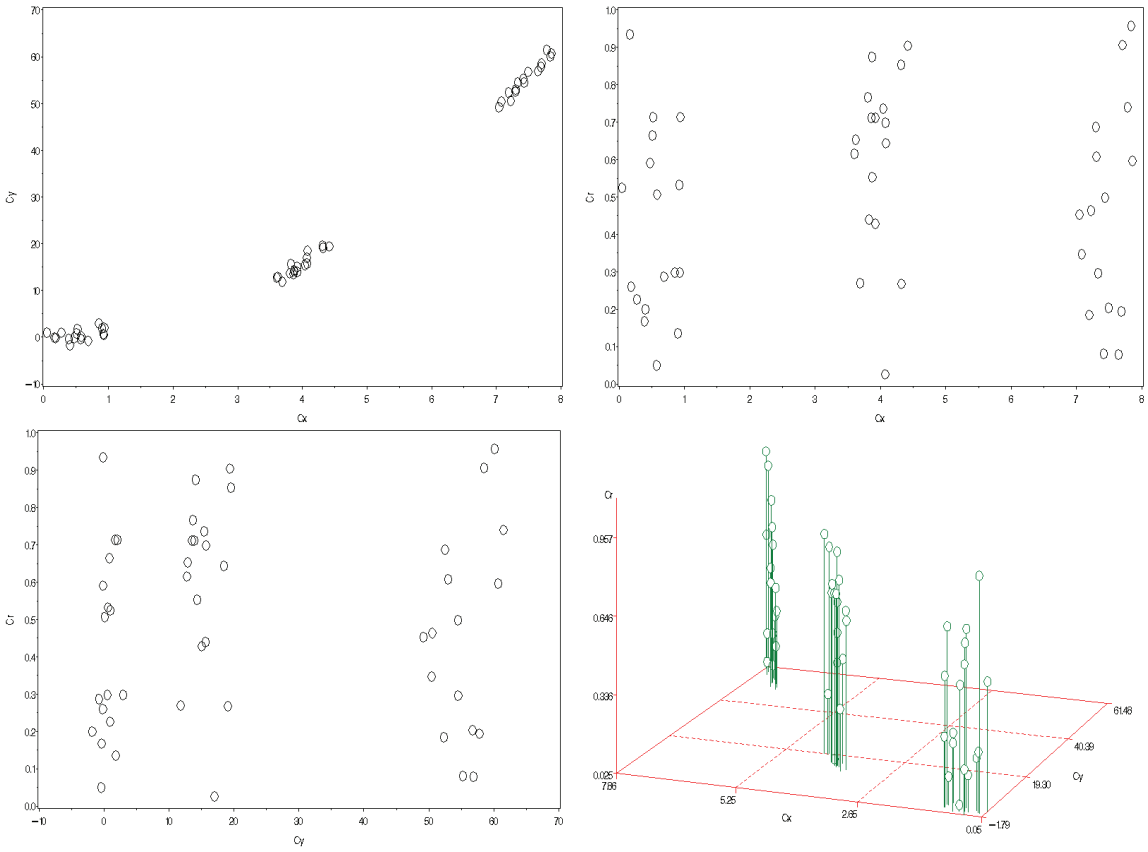
Table 59. Results for type C, half-quadratic, 3 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.853 (0.847, 0.860)	1.007 (0.998, 1.017)	1.139 (1.128, 1.151)
P5, P50, P95	0.800, 0.857, 0.875	0.916, 1.008, 1.040	1.034, 1.139, 1.178

[1] Confidence intervals are normal-based

With larger error, the strength of the misleading estimates is diminished (looking at magnitude). In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 81. Type C, half-quadratic, 3 groups (correlated with), extra wide, small error



Groups: n_{G1}=17, n_{G2}=17, n_{G3}=16
C_x = a random Uniform(0,1) + 3.5*I(G2) + 7*I(G3)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

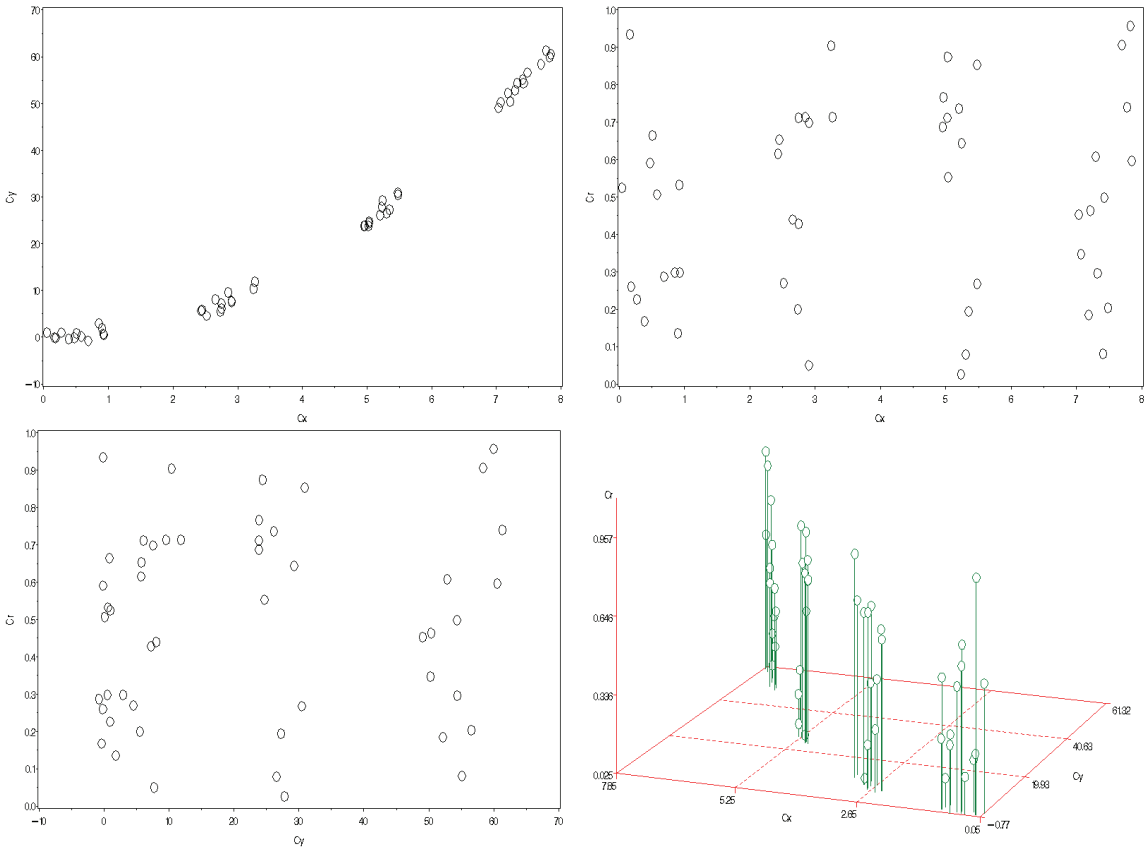
Table 60. Results for type C, half-quadratic, 3 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.740 (0.735, 0.746)	1.105 (1.093, 1.117)	1.155 (1.142, 1.168)
P5, P50, P95	0.700, 0.742, 0.760	1.006, 1.113, 1.146	1.049, 1.153, 1.199

[1] Confidence intervals are normal-based

With wider spread between clusters, the strength of the misleading effect is increased (looking at magnitude), and we begin to see that the placement of the clusters is really rather close to linear. We are experiencing the same problems as we did with those data. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 82. Type C, half-quadratic, 4 groups (correlated with), small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1) + 2.33*I(G2) + 4.66*I(G3) + 6.99*I(G4)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

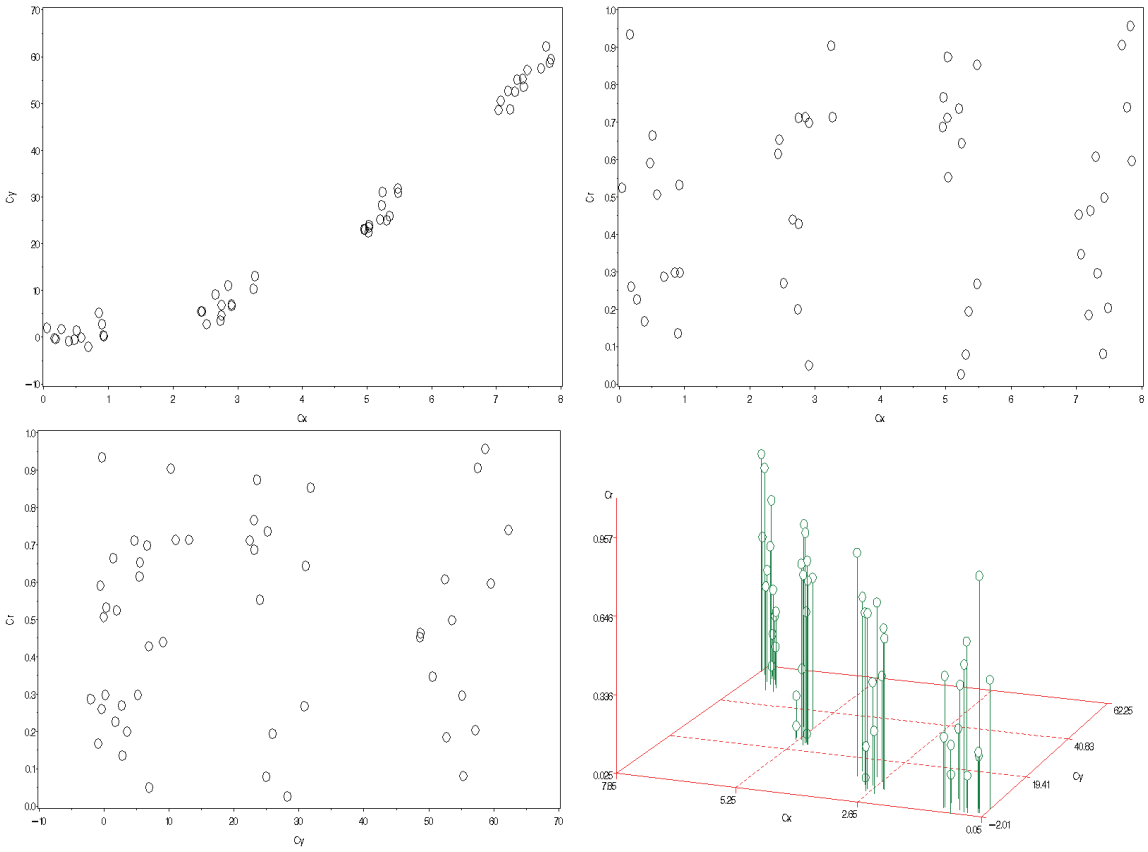
Table 61. Results for type C, half-quadratic, 4 groups (correlated with), small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.863 (0.856, 0.870)	0.913 (0.908, 0.919)	1.224 (1.213, 1.234)
P5, P50, P95	0.811, 0.860, 0.884	0.864, 0.913, 0.933	1.119, 1.231, 1.260

[1] Confidence intervals are normal-based

With four clusters in the half-quadratic (nearly linear) placement, the strength of the misleading effect is again increased (looking at magnitude). In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 83. Type C, half-quadratic, 4 groups (correlated with), large error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,1) + 2.33*I(G2) + 4.66*I(G3) + 6.99*I(G4)
$C_y = C_x^2$ + a random N(0,2) error
C_r = a random Uniform(0,1)

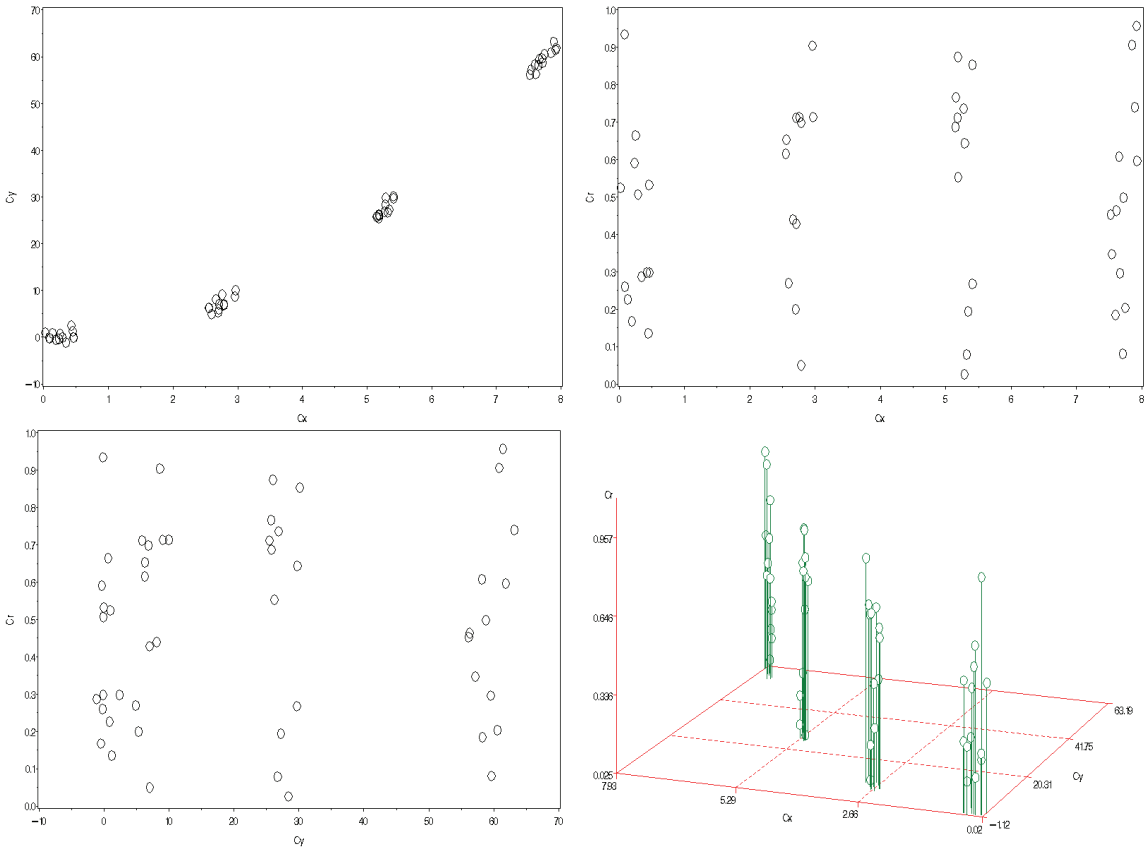
Table 62. Results for type C, half-quadratic, 4 groups (correlated with), large error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.856 (0.849, 0.862)	0.936 (0.930, 0.943)	1.208 (1.197, 1.219)
P5, P50, P95	0.807, 0.852, 0.876	0.881, 0.935, 0.960	1.098, 1.217, 1.242

[1] Confidence intervals are normal-based

With larger error, the strength of the misleading estimates is diminished (looking at magnitude). In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 84. Type C, half-quadratic, 4 groups (correlated with), extra wide, small error



Groups: n_{G1}=13, n_{G2}=12, n_{G3}=12, n_{G4}=13
C_x = a random Uniform(0,.5) + 2.5*I(G2) + 5*I(G3) + 7.5*I(G4)
$C_y = C_x^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

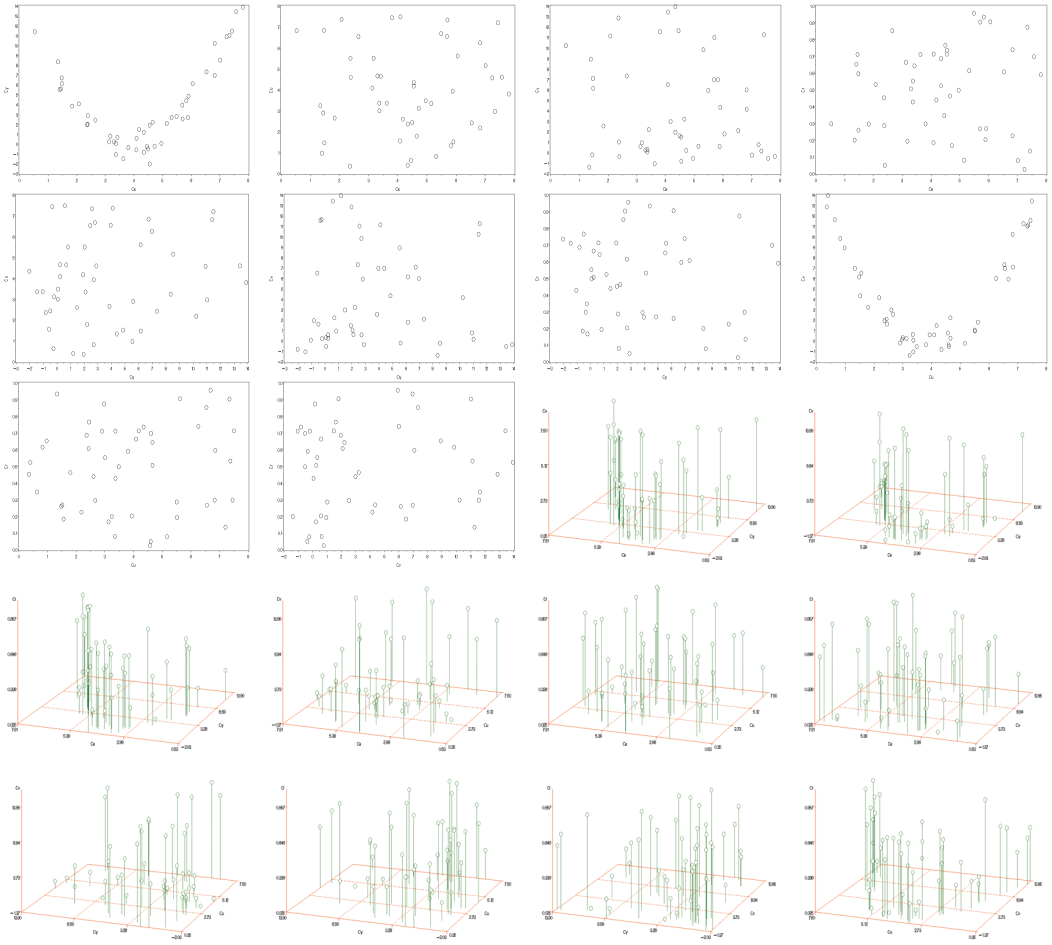
Table 63. Results for type C, half-quadratic, 4 groups (correlated with), extra wide, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cr}
[1]Mean (95% CI)	0.840 (0.833, 0.846)	0.887 (0.881, 0.893)	1.273 (1.263, 1.284)
P5, P50, P95	0.789, 0.837, 0.858	0.839, 0.887, 0.905	1.165, 1.274, 1.309

[1] Confidence intervals are normal-based

With widely spaced clusters, the problem is amplified (looking at magnitude), as we saw before. In 0% of the replicates, $w_{Cr} < w_{Cx}$ and w_{Cy}.

Figure 85. Disjoint relationships: both type C, quadratic, 1 group, small error



C_x = a random Uniform(0,8)
$C_y = (C_x - 4)^2$ + a random N(0,1) error
C_u = a random Uniform(0,8)
$C_v = (C_u - 4)^2$ + a random N(0,1) error
C_r = a random Uniform(0,1)

Table 64. Results for disjoint relationships: both type C, quadratic, 1 group, small error

Summary statistics	w_{Cx}	w_{Cy}	w_{Cu}	w_{Cv}	w_{Cr}
[1]Mean (95% CI)	1.001 (0.991, 1.011)	1.054 (1.043, 1.065)	1.004 (0.994, 1.013)	1.060 (1.047, 1.072)	0.882 (0.872, 0.892)
P5, P50, P95	0.924, 0.998, 1.037	0.964, 1.056, 1.088	0.925, 1.012, 1.032	0.951, 1.065, 1.093	0.791, 0.887, 0.918

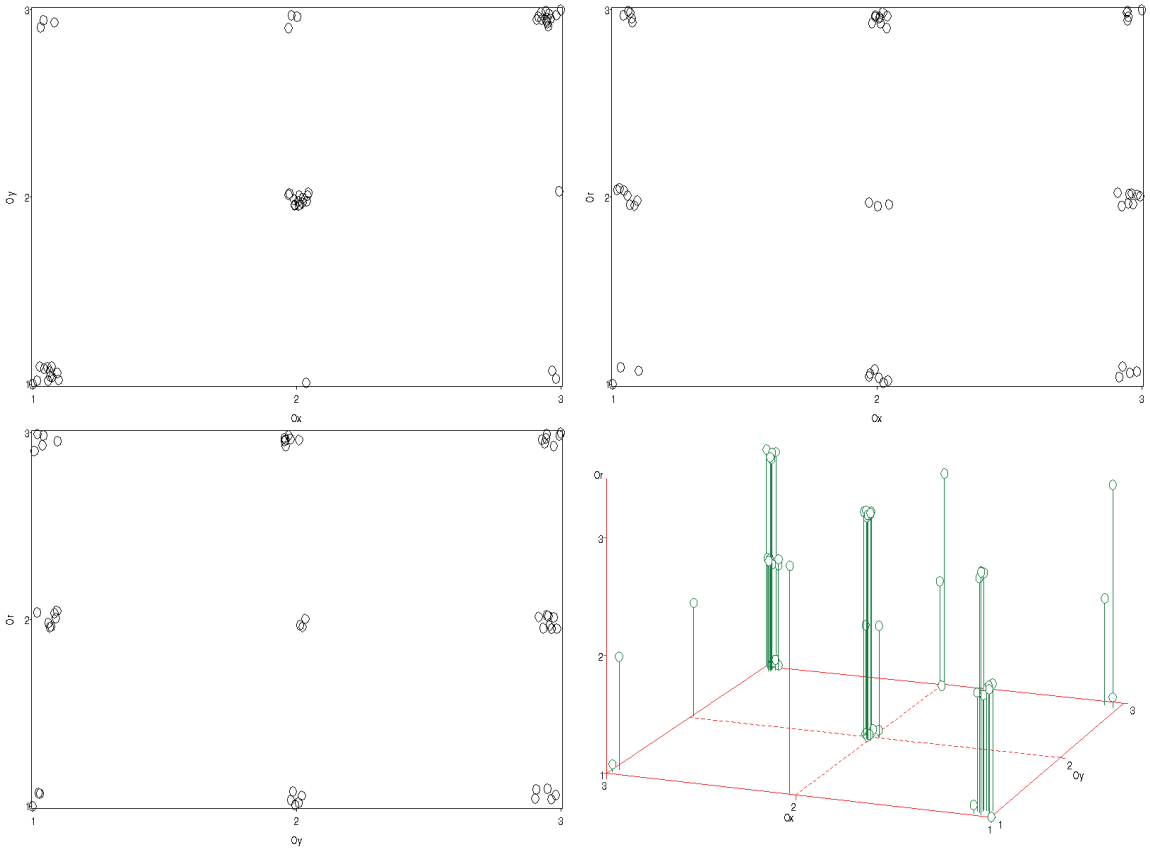
[1] Confidence intervals are normal-based

With disjoint relationships, w_{Cr} is still the smallest weight on average. The remaining weight has been spread across the variables involved in different groupings fairly evenly. In 90% of the replicates, $w_{Cr} < w_{Cx}$, w_{Cy}, w_{Cu} and w_{Cv}.

5.2 Type O data

We should not be surprised to see similar results between type C and type O experiments, considering that the distance formulas are virtually identical (the type O formula using ranks instead of raw data). However, we ran a small subset of the previous experiments, generating 10 type O data sets, replicated 100 times. Figure 86 to Figure 95 show the multivariate distributions of O_x, O_y and O_r in these data sets, as well as O_u and O_v in the last example. Plots are randomly jittered for improved visualization. The captions describe the distributions. "Linear" describes a linear relationship between O_x and O_y. "Quadratic" describes a full parabolic relationship between O_x and O_y. "Half-quadratic" describes a half-parabolic relationship between O_x and O_y. "Small error" versus "large error" describe the relative probabilities of deviating from the prescribed relationship between O_x and O_y. Details of each distribution are given in the footnote below its figure, then a very brief discussion of VWUO-MD's performance on that data set is made.

Figure 86. Type O, linear, 3 levels, small error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.8,.1,.1)*I(O_x=1) + Multinomial(.1,.8,.1)*I(O_x=2) + Multinomial(.1,.1,.8)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

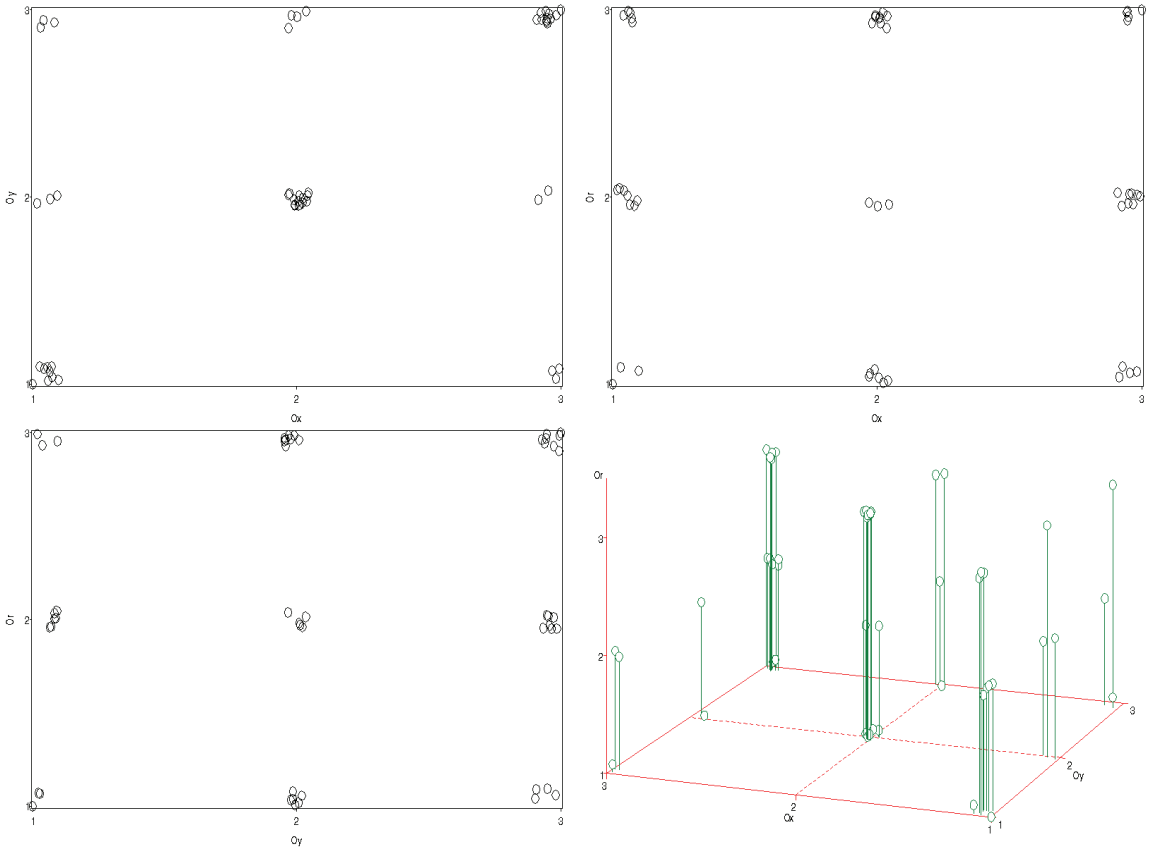
Table 65. Results for type O, linear, 3 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.936 (0.926, 0.947)	0.950 (0.939, 0.962)	1.113 (1.104, 1.122)
P5, P50, P95	0.861, 0.929, 0.985	0.865, 0.973, 0.991	1.013, 1.113, 1.142

[1] Confidence intervals are normal-based

As with type C data, linear relationships between type O variables also appear to present a problem for VWUO-MD. Once again, w_{Or} is the biggest weight on average, the opposite of what would be desired. In 1% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 87. Type O, linear, 3 levels, large error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.6,.2,.2)*I(O_x=1) + Multinomial(.2,.6,.2)*I(O_x=2) + Multinomial(.2,.2,.6)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

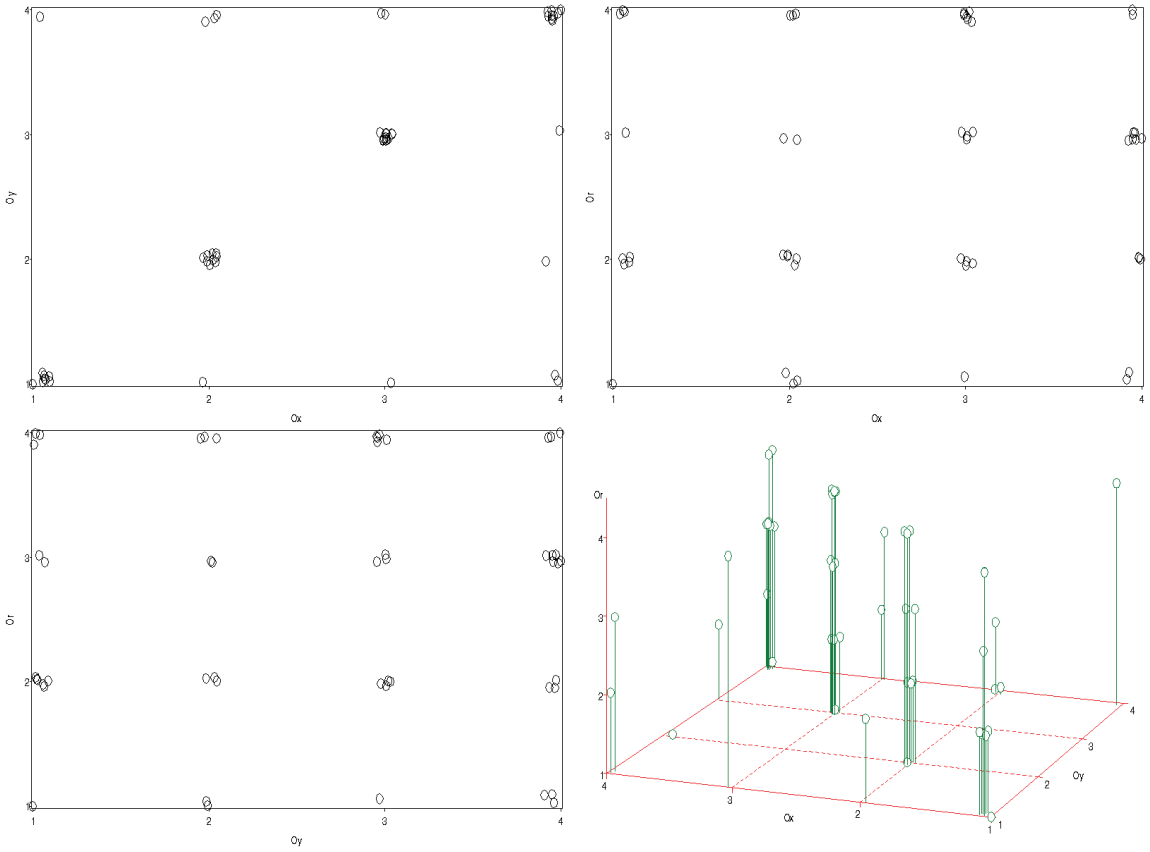
Table 66. Results for type O, linear, 3 levels, large error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.944 (0.930, 0.959)	0.962 (0.944, 0.979)	1.094 (1.080, 1.108)
P5, P50, P95	0.840, 0.929, 0.996	0.840, 0.978, 1.003	0.970, 1.107, 1.149

[1] Confidence intervals are normal-based

With larger error, the strength of the misleading estimates is diminished. In 3% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 88. Type O, linear, 4 levels, small error



O_x = Multinomial(.25,.25,.25,.25)
O_y = Multinomial(.7,.15,.15,.15)*I(O_x=1) + Multinomial(.15,.7,.15,.15)*I(O_x=2) + Multinomial(.15,.15,.7,.15)*I(O_x=3) + Multinomial(.15,.15,.15,.7)*I(O_x=4)
O_r = Multinomial(.25,.25,.25,.25)

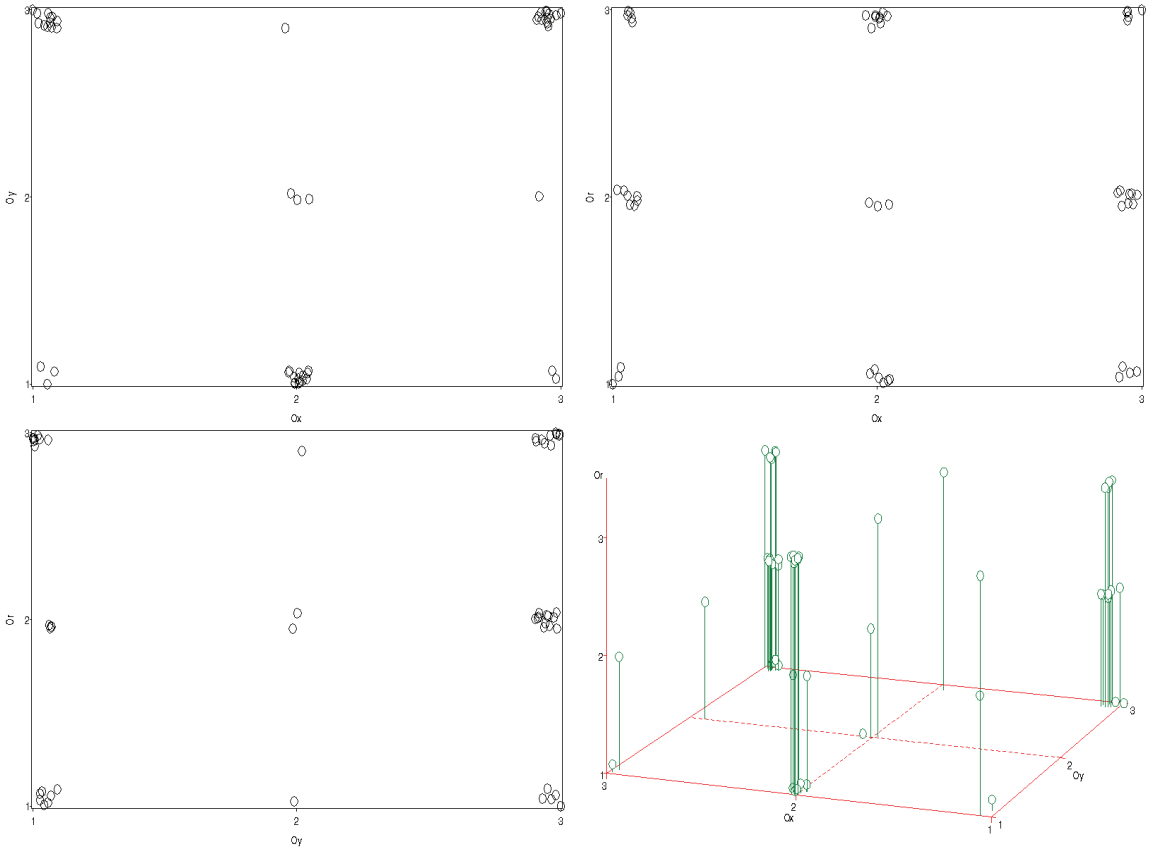
Table 67. Results for type O, linear, 4 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.949 (0.940, 0.958)	0.961 (0.952, 0.971)	1.089 (1.082, 1.096)
P5, P50, P95	0.881, 0.939, 0.981	0.874, 0.972, 0.997	1.039, 1.085, 1.111

[1] Confidence intervals are normal-based

Not surprisingly (thinking back to the clustered type C results), the problem is not alleviated with a greater number of ordinal levels. In 0% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 89. Type O, quadratic, 3 levels, small error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.1,.1,.8)*I(O_x=1) + Multinomial(.8,.1,.1)*I(O_x=2) + Multinomial(.1,.1,.8)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

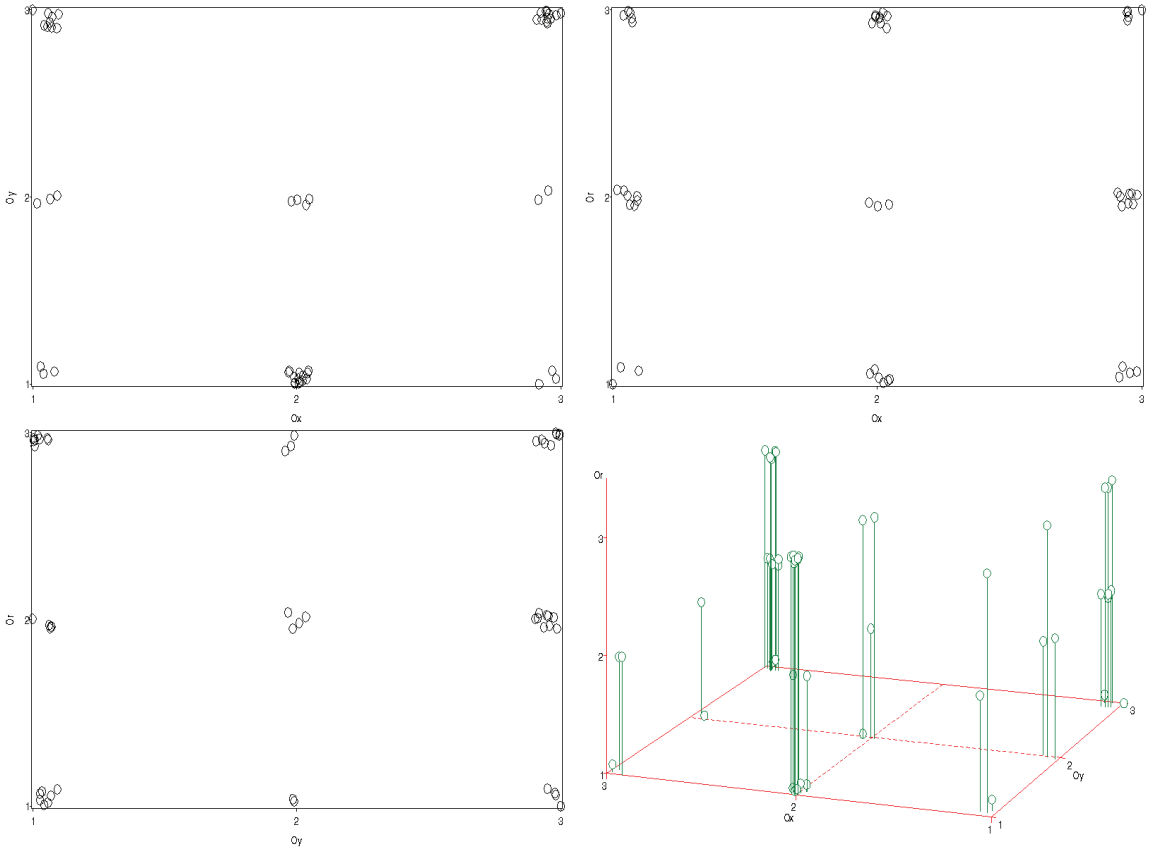
Table 68. Results for type O, quadratic, 3 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	1.152 (1.118, 1.185)	1.172 (1.139, 1.205)	0.677 (0.655, 0.699)
P5, P50, P95	0.891, 1.183, 1.289	0.969, 1.140, 1.247	0.469, 0.682, 0.760

[1] Confidence intervals are normal-based

In keeping with the results on the type C data, VWUO-MD has an easier time correctly identifying quadratic relationships. In 98% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 90. Type O, quadratic, 3 levels, large error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.2,.2,.6)*I(O_x=1) + Multinomial(.6,.2,.2)*I(O_x=2) + Multinomial(.2,.2,.6)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

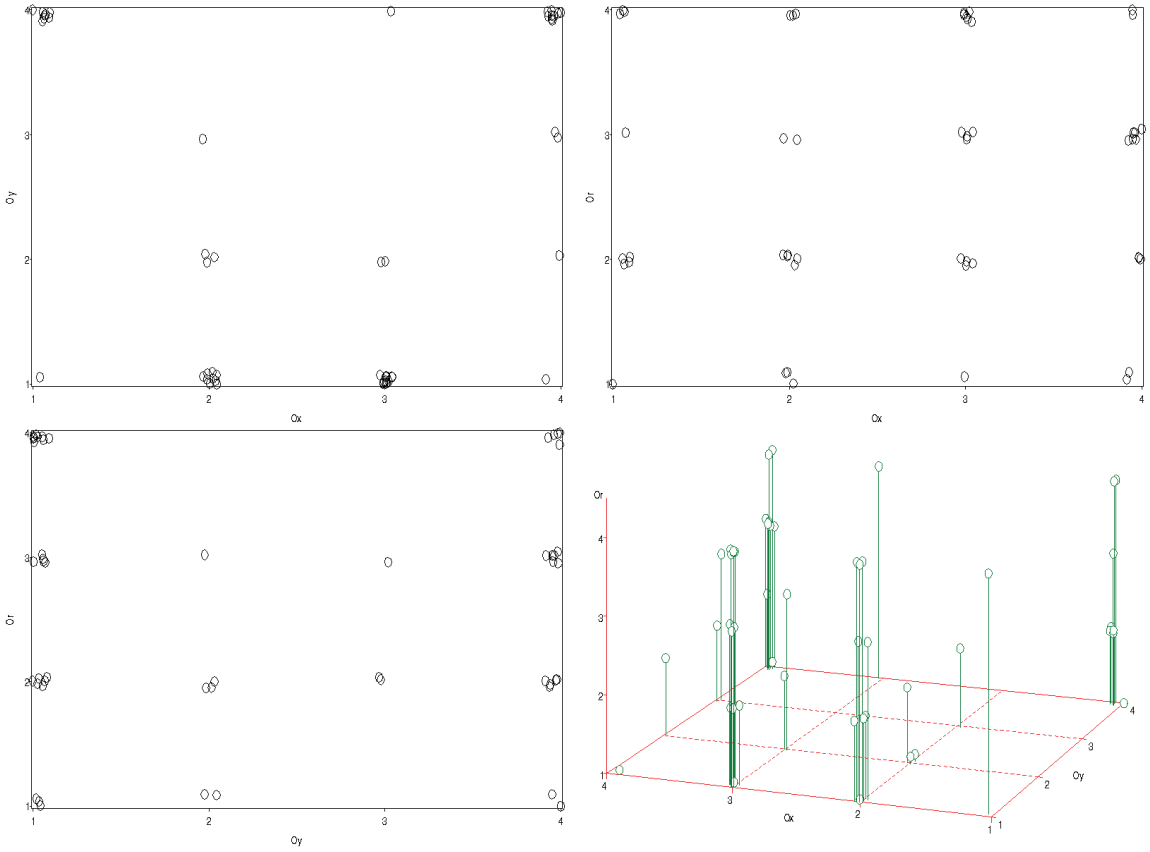
Table 69. Results for type O, quadratic, 3 levels, large error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	1.049 (1.027, 1.071)	1.061 (1.036, 1.087)	0.889 (0.869, 0.909)
P5, P50, P95	0.837, 1.026, 1.149	0.843, 1.036, 1.176	0.751, 0.860, 0.957

[1] Confidence intervals are normal-based

With larger error, the strength of the estimates is diminished. In 68% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 91. Type O, quadratic, 4 levels, small error



O_x = Multinomial(.25,.25,.25,.25)
O_y = Multinomial(.15,.15,.15,.7)*I(O_x=1) + Multinomial(.7,.15,.15,.15)*I(O_x=2) + Multinomial(.7,.15,.15,.15)*I(O_x=3) + Multinomial(.15,.15,.15,.7)*I(O_x=4)
O_r = Multinomial(.25,.25,.25,.25)

Table 70. Results for type O, quadratic, 4 levels, small error

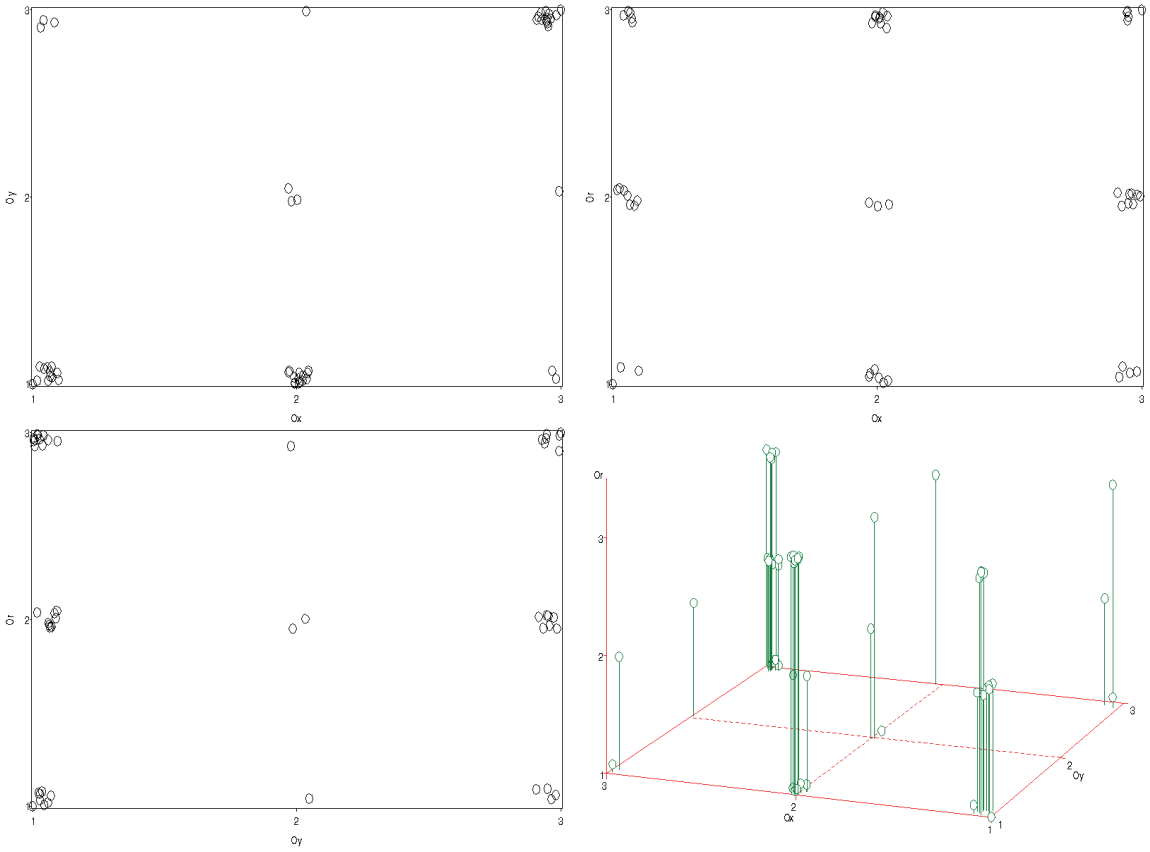
Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	1.099 (1.080, 1.117)	1.055 (1.033, 1.078)	0.846 (0.830, 0.863)
P5, P50, P95	0.922, 1.110, 1.174	0.882, 1.038, 1.115	0.708, 0.840, 0.906

[1] Confidence intervals are normal-based

With four levels, the relationship is still effectively detected by VWUO-MD.

In 83% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 92. Type O, half-quadratic, 3 levels, small error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.8,.1,.1)*I(O_x=1) + Multinomial(.8,.1,.1)*I(O_x=2) + Multinomial(.1,.1,.8)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

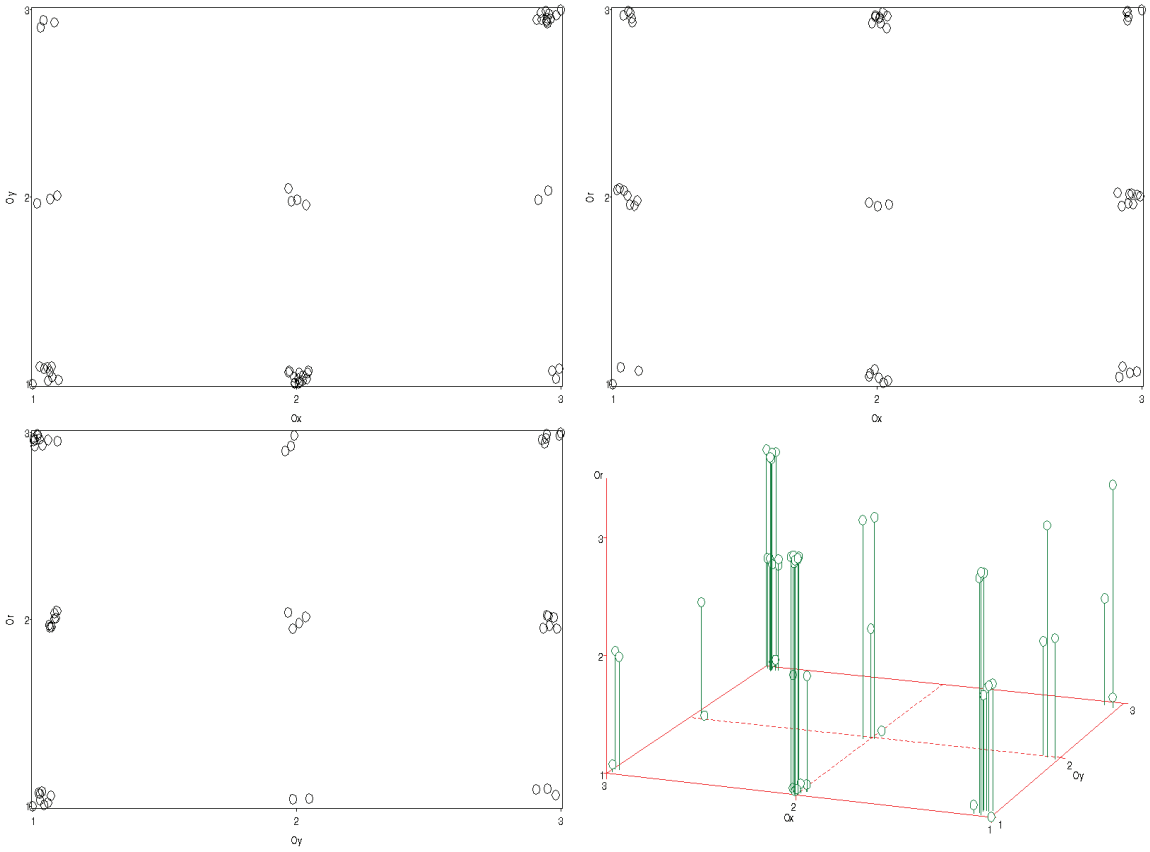
Table 71. Results for type O, half-quadratic, 3 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.898 (0.881, 0.914)	1.351 (1.322, 1.380)	0.752 (0.730, 0.773)
P5, P50, P95	0.740, 0.912, 0.952	1.157, 1.317, 1.444	0.563, 0.763, 0.819

[1] Confidence intervals are normal-based

This result is a welcome departure from the type C results seen earlier. On type O data, a half-quadratic relationship (common in nature) is successfully detected by VWUO-MD. w_{Or} is the lowest variable weight on average. In 85% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 93. Type O, half-quadratic, 3 levels, large error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.6,.2,.2)*I(O_x=1) + Multinomial(.6,.2,.2)*I(O_x=2) + Multinomial(.2,.2,.6)*I(O_x=3)
O_r = Multinomial(.333,.333,.334)

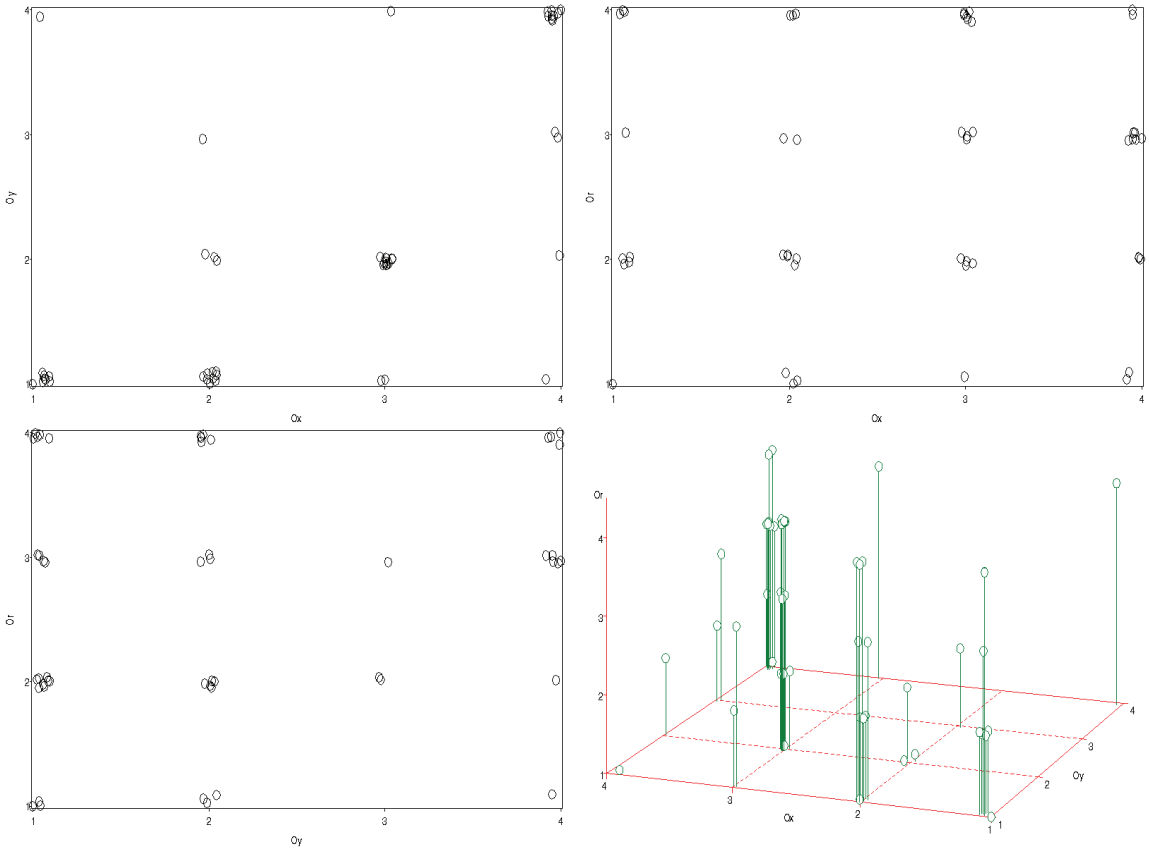
Table 72. Results for type O, half-quadratic, 3 levels, large error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.955 (0.938, 0.973)	1.122 (1.096, 1.148)	0.922 (0.901, 0.944)
P5, P50, P95	0.815, 0.962, 1.009	0.885, 1.143, 1.231	0.757, 0.911, 0.998

[1] Confidence intervals are normal-based

With larger error, the strength of the estimates is diminished. In 51% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 94. Type O, half-quadratic, 4 levels, small error



O_x = Multinomial(.25,.25,.25,.25)
O_y = Multinomial(.7,.15,.15,.15)*I(O_x=1) + Multinomial(.7,.15,.15,.15)*I(O_x=2) + Multinomial(.15,.7,.15,.15)*I(O_x=3) + Multinomial(.15,.15,.15,.7)*I(O_x=4)
O_r = Multinomial(.25,.25,.25,.25)

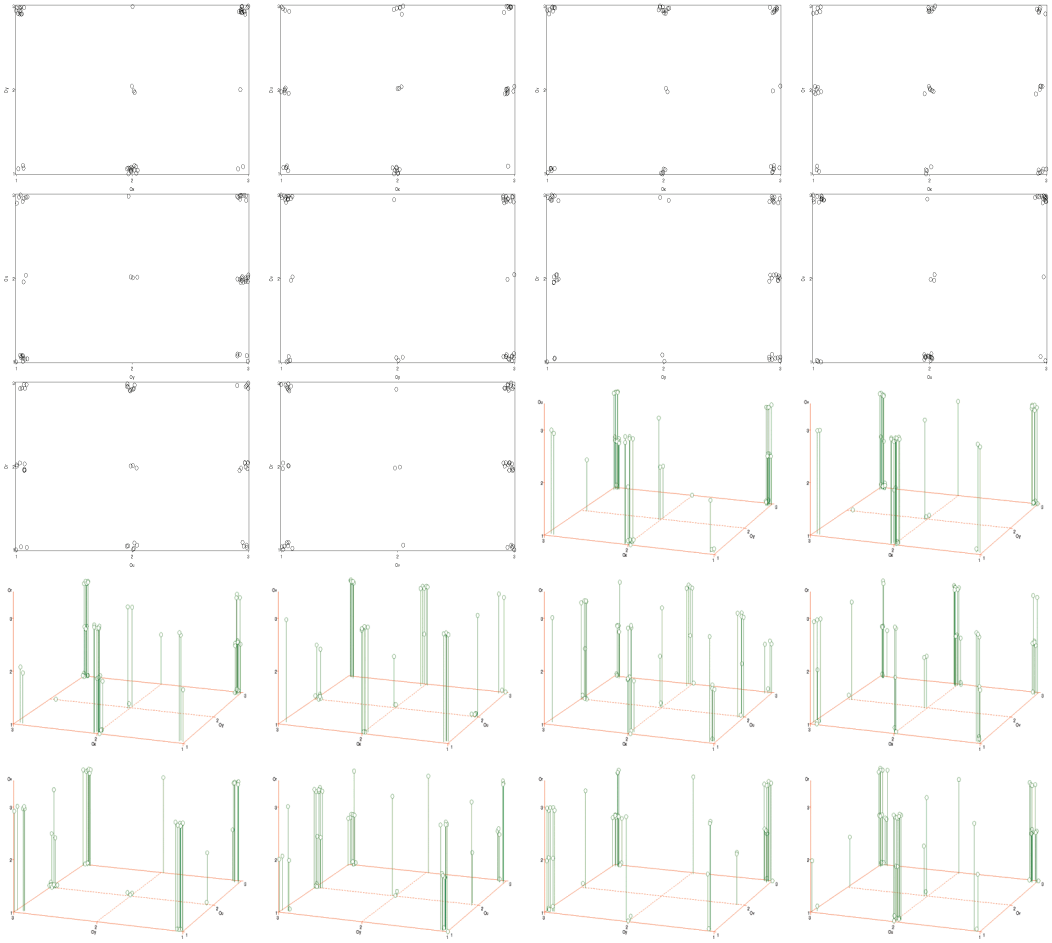
Table 73. Results for type O, half-quadratic, 4 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Or}
[1]Mean (95% CI)	0.956 (0.943, 0.970)	0.990 (0.972, 1.008)	1.053 (1.040, 1.067)
P5, P50, P95	0.878, 0.936, 1.010	0.870, 0.992, 1.069	0.920, 1.067, 1.092

[1] Confidence intervals are normal-based

With four-level variables, the half-quadratic relationship is not as easily detected by VWUO-MD. The quantiles all cover both sides of 1 in this case. In 10% of the replicates, $w_{Or} < w_{Ox}$ and w_{Oy}.

Figure 95. Disjoint relationships: both are type O, quadratic, 3 levels, small error



O_x = Multinomial(.333,.333,.334)
O_y = Multinomial(.1,.1,.8)*I(O_x=1) + Multinomial(.8,.1,.1)*I(O_x=2) + Multinomial(.1,.1,.8)*I(O_x=3)
O_u = Multinomial(.333,.333,.334)
O_v = Multinomial(.1,.1,.8)*I(O_u=1) + Multinomial(.8,.1,.1)*I(O_u=2) + Multinomial(.1,.1,.8)*I(O_u=3)
O_r = Multinomial(.333,.333,.334)

Table 74. Results for disjoint relationships: both are type O, quadratic, 3 levels, small error

Summary statistics	w_{Ox}	w_{Oy}	w_{Ou}	w_{Ov}	w_{Or}
[1]Mean (95% CI)	1.072 (1.039, 1.104)	1.008 (0.972, 1.043)	1.053 (1.025, 1.082)	0.977 (0.941, 1.012)	0.891 (0.875, 0.906)
P5, P50, P95	0.857, 1.018, 1.175	0.734, 1.003, 1.137	0.851, 1.030, 1.122	0.651, 0.988, 1.106	0.741, 0.889, 0.951

[1] Confidence intervals are normal-based

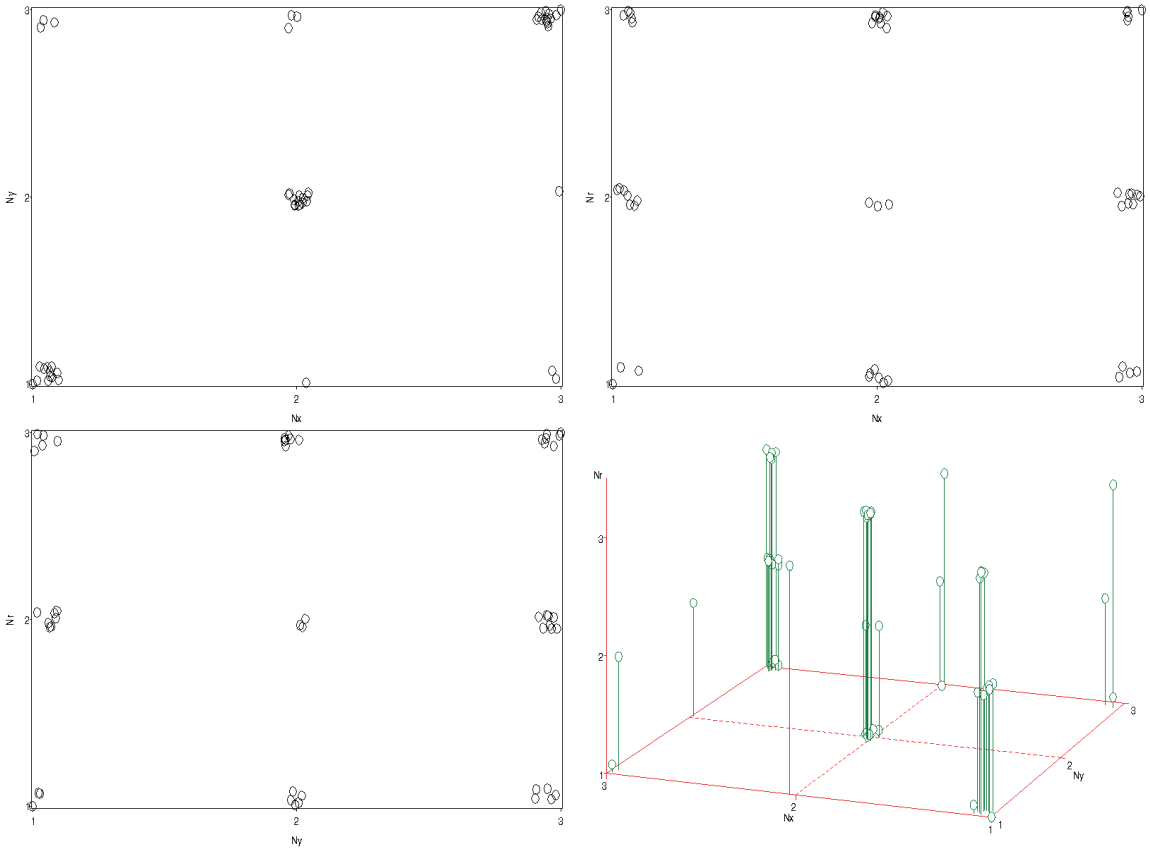
With disjoint relationships, w_{Or} is still the smallest weight on average. The remaining weight has been spread across the variables involved in different

groupings fairly evenly. Unfortunately, variability has increased. In only 29% of the replicates, $w_{Or} < w_{Ox}$, w_{Oy}, w_{Ou} and w_{Ov}.

5.3 Type N data

We generated four type N data sets, replicated 100 times. Figure 96 to Figure 99 show the multivariate distributions of N_x, N_y and N_r in these data sets, as well as N_u and N_v in the last example. Plots are randomly jittered for improved visualization. The captions describe the distributions. The descriptors "linear", "quadratic" and "half-quadratic" that were previously used do not apply with type N data since order is arbitrary. "Small error" versus "large error" describe the relative probabilities of deviating from the prescribed relationship between N_x and N_y. Details of each distribution are given in the footnote below its figure, then a very brief discussion of VWUO-MD's performance on that data set is made.

Figure 96. Type N, 3 levels, small error



N_x = Multinomial(.333,.333,.334)
N_y = Multinomial(.8,.1,.1)*I(N_x=1) + Multinomial(.1,.8,.1)*I(N_x=2) + Multinomial(.1,.1,.8)*I(N_x=3)
N_r = Multinomial(.333,.333,.334)

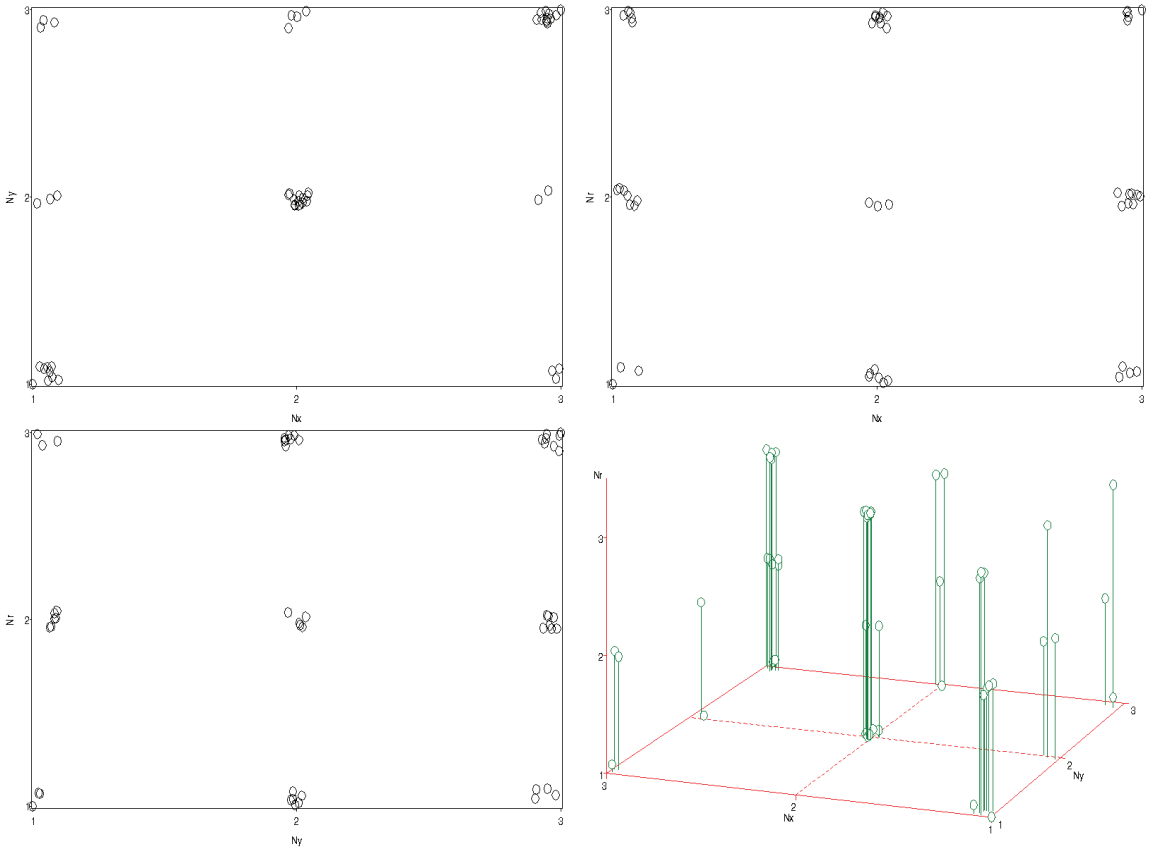
Table 75. Results for type N, 3 levels, small error

Summary statistics	w_{Nx}	w_{Ny}	w_{Nr}
[1]Mean (95% CI)	1.253 (1.201, 1.306)	1.287 (1.234, 1.339)	0.460 (0.444, 0.476)
P5, P50, P95	0.926, 1.076, 1.536	0.944, 1.458, 1.536	0.328, 0.457, 0.520

[1] Confidence intervals are normal-based

With three-level type N variables, the relationship is easily detected by VWUO-MD. In this case the 95th percentile of w_{Nr} is far below the 5th percentile of the two competing variable weights. Note that the linear shape on the graph is an arbitrary choice, since we recall that any reordering of type N (nominal) variable levels produces the same results. In 100% of the replicates, $w_{Nr} < w_{Nx}$ and w_{Ny}.

Figure 97. Type N, 3 levels, large error



N_x = Multinomial(.333,.333,.334)
N_y = Multinomial(.6,.2,.2)*I(N_x=1) + Multinomial(.2,.6,.2)*I(N_x=2) + Multinomial(.2,.2,.6)*I(N_x=3)
N_r = Multinomial(.333,.333,.334)

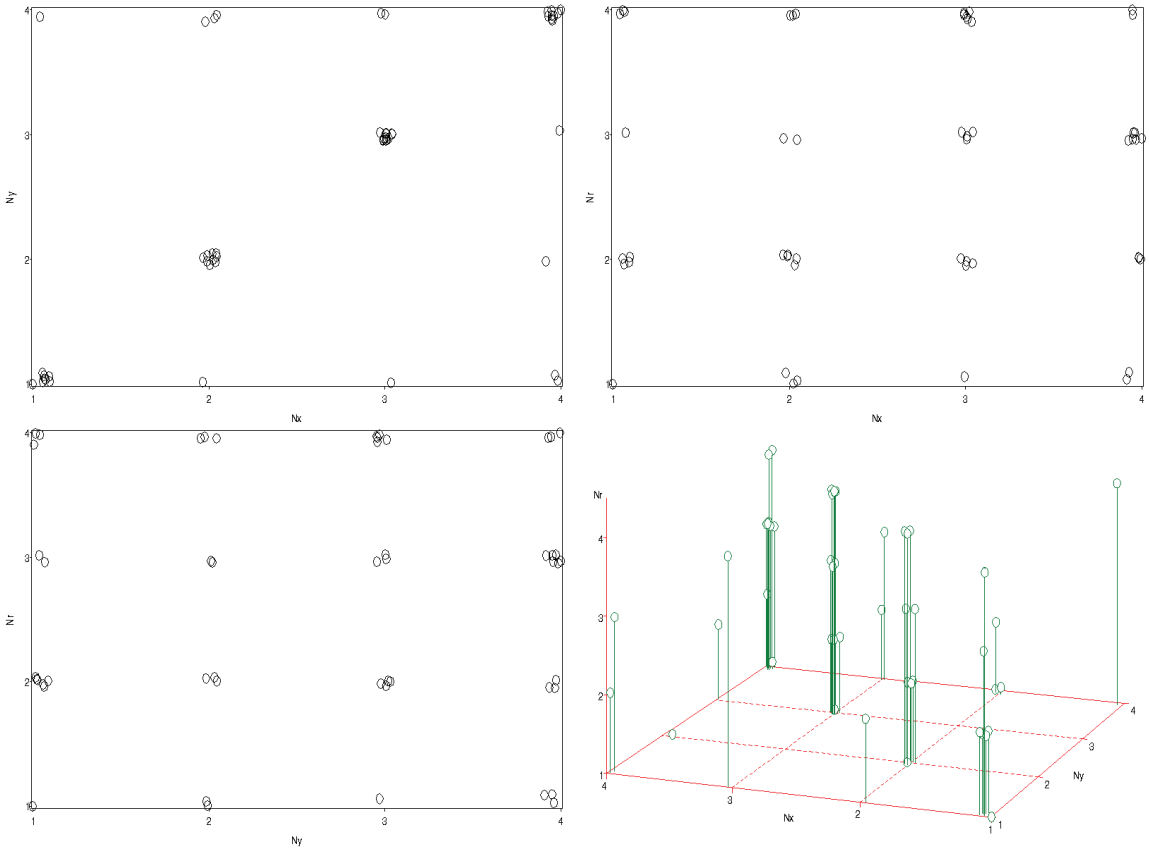
Table 76. Results for type N, 3 levels, large error

Summary statistics	w_{Nx}	w_{Ny}	w_{Nr}
[1]Mean (95% CI)	1.196 (1.143, 1.249)	1.163 (1.112, 1.214)	0.640 (0.626, 0.655)
P5, P50, P95	0.885, 1.383, 1.453	0.883, 0.972, 1.431	0.538, 0.635, 0.669

[1] Confidence intervals are normal-based

With larger error terms, the estimated weights for N_x and N_y are diminished, but remain very strong. In 95% of the replicates, $w_{Nr} < w_{Nx}$ and w_{Ny}.

Figure 98. Type N, 4 levels, small error



N_x = Multinomial(.25,.25,.25,.25)
N_y = Multinomial(.7,.15,.15,.15)*I(N_x=1) + Multinomial(.15,.7,.15,.15)*I(N_x=2) + Multinomial(.15,.15,.7,.15)*I(N_x=3) + Multinomial(.15,.15,.15,.7)*I(N_x=4)
N_r = Multinomial(.25,.25,.25,.25)

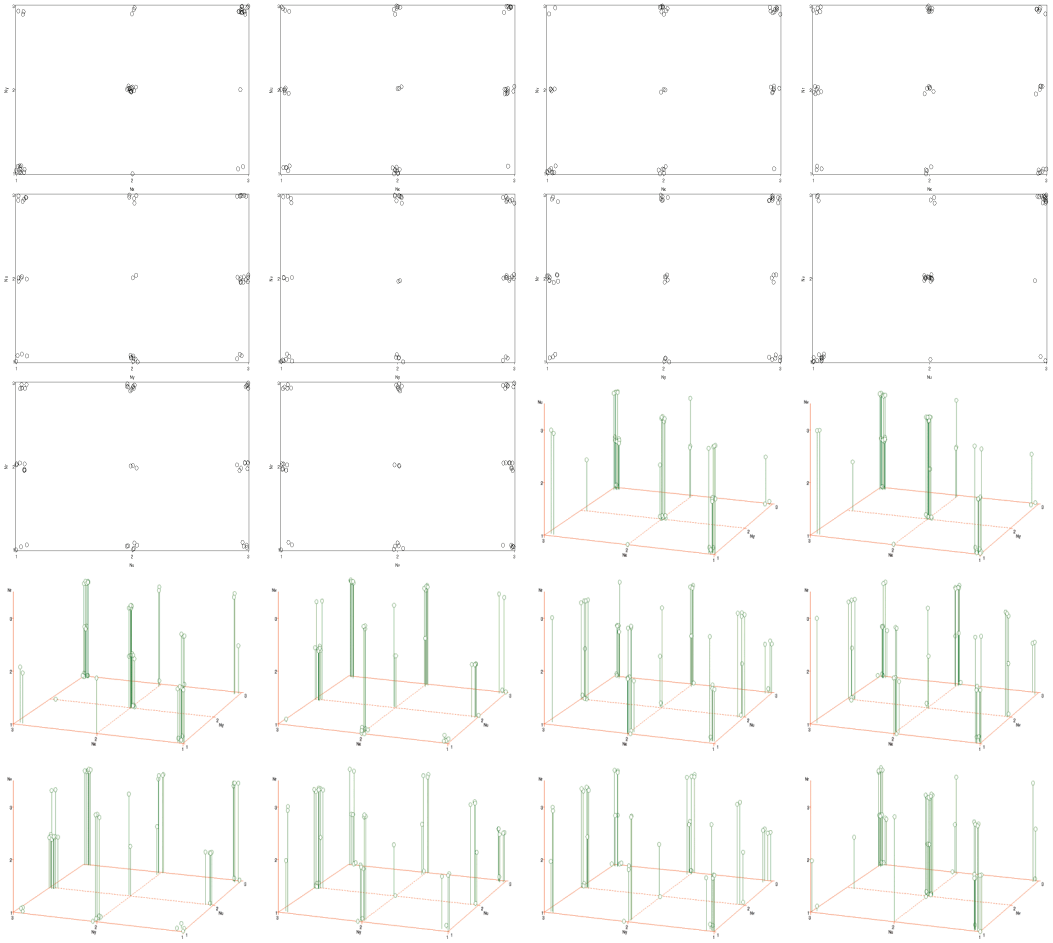
Table 77. Results for type N, 4 levels, small error

Summary statistics	w_{Nx}	w_{Ny}	w_{Nr}
[1]Mean (95% CI)	1.255 (1.194, 1.317)	1.263 (1.200, 1.326)	0.482 (0.471, 0.493)
P5, P50, P95	0.898, 1.262, 1.564	0.903, 1.246, 1.581	0.389, 0.484, 0.528

[1] Confidence intervals are normal-based

Adding another variable level does not have a big effect on the estimates (looking at magnitude), compared to the three-level example above (non-widened data). VWUO-MD continues to perform well in detecting type N relationships. In 100% of the replicates, $w_{Nr} < w_{Nx}$ and w_{Ny}.

Figure 99. Disjoint relationships: both are type N, 3 levels, small error



N_x = Multinomial(.333,.333,.334)
N_y = Multinomial(.8,.1,.1)*I(N_x=1) + Multinomial(.1,.8,.1)*I(N_x=2) + Multinomial(.1,.1,.8)*I(N_x=3)
N_u = Multinomial(.333,.333,.334)
N_v = Multinomial(.8,.1,.1)*I(N_u=1) + Multinomial(.1,.8,.1)*I(N_u=2) + Multinomial(.1,.1,.8)*I(N_u=3)
N_r = Multinomial(.333,.333,.334)

Table 78. Results for disjoint relationships: both are type N, 3 levels, small error

Summary statistics	W_{Nx}	W_{Ny}	W_{Nu}	W_{Nv}	W_{Nr}
[1]Mean (95% CI)	1.019 (0.990, 1.047)	1.019 (0.991, 1.047)	0.997 (0.969, 1.025)	0.997 (0.969, 1.024)	0.969 (0.963, 0.974)
P5, P50, P95	0.844, 1.102, 1.162	0.842, 1.099, 1.152	0.837, 0.914, 1.147	0.838, 0.914, 1.147	0.922, 0.972, 0.984

[1] Confidence intervals are normal-based

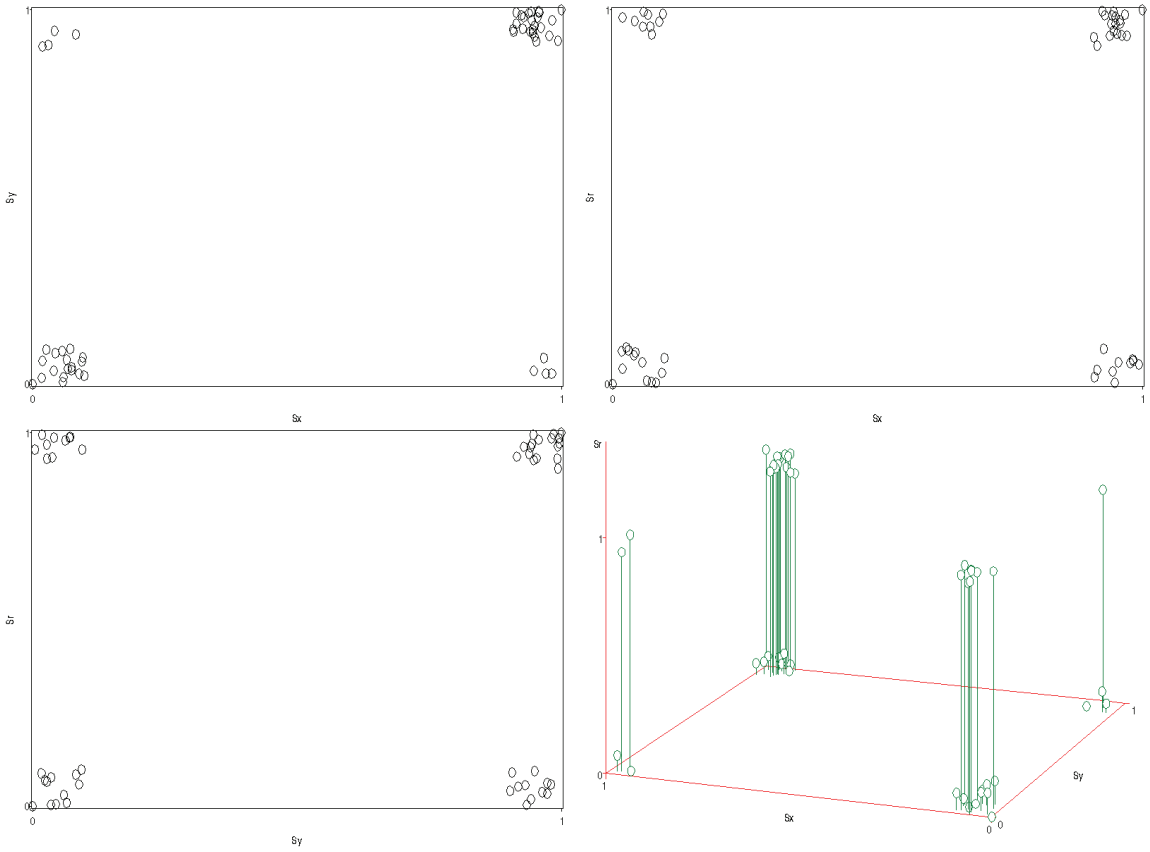
With disjoint relationships, w_{Nr} is still the smallest weight on average. The remaining weight has been spread across the variables involved in different

groupings fairly evenly. Unfortunately, variability has increased. This time in only 2% of the replicates, $w_{Nr} < w_{Nx}$, w_{Ny}, w_{Nu} and w_{Nv}.

5.4 Type S data

We generated five type S data sets, replicated 100 times. Figure 100 to Figure 104 show the multivariate distributions of S_x, S_y and S_r in these data sets, as well as S_u and S_v in the last example. Plots are randomly jittered for improved visualization. The captions describe the distributions. The descriptors "linear", "quadratic" and "half-quadratic" that were previously used do not apply with type S data since order is arbitrary and there are only two levels. "Small error" versus "large error" describe the relative probabilities of deviating from the prescribed relationship between S_x and S_y. Details of each distribution are given in the footnote below its figure, then a very brief discussion of VWUO-MD's performance on that data set is made.

Figure 100. Type S, equal probability levels in S_x, small error



S_x = Bernoulli(.5)
S_y = Bernoulli(.1)*I(S_x=0) + Bernoulli(.9)*I(S_x=1)
S_r = Bernoulli(.5)

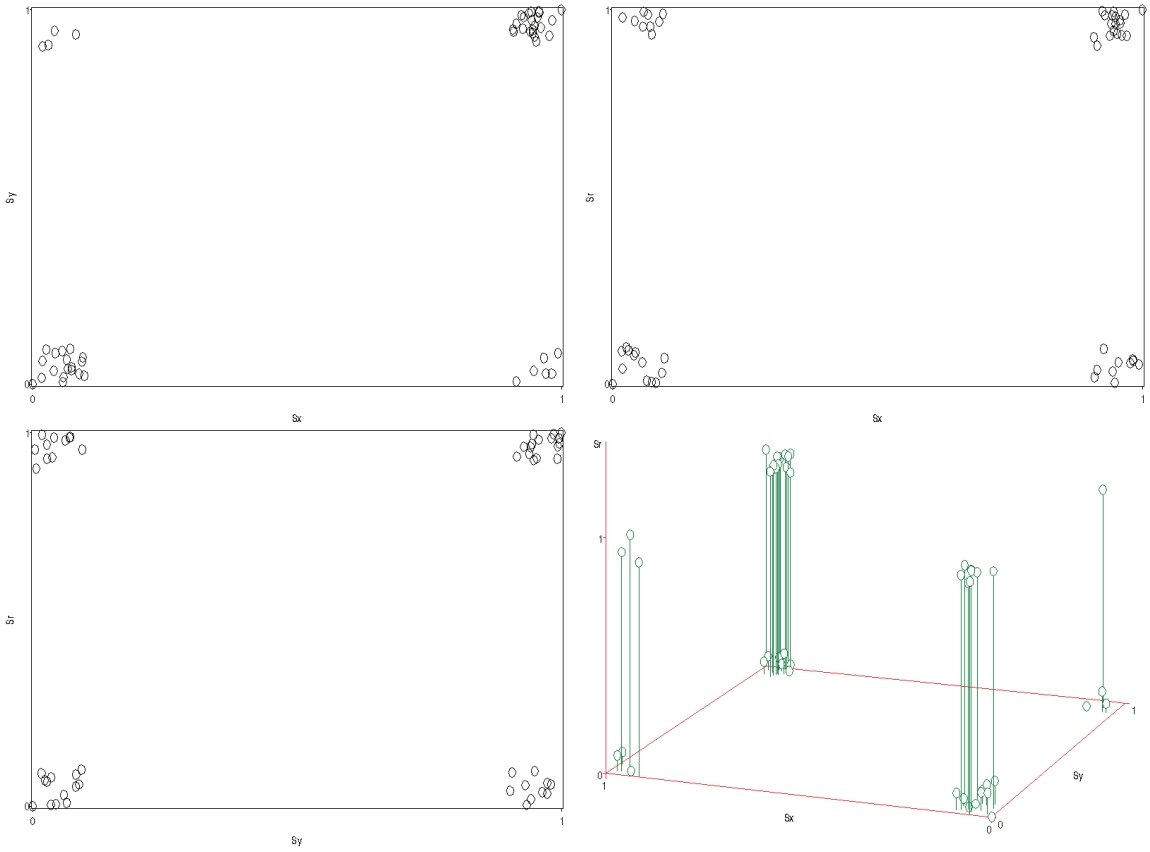
Table 79. Results for type S, equal probability levels in S_x, small error

Summary statistics	w_{Sx}	w_{Sy}	w_{Sr}
[1]Mean (95% CI)	1.292 (1.249, 1.336)	1.271 (1.223, 1.318)	0.437 (0.411, 0.463)
P5, P50, P95	0.996, 1.378, 1.493	0.970, 1.183, 1.501	0.156, 0.448, 0.523

[1] Confidence intervals are normal-based

VWUO-MD effectively detects the relationship between S_x and S_y. The VWUO-MD estimates are very convincing (looking at magnitude). In 100% of the replicates, $w_{Sr} < w_{Sx}$ and w_{Sy}.

Figure 101. Type S, equal probability levels in S_x, large error



S_x = Bernoulli(.5)
S_y = Bernoulli(.2)*I(S_x=0) + Bernoulli(.8)*I(S_x=1)
S_r = Bernoulli(.5)

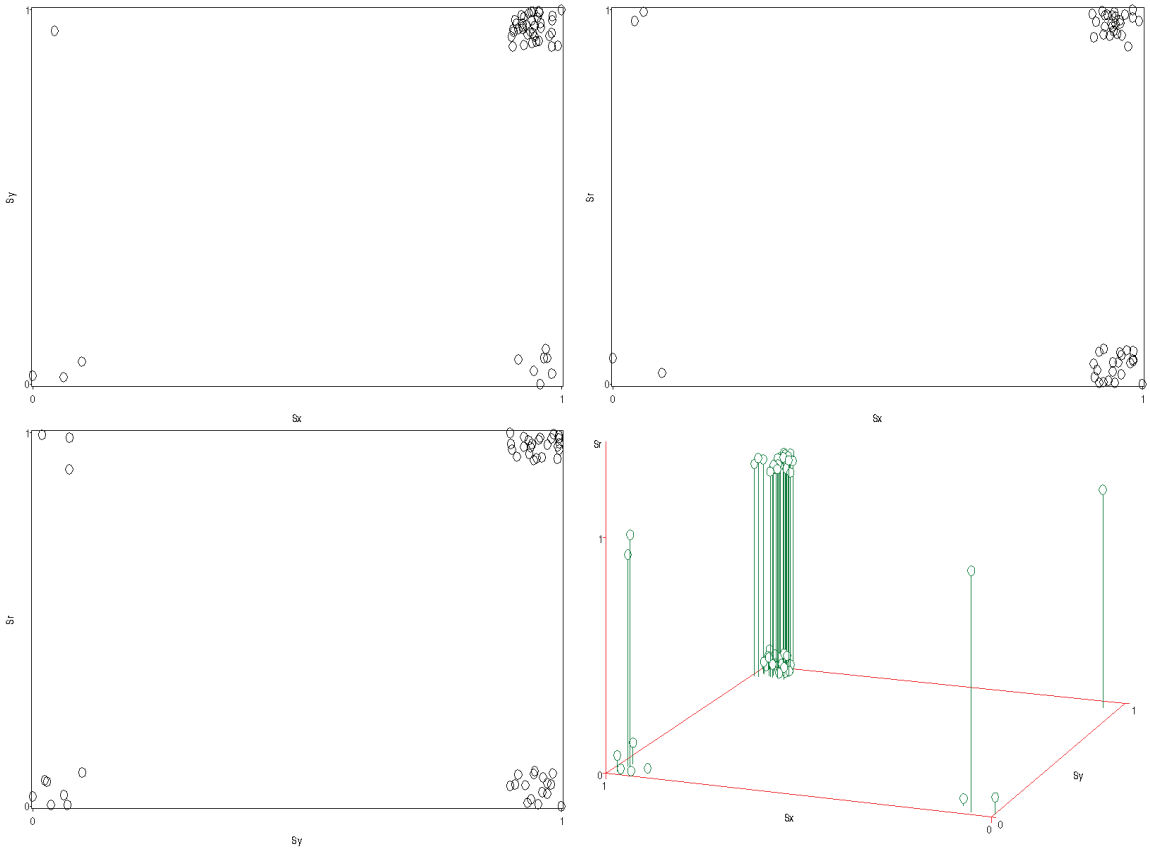
Table 80. Results for type S, equal probability levels in S_x, large error

Summary statistics	w_{Sx}	w_{Sy}	w_{Sr}
[1]Mean (95% CI)	1.187 (1.146, 1.228)	1.199 (1.154, 1.243)	0.614 (0.597, 0.632)
P5, P50, P95	0.951, 1.083, 1.397	0.919, 1.332, 1.407	0.454, 0.615, 0.666

[1] Confidence intervals are normal-based

With larger error terms, the estimated weights for S_x and S_y are diminished, but remain very strong compared to S_r, whose average variable weight is well below 1. In 99% of the replicates, $w_{Sr} < w_{Sx}$ and w_{Sy}.

Figure 102. Type S, higher probability of level 1 vs. 0 in S_x, small error



S_x = Bernoulli(.9)
S_y = Bernoulli(.1)*I(S_x=0) + Bernoulli(.9)*I(S_x=1)
S_r = Bernoulli(.5)

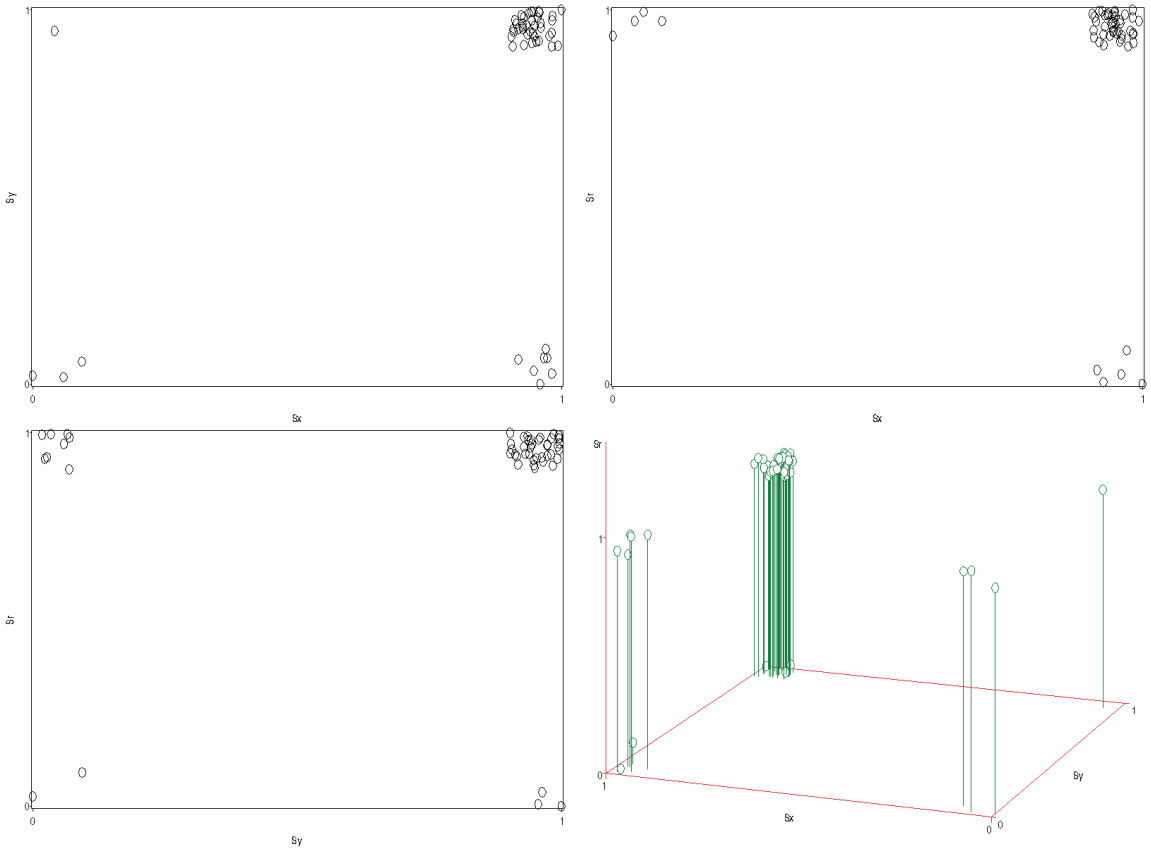
Table 81. Results for type S, higher probability of level 1 vs. 0 in S_x, small error

Summary statistics	w_{Sx}	w_{Sy}	w_{Sr}
[1]Mean (95% CI)	1.418 (1.369, 1.467)	1.281 (1.228, 1.333)	0.302 (0.281, 0.323)
P5, P50, P95	1.037, 1.517, 1.602	1.032, 1.154, 1.552	0.145, 0.308, 0.363

[1] Confidence intervals are normal-based

With unequal probability levels for S_x, but equal probability levels for S_r, VWUO-MD detects the relationship between S_x and S_y extremely convincingly, with an average weight for S_r barely above 0.3 and the remaining weight spread evenly between S_x and S_y. In 99% of the replicates, $w_{Sr} < w_{Sx}$ and w_{Sy}.

Figure 103. Type S, higher probability of level 1 vs. 0 in S_x and S_r, small error



S_x = Bernoulli(.9)
S_y = Bernoulli(.1)*I(S_x=0) + Bernoulli(.9)*I(S_x=1)
S_r = Bernoulli(.9)

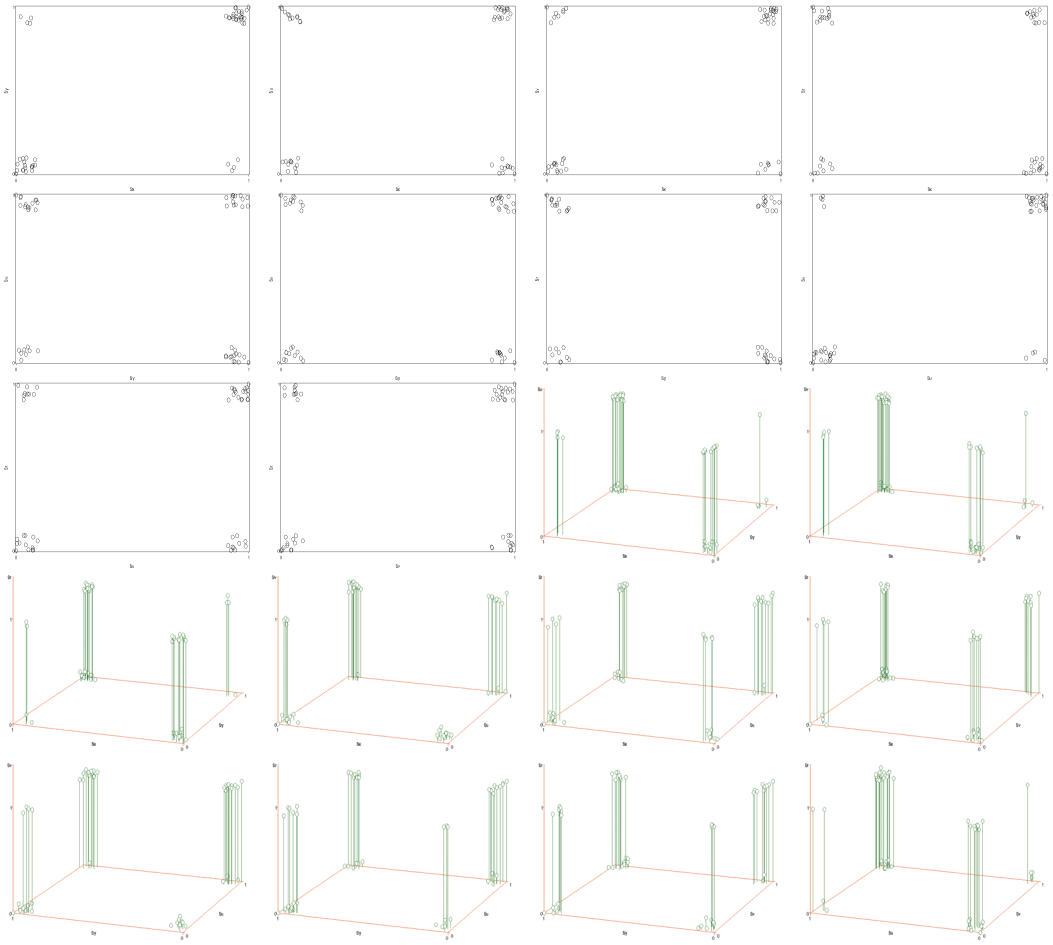
Table 82. Results for type S, higher probability of level 1 vs. 0 in S_x and S_r, small error

Summary statistics	w_{Sx}	w_{Sy}	w_{Sr}
[1]Mean (95% CI)	1.567 (1.493, 1.642)	0.798 (0.696, 0.901)	0.634 (0.581, 0.687)
P5, P50, P95	0.637, 1.696, 1.782	0.329, 0.551, 0.916	0.262, 0.684, 0.892

[1] Confidence intervals are normal-based

When unequal probability is assigned to the levels of both S_x and S_r, VWUO-MD still detects the relationship between S_x and S_y (looking at magnitude), with w_{Sr} being the lowest weight on average, although admittedly w_{Sy} has also become rather low. Unfortunately, in only 47% of the replicates, $w_{Sr} < w_{Sx}$ and w_{Sy}.

Figure 104. Disjoint relationships: both are type S, equal probability levels in S_x, small error



S_x = Bernoulli(.5)
S_y = Bernoulli(.1)*I(S_x=0) + Bernoulli(.9)*I(S_x=1)
S_u = Bernoulli(.5)
S_v = Bernoulli(.1)*I(S_u=0) + Bernoulli(.9)*I(S_u=1)
S_r = Bernoulli(.5)

Table 83. Results for disjoint relationships: both are type S, equal probability levels in S_x, small error

Summary statistics	w_{Sx}	w_{Sy}	w_{Su}	w_{Sv}	w_{Sr}
[1]Mean (95% CI)	0.995 (0.969, 1.020)	0.992 (0.966, 1.018)	1.000 (0.974, 1.026)	1.004 (0.977, 1.031)	1.010 (1.002, 1.017)
P5, P50, P95	0.836, 0.915, 1.126	0.838, 0.911, 1.124	0.829, 1.072, 1.131	0.836, 1.081, 1.135	0.949, 1.013, 1.035

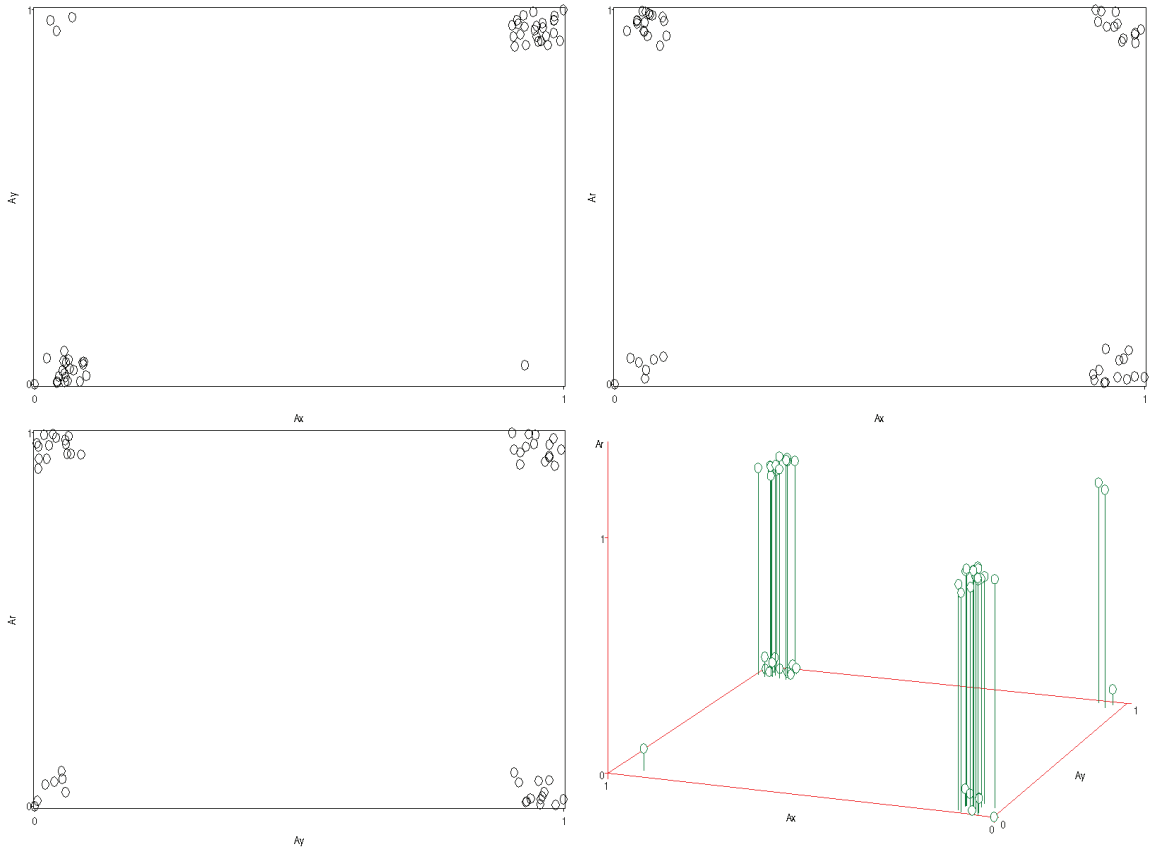
[1] Confidence intervals are normal-based

With disjoint relationships, w_{Sr} is no longer the smallest weight on average. VWUO-MD is having trouble with disjoint type S definitions. In only 1% of the replicates, $w_{Sr} < w_{Sx}$, w_{Sy}, w_{Su} and w_{Sv}.

5.5 Type A data

Finally, we generated six type A data sets, replicated 100 times. Figure 105 to Figure 110 show the multivariate distributions of A_x, A_y and A_r in these data sets, as well as A_u and A_v in the last example. Plots are randomly jittered for improved visualization. The captions describe the distributions. The descriptors "linear", "quadratic" and "half-quadratic" that were previously used do not apply with type A data since there are only two levels. "Small error" versus "large error" describe the relative probabilities of deviating from the prescribed relationship between A_x and A_y. Details of each distribution are given in the footnote below its figure, then a very brief discussion of VWUO-MD's performance on that data set is made.

Figure 105. Type A, equal probability levels in A_x, small error



A_x = Bernoulli(.5)
A_y = Bernoulli(.1)*I(A_x=0) + Bernoulli(.9)*I(A_x=1)
A_r = Bernoulli(.5)

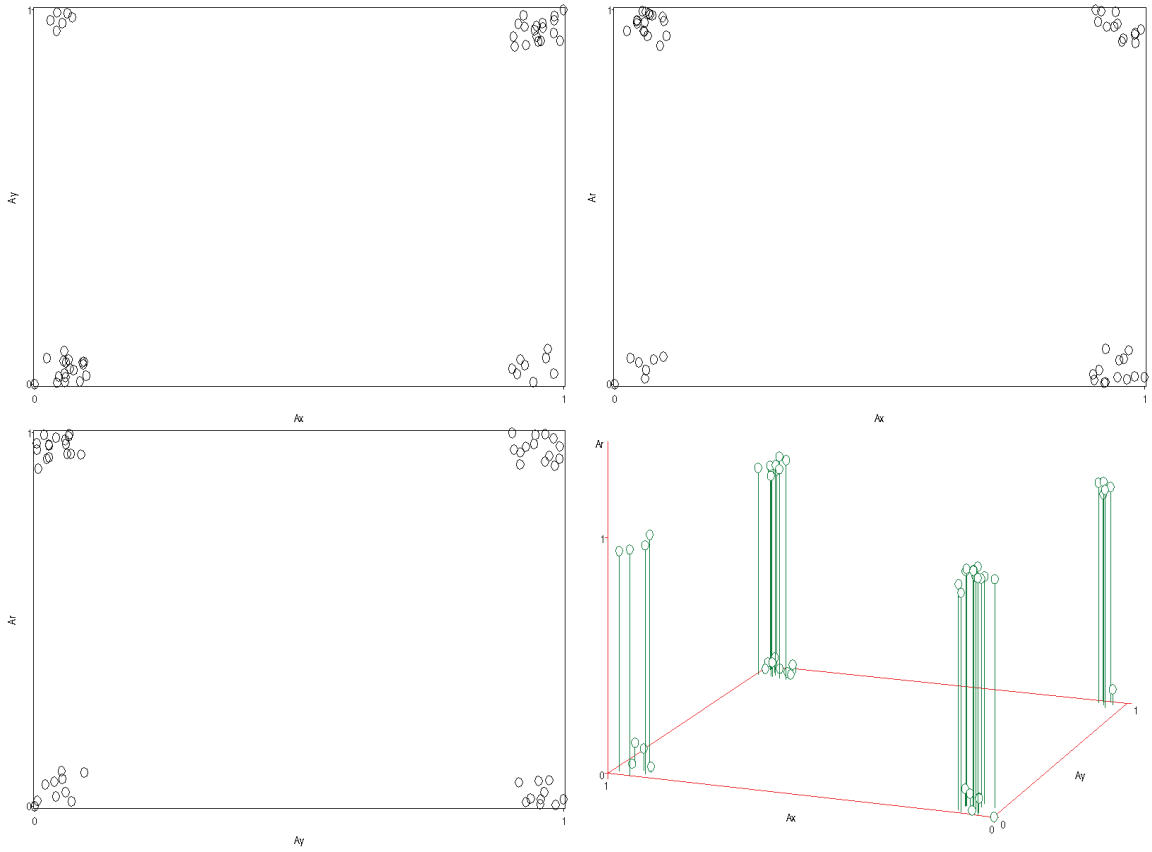
Table 84. Results for type A, equal probability levels in A_x, small error

Summary statistics	w_{Ax}	w_{Ay}	w_{Ar}
[1]Mean (95% CI)	1.370 (1.311, 1.429)	1.258 (1.202, 1.313)	0.372 (0.344, 0.401)
P5, P50, P95	1.018, 1.310, 1.507	0.821, 1.201, 1.455	0.079, 0.373, 0.454

[1] Confidence intervals are normal-based

VWUO-MD very effectively detects the relationship between A_x and A_y. In 100% of the replicates, $w_{Ar} < w_{Ax}$ and w_{Ay}.

Figure 106. Type A, equal probability levels in A_x, large error



A_x = Bernoulli(.5)
A_y = Bernoulli(.2)*I(A_x=0) + Bernoulli(.8)*I(A_x=1)
A_r = Bernoulli(.5)

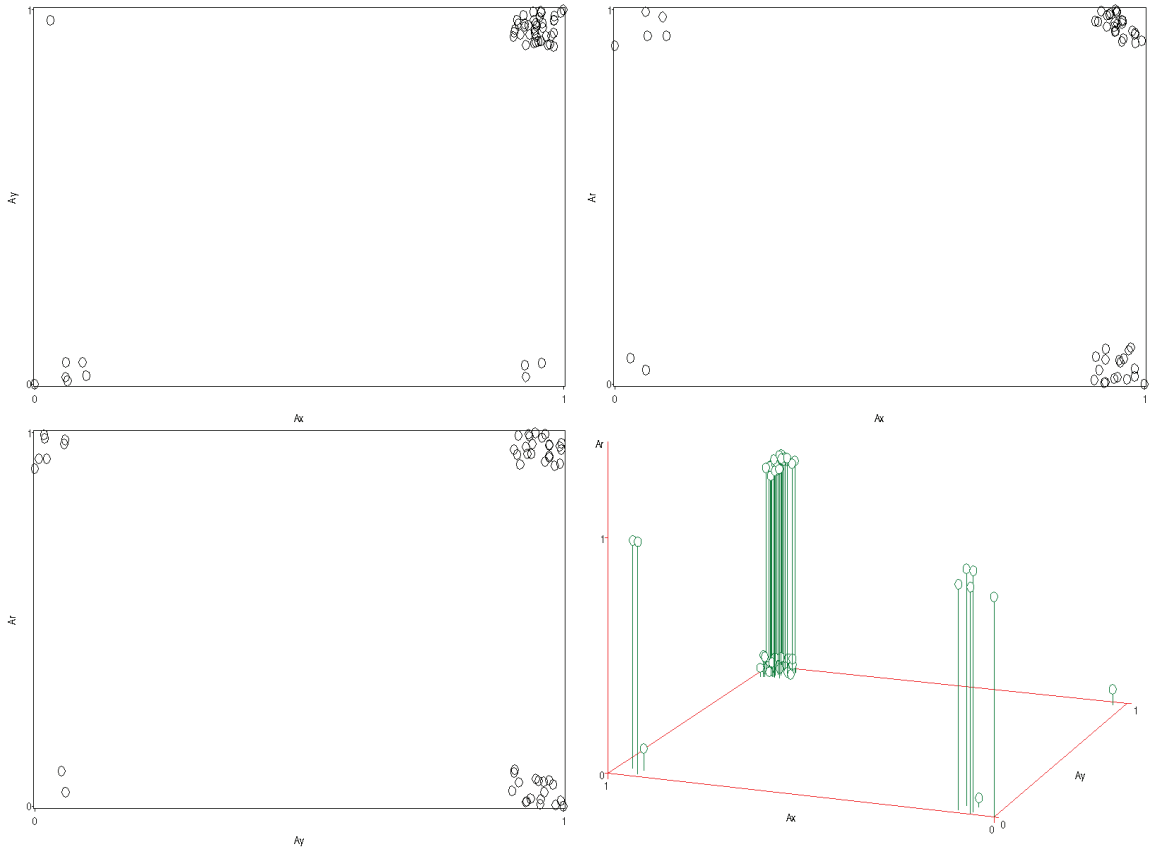
Table 85. Results for type A, equal probability levels in A_x, large error

Summary statistics	w_{Ax}	w_{Ay}	w_{Ar}
[1]Mean (95% CI)	1.224 (1.181, 1.267)	1.190 (1.147, 1.234)	0.586 (0.554, 0.617)
P5, P50, P95	0.973, 1.190, 1.283	0.815, 1.169, 1.292	0.359, 0.553, 0.735

[1] Confidence intervals are normal-based

With larger error terms, the estimated weights for A_x and A_y are diminished, but remain very strong compared to A_r, whose average variable weight is well below 1. In 99% of the replicates, $w_{Ar} < w_{Ax}$ and w_{Ay}.

Figure 107. Type A, higher probability of level 1 vs. 0 in A_x, small error



A_x = Bernoulli(.9)
A_y = Bernoulli(.1)*I(A_x=0) + Bernoulli(.9)*I(A_x=1)
A_r = Bernoulli(.5)

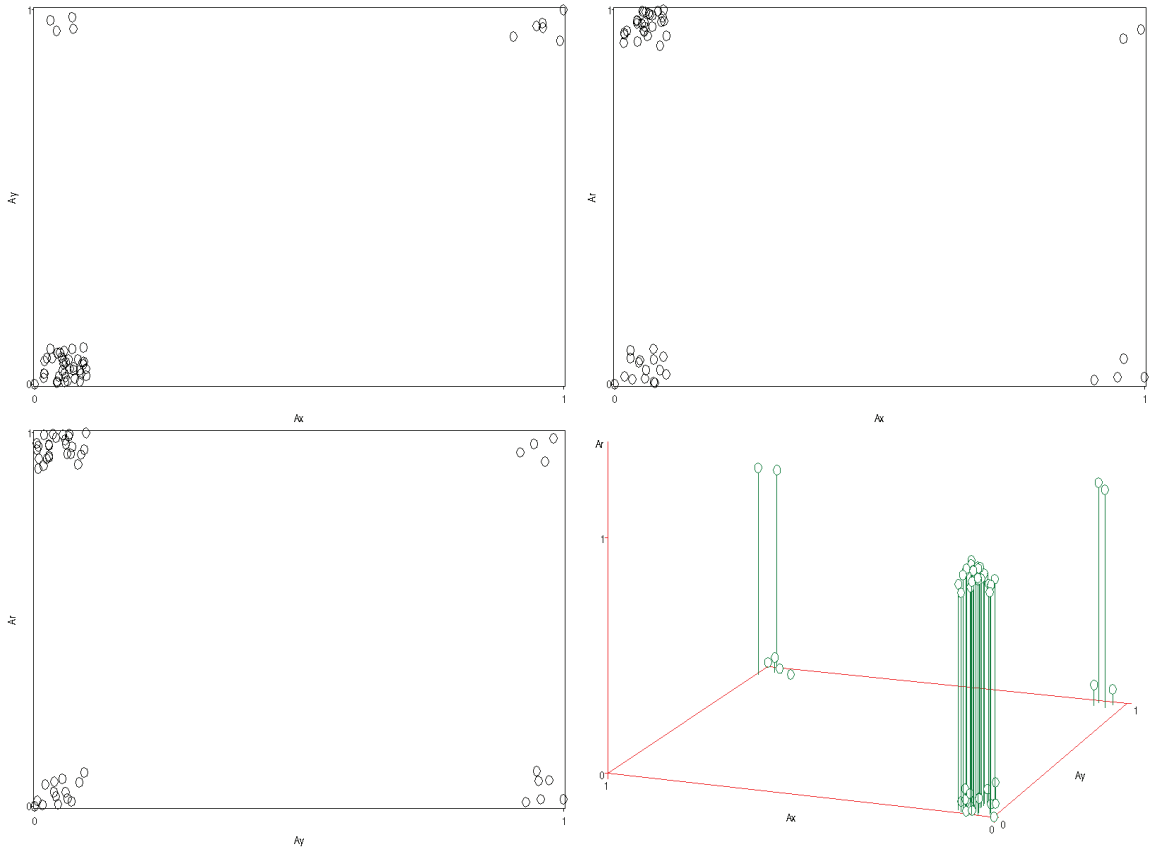
Table 86. Results for type A, higher probability of level 1 vs. 0 in A_x, small error

Summary statistics	w_{Ax}	w_{Ay}	w_{Ar}
[1]Mean (95% CI)	1.798 (1.745, 1.852)	0.999 (0.958, 1.040)	0.203 (0.182, 0.224)
P5, P50, P95	1.305, 1.960, 1.988	0.797, 0.925, 1.114	0.066, 0.167, 0.292

[1] Confidence intervals are normal-based

Increasing the probability of A_x=1 has dramatically increased VWUO-MD's ability to detect the relationship (looking at magnitude), when the noise variable is left evenly distributed between levels 0 and 1. In 100% of the replicates, w_{Ar} < w_{Ax} and w_{Ay}.

Figure 108. Type A, lower probability of level 1 vs. 0 in A_x, small error



A_x = Bernoulli(.1)
A_y = Bernoulli(.1)*I(A_x=0) + Bernoulli(.9)*I(A_x=1)
A_r = Bernoulli(.5)

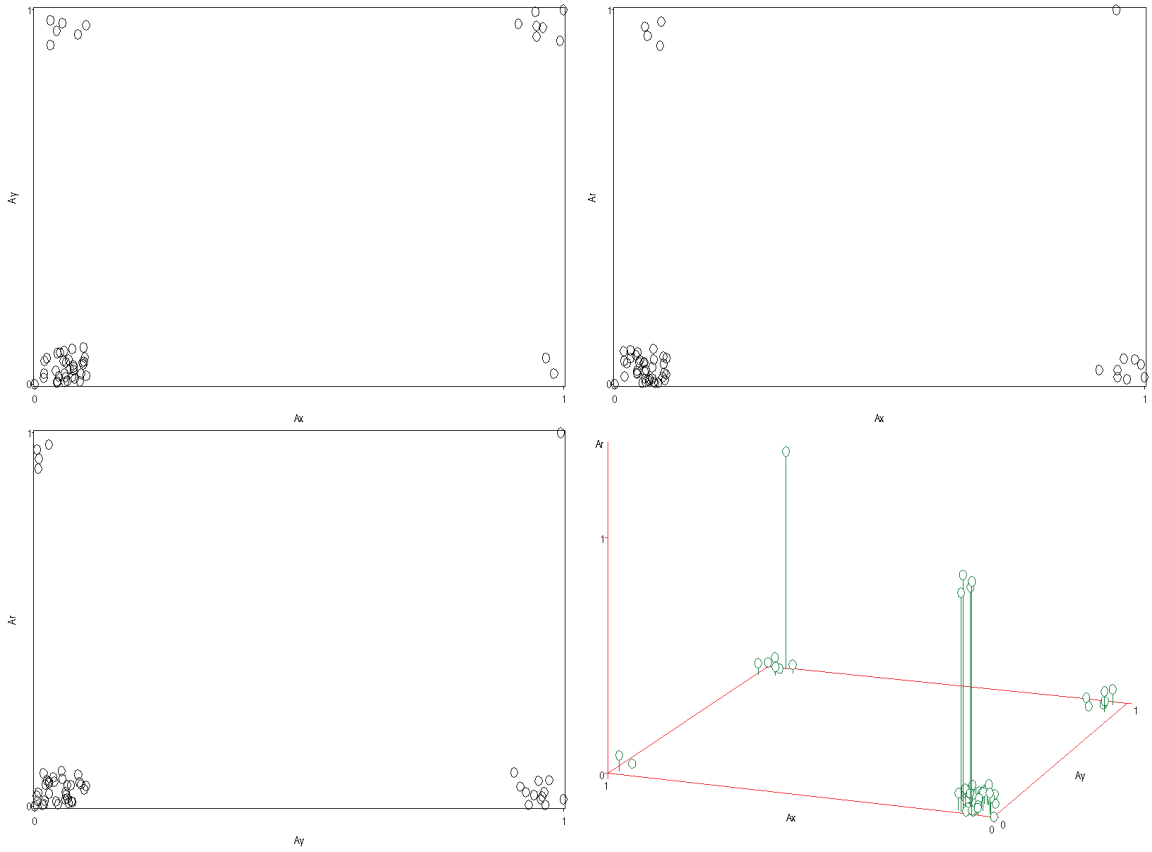
Table 87. Results for type A, lower probability of level 1 vs. 0 in A_x, small error

Summary statistics	w_{Ax}	w_{Ay}	w_{Ar}
[1]Mean (95% CI)	0.984 (0.919, 1.049)	1.487 (1.347, 1.627)	0.529 (0.418, 0.641)
P5, P50, P95	0.651, 0.874, 1.100	0.348, 1.717, 2.150	0.073, 0.233, 0.942

[1] Confidence intervals are normal-based

On the other hand, decreasing the probability of A_x=1 decreases VWUO-MD's ability to detect the relationship (when the noise variable is left evenly distributed between levels 0 and 1). In 69% of the replicates, $w_{Ar} < w_{Ax}$ and w_{Ay}.

Figure 109. Type A, lower probability of level 1 vs. 0 in A_x and A_r, small error



A_x = Bernoulli(.1)
A_y = Bernoulli(.1)*I(A_x=0) + Bernoulli(.9)*I(A_x=1)
A_r = Bernoulli(.1)

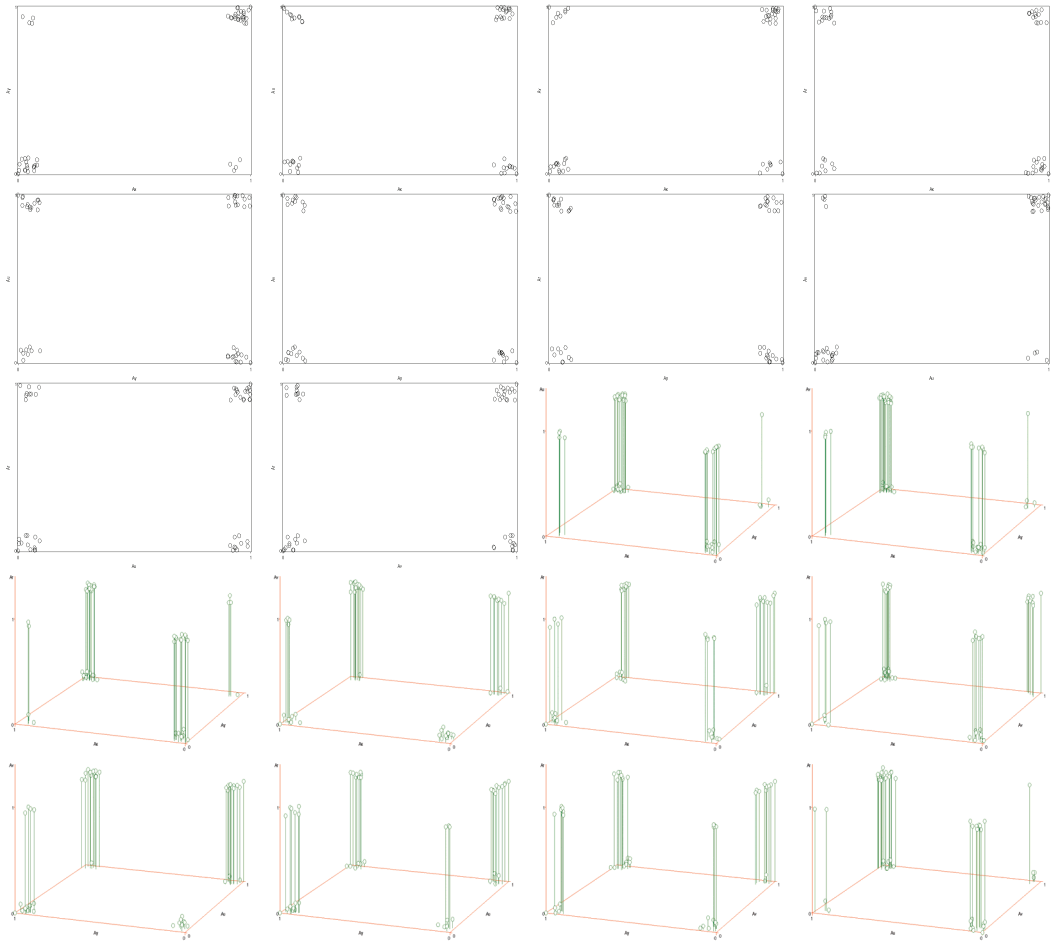
Table 88. Results for type A, lower probability of level 1 vs. 0 in A_x and A_r, small error

Summary statistics	w_{Ax}	w_{Ay}	w_{Ar}
[1]Mean (95% CI)	0.578 (0.509, 0.646)	1.995 (1.914, 2.075)	0.428 (0.381, 0.474)
P5, P50, P95	0.223, 0.521, 0.596	0.851, 2.147, 2.203	0.185, 0.362, 0.525

[1] Confidence intervals are normal-based

Finally, decreasing the probability of A_x=1 and A_r=1 further diminishes VWUO-MD's ability to detect the relationship. Now, A_r and A_x are both weighted low compared to A_y. However, A_r is still weighted below A_x, which is correct. In 65% of the replicates, $w_{Ar} < w_{Ax}$ and w_{Ay}.

Figure 110. Disjoint relationships: both are type A, equal probability levels in A_x, small error



A_x = Bernoulli(.5)
A_y = Bernoulli(.1)*I(A_x=0) + Bernoulli(.9)*I(A_x=1)
A_u = Bernoulli(.5)
A_v = Bernoulli(.1)*I(A_u=0) + Bernoulli(.9)*I(A_u=1)
A_r = Bernoulli(.5)

Table 89. Results for disjoint relationships: both are type A, equal probability levels in A_x, small error

Summary statistics	w_{Ax}	w_{Ay}	w_{Au}	w_{Av}	w_{Ar}
[1]Mean (95% CI)	1.006 (0.992, 1.020)	1.006 (0.992, 1.020)	1.009 (0.995, 1.023)	1.011 (0.997, 1.025)	0.968 (0.961, 0.976)
P5, P50, P95	0.906, 0.991, 1.075	0.912, 1.001, 1.072	0.907, 1.043, 1.071	0.910, 1.053, 1.074	0.898, 0.973, 0.996

[1] Confidence intervals are normal-based

With disjoint relationships, w_{Ar} is still the smallest weight on average.

However, in only 28% of the replicates, $w_{Ar} < w_{Ax}$, w_{Ay}, w_{Au} and w_{Av}.

5.6 Summary of performance on a variety of data shapes

In this chapter, we discovered some of the strengths and limitations of VWUO-MD as a hypothesis generating methodology when applied to data of a variety of types and shapes, with or without the involvement of clustering. There are some data types (notably types N, S and A) that VWUO-MD seems to be able to handle in a variety of scenarios. However, there are certain scenarios under other types that give VWUO-MD trouble. For example, a simple linear relationship between unclustered variables, or $k>2$ linearly placed clusters in a two-dimensional plane cause VWUO-MD to perform badly. On the other hand, quadratic relationships between type C or type O variables (as well as half-quadratic relationships between type O variables) are discovered easily by VWUO-MD, with or without clustering. When type S variables all have low probabilities (e.g., 10%), VWUO-MD experiences some trouble detecting known relationships. Fortunately, binary variables with low probabilities should be treated as type A (since matching 0s generally means less than matching 1s in such cases), and when type A variables all have low probabilities, VWUO-MD performs somewhat better. We also looked the effect of disjoint relationships on performance. Type C estimates held up well, both considering average variable weights, and the percentage of replicates for which the noise variable was weighted the lowest. Types O, N and A estimates also held up well considering average magnitude of the weights, but suffered from a reduced percentage of replicates for which the noise variable was weighted the lowest. Type S

estimates did not hold up well on either metric. Disjoint relationships between type S variables seems to be a real problem for VWUO-MD.

CHAPTER 6: AN APPLICATION OF VWUO-MD

6.1 The Joint Canada/United States Survey of Health

In this chapter we will perform a VWUO-MD analysis of the Joint Canada/United States Survey of Health (JCUSH), a collaborative project undertaken in 2004 by the Health Statistics Division of Statistics Canada and the National Center for Health Statistics (NCHS) of the United States Centers for Disease Control and Prevention.[77] With a total sample size of 8688, the JCUSH collected information from many categories, including:

- Health status
- Limitation of activities
- Asthma, arthritis, heart disease, diabetes, and depression
- Contact with mental health professionals
- Smoking
- Height and weight
- Health care utilization
- Dental visits
- Insurance, including single service plans
- Patient satisfaction

- Physical activities

6.2 Strategy for analysis

We have intended VWUO-MD to be a hypothesis generating method, but being order n^3 the method is practically restricted to small data sets. Therefore, in general VWUO-MD should either be used to mine information in a selected segment of the population where data are not plentiful, or be performed on many smaller sub-samples while looking for consistency of results. In this analysis, we will use VWUO-MD to mine information in a selected segment of the population. We will try to develop new hypotheses and tease out the relationships between variables that define this group, from the plethora of variables available on the JCUSH.

In addition to being sensitive to *n*, the speed of VWUO-MD is also sensitive to *p*, the number of variables, due to the large matrices involved in estimation. In this analysis we will reduce the dimensionality to ≤20 variables on any given run. This can be accomplished by splitting the set of variables into smaller subspaces and performing VWUO-MD analyses of each smaller subspace as a preprocessing step. The least important variables will be dropped from each subspace, and the remaining variables will be combined into the final stage subspace for analysis. This procedure is potentially sensitive to the choice of initial subspaces, since the clustering on one set of variables is not necessarily the same as on another set of variables.[39,43,47,59] Our choice will be guided by a result we obtained earlier. Recall that with multi-type subspaces involving more types, the normalizing constants did not perform as well. This makes type-

specific subspaces a natural choice for preprocessing. After reducing the dimensionality to p=19 in preprocessing, we will use backwards elimination to reduce the dimensionality by an additional five variables.

While the U-statistic-based covariance matrix estimator was shown to be good in data with at least four variables involving only types C and N, it does not work as well on types O or A, but furthermore we are reminded that on complex survey data (like JCUSH) for which bootstrap weights are developed, the bootstrap estimator should always be used, because the U-statistic-based estimator assumes that the data were collected in an SRS. Our final bootstrap analysis will be performed on the reduced subspace with p=14 variables.

Earlier we quoted a study finding that for complex regression models, at least 400 bootstrap weights is required for stability of p-values.[75] However, no such study can cover all possible techniques under all possible scenarios. Therefore, if 1000 bootstrap weights are available on a data set (which is true on many large, complex sample surveys today), then all 1000 should be used. Being from a complex survey design, the JCUSH data include a full sample weight as well as 1000 bootstrap weights, and we will use all 1001 sample weights in our analyses. For numerical stability, every sample weight will be rescaled to sum to the sample size before analysis; this should not affect the estimates except to better ensure numerical stability.

We will study the segment of the population characterized as working, mature students 50 years or older who received health care services in the past 12 months (from the interview date). This is an unusual group, and may present

unusual characteristics with respect to demographics, physical and mental health, health care utilization and satisfaction, and habits such as smoking. In additional, any of these characteristics or the relationships between them may be affected by country (Canada versus the United States).

To begin with, we will select a representative set of variables from each area of subject matter with which to develop hypotheses. We will randomly split our data into two halves, developing hypotheses on the training half, and testing them on the testing half. This is important for avoiding spurious statistical significance. Statistical comparisons between variable weights in the final, mixed-type analysis will be made with bootstrap percentile confidence intervals, Bonferroni-adjusted for multiple comparisons.

6.3 Description of the data

Twenty-five variables were extracted from JCUSH, covering all the subject categories listed earlier. Table 90 lists the variables included in the analysis data set.

Table 90. JCUSH variables included in the analysis, ordered by types C, O, N and A and then alphabetical order

VWUO-MD variable (based on JCUSH variable)	Concept (units and range, or categories)
cAge (DHJ1GAGE)	Age (years; 50 to 81)
cBMI (HWJ1DBMI)	Body mass index (kg/m^2; 17.3 to 59.0)
oCarequal (SAJ1_11A)	Quality of health care services received in the past 12 months (1=Excellent; 2=Good; 3=Fair/poor)
oDentist (DEJ1_2)	Last time visited dentist (1=Less than 1 year ago; 2=1 year to less than 2 years ago; 3=2+ years ago)
oEduc (SDJ1GHED)	Highest level of post-secondary education attained (1=Less than high school; 2=High school degree or equivalent (GED); 3=Trades certificate, vocational school, community college; 4=University or college including below Bachelor's degree)
oGenhealth (GHJ1DHDI)	Health description index (1=Poor/fair; 2=Good; 3=Very good; 4=Excellent)
oHhldsz (DHJ1GNHH)	Number of persons in household (1=1 person; 2=2 persons; 3=3 persons; 4=4+ persons)
oIncome (IWJ1DTHI)	Total household income from all sources (1=0-$19,999; 2=$20,000-$39,999; 3=$40,000-$59,999; 4=$60,000-$79,999; 5=$80,000+)
oPhysact (PAJ1DIND)	Physical activity index (1=Active; 2=Moderate; 3=Inactive)
oUsualact (PAJ1_6)	Level of physical activity for usual day (1=Usually sit; 2=Stand or walk quite a lot; 3=Usually lift or carry light loads/Do heavy work or carry very heavy loads)
nArthritis (CHJ1_3)	Has arthritis excluding fibromyalgia (0=No; 1=Yes)
nCanadian (SPJ1_TYP)	Sample type (0=United States sample; 1=Canada sample)
nEthnic (SDJ1DRC & SDJ1DRUS)	Racial origin (0=White only; 1=Other race or a multiple race)
nHypertens (CHJ1_5)	Has high blood pressure (0=No; 1=Yes)
nMale (DHJ1_SEX)	Sex (0=Female; 1=Male)
nMarital (SDJ1GMS)	Marital status (1=Married/common-law/partner; 2=Widowed; 3=Separated/divorced; 4=Single, never married)
nNoinsurdt (ISJ1_2)	No insurance - dental expenses (0=Has dental insurance; 1=Does not have dental insurance)
nSmoker (SMJ1_4)	Type of smoker (0=Not at all; 1=Some days/Every day)
aDepressed (DPJ1DPP)	Depression Scale - predicted probability (0=Less than 50%; 1=50+%)

aDiabetes (CHJ1_7A & CHJ1_7B)	Diagnosed with diabetes other than during pregnancy (0=No; 1=Yes)
aJogging (PAJ1_1J)	Activity in last 3 months - jogging or running (0=No; 1=Yes)
aRedacthm (RAJ1_2A)	Frequency - reduction in activities at home due to a long-term physical condition or mental condition or health problem (0=Never; 1=Sometimes/often)
aRedactsc (RAJ1_2B1)	Frequency - reduction in activities at school due to a long-term physical condition or mental condition or health problem (0=Never; 1=Sometimes/often)
aRedactwk (RAJ1_2B2)	Frequency - reduction in activities at work due to a long-term physical condition or mental condition or health problem (0=Never; 1=Sometimes/often)
aServment (CMJ1_01K)	Mental health - has consulted with a health professional in the past 12 months (0=No; 1=Yes)

The data set characterized as working, mature students 50 years or older who received health care services in the past 12 months had 167 records with no missing variables among the 25 in our analysis. Every sample weight in the full sample was rescaled to sum to the sample size. Next we randomly split the data set into two parts, the first (training segment) with 83 records and the second (testing segment) with 84 records. With the exception of nCanadian, two-level discrete variables were treated as type A if the weighted prevalence in the full sample was ≤20%, otherwise they were treated as type N. For obvious reasons, nCanadian had a weighted prevalence of 10.3% Canadian, but was treated as type N under the conceptual considerations discussed earlier with the type A distance formula, namely that two Canadian citizens are conceptually no more similar than two US citizens. For numerical stability, categorical variables were collapsed where feasible to ensure at least 10 records per level in the full data, so that in most cases we would have at least five records per level in each half of

the data. The exceptions in the full data were oEduc with nine persons having less than high school, and aDepressed with nine persons being probably depressed. Exceptions in the halved data were oEduc, nMarital and aDepressed with four records in their smallest groups.

6.4 Preprocessing and backwards elimination to reduce dimensionality

The next step was to reduce dimensionality by performing VWUO-MD analyses of each type-specific subspace in the training data set. Since the first stage of reduction is a pre-processing step only, variables should be removed conservatively. It was decided to keep both type C variables at this stage, and utilize preprocessing to drop the least important two variables of each of the other three types.

Each type-specific subspace in the training data set was analyzed with VWUO-MD starting from **w=1** and utilizing 10 random restarts. The full sample weight was used. Bootstrapping was not performed in this preprocessing stage, because the number of variables to be retained from each type was decided in advance and would not depend on statistical significance. Although preprocessing was performed by design using the training data set, as a consistency check (that would not influence methodology) the same analyses were performed on the testing data set as well.

Table 91 lists the solution vectors from each subspace on both data sets. Based on the results in the training data, we will drop oPhysact, oUsualact, nMale, nHypertens, aDiabetes and aJogging. Type C results are not consistent

between the training and testing data sets. However, there only being two type C variables, we are not dropping either one in preprocessing based on these results, and so far we have not performed bootstrapping and so do not yet have any idea about statistical significance. Type O, type N and type A results are consistent between the training and testing data sets at least as far as the set of two smallest variable weights in each type. The order of the more important variables differs somewhat.

Table 91. VWUO-MD solutions on each type-specific subspace of the training and testing JCUSH data sets; analyses were weighted with the full sample weight, started at w=1, with 10 random restarts

Subspace type	Variable X	w_X on training data	w_X on testing data
C	cAge	1.157038	0.779702
	cBMI	0.842962	1.220298
O	oCarequal	0.975506	1.031765
	oDentist	1.029953	1.060696
	oEduc	1.140456	1.056183
	oGenhealth	1.022388	1.016197
	oHhldsz	1.011677	0.993640
	oIncome	1.058515	0.983269
	oPhysact (D_0)	0.811077	0.903510
	oUsualact (D_0)	0.950428	0.954741
N	nArthritis	0.958692	1.005879
	nCanadian	1.131308	1.093556
	nEthnic	0.911462	0.975950
	nHypertens (D_0)	0.890249	0.921929
	nMale (D_0)	0.885718	0.867019
	nMarital	1.183638	1.174143
	nNoinsurdt	1.041005	0.992037
	nSmoker	0.997928	0.969487
A	aDepressed	1.122616	1.098128
	aDiabetes (D_0)	0.500541	0.949799
	aJogging (D_0)	0.695837	0.748518
	aRedacthm	1.205922	1.050566
	aRedactsc	1.196060	0.995158
	aRedactwk	1.187510	1.094812
	aServment	1.016551	0.984988

(D_0)=Dropped in preprocessing

The second stage of reduction is to combine the remaining variables that passed the prescreening into a mixed-type data set, and perform backwards

elimination to reduce the dimensionality by another five variables. The backwards elimination analyses were run on the training data set with the full sample weight, started at **w=1**, with 10 random restarts. Table 92 lists the six solution vectors starting with the *p*=19 subspace obtained above and ending in the final *p*=14 subspace on which our bootstrap analyses will be performed. The five variables removed at this stage, in order, were oCarequal, oHhldsz, oGenhealth, cBMI and oDentist.

Table 92. VWUO-MD solutions during backwards elimination of the JCUSH training data set from *p*=19 to *p*=14; analyses were weighted with the full sample weight, started at w=1, with 10 random restarts

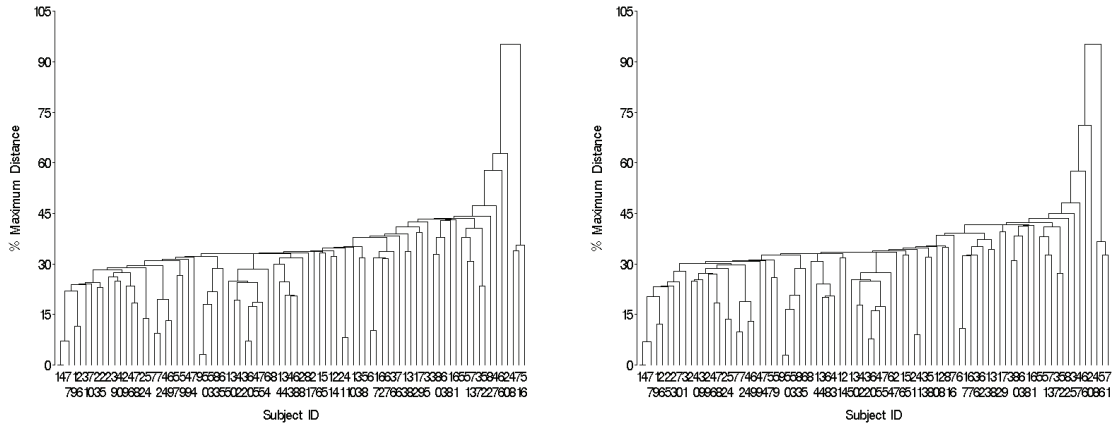
Variable X	Step 0 w_X	Step 1 w_X	Step 2 w_X	Step 3 w_X	Step 4 w_X	Step 5 w_X
cAge	0.959703	0.954556	0.948325	0.929032	0.912336	0.901920
cBMI	0.945643	0.917335	0.902211	0.860729	(D_4)	(D_4)
oCarequal	0.860661	(D_1)	(D_1)	(D_1)	(D_1)	(D_1)
oDentist	0.927313	0.908147	0.891889	0.886171	0.855580	(D_5)
oEduc	1.013775	0.999294	0.955457	0.938581	0.932663	0.954633
oGenhealth	0.922035	0.919052	0.883479	(D_3)	(D_3)	(D_3)
oHhldsz	0.873220	0.849418	(D_2)	(D_2)	(D_2)	(D_2)
oIncome	0.982626	0.947424	0.908006	0.904055	0.877777	0.831205
oPhysact	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
oUsualact	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
nArthritis	1.009608	0.994064	0.981263	0.962177	0.940500	0.912388
nCanadian	1.026119	1.026586	1.031201	1.029416	1.021583	1.014939
nEthnic	0.990893	0.973630	0.963638	0.937799	0.918241	0.872531
nHypertens	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
nMale	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
nMarital	1.066463	1.065914	1.075728	1.072797	1.080908	1.083232
nNoinsurdt	1.003048	0.998386	0.995184	0.971607	0.957677	0.948078
nSmoker	0.990825	0.988890	0.982326	0.976380	0.952456	0.897468
aDepressed	1.085118	1.088844	1.094072	1.100919	1.105403	1.107812
aDiabetes	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
aJogging	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)	(D_0)
aRedacthm	1.091594	1.098514	1.106215	1.115883	1.122547	1.135694
aRedactsc	1.091293	1.098269	1.105731	1.115121	1.122122	1.133663
aRedactwk	1.088843	1.095146	1.103461	1.112057	1.117690	1.128702
aServment	1.071220	1.076530	1.071814	1.087276	1.082518	1.077734

(D_i)=Dropped after *i* steps of backwards elimination (*i*=0 if dropped in preprocessing)

6.5 VWUO-MD analysis of the reduced JCUSH data set

The dendrograms (single linkage) based on the unweighted versus variable-weighted (with the reduced variable weights solution obtained above) distance matrices are shown in Figure 111. There is no discernable difference, which we have come to expect.

Figure 111. Dendrograms (single linkage) based on unweighted (left) versus variable-weighted (right) distance matrices from reduced JCUSH data set



During backwards elimination, all six analyses found their best local minima at the first (default) solution, and found no other local minima at any step in 10 random restarts. For this reason, we will perform bootstrap analyses of the final reduced subspace *without* random restarts, to improve efficiency. VWUO-MD was thus run on the reduced JCUSH data set using the 1000 bootstrap weights. The bootstrap correlation matrix calculated from $\hat{Var}_{BS}(\hat{\mathbf{w}})$ is shown in Table 93. This is not based on exactly the estimator $\hat{Var}_{BS(p-1)}(\hat{\mathbf{w}})$ developed earlier, since $\hat{Var}_{BS}(\hat{\mathbf{w}})$ includes the p^{th} variable and is therefore a singular matrix. However, for estimating the variance of individual weight estimates (or in fact any

contrast of weights not involving all of them at once), use of the full matrix is asymptotically equivalent, more convenient, and possibly more stable than calculating the variance of the last variable weight using all the entries in $\hat{Var}_{BS(p-1)}(\hat{\mathbf{w}})$. The asymptotic equivalence can be seen by an examination of the bootstrap covariance matrix estimator, the fact that the last variable weight is *p* minus the sum of the other weights, and the near invariance to changes in variable order that involve the last variable. The bootstrap correlation matrix shows us that the biggest correlation is between the variable weights for aRedacthm and aRedactsc, ρ=0.949. Considering the meaning of these variables, this is not surprising, and will affect our analysis below.

Table 93. $\hat{C}orr_{BS}(\hat{\mathbf{w}})$ **from reduced JCUSH data set**

Variable	cAge	oEduc	oIncome	nArthritis	nCanadian	nEthnic	nMarital	nNoinsurdt	nSmoker	aDepressed	aRedacthm	aRedactsc	aRedactwk	aServment
cAge	1.000	-0.239	-0.428	-0.280	-0.150	-0.355	-0.083	-0.204	-0.249	-0.116	0.037	0.028	-0.042	-0.147
oEduc	-0.239	1.000	0.119	-0.148	-0.028	-0.134	-0.198	-0.135	-0.164	-0.098	-0.110	-0.086	-0.136	-0.231
oIncome	-0.428	0.119	1.000	0.008	-0.083	0.007	0.231	-0.195	-0.219	-0.005	-0.027	-0.023	0.057	-0.214
nArthritis	-0.280	-0.148	0.008	1.000	0.053	0.257	0.009	0.009	-0.233	-0.172	0.036	0.029	-0.071	-0.115
nCanadian	-0.150	-0.028	-0.083	0.053	1.000	-0.091	0.108	0.143	-0.139	0.092	-0.098	-0.064	-0.018	-0.017
nEthnic	-0.355	-0.134	0.007	0.257	-0.091	1.000	-0.200	-0.170	0.313	-0.263	0.035	0.005	-0.081	-0.337
nMarital	-0.083	-0.198	0.231	0.009	0.108	-0.200	1.000	0.127	-0.222	0.168	-0.116	-0.095	0.049	0.070
nNoinsurdt	-0.204	-0.135	-0.195	0.009	0.143	-0.170	0.127	1.000	-0.070	0.111	-0.228	-0.215	-0.014	0.404
nSmoker	-0.249	-0.164	-0.219	-0.233	-0.139	0.313	-0.222	-0.070	1.000	-0.058	-0.075	-0.103	-0.121	0.156
aDepressed	-0.116	-0.098	-0.005	-0.172	0.092	-0.263	0.168	0.111	-0.058	1.000	-0.171	-0.137	-0.024	0.356
aRedacthm	0.037	-0.110	-0.027	0.036	-0.098	0.035	-0.116	-0.228	-0.075	-0.171	1.000	0.949	0.535	-0.167
aRedactsc	0.028	-0.086	-0.023	0.029	-0.064	0.005	-0.095	-0.215	-0.103	-0.137	0.949	1.000	0.514	-0.160
aRedactwk	-0.042	-0.136	0.057	-0.071	-0.018	-0.081	0.049	-0.014	-0.121	-0.024	0.535	0.514	1.000	-0.039
aServment	-0.147	-0.231	-0.214	-0.115	-0.017	-0.337	0.070	0.404	0.156	0.356	-0.167	-0.160	-0.039	1.000

We generated unadjusted and Bonferroni-adjusted simultaneous confidence intervals for individual variable weights based first on bootstrap percentile confidence intervals, and then for comparison, the same intervals based on univariate normal distributions and the bootstrap standard error estimates. These are listed in Table 94. The bootstrap percentile confidence intervals are extremely close to the normal-based intervals with bootstrap standard errors, different by only about 1%. This is not altogether surprising, considering our earlier observation that more and mixed-type variables produced straighter Q-Q plots and less significant Henze-Zirkler T-tests for multivariate normality of $\hat{\mathbf{w}}_{(\mathbf{p-1})}$. This observation offers support to the idea of using the U-statistic-based variance estimator in high-dimensional scenarios that are sampled in an SRS, at a sizable savings in computational resources.

Table 94. Unadjusted and Bonferroni-adjusted simultaneous 95% confidence intervals for individual variable weights from reduced JCUSH data set

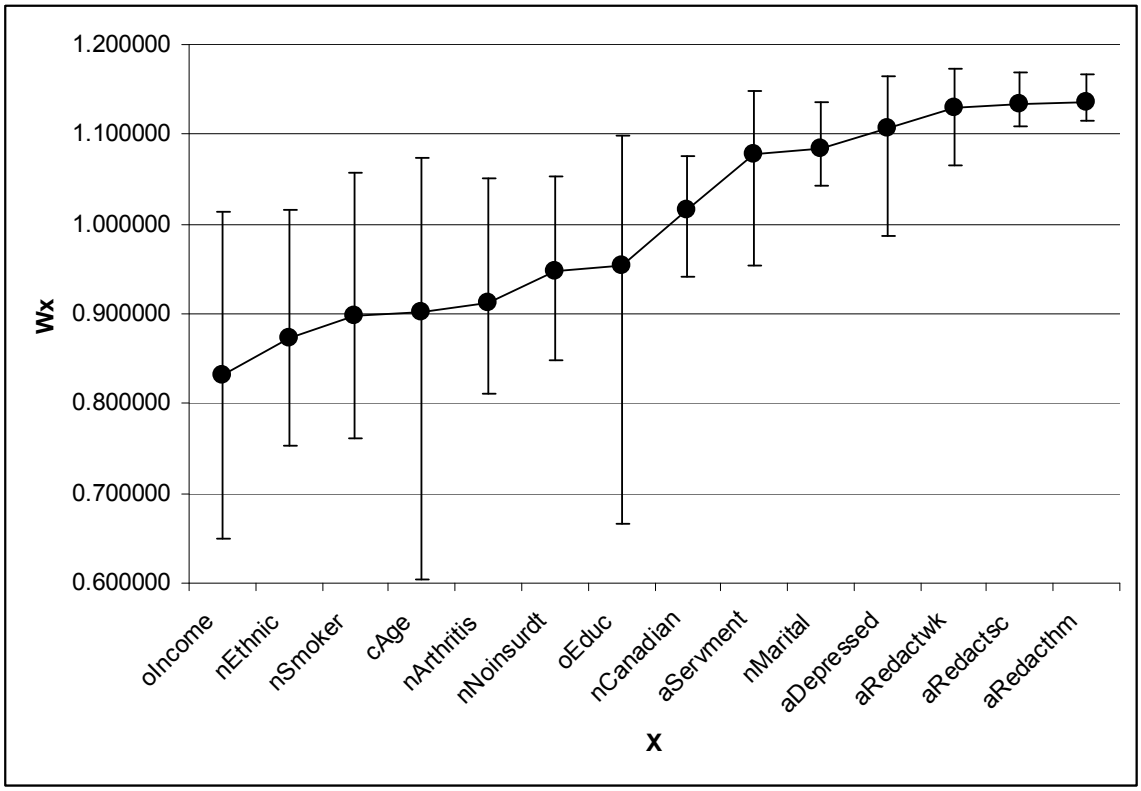
Variable X	Bootstrap percentile 95% CI for w_X		Normal-based (with bootstrap SE) 95% CI for w_X	
	Unadjusted	Bonferroni-adjusted	Unadjusted	Bonferroni-adjusted
cAge	(0.687427, 1.027463)	(0.603144, 1.074519)	(0.711609, 1.092230)	(0.619000, 1.184839)
oEduc	(0.850708, 1.061456)	(0.667105, 1.099406)	(0.840146, 1.069121)	(0.784433, 1.124833)
oIncome	(0.722102, 0.941596)	(0.649583, 1.014692)	(0.719531, 0.942879)	(0.665189, 0.997222)
nArthritis	(0.838832, 1.002171)	(0.810996, 1.051303)	(0.829631, 0.995146)	(0.789360, 1.035417)
nCanadian	(0.965803, 1.057615)	(0.941369, 1.075240)	(0.968172, 1.061707)	(0.945413, 1.084466)
nEthnic	(0.798378, 0.965081)	(0.753100, 1.016674)	(0.786887, 0.958174)	(0.745212, 0.999850)
nMarital	(1.055552, 1.108524)	(1.043143, 1.135498)	(1.054480, 1.111984)	(1.040489, 1.125975)
nNoinsurdt	(0.892650, 1.018919)	(0.848811, 1.052219)	(0.884400, 1.011757)	(0.853413, 1.042744)
nSmoker	(0.814952, 1.007351)	(0.761916, 1.057962)	(0.798307, 0.996629)	(0.750054, 1.044883)
aDepressed	(1.053082, 1.141697)	(0.986039, 1.164237)	(1.064759, 1.150865)	(1.043808, 1.171815)
aRedacthm	(1.120952, 1.156084)	(1.115901, 1.167558)	(1.117771, 1.153617)	(1.109049, 1.162339)
aRedactsc	(1.115135, 1.156366)	(1.108674, 1.169804)	(1.113526, 1.153800)	(1.103727, 1.163599)
aRedactwk	(1.099529, 1.154131)	(1.065904, 1.172619)	(1.102746, 1.154659)	(1.090115, 1.167290)
aServment	(1.001860, 1.131790)	(0.954111, 1.147754)	(1.011531, 1.143936)	(0.979316, 1.176152)

The Bonferroni-adjusted bootstrap percentile confidence intervals are plotted in Figure 112, arranged in order of increasing variable weight. We do not know in advance how many variables we are looking for, but the graph can guide us in this, similar to the way a scree plot helps one decide on the appropriate number of factors during exploratory factor analysis.[12] The scree plot of VWUO-

MD weights illustrates a potential problem. All the type A variable weights are quite large. This could indicate a need to revisit the calibration of the normalizing multipliers, however we may also recall that excluding the two lowest variables in the initial type A subspace analysis, the remaining weights in that subspace *were* relatively large. This might indicate a strong grouping amongst the remaining type A variables, which would lend credibility to the relatively large type A weights observed in the final, multi-type solution.

As VWUO-MD is a hypothesis generating methodology, we are afforded some flexibility in selecting a set of variables for analysis. One might select the best k variables for some number k, all variables whose weights are $>t$ for some threshold t, or some subset of the variables that are both not weighted too low and interesting to the analyst. As long as one selects from those variables whose weights are not obviously *low*, one should have an improved chance of finding associations in the data, which is the entire point. In our example, the variable weights plotted in Figure 112 could be split into two groups: the bottom seven weights whose 95% CIs clearly cover values <0.9, and the top seven weights whose 95% CIs lie entirely above 0.9.

Figure 112. Scree plot of VWUO-MD weights from reduced JCUSH data set, with Bonferroni-adjusted 95% bootstrap percentile CIs



In a pure data mining analysis of these data without any additional exploration of the VWUO-MD method itself, we might retain only the upper scree plot group of variables for further study regardless of balance between groups. In our case, having an equal number of variables in the lower and upper scree plot groups will be helpful for investigating additional properties of the VWUO-MD method by this practical example. However, when models were fit to the upper set of variables, near colinearity was revealed between aRedacthm, aRedactsc and aRedactwk. This is not surprising considering the high bootstrap correlation we calculated between these variables' VWUO-MD weights, and it almost certainly contributed to these variables' high weights since near colinearity is in a way equivalent to strong clustering. This raises the idea that it might be a viable

option to eliminate such relationships in the data before one performs VWUO-MD analysis. However, that would be counter to the theme of VWUO-MD, which is to *find* such relationships and generate hypotheses with minimal intervention. The solution in our case is to combine the top three variables into one named aRedact, which is equal to 1 if any of the three variables are 1, or else 0. Since for this example we want to compare models between lower and upper scree plot groups, we will then even out the number of variables between groups by moving oEduc into the upper scree plot group model, for purposes of a more even comparison between groups.

Consideration of the upper scree plot group could suggest various hypotheses or models for further investigation. One logical choice is to fit a model predicting the variable (derived from those) with the highest weight, aRedact, from the remaining five variables in the upper scree plot group. For this analysis, we will use the testing data set in order to avoid spurious findings. With the training and testing data sets randomly selected from the full sample independently and without replacement, a random association in the training data set that may have led to a given set of variable weights is unlikely to also appear in the testing data set. Without taking this measure or an equivalent adjustment, p-values from statistical analyses based on hypotheses generated by VWUO-MD would not be valid estimates of Type I error probability.

The upper scree plot group logistic regression model predicts the logit transformed expectation of the probability of aRedact=1 from the regressors. The results, expressed as odds ratios and 95% CIs, are presented in Table 95. For

comparison, we fit three additional binary logistic regression models: 1) the lower scree plot group model predicting its highest weighted variable, nNoinsurdt, from the remaining five variables in that group; 2) the model predicting a function of the highest weighted variable in that group, oDentist=1+ years ago, from the other four variables dropped during backwards elimination; and 3) the model predicting the second highest weighted variable of those dropped in prescreening, nHypertens, from the other five variables dropped in prescreening. (The outcome variables in these models were chosen for optimal comparison between models; all models would be binary logistic models predicting the highest or second highest weighted variable in that group.) The lower scree plot group model is presented in Table 96. Note that age had to be categorized into the ordinal variable oAge in this model due to extremely poor fit (p-value<0.001 on the Hosmer-Lemeshow goodness of fit (GoF) test for logistic models[78]) when cAge (with or without an additional quadratic term) was included instead. The final model only just fails the fit test (p-value=0.027), and as that test is not bootstrap adjusted, we are comfortable with this result. In addition, fit will actually be one of the criteria of comparison. The backwards elimination rejects group model is presented in Table 97. We retained $cBMI^2$ as it was borderline statistically significant. This model barely fails the fit test (p-value=0.047), and as that test is not bootstrap adjusted, we are comfortable with this result. The prescreening rejects group model is presented in Table 98. All four models were fit using the Estimating Equations Bootstrap (EEB), a method of bootstrap variance estimation for logistic regression developed at Statistics Canada by

Roberts et al.[79] The EEB method calculates the bootstrap covariance matrix of a kernel within the logistic regression estimating equation rather than on the model coefficients directly. As such, there is "no problem with ill-conditioned matrices, provided that an ill-conditioned matrix is not encountered when fitting the model with the full sample; this a particular advantage with small samples."[79]

Table 95. Odds ratios and 95% CIs from the upper scree plot group logistic regression model; aRedact (derived from the three highest weighted variables) is predicted

Predictor variable	P-value	Odds Ratio (95% CI)
aDepressed	0.889	0.76 (0.02, 36.39)
aServment	0.014	12.25 (1.67, 90.02)
nCanadian	0.388	1.86 (0.46, 7.55)
[1]nMarital		
1=Married/common-law/partner	-	1.00
2=Widowed	0.545	2.37 (0.15, 38.81)
3=Separated/divorced	0.094	3.60 (0.80, 16.12)
4=Single, never married)	0.188	0.19 (0.02, 2.26)
[1,2]oEduc		
1/2=Less than high school/High school degree or GED	0.499	1.75 (0.34, 8.93)
3=Trades certificate, vocational school, community college	0.744	0.74 (0.12, 4.66)
4=University or college including below Bachelor's degree	-	1.00

[1] Reference categories were selected based on largest cell size
[2] Categories were collapsed for numerical stability

Table 96. Odds ratios and 95% CIs from the lower scree plot group logistic regression model; nNoinsurdt is predicted

Predictor variable	P-value	Odds Ratio (95% CI)
nArthritis	0.962	1.04 (0.19, 5.61)
nEthnic	0.638	0.65 (0.11, 3.83)
nSmoker	0.059	0.13 (0.02, 1.08)
[1]oAge		
1=50-54 years	-	1.00
2=55-59 years	0.113	4.53 (0.70, 29.46)
3=60-64 years	0.373	2.51 (0.33, 19.10)
4=65+ years	0.023	8.31 (1.33, 51.78)
[1,2]oIncome		
1/2=0-$19,999/$20,000-$39,999	0.093	5.33 (0.76, 37.49)
3=$40,000-$59,999	0.523	1.77 (0.31, 10.28)
4=$60,000-$79,999	0.693	1.51 (0.20, 11.63)
5=$80,000+	-	1.00

[1] Reference categories were selected based on largest cell size
[2] Categories were collapsed for numerical stability

Table 97. Odds ratios and 95% CIs from the backwards elimination rejects group logistic regression model; oDentist=1+ years ago is predicted

Predictor variable	P-value	Odds Ratio (95% CI)
cBMI	0.080	6.32 (0.8, 49.84)
cBMI[2]	0.070	0.96 (0.93, 1.00)
[1,2]oCarequal		
1=Excellent	-	1.00
2/3=Good/Fair/poor	0.433	1.69 (0.45, 6.31)
[1,2]oGenheath		
1/2=Poor/fair/Good	0.450	2.15 (0.29, 15.65)
3=Very good	-	1.00
4=Excellent	0.506	1.78 (0.32, 9.81)
[1]oHhldsz		
1=1 person	0.531	0.61 (0.13, 2.90)
2=2 persons	-	1.00
3=3 persons	0.320	3.70 (0.28, 48.81)
4=4+ persons	0.374	0.36 (0.04, 3.39)

[1] Reference categories were selected based on largest cell size
[2] Categories were collapsed for numerical stability

Table 98. Odds ratios and 95% CIs from the prescreening rejects group logistic regression model; nHypertens is predicted

Predictor variable	P-value	Odds Ratio (95% CI)
aDiabetes	0.468	2.10 (0.28, 15.61)
aJogging	0.251	0.31 (0.04, 2.31)
nMale	0.259	2.06 (0.59, 7.26)
[1]oPhysact		
1=Active	0.709	0.72 (0.13, 3.94)
2=Moderate	0.841	0.86 (0.19, 3.83)
3=Inactive	-	1.00
[1]oUsualact		
1=Usually sit	0.857	0.87 (0.19, 3.9)
2=Stand or walk quite a lot	-	1.00
3=Usually lift or carry light loads/Do heavy work or carry very heavy loads)	0.275	2.25 (0.52, 9.69)

[1] Reference categories were selected based on largest cell size

If the VWUO-MD weights helped to generate viable hypotheses, the upper scree plot group model ought to produce more or stronger associations than the lower scree plot group model, which in turn ought to produce more or stronger associations than the backwards elimination rejects group model, followed by the prescreening rejects group model. In this example, we find that this hierarchy is approximately satisfied according to a number of metrics (although not all monotonic). These are summarized in Table 99. Much, but not all the gains of VWUO-MD were attained in the prescreening and backwards elimination stages, after which the number of significant variables increased from 0 to 1. While the number of significant variables did not increase further between the lower scree plot group model and the upper scree plot group model, the overall statistical significance of the model (assessed with a likelihood ratio test (LRT) of global $\beta = \mathbf{0}$) was increased. Overall statistical significance increased from a p-value of 0.377 in the prescreening rejects group model to 0.027 in the backwards

elimination rejects group model, then stayed about constant at 0.029 in the lower scree plot group model (bear in mind that there was an extra independent variable in this model), and finally increased to 0.004 in the upper scree plot group model. In addition, model fit was improved between the lower scree plot group model and the upper scree plot group model, with the Hosmer-Lemeshow GoF test p-value for rejection increasing from 0.027 to 0.747. (The lower scree plot group model also had to be optimized by use of oAge instead of cAge, recall, before which the GoF p-value was <0.001). These results show a general increase in statistical significance as well as improved fit, as the VWUO-MD variable weights of those variables included in the model increase. This general pattern suggests that in this example, VWUO-MD has served as a useful tool for mining the JCUSH data set and producing viable, testable hypotheses.

Table 99. Comparing logistic regression models built on three different groups defined by VWUO-MD variable weights; all models have 5 independent variables

Metric	Upper scree plot group model	Lower scree plot group model	Backwards elimination rejects group model	Pre-screening rejects group model
No. of variables with p-value≤0.05	1	1	0	0
[1]LRT test of global $\beta = \mathbf{0}$ (p-value)	0.015	0.029	0.027	0.377
[1]Hosmer-Lemeshow GoF test (p-value)	0.747	0.027	0.047	0.551

[1] LRT test of overall statistical significance and Hosmer-Lemeshow GoF test are not bootstrap-adjusted

6.6 Summary

The analyses performed in this chapter may have revealed aspects of VWUO-MD that need additional study, such as the calibration of normalizing

multipliers. However, even in its present form, VWUO-MD produced viable hypotheses for future research from the JCUSH data. In this example, we discovered something new about the reduction in activities due to a long term health condition amongst working, mature students 50 years or older who received health care services in the past 12 months. Specifically, we learned that such limitations are strongly positively associated with consultations with a mental health professional in the past 12 months (odds ratio=12.25, 95% CI=1.67, 90.02). Our model was adjusted for the previously mentioned three variables, as well as country of residence (Canada vs. USA) and depression. Inclusion of other potentially confounding variables that could alter statistical significance is probably warranted (e.g., many models ought to be adjusted for age and gender). We did not perform that additional step in this example, in order to more clearly perform an analysis of the VWUO-MD method via a comparison of models built out of groups of variables differently ranked by their VWUO-MD variable weights. In actual practice, additional adjustment of models would usually need to be done. Eventually, the information obtained in an adjusted upper scree plot group model on a testing data set could either be published as-is, and/or be used to direct more specific research with other, larger data sets by interested researchers.

CHAPTER 7: DISCUSSION

In this thesis, we developed a data mining methodology for generating hypotheses based on the variable-weighted ultrametric optimization of mixed-type data. VWUO-MD supports the analysis of any mixture of continuous, ordinal, nominal (including binary symmetric) and binary asymmetric variables. The variable weights that are produced by VWUO-MD have been shown to be informative about which variables participate most strongly in the clustering within the data, as well as otherwise related variables.

We developed two covariance matrix estimators for the variable weight estimates, a bootstrap estimator, and one based on U-statistics. The bootstrap covariance matrix was consistently found to be a slightly conservative estimator under a broad variety of scenarios. Unfortunately, bootstrap replication increases the run time by a big factor, but at worst it is only a *linear* increase, and in reality the factor is not as high as the number of weights. This is because every bootstrap replicate sample contains many zero-weighted records, and those are dropped before analysis, so the replicate estimates are obtained much faster ("thanks" to the order n^3 algorithm) than the point estimates. Additionally, the VWUO.exe software easily allows one to perform replication on several computers or several processors within one computer simultaneously, further decreasing the burden of bootstrap replication. There is an advantage to the bootstrap approach: we can estimate variance even on complex survey samples

for which bootstrap weights have been produced. If the data are from an SRS with only variables of types C and N, the U-statistic-based estimator performs well with at least four variables. However, in low-dimensional data or mixtures of type O or A variables, it underestimates variance by factors in the low to mid single digits. For such situations, as well as any analyses of complex survey data, the bootstrap estimator should be used. Bootstrap percentile confidence intervals ought to be used over normal-based intervals at least in low-dimensional problems. Besides simple random sample data with at least four variables involving only types C and N, where the U-statistic-based estimator and normal-based confidence intervals ought to produce results that are extremely close to comparable percentile-based intervals, the U-statistic-based estimator may also be useful for preliminary analyses of other types of SRS data on which approximate, somewhat liberal variance estimates are sufficient. The U-statistic-based estimator can dramatically save computing time, by a factor on the order of several hundred to a thousand—the typical number of bootstrap weights.

The method developed in this thesis ought to be compared at least generally to other methods that do related things, for example, cluster analysis. VWUO-MD is not a cluster analysis method, however, it is built from an optimization that was originally intended for enhancement of clusters in data, with hierarchical clustering being a specific focus. As such, here we will briefly review some alternative cluster analysis methods and discuss their relevance to VWUO-MD. Model-based CA is a modern, statistical approach to CA involving maximizing a likelihood typically comprised of a product of MVN densities. This is

described in Fraley et al (2002).[80] Estimation of the mixture models is typically done with expectation-maximization, combined with a Bayesian prior imposed on the likelihood to help determine the number of clusters (via the Bayesian Information Criterion). There are limitations of model-based CA, which include situations with non-Gaussian data. However, Fraley describes how such data may be well approximated by multiple MVN clusters. For example, a linear relationship could be represented by several Gaussian clusters in series. Another limitation as described by Fraley is that large data sets are not easily handled in the expectation-maximization algorithm they describe, being order n^2 (although we note that this is a full order lower than VWUO-MD). They offer a solution involving cluster analyzing an initial random subsample followed by discriminant analysis to relate the classifications to the remaining sample. Model-based CA involves maximizing a (typically Gaussian) likelihood on continuous data. However, it is not difficult to envision adding multiplicands to the likelihood function with multinomial probability mass functions to accommodate mixed-type data. Besides the addition of multinomial probability mass functions, modification of model-based CA for the purpose of HG would probably involve variable weights. It is not immediately clear how a function of the likelihood could be designed so that the weights corresponded to relationships in the data as they purport to in VWUO-MD, but this would need to designed as well. Other CA methods are less promising for variable weights due to the subjectivity described in the introduction associated with linkage and stopping rules. For example, in k-means CA, some number of clusters k is decided on *a priori*, centroids (means)

are assigned for each cluster, and objects are assigned to the nearest centroid. In subsequent steps, centroids are recalculated and objects are reassigned, and the process continues until convergence. Several variable weighting methods have already been developed for use with this method, also described in the introduction. However, k-means CA has the subjectivity issues described in the introduction.

Besides CA, VWUO-MD could be compared to other machine learning techniques. Such methods include principal components and exploratory factor analysis.[12] These methods are designed to extract information from the sample correlation matrix as opposed to the raw data, and this is a double-edged sword. On the one hand, being designed to operate on correlation matrices ensures that linear relationships in continuous variables (one of the biggest existing holes in VWUO-MD) are easily detected. On the other hand, it also means that quadratic relationships, especially non-lopsided parabolas such as those analyzed in *Chapter 4: Additional exploratory analyses of artificial, clustered data*, are not easily detectable. There is a useful connection with VWUO-MD that we can draw from these ideas. Until VWUO-MD is "fixed" such that it can detect plain linear relationships in type C data, a VWUO-MD analysis could (and probably should) be augmented with a complementary, parallel analysis by something like principal components or exploratory factor analysis. The strengths of each approach would help to offset the other's weaknesses.

Another data mining method is multidimensional scaling (MDS).[81] The goal in MDS is to represent in as few dimensions as possible the distances

between N items, such that the approximate distances match the unscaled distances as closely as possible. With Euclidean distance, the goal is similar to principal components analysis. The method involves finding a *q*-dimensional configuration of the data such that the order and magnitude of the pairwise distances is preserved as closely as possible. Closeness is assessed with a stress function that is a scaled sum of squared deviations of two distances (or squared distances): one is assessed on the actual transformation, and the other are reference numbers designed to reflect the desired monotonicity. The stress function is minimized with respect to the *q*-dimensional distances. This is done for each value of *q*, and the stress function is plotted against *q* on a scree plot. From this the optimal value of *q* can be determined. MDS may reduce the effect of noisy variables, and cluster recovery, while comparable between unscaled and scaled data, is found to be several times faster in low-dimensional scaled data. The reduction of noise in MDS is a similar goal to that of De Soete, but less so with VWUO-MD, where we focus on the variable weights. Nevertheless, there may be a way in which MDS could serve as a preprocessing step for VWUO-MD. Generation of variable weights that are proportional to the dependence between variables in the data would require additional analyses, however, and there is also unfortunate subjectivity associated with selection of the monotonicity function in MDS.

Another technique relevant to data mining is bootstrap aggregating (also called "bagging").[82] Bagging is based on the combination of models fit to bootstrap samples taken from the training segment of the data set. In bagging,

the goal is reduce the error in prediction and classification models. The algorithm is performed in each bootstrap sample, and the result is taken to equal the mean in regression problems, or by majority in classification problems. There are two ways we might relate this approach to VWUO-MD. The most direct idea is that actual bagging could theoretically be applied to the variable weights. However, we ought to mention that the variance of the variable weights is proportional to the variance of the gradient (which is essentially a U-statistic), and previous research has suggested that bagging is not generally beneficial in reducing the error of U-statistics.[82,83] It still may be worth investigating, however. A less direct analogy to bagging that would be useful for VWUO-MD is the idea of splitting a training segment into several smaller pieces and aggregating the results from analyzing those. With the order n^3 algorithm, this could produce an important time savings, allow larger training segments to be analyzed, and therefore improve precision. There are other machine learning methods, and some are "master methods", in that they are algorithms for utilizing other algorithms. For example, "boosting" involves incrementally applying machine learning algorithms to a data set, each step weighting most heavily previously misclassified objects. This method assumes that the training data set contains "true" classifications on which to judge misclassification.[84] Another master model is "stacking", which involves a decision based on the results from a set of other machine learning algorithms.[85] Like boosting, this approach also assumes that the training data set contains "true" classifications on which to judge misclassification. For this reason,

these master models are not useful in the context of problems for which VWUO-MD would be applied.

There are many aspects of VWUO-MD that could benefit from additional research:

1. The improved penalty in the denominator of L_U was meant to preclude the possibility of 0 weights. However, while a variable weight solution of exactly 0 is not possible, in this thesis we have not considered the possibility of solutions diverging *towards* 0 (i.e., the numerator going towards 0 faster than the denominator). Unfortunately, we have encountered some trivial situations (e.g., the two-variable type N and type S examples in *Chapter 4: Additional exploratory analyses of artificial, clustered data*) where this happened. Fortunately, no practical application of VWUO-MD ought to involve only two variables, but nevertheless, this topic deserves more research.

2. The normalizing multipliers were obtained on a calibration data set, with the intention that they would provide a fair comparison between variables of different types. However, the analysis of the JCUSH data revealed the possibility that type A variables are too highly weighted on average, all else being equal. The “all else being equal” clause is actually hard to test because of the very different forms the distance formulas assume depending on variable type. Our approach was to construct a calibration data set with two grouping variables and two noise variables of each type (16 variables in total) and adjust the normalizing multipliers in an iterative procedure until each type’s four variables had an average weight of exactly 1. This approach is sensitive to the definition of the

variables in the calibration data set, and developing a true one-size-fits-all calibration data set is probably a complicated matter, if it is even possible. A potential solution that is available to the VWUO-MD analyst is to calibrate their own normalizing multipliers on artificial data that mimic the extraneous characteristics of their target data set, e.g., the number of categories of type O and type N variables, and the probability distributions of the grouping (if any) and noise variables. The danger with that approach is that it may become a self-fulfilling prophecy if one creates "too good" a representation of the target data, and where to draw the line at "extraneous" would not be a simple matter. For now, even using the normalizing multipliers developed in this thesis, VWUO-MD appears to be entirely informative within types, and mostly informative across types in various multi-type (particularly two-type) scenarios.

3. VWUO-MD could benefit from additional research on algorithm speed. VWUO-MD compares the three distances between objects in every triple in the data, an order n^3 method. This limits the size of the training data set that one may analyze with VWUO-MD to realistically not much more than n=100. The number of variables also places a burden on the software, due to the memory requirements of the matrix calculations performed in the estimation of $\hat{\mathbf{w}}$. It may not be logically possible to decrease the order from n^3 without fundamentally changing the approach. However, additional optimization of the software may be beneficial. In addition, an algorithm similar to bagging as discussed above could be used to aggregate results obtained on smaller pieces of a larger training

segment, both improving run time and precision simultaneously. Such algorithms should be explored.

4. Solutions to shortcomings (such as VWUO-MD's difficulty with simple linear relationships in type C data) should be sought. Until such time as the problem with linear relationships in type C data is solved, VWUO-MD ought to be used in parallel with a complementary method that is well suited to finding linear relationships in type C data, e.g., principal components or factor analysis. VWUO-MD could complement either approach.

5. Finally, how the relationships between variables, as well as the number, shape and placement of clusters within the data affect $\hat{\mathbf{w}}$ should be more fully explored. In *Chapter 5: Exploratory analyses of distributions for hypothesis generation*, we also considered examples with disjoint relationships that depend on which subset of variables one focuses on, and found that for types C, O, N and A, VWUO-MD compromises, spreading the weight between those variables that participate in *some* clustering. Type C estimates held up very well, while types O, N and A had a little more trouble, considering the percentage of replicates for which the noise variable was not weighted the lowest. Type S estimates were quite adversely affected by disjoint relationships however, in the example we considered, and the noise variable was weighted the highest on average. One area for improvement in VWUO-MD would be its ability to handle such scenarios, particularly with type S data. Going beyond this, it has been suggested by some that it would also be informative to know which variables are involved in different groupings than others.[39,43,47,59] This can be accomplished

with multiple parallel analyses of VWUO-MD using different groups of input variables, and the results of the type S example illustrate that this is not a bad idea.

There are probably other aspects of VWUO-MD that require additional research. But even so, VWUO-MD has already shown promise as a data mining tool for generating new hypotheses that potentially involve varied mixtures of data types. Should more researchers choose to use this approach and/or improve upon it, it will be exciting to witness all the new and previously unthought-of hypotheses defined on subspaces of mixed-type variables, wherever that may lead us.

REFERENCES

1. Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques (Second Ed.). Morgan Kaufmann Publishers. San Francisco, CA, USA. 2005. p. 5.
2. Gilman EA, Knox EG. Childhood cancers: space-time distribution in Britain. Journal of Epidemiology and Community Health. 1995;49:158-163.
3. Pfizer's Data Aggregation Cross Functional Team. Data Analytic Principles. June 15, 2007.
4. Theodoridis S, Koutroumbas K. Pattern recognition (Third Ed.). Academic Press. Burlington, MA, USA. 2006.
5. Bacardit J, Stout M, Hirst JD, Valencia A, Smith RE, Krasnogor N. Automated alphabet reduction for protein datasets. BMC Bioinformatics. 2009 Jan 6;10(1):6.
6. Huang da W, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc. 2009;4(1):44-57.
7. Kerstens HH, Kollers S, Kommadath A, Del Rosario M, Dibbits B, Kinders SM, Crooijmans RP, Groenen MA. Mining for Single Nucleotide Polymorphisms in Pig genome sequence data. BMC Genomics. 2009 Jan 6;10(1):4.
8. Moore M, Chan E, Lurie N, Schaefer AG, Varda DM, Zambrano JA. Strategies to improve global influenza surveillance: a decision tool for policymakers. BMC Public Health. 2008 May 28;8:186.
9. Bredel M, Bredel C, Sikic BI. Genomics-based hypothesis generation: a novel approach to unravelling drug resistance in brain tumours? Lancet Oncology. 2004 Feb;5(2):89-100.
10. Jacquez, GM. Spatial Cluster Analysis. In Fotheringham S, Wilson J (Eds.), The Handbook of Geographic Information Science. Blackwell Publishing, Edinburgh, UK. 2008. pp. 395-416.
11. Stegmann J, Grohmann G. Hypothesis generation guided by co-word clustering. Scientometrics. 2003;56(1):111-35.
12. Johnson RA, Wichern DW. Applied Multivariate Statistical Analysis (Fifth Ed.). Prentice Hall, Inc. Upper Saddle River, NJ, USA. 2002.
13. Kwek S. Cluster Analysis. Presented at the Human Genome Laboratory in the Department of Computer Science at the University of Texas at San Antonio. 2005.
14. Hogeweg P. Iterative character weighing in numerical taxonomy. Computers in Biology and Medicine. 1976;6(3):199-223.
15. Art D, Gnanadesikan R, Kettenring JR. Data-Based Metrics for Cluster Analysis. Utilitas Mathematica. 1982;21A:75-99.

16. DeSarbo WS, Carroll JD, Clark LA, Green PE. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. Psychometrika. 1984 March;49(1):57-78.
17. De Soete G, DeSarbo WS, Carroll JD. Optimal variable weighting for hierarchical clustering: An alternating least-squares algorithm. Journal of Classification. 1985 Dec;2(1):173-92.
18. De Soete G. Optimal variable weighting for ultrametric and additive tree clustering. Journal Quality and Quantity. 1986 June. 20(2-3):169-80.
19. Arabie P, Hubert LJ. Combinatorial data analysis. Annual Review of Psychology. 1992;43(1):169-203.
20. Arabie P, Hubert LJ. Clustering from the Perspective of Combinatorial Data Analysis. In Krzanowski WJ (Ed.), Recent advances in descriptive multivariate analysis. Oxford University Press. Oxford, UK. 1995. pp. 1-13.
21. Breckenridge JN. Validating Cluster Analysis: Consistent Replication and Symmetry. Multivariate Behavioral Research. 2000;35(2):261-85.
22. Brusco MJ, Cradit JD. A variable-selection heuristic for K-means clustering. Psychometrika. 2001 June;66(2):249-70.
23. Brusco MJ. Clustering Binary Data in the Presence of Masking Variables. Psychological Methods. 2004 Dec;9(4):510-23.
24. Bull JK, Basford KE, DeLacy IH, Cooper M. Classifying genotypic data from plant breeding trials: a preliminary investigation using repeated checks. Theoretical and Applied Genetics. 1992 Dec;85(4):461-9.
25. Carmone FJ Jr., Kara A, Maxwell S. HINoV: A New Model to Improve Market Segment Definition by Identifying Noisy Variables. Journal of Marketing Research. 1999 Nov;36(4):501-9.
26. Chun J. Computer Assisted Classification and Identification of Actinomycetes. University of Newcastle, Department of Microbiology, Doctor of Philosophy thesis, 1995.
27. Chung J, Choi I. A Non-parametric Method for Data Clustering with Optimal Variable Weighting. In Corchado E, Yin H, Botti V, Fyfe C (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2006. Springer. Berlin/Heidelberg, Germany. 2006. pp. 807-14.
28. Corter JE. Tree Models of Similarity and Association (Quantitative Applications in the Social Sciences). Sage Publications, Inc. California, USA. 1996.
29. Debska B, Guzowska-Swider B. Analysis of the relationship between the structure and aromatic properties of chemical compounds. Analytical and Bioanalytical Chemistry. 2003 Apr;375(8):1049-61.
30. DeSarbo WS, Cron WL. A maximum likelihood methodology for clusterwise linear regression. Journal of Classification. 1988 Sep;5(2):249-82.
31. DeSarbo WS, Oliver RL, Rangaswamy A. A simulated annealing methodology for clusterwise linear regression. Psychometrika. 1989 Sep;54(4):707-36.

32. De Soete G, Carroll JD, DeSarbo WS. Least squares algorithms for constructing constrained ultrametric and additive tree representations of symmetric proximity data. Journal of Classification. 1987 Sep;4(2):155-73.
33. De Soete G. OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting. Journal of Classification. 1988 March;5(1):101-4.
34. Donoghue, JR. The Effects of Within-group Covariance Structure on Recovery in Cluster Analysis: I. The Bivariate Case. Multivariate Behavioral Research. 1995;30(2):227-54.
35. Donoghue JR. Univariate Screening Measures for Cluster Analysis. Multivariate Behavioral Research. 1995 July;30(3):385-427.
36. Everitt BS, Landau S, Leese M. Cluster Analysis (Fourth Ed.). Oxford University Press, Inc. New York, NY, USA. 2001.
37. Fovell RG, Fovell MC. Climate Zones of the Conterminous United States Defined Using Cluster Analysis. Journal of Climate. 1993;6(11):2103-35.
38. Fowlkes EB, Gnanadesikan R, Kettenring JR. Variable selection in clustering. Journal of Classification. 1988;5(2):205-28.
39. Friedman JH, Meulman JJ. Clustering Objects on Subsets of Attributes. Journal of the Royal Statistical Society. Series B (Statistical Methodology). 2004; 66(4):815-49.
40. Gnanadesikan R, Kettenring JR, Tsao SL. Weighting and selection of variables for cluster analysis. Journal of Classification. 1995;12(1):113-36.
41. Gordon AD. Constructing dissimilarity measures. Journal of Classification. 1990 Sep;7(2):257-69.
42. Green PE, Kim J, Carmone FJ. A preliminary study of optimal variable weighting in k-means clustering. Journal of Classification. 1990 Sept;7(2):271-85.
43. Hand DJ, Heard NA. Finding groups in gene expression data. Journal of Biomedicine and Biotechnology. 2005 Jun;2:215-25.
44. Huang JZ, Ng MK, Rong H, Li Z. Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005 May;27(5):657-68.
45. Huang JZ, Xu J, Ng M, Ye Y. Weighting Method for Feature Selection in K-Means. In Liu H, Motoda H (Eds.), Computational Methods of Feature Selection. Chemical Rubber Company Press. New York, NY. 2007. pp. 193-210.
46. Jedidi K, DeSarbo WS. A stochastic multidimensional scaling procedure for the spatial representation of three-mode, three-way pick any/J data. Psychometrika. 1991 Sep;56(3):471-94.
47. Jing L, Ng MK, Huang JZ. An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. IEEE Transactions on Knowledge and Data Engineering. 2007 June;19(8):1026-41.
48. Lapointe FJ, Legendre P. A Statistical Framework to Test the Consensus Among Additive Trees (Cladograms). Systematic Biology. 1992 Jun;41(2):158-71.

49. Leonard S, Droege M. The uses and benefits of cluster analysis in pharmacy research (Research in Social and Administrative Pharmacy). 2008 Mar;4(1):1-11.
50. Makarenkov V, Legendre P. Optimal Variable Weighting for Ultrametric and Additive Trees and K -means Partitioning: Methods and Software. Journal of Classification. 2001 Feb 1;18(2):245-71.
51. Meulman JJ, Verboon P. Points-of-view analysis revisited - fitting multidimensional structures to optimal distance components with cluster restrictions on the variables. Psychometrika. 1993 Mar;58(1):7-35.
52. Milligan GW, Cooper MC. Methodology Review: Clustering Methods. Applied Psychological Measurement. 1987;11(4):329-54.
53. Milligan GW, Cooper MC. A study of standardization of variables in cluster analysis. Journal of Classification. 1988 Sep;5(2):181-204.
54. Milligan GW. A validation study of a variable weighting algorithm for cluster analysis. Journal of Classification. 1989 Dec;6(1):53-71.
55. Milligan GW. Clustering Validation: Results and Implications for Applied Analyses. In Arabie P, Hubert LJ, De Soete G (Eds.), Clustering and Classification. World Scientific Publ. River Edge, NJ. 1996. pp. 341-76.
56. Milligan GW, Hirtle SC. Clustering and Classification Methods. In Weiner IB, Freedheim DK, Schinka JA (Eds.), Handbook of Psychology. John Wiley and Sons. 2003. pp. 165-86.
57. Morris L, Schmolze R. Consumer archetypes: A new approach to developing consumer understanding frameworks. Journal of Advertising Research. 2006 Sep;46(3):289-300.
58. Schweinberger M, Snijders TAB. Settings in Social Networks: A Measurement Model. Sociological Methodology. 2003;33:307-41.
59. Soffritti G. Identifying multiple cluster structures in a data matrix. Communications in Statistics-Simulation and Computation. 2003;32(4):1151-77.
60. Sokal RR. Phenetic taxonomy: Theory and methods. Annual Review of Ecology & Systematics. 1986;17:423-42.
61. Steinley D, Henson R. OCLUS: An Analytic Method for Generating Clusters with Known Overlap. Journal of Classification. 2005 Sept;22(2):221-50.
62. Steinley D. K-means clustering: A half-century synthesis. British journal of mathematical & statistical psychology. 2006;59(1):1-34.
63. Steinley D, Brusco MJ. Selection of variables in cluster analysis: An empirical comparison of eight procedures. Psychometrika. 2008 Mar;73(1):125-144.
64. Taylor & Francis, Ltd. for the Society of Systematic Biologists. 21st Numerical Taxonomy Conference. Systematic Zoology. 1988 Mar;37(1):91-3.
65. Tsai CY, Chiu CC. Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. Computational Statistics & Data Analysis. 2008 Jun;52(10):4658-72.

66. van Buuren S, Heiser WJ. Clustering n objects into k groups under optimal scaling of variables. Psychometrika. 1989 Sep;54(4):699-706.
67. Cochran WG. Sampling Techniques (Third Ed.). John Wiley & Sons, Inc. New York, NY, USA. 1977.
68. Casella G, Berger R. Statistical Inference (Second Ed.). Duxbury. Pacific Grove, CA, USA. 2002. pp. 240-44.
69. Serfling RJ. Approximation Theorems of Mathematical Statistics. John Wiley & Sons, Inc. New York, NY, USA. 1980.
70. Thomas PY, Sreekumar NV. Estimation of location and scale parameters of a distribution by U-statistics based on best linear functions of order statistics. Journal of Statistical Planning and Inference. 2008 Jul;138(7):2190-2200.
71. Lee AJ. U-statistics: theory and practice. Chemical Rubber Company Press. New York, NY. 1990.
72. Efron B, Gong G. A Leisurely look at the bootstrap, the jackknife, and cross-validation. The American Statistician. 1983 Feb;37(1):36-48.
73. SAS Institute Inc., Cary, NC, USA.
74. Henze N, Zirkler B. A class of invariant consistent tests for multivariate normality. Comm. Statist. Theory Methods. 1990;A19:3595-3617.
75. Yeo D, Mantel H, Liu T. Bootstrap variance estimation for the National Population Health Survey. Proceedings of the Survey Research Methods Section, ASA. 1999;778-83.
76. Vardeman S. Bootstrap percentile confidence intervals. [Available at http://www.public.iastate.edu/~vardeman/stat511/BootstrapPercentile.pdf]
77. The Joint Canada/United States Survey of Health. [Available at http://www.cdc.gov/nchs/about/major/nhis/jcush_mainpage.htm]
78. Hosmer DW, Lemeshow S. Applied Logistic Regression (Second Ed.). John Wiley & Sons, Inc. USA. 2000.
79. Roberts G, Binder D, Kovacevic M, Pantel M, Phillips O. Using an estimating function bootstrap approach for obtaining variance estimates when modeling complex health survey data. Proceedings of the Survey Methods Section, SSC Annual Meeting. June, 2003.
80. Fraley C, Raftery AE. Model-Based Clustering, Discriminant Analysis, and Density Estimation. Journal of the American Statistical Association. 2002;97:611-31.
81. Tzeng J, Lu HHS, Li WH. Multidimensional scaling for large genomic data sets. BMC Bioinformatics. 2008;9:179.
82. Pino-Mejias R, Jimenez-Gamero MD, Cubiles-de-la-Vega MD, Pascual-Acosta A. Reduced bootstrap aggregating of learning algorithms. Pattern recognition letters. 2008;29(3):265-271.
83. Buja A, Stuetzle W. The effect of bagging on variance, bias, and mean squared error. 2000. AT& T Labs-Research.
84. Shen SH, Liu YC. Efficient multiple faces tracking based on Relevance Vector Machine and Boosting learning. Journal of Visual Communication and Image Representation. 2008;19(6):382-91.

85. Ting KM, Witten IH. Issues in stacked generalization. Journal of Artificial Intelligence Research. 1999;10:271-89.