NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

VISUALIZING MIXED VARIABLE-TYPE MULTIDIMENSIONAL DATA USING TREE DISTANCES

by

Yoav Shaham

September 2015

Thesis Advisor:	Lyn R. Whitaker
Second Reader:	Samuel E. Buttrey

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704–0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2015	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE VISUALIZING MIXED VARIABLE-TYPE MULTIDIMENSIONAL DATA USING TREE DISTANCES			**5. FUNDING NUMBERS**
6. AUTHOR(S) Shaham, Yoav			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This research explores the use of the tree distances of Buttrey and Whitaker to visualize multidimensional data of mixed-variable types, having both numerical and categorical data. Tree distances measure dissimilarities among observations in a data set while exploiting desirable properties of classification and regression trees: ease of handling of most variable types, indifference to variable scaling, resistance to noise and outliers, accommodations for missing values, and computational ease. In this research, we map the dissimilarities using Classical Multidimensional Scaling to a lower-dimensional Euclidean space in order to provide an analyst with a comfortable framework, which supplies visual cues in order to help find patterns and gain insights about the data. We offer in this thesis several algorithms for coloring observations in the lower-dimensional mappings in order to focus the analyst's attention on the most important and interesting relationships in the data set. In addition, through our visualization, we gain a deeper understanding of the properties of tree distances and propose a modification. Our framework can be used on any military data set that involves mixed or non-mixed variables and is valuable for analysts who wish to shed light on data during the exploratory phase of analysis.

14. SUBJECT TERMS Visualization, tree distances, mixed data, 2-D,3-D, insights, distances, dissimilarities, categorical, numerical			**15. NUMBER OF PAGES** 125
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

VISUALIZING MIXED VARIABLE-TYPE MULTIDIMENSIONAL DATA USING TREE DISTANCES

Yoav Shaham
Captain, Israel Defence Forces
B.Sc., The Hebrew University of Jerusalem, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Approved by: Lyn R. Whitaker
Thesis Advisor

Samuel E. Buttrey
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This research explores the use of the tree distances of Buttrey and Whitaker to visualize multidimensional data of mixed-variable types, having both numerical and categorical data. Tree distances measure dissimilarities among observations in a data set while exploiting desirable properties of classification and regression trees: ease of handling of most variable types, indifference to variable scaling, resistance to noise and outliers, accommodations for missing values, and computational ease. In this research, we map the dissimilarities using Classical Multidimensional Scaling to a lower-dimensional Euclidean space in order to provide an analyst with a comfortable framework, which supplies visual cues in order to help find patterns and gain insights about the data. We offer in this thesis several algorithms for coloring observations in the lower-dimensional mappings in order to focus the analyst's attention on the most important and interesting relationships in the data set. In addition, through our visualization, we gain a deeper understanding of the properties of tree distances and propose a modification. Our framework can be used on any military data set that involves mixed or non-mixed variables and is valuable for analysts who wish to shed light on data during the exploratory phase of analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. TREE DISTANCE VISUALIZATION—IRIS EXAMPLE 2
B. RELATED WORK 5
1. Dissimilarities Calculation Methods for Mixed Data Type 5
2. Mapping techniques 7
3. Visualization Techniques for Mixed Data Sets 8
C. THESIS OUTLINE 8

II. BACKGROUND 11
A. TREE DISTANCES 11
B. CLASSICAL MULTIDIMENSIONAL SCALING 14
C. THE TREE DISTANCE VISUALIZATION PROCESS 15
D. THE DATA SETS 17
E. SUMMARY 19

III. COLORING THE TREE DISTANCES MAPPINGS FOR INSIGHTS 21
A. COLORING SPLICE BY ITS CLASS AND THE V35 VARIABLE 22
B. CHOOSING VARIABLES FOR COLORING 24
1. The Regression Method 25
2. Maximum Deviance Ratio Method 27
3. The Purity Method 29
C. CHOOSING VALUES FOR COLORING (PRUNING METHOD) 36
D. CONCLUSIONS 39

IV. STRONG DEPENDENCE AMONG VARIABLES 41
A. THE ISSUE OF STRONG DEPENDENCE 41
.1 Theoretical Analysis 41
2. Case Study: Splice Data Set 43
a. Splice Data Mapped Using d1 and d4 43
b. Splice Data With Constructed Dependence 46
3. Discussion 49
4. How Does the Deviance Change as a Factor of the Amount of Correlation? 50
B. A PROPOSED SOLUTION 51

1. Advantages and Disadvantages 52
2. Proposed Solution Experiments 53
3. Credit Data Set Experiment 54
4. Artificial Splice Data Set Experiment 60
C. SUMMARY 63

V. DISCUSSION 65
A. CLUSTER PROPERTIES OF THE TREE DISTANCE ALGORITHM 65
1. Observations' Collapsing Tendency 67
2. Nearly-Equal Distances 72
3. "Snake" Shape Mapping 76
B. THE VARIANTS' INFLUENCE ON THE MAPPING 79
1. Splice Mappings Using the Different Variants 80
2. Similar Mapping for Different Variants of the Tree Distances 92

VI. CONCLUSIONS 95

LIST OF REFERENCES 97

INITIAL DISTRIBUTION LIST 101

LIST OF FIGURES

Figure 1. Iris data set mapping using d1 colored by the iris type 3

Figure 2. Iris data set mapping using d1 colored by the petal length 3

Figure 3. Iris data set mapping using d1 colored by the sepal length..... 4

Figure 4. Iris data mapping using d1 colored by the Iris class; without noise 17

Figure 5. Iris data mapping colored by the Iris class; with noise 17

Figure 6. Splice data mapping using d4; colored by Splice Class 22

Figure 7. Splice data mapping using d4; colored by levels of V35 23

Figure 8. Iris data map by d1 color-coded by petal length 26

Figure 9. Seeds data set mapped using d1; colored by V1..... 28

Figure 10. Seeds data set mapped using d1; colored by V6..... 28

Figure 11. Bar plot of the percent of boxes with purity above 0.9 per variable in the Splice data mapped using d1 31

Figure 12. Splice data map by d1 color-coded by V35..... 32

Figure 13. Purity histogram of V35 in Splice mapping using d1 33

Figure 14. Splice data mapping using d1 color coded by V32 34

Figure 15. Purity histogram - V32 in Splice mapping using d1..... 34

Figure 16. Splice data mapping using d1 color coded by V61 35

Figure 17. Purity histogram – V61 in Splice mapping using d1..... 36

Figure 18. Reduction in R^2 analog (deviance ratio-DevRat) per variable for the Splice data 43

Figure 19. Splice data mapped using d1 colored by Splice class levels 44

Figure 20. Splice data mapped using d4 colored by Splice class levels 44

Figure 21. Splice data mapped by d1 colored by V4..... 45

Figure 22. Splice data mapped by d4 colored by V4..... 45

Figure 23. Reduction in deviance ratio per variable for the Splice data with additional V0 variable, a constructed dependence variable to V4 46

Figure 24. The V4 associated tree..... 47

Figure 25. Splice with additional correlated variable V0 mapping based on d1 colored by Splice class levels 48

Figure 26. Splice with additional correlated variable V0 mapped with d4 colored by Splice class levels 48
Figure 27. Splice with additional correlated variable V0 mapping based on d4 colored by V4 values 49
Figure 28. Reduction in R^2 analogy (Deviance Reduction Dev-Rat) for V4 as a function of the percent of permutation 51
Figure 29. R^2 analog (DevRat) per variable of Credit data set using the current tree distance algorithm 54
Figure 30. Mapping Credit using d1 of the current solution, colored by Credit class 54
Figure 31. Mapping Credit using d1 of the current solution colored by V5 55
Figure 32. Mapping Credit using d1 of the current solution colored by V10 56
Figure 33. R^2 analog (DevRat) per variable of Credit data using the proposed algorithm 57
Figure 34. Mapping Credit using d1 of the proposed solution colored by Credit class 58
Figure 35. Mapping Credit using d1 of the proposed solution colored by V5 59
Figure 36. Mapping Credit using d1 of the proposed solution colored by V10 60
Figure 37. R^2 analog (DevRat) per variable of Artificial Splice Data using the proposed algorithm 61
Figure 38. Mapping Artificial Splice data using d4 of the proposed solution, colored by Splice class 62
Figure 39. Mapping Artificial Splice data using d4 of the proposed solution, colored by V4 62
Figure 40. The artificial data set colored by the type 66
Figure 41. The trees created to compute tree distances algorithm on the artificial data set 68
Figure 42. Artificial data set mapping using d1 69
Figure 43. Iris data set, colored by the Iris class. 70
Figure 44. Iris data mapping using d1 71
Figure 45. The new artificial data set consisting of an additional green cluster 73
Figure 46. The trees created for tree distances on the new artificial data set; each leaf is colored by the corresponding cluster 74

Figure 47. The new artificial data set mapping using d1 75
Figure 48. Seeds data set mapping using d1 colored by (a) Seed class, (b) V1, and (c) V2. 78
Figure 49. The monotonic increasing dependence between V1 and V2 in the Seeds data set 79
Figure 50. Splice data set mapping using d1 colored by Splice class 81
Figure 51. Splice data set mapping using d1 colored by the PAM algorithm clusters results 82
Figure 52. Splice data set mapping using d2 colored by Splice class 83
Figure 53. Splice data set mapping using d3 colored by Splice class 85
Figure 54. Splice data set mapping using d3 colored by V35 86
Figure 55. Splice data set mapping using d3 colored by the PAM algorithm clusters results 87
Figure 56. Splice data set mapping using d4 colored by Splice class 89
Figure 57. Splice data set mapping using d4 colored by V35 90
Figure 58. Splice data set mapping using d4 colored by V32 91
Figure 59. Splice data set mapping using d4 colored by PAM clustering algorithm's results 92
Figure 60. Seeds data set mapping using the four variants 94

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. The relationship between the “T” level for variable V35 and the “EI” class for Splice class 24

Table 2. Deviance ratio deduction per variable; Seeds data set 27

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CMDS	Classical Multidimensional Scaling
MCA	Multiple Correspondence Analysis
PAM	Partitioning Around Medoids
SNE	Stochastic Neighbor Embedding

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Visualization is a key tool for an analyst who wishes to explore data sets. Large number of military data sets are of mixed-type data. They contain more than one type of variables, numerical, ordinal or categorical. Johansson et al. (2008) mention that there is no agreed similarity measurement for mixed-type data and therefore no generalize framework for visualization of this type of data. In this thesis, we propose a novel visualization technique by the name “tree distance visualization” for mixed-type data based on the tree distances developed by Buttrey (2006) and expanded by Buttrey and Whitaker (2015a, 2015b). The tree distances measure the dissimilarities of a mixed-type data using trees. They calculate the dissimilarities while exploiting the relationships between the different variables. They have several advantages: ease of handling of most variable types, indifference to variable scaling, resistance to noise and outliers and accommodations for missing values (Buttrey and Whitaker 2015a).

Tree distance visualization is based on two major components: the tree distances and Classical Multidimensional Scaling, or CMDS (Gower, 1966). The algorithm calculates the dissimilarities among the observations in the data using one of the four variants of the tree distances (Buttrey and Whitaker 2015a). Then the algorithm maps them into lower-Euclidean space using CMDS while minimizing the stress (Kruskal and Wish 1978), which is a measure of the difference between the original dissimilarities and the distances among observations in the new space.

Coloring the mappings of the tree distance visualization is an essential tool for providing an analyst visual cues which helps create insights about the data. We provide in this thesis three algorithms that we developed in order to automate the coloring task. They include the maximum deviance ratio that uses the deviance reduction assessment for the quality of the trees created to compute the dissimilarities, in order to identify the variables to color the mapping by. The second is the purity method that finds categorical variables to color the

mapping by using the spatial properties of the mapping with respect to the variables' values. The third method we developed is the pruning method that exploits the structure of the trees created in order to compute the dissimilarities. The method assigns colors for ranges of values for a specific variable in interest.

We continue the thesis by explaining the strong dependence issue, which we discovered while using the tree distance visualization. The tree distances for a data set that contain variables with strong dependence among them will have biased distances if the strong dependence is a constructed dependence. Examples for constructed dependence include a variable measured more than once in different units, summary variables, and a monotonic function of a variable. We suggest a modification to the tree distances that identify strong dependence variables using the trees created to compute the tree distances. Our modification is implemented in the tree distances R package ("treeClust," Buttrey 2015).

The end of the thesis contains a discussion about some of the properties of the tree distance visualization. We consider the tendency of observations to collapse into several distinct points in the mapping. We discuss the equal distance property for different clusters in the tree distance mapping having an equal distance among them, regardless of the structure of the original data set. We consider a special shape that appears in several numeric data sets' mappings: the "snake" shape. We finish the thesis by providing a visual representation of the differences among the tree distances variants (Buttrey and Whitaker 2015a).

List of references

Buttrey, Samuel E. 2006. A Scale-Independent Clustering Method with Automatic Variable Selection Based on Trees. Presented at the Joint Statistical Meetings, Seattle, WA.

Buttrey, Samuel E. 2015. treeClust: Create a measure of inter-point dissimilarity useful for clustering mixed data, and, optionally, perform clustering. R package version 1.1-1.

Buttrey, Samuel E., and Lyn. R. Whitaker. 2015a. “A Scale-Independent, Noise-Resistant Dissimilarity for Tree-Based Clustering of Mixed Data.” (submitted), Naval Postgraduate School, Monterey, CA.

Buttrey, S. E. and Lyn R. Whitaker. 2015b. “treeClust: An R Package for Tree-Based Clustering Dissimilarities.” (To appear in *The R Journal.*)

Gower, J. C. 1966. “Some Distance Properties of Latent Root and Vector Methods used in Multivariate Analysis.” *Biometrika* 53: 325–338.

Johansson, Sara, Mikael Jern, and Jimmy Johansson. 2008. “Interactive Quantification of Categorical Variables in Mixed Data Sets.” In *12th International Conference on Information Visualisation*, 3–10. Los Alamitos, CA: IEEE Computer Science Press.

Kruskal, Joseph B., and Myron Wish. 1978. *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Whitaker and Professor Buttrey for their guidance throughout the thesis. It was an enjoyable journey, and I would like to thank them for the invitation to join it, and their enthusiastic guidance and leadership while I worked on the thesis.

I would also like to thank my parents who encouraged me to pursue my dreams, and to my commanders at the IDF who believed in me and provided me the opportunity to extend my education.

I would like to give a special thanks to Sivan, who supported me through this journey. I am glad we had this time together at NPS.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Gaining insights about and from data is one of the major tasks of a data analyst. The world is flooded with information in a larger volume and with more variety and complexity than ever before. In the military domain, large numbers of data sets are of mixed data types, having both numerical and categorical variables. Examples of these data sets include manpower data, which can include numerical variables such as age and time in service, and categorical variables such as service, military occupational specialty, and pay grade.

Over the years, many techniques have been developed to analyze and find patterns in multidimensional data. These techniques can be grouped into tasks such as classification, regression, clustering, visualization, and more. Visualization of data exploits the human ability to recognize patterns using the sense of sight. An analyst can gain insights about patterns such as clusters, trends, outliers, and more by observing a mapping of multidimensional data in lower-dimensional space. Visualization of mixed data types is a nontrivial task. There are techniques for visualization of categorical data sets (e.g. Meyer, Zelies, and Hornik 2006) and also for numerical data sets such as parallel coordinate plots (Inselberg and Dimsdale 1990) and projection pursuit (Friedman and Tukey 1974), but only a few for mixed data because it is more difficult to combine categorical and numerical variables (for examples see Johansson 2008).

In this thesis, we suggest a new technique for the visualization of mixed data. The method combines the tree distances of Buttrey (2006) and Buttrey and Whitaker (2015a) implemented in the R statistical Environment (R Core Team 2013) by the package treeClust (Buttrey 2015) and Classical Multidimensional Scaling, or CMDS of Gower (1966), to visualize mixed data in a lower-dimensional mapping in order to supply an analyst with visual tools for gaining insights. Tree distance visualization can be used for any type of data—numerical, categorical, or mixed data types. The visualization is

resistant to noise variables in the data and outliers and indifferent to linear transformations of the data.

This chapter is organized as follows: Section A contains an example of the tree distance visualization technique using the Iris data set. Section B covers related works for distance measurement and visualization of mixed data types. Section C outlines the structure of the rest of the thesis.

A. TREE DISTANCE VISUALIZATION—IRIS EXAMPLE

In this section, we demonstrate some of the properties of tree distance visualization. We provide in this section only sufficient details about the process in order to display the main ideas of the thesis. In later chapters, we provide a much more in-depth description of the data and the process. In this example, we are using the well-known Iris data set (for details, see Chapter II). The Iris data set contains three types of irises and has four numerical variables describing different properties of them. We demonstrate in this section some of the insights that can be gained by using the tree distance visualization technique.

Figure 1 shows the Iris data set mapped into a two-dimensional Euclidean space based on the tree distances variant, d1 (for details, see Chapter II). We label the axes of mapped Euclidean space by “a1,” “a2,” and “a3” if needed so as not to confuse these axes with the variables in the data set. Figures 2 and 3 show the same mapping colored by two variables of the Iris data set, the sepal and petal lengths. The process by which we select the most promising variables to color by and how to color those variables are described in Chapter IV.

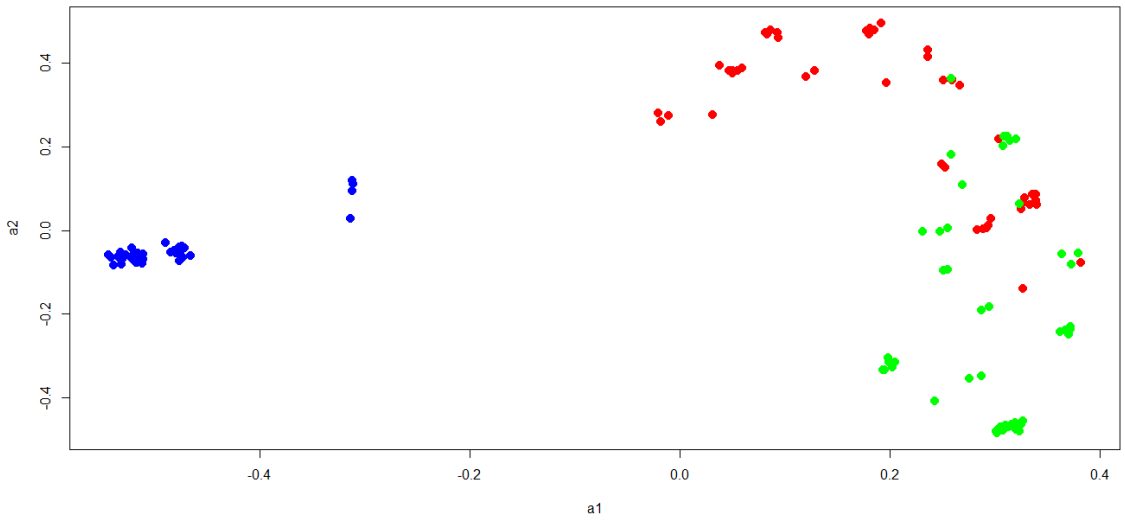


Figure 1. Iris data set mapping using d1 colored by the iris type

Legend: Setosa – blue, Versicolor – red, and Virginica - green

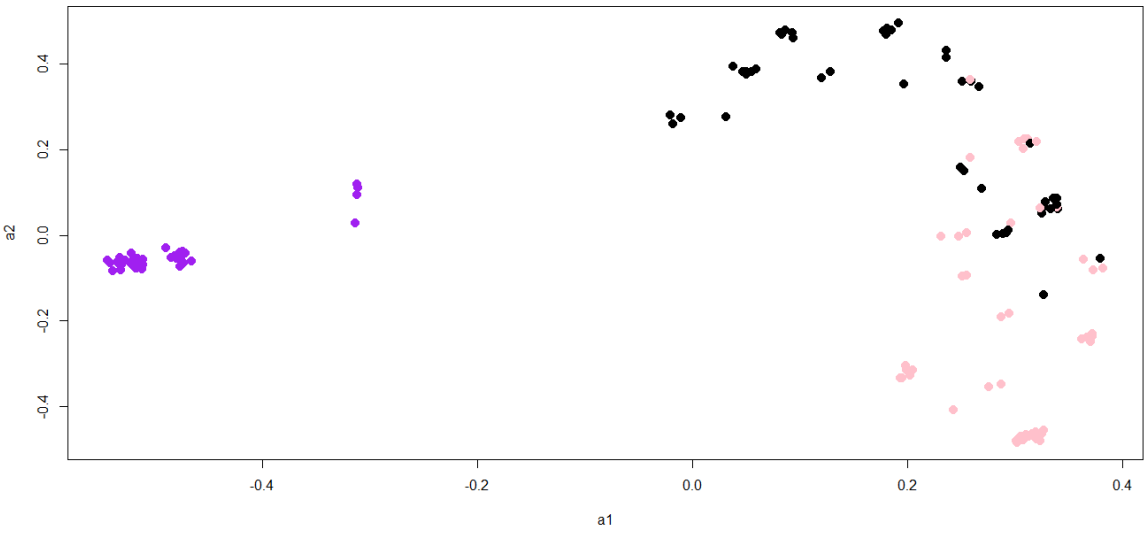


Figure 2. Iris data set mapping using d1 colored by the petal length

Legend: petal length (cm) - (1, 1.9] – purple, (1.9, 4.8] – black, and (4.8, 6.9] – pink

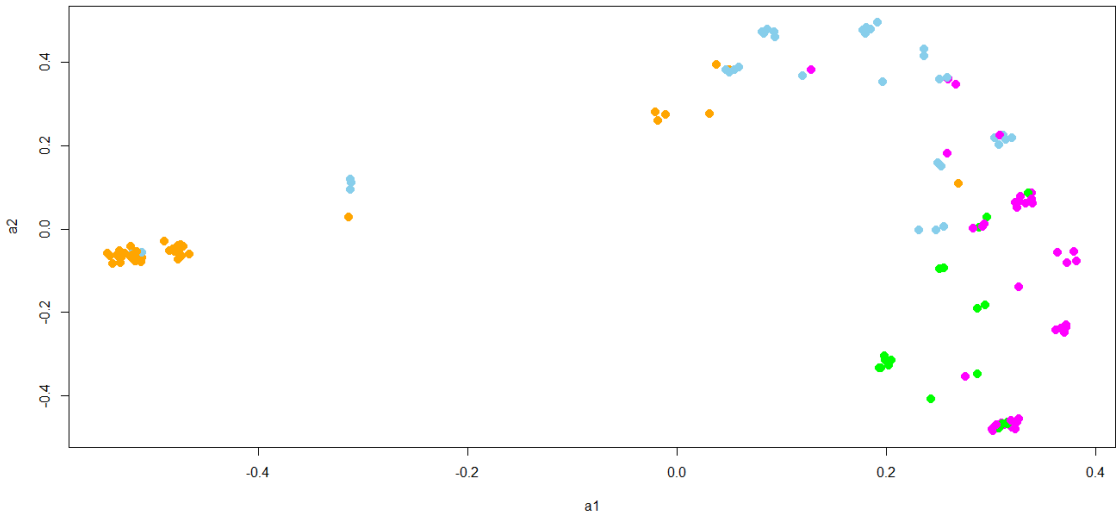


Figure 3. Iris data set mapping using d1 colored by the sepal length

Legend: (4.3, 5.4] – orange, (5.4, 6] – sky blue, (6, 6.7] – magenta, and (6.7, 7.9] - green

Viewing all three figures, an analyst can deduce several insights about the data. We state several of them here. First, the three types of irises are separated in the data set. The tree distance visualization algorithm does not perfectly separate the Versicolor and Virginica irises; therefore, they probably share some key features in the original data set. The Setosa irises seem to be well separated from the others. Second, there is a clear connection between the iris's type and petal length. The Setosa irises have petal lengths between 1.0 cm and 1.9 cm, while the Versicolor's petal lengths are longer, and the Virginica's petal lengths are the longest of them all. Third, there is a connection between the iris's type and its sepal length. For example, the majority of the Virginica irises do not have sepal lengths shorter than 6.0 cm. Fourth, there is a connection between the sepal and the petal length. Irises with long petal lengths have long sepal lengths and vice versa.

The tree distance visualization produces these plots automatically for a requested data set, choosing the variables to color by and the ranges to assign for each color. The insights stated previously about the data can be gained in a couple of minutes of analyzing the three figures. They provide an important understanding about a four-dimensional data set using two-

dimensional visualization, and they do not require an analyst to do any calculations.

B. RELATED WORK

The tree distance visualization maps a mixed-type data set into a low-dimensional Euclidean space in order to communicate insight about patterns in the data to an analyst. The key is to put variables of different types and different scale on the same footing before mapping the data into a two- or three- dimensional space. We describe in this section three types of methods and algorithms: dissimilarities calculations for mixed data types, mapping algorithms of dissimilarities into lower-dimensional Euclidean spaces, and techniques for mapping mixed data types directly into lower-dimensional Euclidean spaces.

1. Dissimilarities Calculation Methods for Mixed Data Type

In this subsection, we review two methods for computing dissimilarities for mixed data types. The first, the Gower dissimilarity (Gower 1971), is the most commonly used dissimilarity for mixed data. The second, random forest proximity (Breiman 2001), is the dissimilarity most similar to tree distances. We note that the distances in this thesis are not necessarily metric, but they are dissimilarities. They are non-negative, symmetric, and the dissimilarity between a point and itself is zero (Hastie, Tibshirani, and Friedman 2009).

The Gower dissimilarity (Gower 1971) is a weighted sum of dissimilarities computed for each variable in a data set. For numerical variables, the method calculates the Manhattan distance between two observations while scaling the distance to be between 0 and 1 by a linear scaling. For categorical variables, the value of the dissimilarity component is 0 if the values of the observations are equal for that variable and 1 otherwise. For observations x_i and x_j, the *k*th component is calculated as described on in (1).

$$d_G^k = \frac{\left| x_{i,k} - x_{j,k} \right|}{\max(x_k) - \min(x_k)} \quad \text{when } x_k \text{ is numeric,}$$

(1)

$$= \begin{Bmatrix} 1 \, if \; x_{i,k} \neq x_{j,k} \\ 0 \, if \; x_{i,k} = x_{j,k} \end{Bmatrix} \quad \text{when } x_k \text{ is categorical.}$$

Letting w_k be the weight of the *k*th component, the Gower dissimilarity between observations *i* and *j* is the weighted sum

$$d_G(i, j) = \frac{\sum_{k=1}^{p} w_k \delta_k(i, j) \, \mathrm{d}_G^k(i, j)}{\sum_{k=1}^{p} w_k \delta_k(i, j)} \quad , \qquad (2)$$

where to accommodate missing values $\delta_{i,j}$ is 0 if either observation is missing variable *k* and 1 otherwise.

Computation of Gower dissimilarities is fast and scaling of numeric variables is automatic. One of its disadvantages with respect to the visualization task is its problem in handling skewed numeric variables and outliers. The scaling technique of the Gower dissimilarities is linear and depends on the extreme values of the variable. If the variable's distribution is skewed or there is an outlier that significantly changes the range of the variable, most of the measured dissimilarities components of this variable will be very small, and a few of them will be close to 1. This phenomenon clusters most of the data observations very close to one another in the visualization mapping and reduces the ability of an analyst to identify the true clusters in the data. An example of the use of Gower dissimilarities and CMDS for visualization can be found in Kagie, van Wezel and Groenen's (2007) example of visualizing online shopping.

The random forests method (Breiman 2001) is a supervised learning method based on ensembles of classification or regression trees. Nonetheless, it can be used for unsupervised proximities calculation (Breiman and Cutler 2003) of mixed data types and therefore for unsupervised

visualization (Liaw and Wiener 2002). In order to calculate the proximities, the method creates a simulated space using the original data set. In the new space, the original observations have a response variable value of 1. In addition, artificial observations are created using the empirical marginal distribution of the original variables, with a response variable value of 0. The random forest is grown over the simulated data. As with the tree distances, two observations are similar if they fall in the same leaves. The calculation of the dissimilarity is done by counting the number of shared leaves for two observations and normalizing it by dividing by twice the number of trees. The proximities are between 0 and 1, and the dissimilarities are calculated by taking the complimentary value. Liaw and Wiener (2002) demonstrate the use of random forests proximities for visualization. They used the CMDS for mapping the dissimilarities.

2. Mapping techniques

There are many mapping techniques for data in high dimensions into lower-dimensional Euclidean space. These include principal component analysis, factor analysis, projection pursuit, and independent component analysis, among others. For a review, see Fodor (2002). In this subsection, we review a new technique desigined to visualize clusters in high-dimensional data, t-Distributed Stochastic Neighbor Embedding (t-SNE; van der Maaten and Hinton 2008).

The t-SNE algorithm is a mapping technique that aims to remedy the phenomenon of unseparated clusters in lower dimension after scaling: the crowding problem (van der Maaten and Hinton 2008). This phenomenon occurs because small dissimilarities measured in high dimensions among many observations do not have enough space when algorithms map them into lower dimensions. Therefore, most of the algorithms solve this problem by filling the gaps between different groups in the mapping, exploiting the fact that there are large dissimilarities among the groups. This solution reduces the ability of an analyst to differentiate among the clusters in the data set. The t-SNE algorithm is based on Stochastic Neighbor Embedding (SNE), which uses conditional distributions between the observations in the data. t-SNE

exploits the heavy-tail property of the Student's T distribution in order to solve the crowding problem (van der Maaten and Hinton 2008).

3. Visualization Techniques for Mixed Data Sets

There are few recent works about visualization techniques of mixed data types. Johansson, Jern, and Johansson (2008) describe an interactive algorithm for the visualization of mixed data sets. Their algorithm quantifies the categorical variables and converts them to a numerical representation before visualizing them. The framework uses user feedback in order to adjust the visualization. The framework calculates the Multiple Correspondence Analysis (MCA; Johansson, Jern, and Johansson 2008) of the categorical variables by creating contingency tables of the relationships between the categorical variables. The numerical variables are converted into categorical variables manually by the analyst or as a result of a clustering process. The MCA coverts the categorical variables into a numerical representation. The last step of the MCA is to visualize the numerical representation. The parallel variables visualization algorithm is the chosen algorithm for this part. The manual interface enables the analyst to use his or her subject matter knowledge about the data.

C. THESIS OUTLINE

This section presents the outline of the rest of the thesis. Chapter II contains the background for this thesis. It describes the components of the tree distance visualization method, including tree distances (Buttrey and Whitaker 2015a) and CMDS (Gower 1966). The second part of Chapter II describes the different data sets we used in this thesis in order to test, evaluate and demonstrate the different properties of the tree distance visualization method.

Chapter III discusses the different coloring techniques of the tree distance visualization's mappings in order to assist the analyst who explores the data looking for patterns. In the beginning of the chapter, we demonstrate the importance of coloring in the process of making insights about the data using the Splice data set (Noordewier, Towell, and Shavlik 1991). We

continue by describing three methods for identifying the appropriate variables by which to color the mappings. The first method is the regression method of Kruskal and Wish (1978), and the other two were developed specifically for tree distance visualization: the maximum deviance ratio and the purity methods. At the end of Chapter III, we introduce our pruning method for choosing values of a variable for coloring.

Chapter IV discusses the issue of strong dependence among variables in a data set and their influence on the tree distances and the tree distance visualization method. We describe the problem of having constructed dependence among variables in a data set, and demonstrate it using the Splice data set. We offer a remedy for the problem by adding an additional step for identifying and removing constructed dependence in a data set and discuss its advantages and disadvantages. We demonstrate the positive effect of the proposed solutions on the mappings of the Credit data set (Lichman, 2013) and the Splice data set with a constructed dependence addition.

Chapter V includes a discussion about the tree distance visualization method. The beginning of the chapter discusses the reasons that tree distance visualization is a suitable visualization method for an analyst who desires to study the clustering properties of the data. We mention a couple of properties of the method, such as the collapsing tendency of similar observations, as well as the equal distance property. These properties help the analyst to understand the relationships inside the data regardless of the scale of the variables. We continue the chapter by discussing the different mappings generated for the same data set by the different variants of the tree distances. Chapter VI contains our summary and conclusions. In this chapter, we point to a number of possible questions that can be researched for future work. This section includes questions about the dimension in which the mappings should be created, the result of repeating the process of the tree distance visualization method on the mapping of the data, and the effect of different visualization techniques using tree distances.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

In this chapter, we review the two major steps of our visualization approach: the tree distances algorithm developed by Buttrey (2006) and expanded and implemented by Buttrey and Whitaker (2015a, 2015b) and the visualization of the dissimilarities in a lower-dimensional Euclidean space using Gower's CMDS (Gower 1966). We describe the entire tree distance visualization process composed of these two steps. We also describe in this chapter the data sets used to test our method.

The chapter is organized as follows: Section A describes tree distances, their four variants, and the motivation for using them. Section B describes CMDS. Section C describes the whole process involving both tree distances and CMDS. We also describe in Section C the additional process of adding an artificial noise for enhancing visualization. Section D includes a short description of the data sets used in this thesis. Section E is the chapter's summary.

A. TREE DISTANCES

Tree distances (Buttrey and Whitaker 2015a) are at the core of our visualization method. The four variants of tree distances are discussed in this section. We finish the section with the motivation for using tree distances for measuring dissimilarities between observations in mixed data–type data sets.

When looking at classification or regression trees, "two observations are similar if they fall in the same leaf" (Buttrey and Whitaker 2015a). For each variable in the data set, the algorithm to compute tree distances builds a tree with that variable as the response variable and the rest of the variables as predictor variables. Mixed variable data sets are allowed because regression trees are built for numeric responses and classification trees are built for categorical responses. Any variable type can act as a predictor variable. Thus, a data set with p variables results in p trees. Each tree is pruned to its "optimal" size using cross-validation (Buttrey and Whitaker 2015a).

The “deviance reduction ratio” is a measure of the goodness of a tree. The deviance of a node in a tree is “measured by the sum of squares of deviations from the mean for a numeric response variable, and by the multinomial deviance for a categorical one” (Buttrey and Whitaker 2015a). The deviance of the tree, is the sum of deviances for each leaf (terminal node). The deviance reduction ratio is calculated by the ratio of the difference between the root node deviance and the tree deviance to the root node deviance. The larger the ratio, the better the tree reduces the deviance, and therefore the better the explanation of the response variable by the predictors. The ratio is between 0 and 1. If the ratio equals 1, then the response variable is explained completely by the tree. The deviance reduction ratio is similar to R^2, which is a classical statistical measurement for the goodness of a linear model. We use the term R^2 analog interchangeably with the deviance reduction ratio.

Once the trees are built, with each tree corresponding to a different variable in the data set, the dissimilarity between two observations is measured based on the number of trees in which those observations fall into the same leaf. The calculation of the dissimilarities can be represented as follows: For data sets with *n* observations and *p* variables, $i = 1,..,n, t = 1,..,p$ denotes the leaf of the t^{th} tree into which the i^{th} observation falls by $L_t(i)$: w_t is the weight of the t^{th} tree; $d^t(i,j)$ is the contribution of the tree *t* to the dissimilarity between observations *i* and *j*. The tree distance dissimilarity between observations *i* and *j* is calculated by the following formula:

$$d(i,j) = \sum_{t=1}^{p} w_t \, d^t(i,j) \, I(L_t(i) \neq L_t(j)) \qquad (3)$$

where $I(\bullet)$ is equal to 1 if the condition is true and equal to 0 otherwise. There are four variants of the tree distances algorithm. The differences between the variants are the weights, w_t, and the tree’s contributions, $d^t(i,j)$. The d1 variant’s weight and tree contribution is equal to 1 across all trees and pairs of observations. The d2 variant also has a tree contribution of 1 across all pairs of observations, but it differs from d1 in the weighting factor. For d2,

the weight of each tree equals the ratio between the R^2 analog of the tree and the maximum R^2 analog across all the trees. This variation gives larger weight for "better" trees. In the third variation, d3, weights of the trees are equal to 1, whereas the $d^t(i, j)$ are computed to reflect how "far apart" the leaves $L_t(i)$ and $L_t(\mathrm{j})$ are; the distance is calculated by the ratio of the decreased deviance between each leaf and their first shared parent node. The last variation, d4, is a combination of d2 and d3. The d4 variant has the same weights as d2 and the same $d^t(i, j)$ calculation as d3. A more detailed discussion about the four variations, including examples, can be found in Buttrey and Whitaker (2015a).

Using tree distances has several advantages for measuring dissimilarities between observations. The first advantage is that the algorithm works on mixed data type data sets, which can include numerical, categorical, or ordinal variables, and any mix of them. The second advantage is the resistance to noise variables. The tree associated with a noise variable usually consists only of a root node. The tree classifies all the observations to the same leaf, and therefore, there is no contribution of that noise variable to the total dissimilarities calculation. The third advantage is the "invariance" to different scales of the data. The tree grows the nodes with respect to the deviance reduction. The tree's splits are not changed by a scale (or location change) for either the response variables or the predictor variable. Therefore, the same dissimilarities are measured for a variable and a linear function of it. Thus, choice of scale does not influence the measurement of dissimilarities. For example a data set may contain one variable measured in kilometers and another measured in centimeters. Further a monotonic function of the predictor variables will not change the splits of a tree. The algorithm measures the distances by the leaves of the trees and not by the original scale. Furthermore, unlike Gower dissimilarities, tree distances are resistant to outliers.

Buttrey and Whitaker (2015a) demonstrate the advantages of the tree distances in their paper. They show that the tree distances perform well most

of the time compared to other distance-measuring techniques in clustering tasks.

B. CLASSICAL MULTIDIMENSIONAL SCALING

Gower (1966) introduced CMDS as "principal coordinates analysis." This method maps dissimilarities between high-dimensional items into a Euclidean space in a requested dimension, while keeping the distances between the points in the new space as close as possible to the original dissimilarities (Kruskal and Wish 1978). The scaling allows the discovery and visualization of the hidden structure of the data, which often enables the analyst to gain quick insights about the data, especially when it contains relationships (similarities or dissimilarities) among data points instead of real values. The scaling is used in a variety of applications, such as to reduce numerical multidimensional data for visualization into lower-dimensional Euclidean space, determine the structure of social groups based on the members' perceptions of each other, and to structure products based on consumer reviews (Kruskal and Wish 1978, 6).

"Stress" is the basic concept behind CMDS. It describes the difference between the original dissimilarities and those created by the new configuration (Kruskal and Wish 1978). Assume a configuration of points in the lower-dimensional Euclidean space where every observation *i* from the original data corresponds to a point in the new space. Denote by $\delta_{i,\mathrm{j}}$ the original dissimilarities between observations *i* and *j,* and $d_{i,\mathrm{j}}$ for the distances between the corresponding points in the new Euclidean space. Stress is defined as the square root of the sum of squared differences between the original dissimilarities and the new distances scaled by the sum of the squared distances in the new space (Kruskal and Wish 1978). The mathematical formulation of the stress is as follows:

$$stress = \sqrt{\frac{\sum_{i,j}(\delta_{i,j} - d_{i,\mathrm{j}})^2}{\sum_{i,j}(d_{i,j})^2}} \tag{4}$$

Stress is non-negative. If the stress equals 0, the original dissimilarities equal the new distances, and therefore the mapping captures perfectly the relationships in the original data. Large stress means a bad fit of the data into the lower-dimensional Euclidean space.

CMDS is an optimization program that finds a mapping to minimize stress for a given set of dissimilarities. The optimization is subject to a constraint that the mean values of all axes in the new Euclidean space are 0, so the mapping is centered at the origin.

The scaling is invariant to rotations and reflections (Kruskal and Wish 1978, 82). The reason for this invariance is that rotated or reflected mappings have the same inter-point distances. Therefore, the axes of the new lower-dimensional Euclidean space have no immediate meaning. Kruskal and Wish (1978, 30–45) discussed how to interpret the meaning of the different axes of a mapping. In Chapter III, we discuss these and suggest two new methods for interpreting the mappings based on tree distances using CMDS.

C. THE TREE DISTANCE VISUALIZATION PROCESS

In this section, we describe the tree distance visualization process, whose key steps are using tree distances for calculating dissimilarities and then applying CMDS. An analyst who desires to use the tree distance visualization technique needs to decide the values of two parameters: the tree distances variant (d1, d2, d3, or d4) and the dimension of the space into which the analyst wishes to map the observations. The tree distance visualization algorithm first calculates the dissimilarities according to the requested variant of the tree distances. After the dissimilarities are computed, the algorithm maps the observations into a space with the requested dimensions using CMDS. The output of the algorithm is the configuration of the observations in the new lower-dimensional Euclidean space.

The process can end after the scaling step. We provide two optional additional steps that can be used after scaling. In Chapter III, we show how to add color to the mapping to enhance its interpretability. The second optional step is the addition of artificial noise to create a more understandable

mapping. Many tree distances have value 0. This happens because the tree distances are based on leaves of trees, which form a gross partition of the data. In our experience, it is common that observations that fall in the same leaf in one tree also do so in the other trees. This phenomenon of a large number of zero dissimilarities is good for clustering (because the observations share the same cluster), but hamper visualization. There is a problem of recognizing how many points are in the same location in space if several points are in exactly the same position. A more severe problem occurs when trying to color the points with respect to a given value. If the points in the same location do not all share the same value, what should the color of the point be?

Our solution is to add artificial noise to the dissimilarities before mapping them to the lower dimensional Euclidean space. Adding a small amount of noise does not change the gross structure of the data, but it does separate points in the same location so the analyst can understand the size of clusters and see the correct color assigned to the value of each observation. For each distance, we add an absolute value of a noise that is sampled from a normal distribution, with a mean of 0 and standard deviation equal to one-tenth of the minimum absolute difference between all pairs of dissimilarities computed for the data set.

Figures 4 and 5 show a visualization using d1 of the 150 observations of the Iris data set (described in the next section), colored by the Iris class. Figure 4 is a mapping without artificial noise, and Figure 5 has the added artificial noise. In Figure 4, only 25 points out of 150 unique observations are visible in the mapping because many of the points are in the same position, and therefore an analyst cannot differentiate among them. Some of those points have different values (Versicolor and Virginica) that are only visible with the added noise in Figure 5.

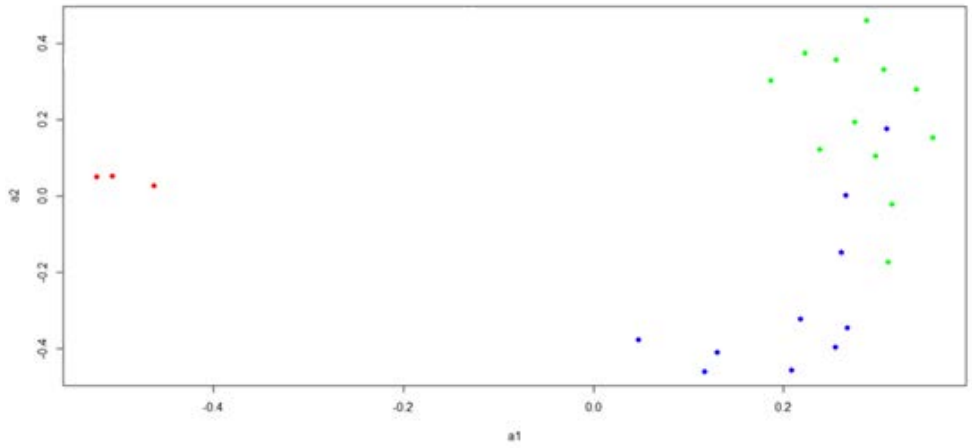


Figure 4. Iris data mapping using d1 colored by the Iris class; without noise

Legend Setosa – red, Versicolor – blue, and Virginica – green

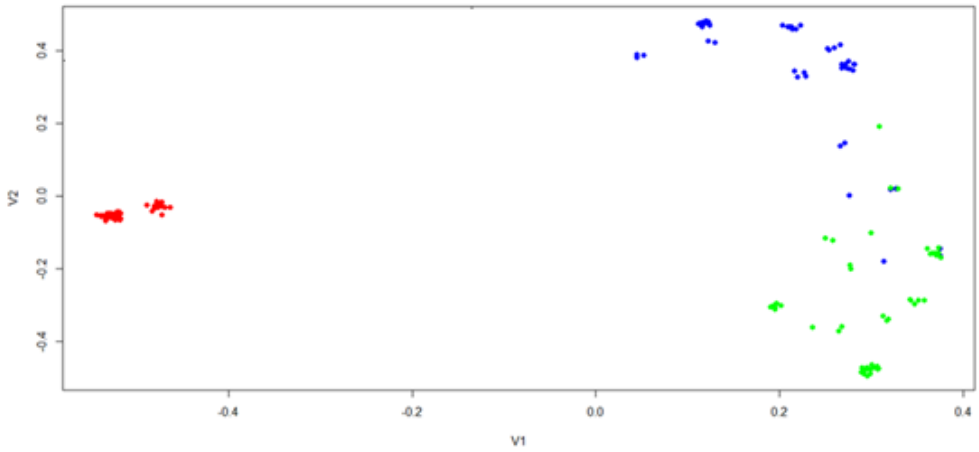


Figure 5. Iris data mapping colored by the Iris class; with noise

Legend Setosa – red, Versicolor – blue and Virginica – green

D. THE DATA SETS

In this thesis, we apply our tree distance visualization to several data sets in order to evaluate its performance. The data sets include all-numeric data sets along with categorical data sets and mixed data type data sets. In this section, we briefly describe the different data sets. All of our data is taken from the UC Irvine Machine Learning Repository (Lichman 2013).

(1) Iris

The Iris data set is one of the best-known data sets in the world for classification. The data set was first introduced by Fisher (Fisher 1936). Each observation is classified into one of three types of irises: “Setosa,” “Versicolor,” and “Virginica.” The data set includes four numeric variables that describe the different flowers’ properties. The variables are: septal length and width, and petal length and width. The data set contains 150 observations, which are split equally among the classes.

(2) Splice

The Splice data set is a genetic data set that contains sequences of DNA (Noordewier, Towell, and Shavlik 1991). There are three different classes that represent two types of genetic splice and their absence. The classes are exon/intron boundaries (“EI”), intron/exon boundaries (“IE”), and neither above (“N”). There are 60 variables (named “V4,” “V5,” ..., “V63”) for the data, which represent the genetic sequence. All variables are categorical, and the most common levels are “A,” “G,” “T,” and “C.” Observations with any other levels for any of the 60 variables are removed from the data set for our purposes. The resulting data contains 3,175 observations. The EI class contains 762 observations, the IE class contains 765 observations, and the N class contains the remaining 1,648 observations.

(3) Credit

The Credit data set is a credit card applications data (Lichman, 2013). The data set is anonymous—all the variables’ names were omitted for privacy reasons. The data consists of 653 observations that are split into two classes. Three hundred fifty seven observations are of type “-,” and the rest are of type “+.” There are 15 variables. Nine of the variables are categorical, and the rest are continuous.

(4) Seeds

The Seeds data set is an agricultural data set that describes the kernels of three different types of wheat: “Kama,” “Rosa,” and “Canadian”

(Charytanowicz et al. 2010). Each class consists of 70 observations; there are 210 in total. The seven continuous variables of the data describe the different properties of the kernels: the area, the perimeter, the length, and more.

E. SUMMARY

In this background chapter, we reviewed the components of the tree distance visualization and the data sets that we used in our research. We reviewed the first step of the process: computing the tree distances for measuring dissimilarities among observations in a mixed data type data set. We reviewed the four variants of the tree distances and the motivation behind using them. Then we reviewed the second part of the visualization algorithm: CMDS. The scaling maps the dissimilarities between all pairs of observations into a lower-dimensional Euclidean space. We defined stress as a function of the difference between the original dissimilarities and the distances in the mapped Euclidean space. CMDS finds a configuration in the mapped Euclidean space that minimizes stress. We then reviewed the complete tree distance visualization process, with two optional additional steps. At the end of the chapter, we described the different data sets that we used in our research, the Iris, the Splice, the Credit, and the Seeds data sets.

THIS PAGE INTENTIONALLY LEFT BLANK

III. COLORING THE TREE DISTANCES MAPPINGS FOR INSIGHTS

One of the big advantages of visualization is the ability to gain insights, most of them quickly and without performing calculations. The tree distances mapping creates a configuration of observations in space. Coloring the observations by the different values of certain data set variables can shed a light on the relationships among the data set's variables. The tree distances measure the distance between observations using the different nodes of trees that are grown as a function of the variables' relationships. Therefore, different spatial regions in the mapping created by the algorithm should represent differences in the relationship among one or more variables of the data set. Insights are easily drawn if the variables for coloring are chosen wisely and an analyst can effectively explore the mapping with respect to the variables.

In this chapter, we discuss how to decide which variables to color by and how to color by their values. We give an example in Section A using the Splice data set. We continue by explaining in Section B the different techniques for automating the coloring. Many data sets have a large number of variables, so it is important to choose those that are dominant in the mapping. We cover several methods for this task, including the regression method suggested by Kruskal and Wish (1978), a method based on trees' R^2 analog that we call the maximum deviance ratio method, and a method based on the "purity" of observations with respect to each variable in a subset of a partition of the lower-dimensional Euclidean space.

In the chapter, we consider a coloring scheme to be good if insights can be made from it. A good coloring scheme has logic—the map is divided by the colors, so a certain partition could be considered as colored mainly by colors that correspond to a subset of values of the data. A good coloring provides information about the spatial configuration of the observations and the data.

A. COLORING SPLICE BY ITS CLASS AND THE V35 VARIABLE

An example of insight that can be drawn from tree distance visualization is given by the Splice data set. The data set, which is described in detail in Chapter II, has three levels to its class: “EI,”“IE,” and “N.” V35 is one of the variables in the data set, and it contains four levels: “A,” “C,” “G,” and “T.” The mapping of the Splice data using d4 is shown in Figures 6 and 7, which are snapshots of interactive 3D plots. Figures 6 and 7 are colored by the Splice class and V35 levels, respectively.

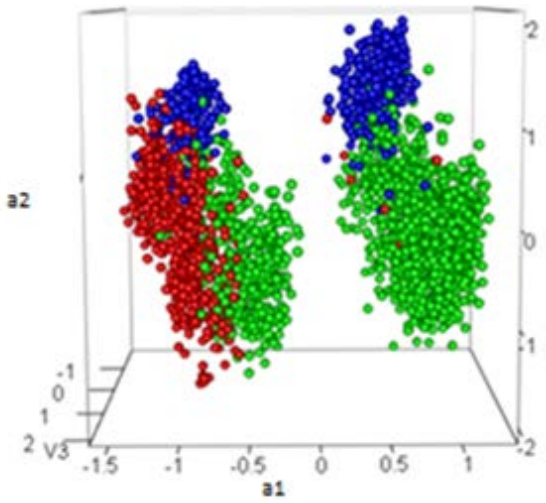


Figure 6. Splice data mapping using d4; colored by Splice Class

Legend “EI” – red, “IE” – blue, and “N” – green

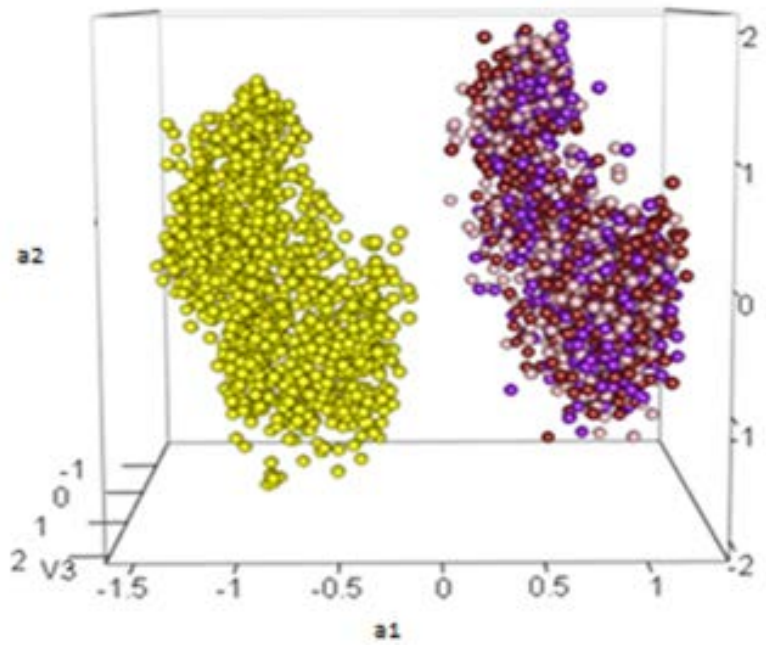


Figure 7. Splice data mapping using d4; colored by levels of V35

Legend “T” – yellow, “C” – brown, “G” – pink, and “A” - purple

Viewing the two mappings in Figures 6 and 7 colored by the Splice class and V35 variable, an analyst can easily conclude that the majority of the “EI” observations have level “T” for the V35 variable. The analyst infers this by observing the lower-left cluster in Figure 7, which has only the level “T” for the V35 variable (the mapping actually splits by the levels of V35) and by observing from Figure 6 that most of the “EI” observations are in the same lower-left cluster. It is also obvious that an observation with level “T” for the V35 variable does not have to be from class “EI” because the lower-left cluster also contains many “N” and “IE” observations. We summarize in Table 1 the distribution of the “EI” class and level “T” in the V35 variable.

Table 1. The relationship between the “T” level for variable V35 and the “EI” class for Splice class

	Class “EI”	Classes “N” and “IE	Total
V35 Level “T”	755	699	1424
V35 Levels “C,” “G” and “A”	7	1744	1751
Total	762	2413	3715

While 99% of the observations of class “EI” have level “T” for the V35 variable, 53% of the “T” observations are from class “EI.” The insight is immediate from comparing the different coloring of the mapping without gathering statistics on the data set such as in Table 1. This example demonstrates the power of colored visualization in gaining quick insights about the relationships among the variables in the data.

B. CHOOSING VARIABLES FOR COLORING

Visualization using tree distances is suited for data sets with large numbers of variables. We want visualizations that help the analyst understand such data. Therefore, when we color, it is important to decide which variables of the whole set of variables to select. We aim to choose those that give the greatest insight with the smallest number of different coloring schemes and with minimum effort by the analyst.

Finding the important variables in a data set, often called variable selection, is a problem that has been extensively researched and documented (e.g., Hastie,, Tibshirani, and Friedman 2009). Most of the known techniques for this task are applied only to numerical data sets. Principal component analysis is a commonly used example for these kinds of numerical techniques (Jolliffe 1986). Breiman (2001) suggests a method for identifying the most important variables in a tree-related domain using random forests for mixed data types.

We focus our discussion on methods that address specifically the characteristics of the tree distance visualization process. We consider three methods in this chapter. We start with the classical regression method suggested by Kruskal and Wish (1978) which fits a regression, linear or

logistic, between the coordinates of the data and the different variables. We discuss its disadvantages, which leads us to develop two new methods: the maximum deviance ratio method, which takes advantage of the information about the trees generated to compute tree distances, and the purity method that finds the variables that have the most pure areas in the data. We continue by considering the advantages and the disadvantages of each method and providing examples.

1. The Regression Method

Regression is a common tool for creating inferences from multidimensional scaling (Kruskal and Wish 1978, 36). If we regress a variable against the coordinates of the map, we have a statistical test for the relationship between the position (coordinates) in the mapping and the variable. If there is a significant statistical relationship, the direction of the relationship can be obtained from normalizing the coefficients of the coordinates (Kruskal and Wish 1978, 37–39). The final result is a linear relationship between a direction in space and the variable. A dummy example of such a relationship could be “increasing the value of axis a1 by 1 increases on average the value of variable Y by 2.1.” The method provides both the important variables for coloring (by statistical significance and weight of coefficients) and the relationship itself. Logistic regression can also be used for categorical values using the same principles as linear variables.

There are several advantages of the regression method. The linear and logistic regression are easy to understand and very common methods in statistical applications. They are well supported theoretically and have a large number of commonly used implementations.

We tested the regression method on several data sets, and our conclusion is that the regression method has a severe drawback that does not make it the best option for use. CMDS applied to the tree distances does not tend to map different well-separated groups or clusters of observations in a linear configuration. One of the common configurations (but not the only one) created by the algorithm is a circle or sphere or a “horseshoe” (Kruskal and

Wish 1978) configuration of the different clusters. One explanation for this phenomenon could be that tree distances create an equal separation between different clusters, and the ideal way to order equal-distance objects in space is a sphere or a circle. A linear regression on the linear representation of the coordinates cannot explain a sphere or circle configuration well. Therefore, linear regression methods are not the right tool for this task.

A simple example of this behavior can be observed in the Iris data set mapping, colored by petal length, shown in Figure 8.

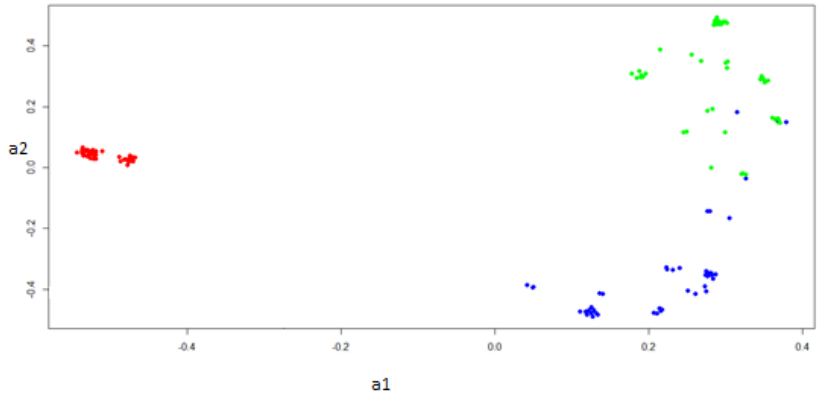


Figure 8. Iris data map by d1 color-coded by petal length

Legend (1, 1.9] – red, (1.9,4.8] – blue and (4.8, 6.9] – green

Although the relationship is not perfect, it is clear that the petal length increases with counterclockwise rotation from a starting point at the lower values of a1 (the red cluster). The relationship is not linear.

Another problem is that the tree distances are calculated using the tree's properties and depends how far apart tree leaves are, not on the values of the variable itself. Therefore, there is no guarantee that the change of the variable by the coordinates is linear and not another increasing function.

These drawbacks are also valid for logistic regression. We recommend not using regression as a variable importance method.

2. Maximum Deviance Ratio Method

The tree distance algorithm produces a deviance reduction ratio, which can be used as a R^2 analog, for each tree that it creates. As mentioned in Chapter II, the R^2 analog is a measure of the performance of a tree. The higher the ratio is (close to 1), the better a tree explains the response variable as a function of the predictor variables.

Trees are grown with each of the variables used in turn as the response variable while the rest of the variables are treated as predictor variables. The R^2 analog indicates how much the predictor variables reduce the deviance of the root node. The better the response is explained by the other variables, the higher its R^2 analog. Therefore, a simple approach for finding important variables is to choose the trees with the highest R^2 analog. In addition, as described in Chapter II, the d2–d4 distance variants weigh the contribution of each variable by the deviance reduction ratio of the associated tree. For these distances, there is an additional reason for using the R^2 analog as an indicator for the important variables.

In Table 2, we demonstrate the maximum deviance ratio method using the Seeds data set, which was introduced in Chapter II, and the d1 variant.

Table 2. Deviance ratio deduction per variable; Seeds data set

Variable	R^2 analog
V1	0.968
V2	0.962
V3	0.878
V4	0.947
V5	0.933
V6	0.445
V7	0.890

Table 2 shows the analog R^2 for each variable's associated tree. It is possible that a variables associated tree can have a 0 for its R^2 analog. This happens when the associated tree is discarded because it contains only a root node. We consider the two extremes: the variable with the highest R^2

analog and the one with the lowest. Those are V1 with 0.968, and V6 with 0.445, respectively. Figure 9 shows the Seeds mapping for two dimensions using d1 colored by V1, while Figure 10 shows the same map colored by V6.

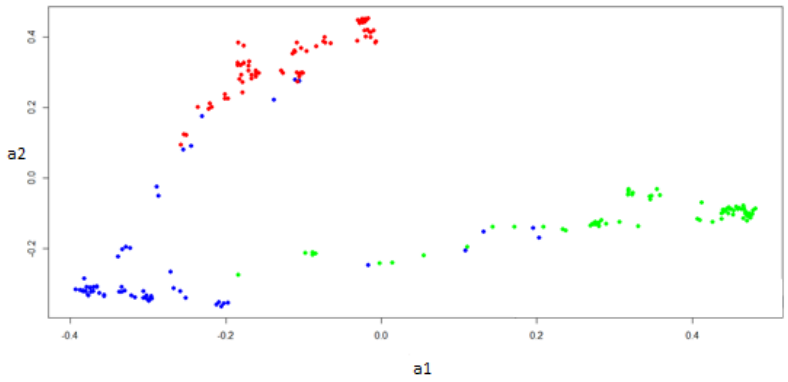


Figure 9. Seeds data set mapped using d1; colored by V1

Legend: (10.6,12.8] – red, (12.8, 15.6] – blue and (15.6, 21.2] – green

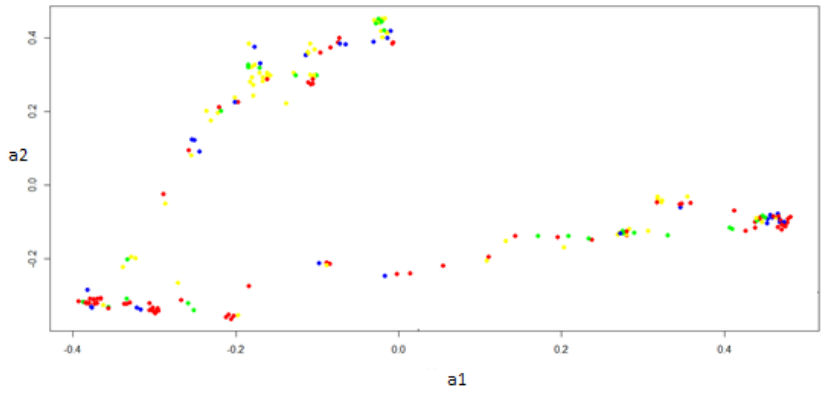


Figure 10. Seeds data set mapped using d1; colored by V6

Legend: (0.765, 3.13] – red, (3.13, 3.69] – blue, (3.69, 4.33] – green, and (4.33, 8.46] - yellow

It is clear that the coloring scheme based on V1, the variable with the maximum R^2 analog, almost partitions the observations in the 2D mapping. Coloring by V6, with the minimum R^2 analog, colors almost randomly. This example fits our hypothesis about the relationship between the analog R^2 and

the coloring. We have observed the successful performance of the maximum deviance ratio method on other data sets, such as Iris, Splice, and Credit.

An advantage of the maximum deviance ratio is that it is very fast. The deviance ratio table is one of the basic outputs of the tree distance algorithm. Ordering the variables by the ratio is an easy, fast task.

For many data sets, the most interesting variables can be identified as those that have the largest R^2 analog. However, this is not always the case. The reason for this is that the relationship between a variable and the distances could be the result of the tree associated with the variable, or the other trees that could be split by the variable values. An example of this is that the most important variable chosen by the purity method for coloring the Splice data is variable V35, See section 3.A.3. The associated tree of V35 was discarded in the algorithmic process because there are so many "T" level values for the variable, so the resulting tree consists only the root node, which predicts "T."

3. The Purity Method

The purity method is our proposed method for identifying which categorical variable to color by. This method can also be used for numerical variables converted to categorical. In the next section we introduce a technique, the pruning method, by which numerical variables can be made categorical using results from the tree distance computations. We start the discussion about the purity method by defining a couple of terms. We define a region of a partition of the mapping to be "pure" with respect to a certain categorical variable (including the data set class) if the region contains only one level of the variable.

"Purity" means how pure a region is. We can calculate a measure of purity by calculating the total number of observations inside a specific region and computing the distribution of them among the different levels of the variable. Purity is then defined by the ratio of the number of observations at the most common level to the total number of observations in the region. The maximum purity is 1, which occurs if all the observations share the same

level. The minimum purity for a region with $n \geq 1$ observations is $1/n$ which occurs when none of the observations share the same level.

We can assume that if we partition the space into reasonably-sized regions and the majority of the partitions have high purity with respect to a certain categorical variable, then coloring observation by the variable will be useful.

There are several drawbacks to this assumption. First, what are reasonably-sized regions? If the size of the regions is too large, the purity value will be low. As an extreme example, if the mapping contains two separate equal-sized clusters, each with a different level, the purity of the total space is 0.5, while a partition which separates the two clusters will have regions with purity 1.

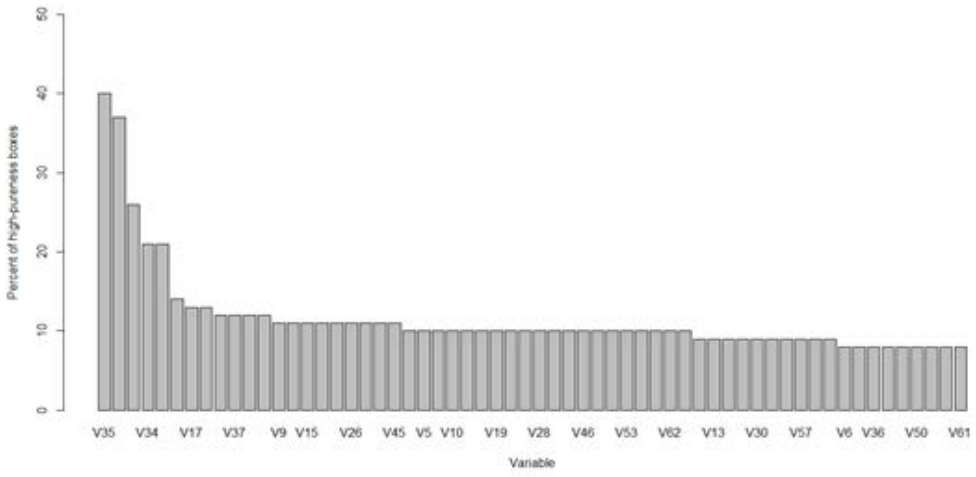
On the other hand, the smaller the size of the region, the less information can be obtained from calculating the purity. For example, if we divide a mapping into regions in which each one contains only one observation, then the purity of all of the regions is the maximum, 1, but this does not aid our understanding. In our research, we find that several trials usually give reasonably-sized partitions, which balances the two extremes of regions which are too large and regions which are too small.

The other drawback is the fact that even if each region has a high purity level, this does not assure that there is a continuity between adjacent partitions. Returning to the example of a region for each observation, high purity does not guarantee that adjacent region will contain the same most frequent value of the variables. We find that although this drawback exists in theory, if we divide the mapping into reasonably-sized regions, we tend to get reasonable results in the data sets we explore. A sensitivity analysis of the size of the partitions can identify the existence of this problem. If the purity does not change severely as a result of a small change in the partition's size, we can assume the data splits solidly.

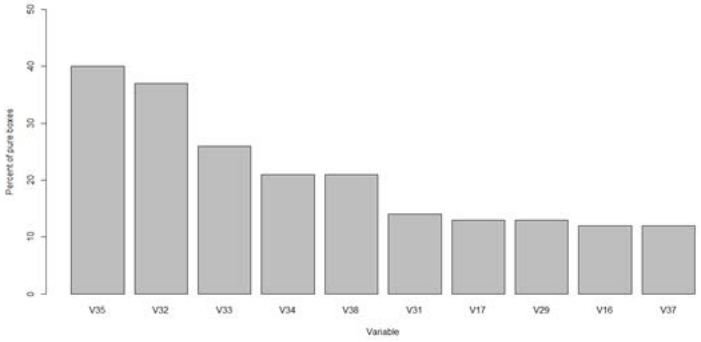
The purity method splits the mapping into equally-sized boxes in two- or three- dimensions, depending on the map dimensionality. Then it calculates

the purity of each box with respect to each variable. The last step is ordering the variables by the number of boxes with purity larger than 0.9. The boxes are common among all the variables.

We demonstrate this method using the Splice data set using d1 mapped into three dimensions. The Splice data set contains 60 categorical variables. We choose to split the map into 216 equal sized boxes, which are five cuts (six areas) per dimension. A bar plot of the percentage of boxes with purity above 0.9 by variable is shown in Figure 11.



(a)



(b)

Figure 11. Bar plot of the percent of boxes with purity above 0.9 per variable in the Splice data mapped using d1

Note. The plot (a) has all the variables of the Splice data, while plot (b) has only the top 10 deviance reduction ratio variables

From the bar plot Figure 11 (b), it is clear that there are five variables with high purity percentages: V35, V32, V33, V34, and V38, while the other variables have low purity percentages. We examine the mapping color-coded by three variables: the top two pure variables, V35 and V32, and the one with the lowest purity, V61.

Figure 12 shows the Splice mapping of d1 color-coded by V35. It is clear V35 has a pure region at the negative values of axis a1 of the level "T." The positive values of axis a1 contain the rest of the levels without a visible ordering.

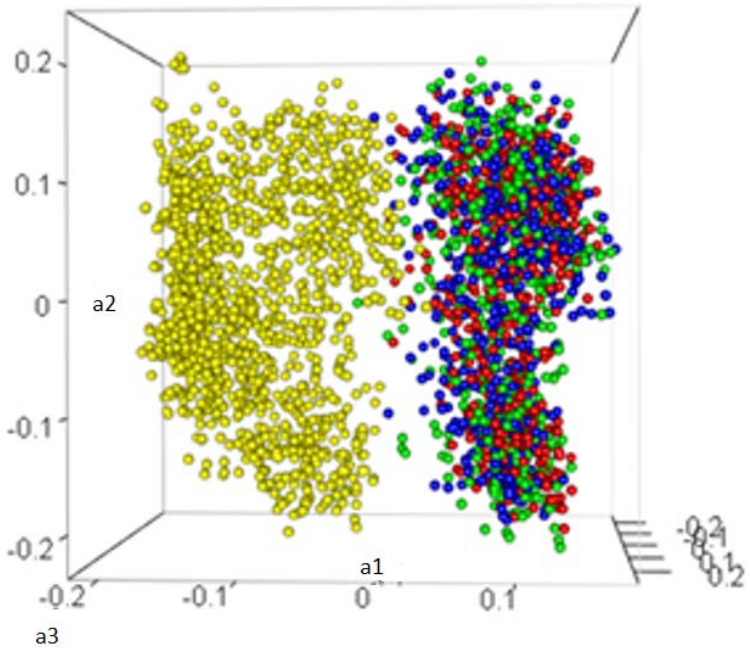


Figure 12. Splice data map by d1 color-coded by V35

Legend "A" – red, "G" – green, "C" – blue, and "T" – yellow

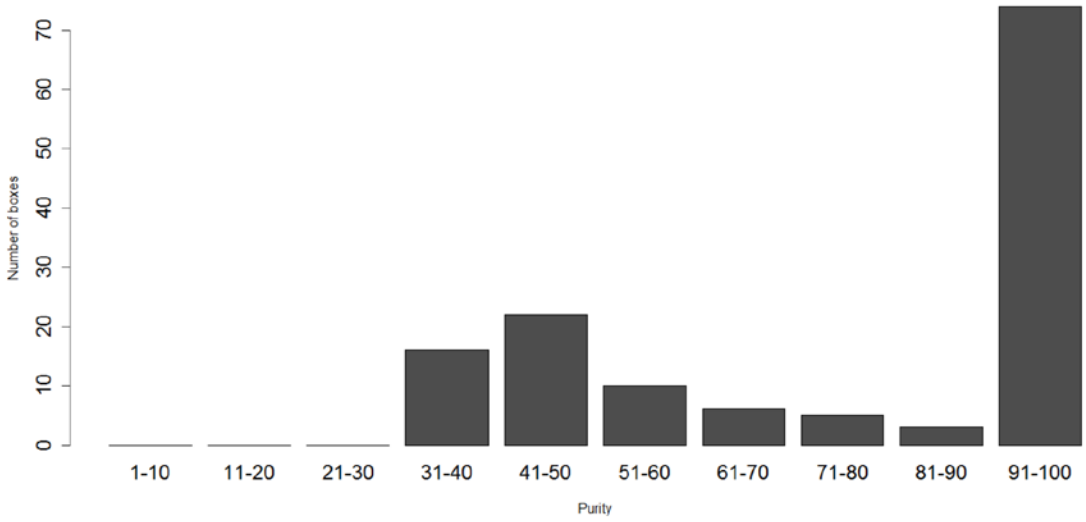


Figure 13. Purity histogram of V35 in Splice mapping using d1

Figure 13 shows the distribution of the purity of V35. The large number of pure regions corresponds to the regions that contain the “T” value of V35 and correspond to the yellow cluster in Figure 12. The other regions are much less pure, and they represent the multicolored cluster of observations in Figure 12.

From the Splice mapping colored by V32 in Figure 14, it is easy to see that the “A” values of V32 are concentrated in a specific region of the mapping. It is also clear that the separation of the “A” values from the other values is not clean, which is the reason V32 has fewer regions that V35 with purity larger than 0.9.

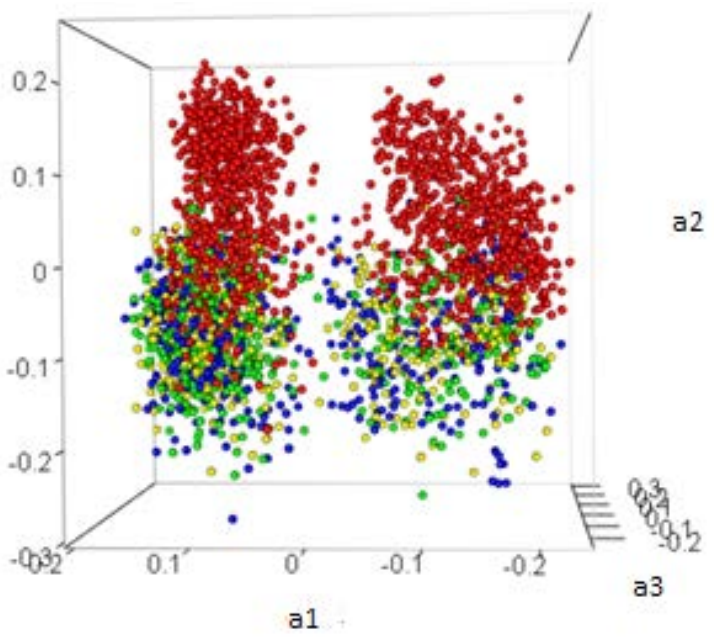


Figure 14. Splice data mapping using d1 color coded by V32

Legend “A” – red, “G” – green, “C” – blue, and “T” – yellow

Figure 15 is a histogram of the purity by V32 of the different regions in the Splice mapping. Figure 15 shows a plot similar to the V35 histogram in Figure 12, but with fewer pure boxes.

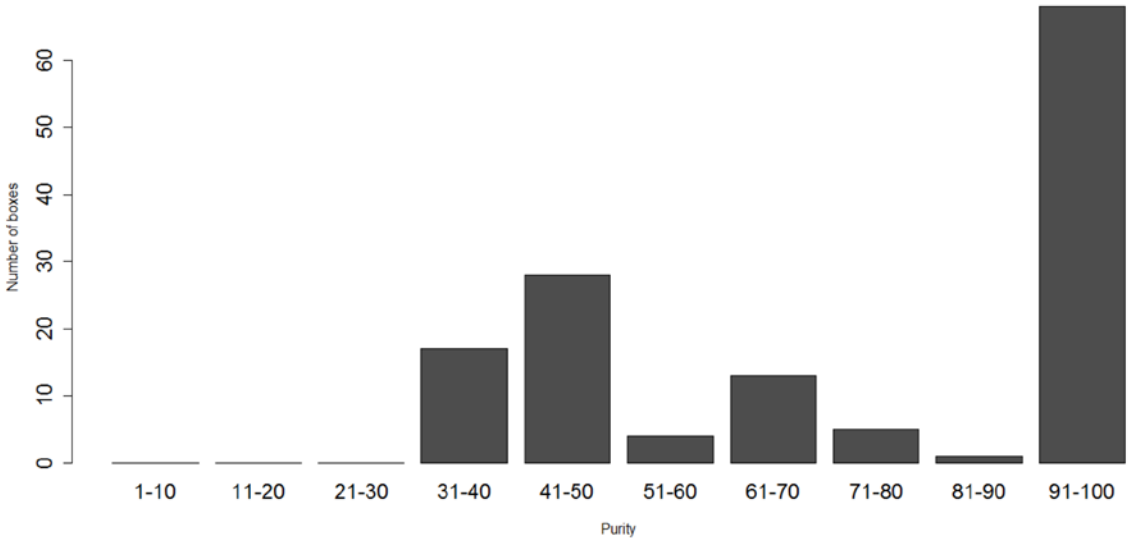


Figure 15. Purity histogram - V32 in Splice mapping using d1

In contrast to both V35 and V32, which have a large number of pure boxes, V63 has the lowest number of pure boxes. Examining the Splice map colored by V61 and the purity histogram in Figures 16 and 17, respectively, one can see that the values of V61 are spread quite randomly over the map. There is still a moderate number of pure boxes, but they are not significant enough or close enough to each other to uncover a pattern in the data with respect to V61.

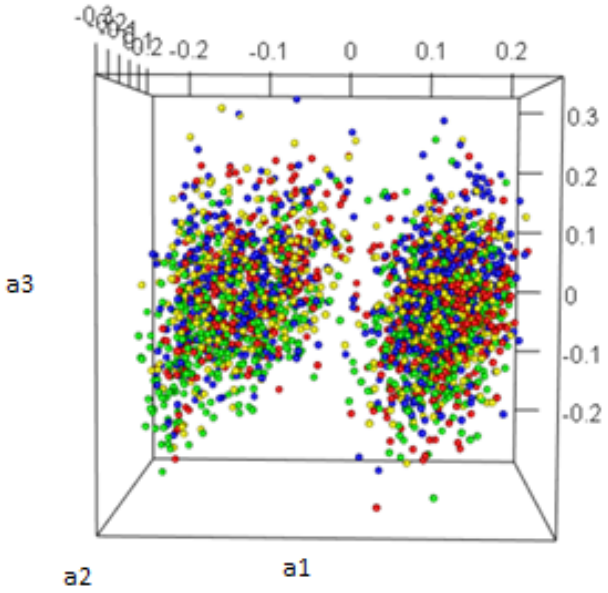


Figure 16. Splice data mapping using d1 color coded by V61

Legend “A” – red, “G” – green, “C” – blue, and “T” – yellow

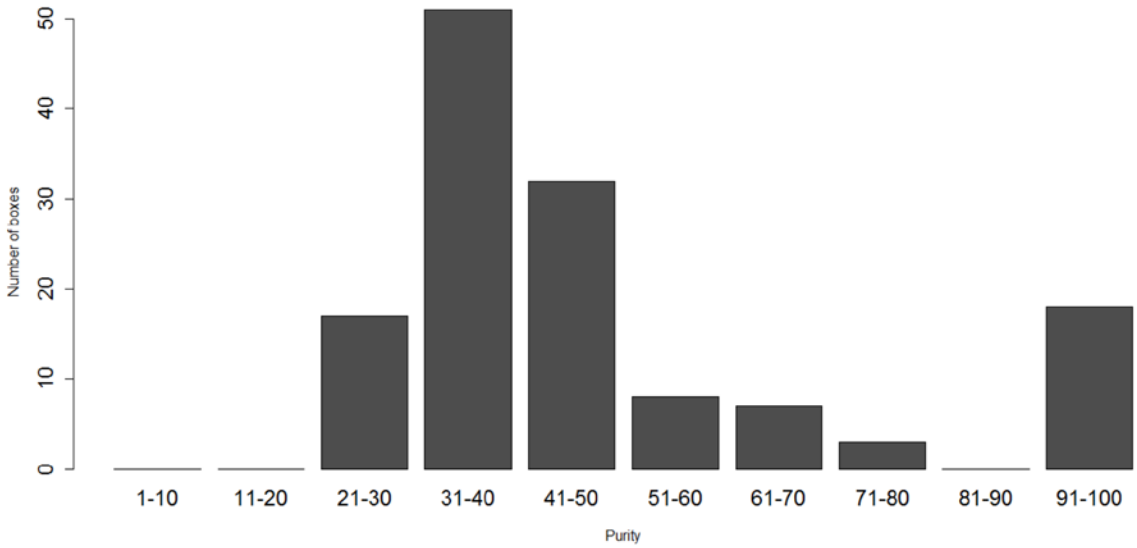


Figure 17. Purity histogram – V61 in Splice mapping using d1

In conclusion, this method can help identify important variables for coloring a lower-dimensional mapping of the data. The graphical colored mapping of the data aids in uncovering patterns and drawing conclusions.

C. CHOOSING VALUES FOR COLORING (PRUNING METHOD)

Once an interesting variable has been chosen, the question is how to color it. For categorical variables, most of the time the answer is simple—assign a color for each level of the variable. Assigning a different color for each value of a numerical variable in the data set creates confusing coloring. The problem also occurs with categorical variables with large numbers of levels.

In this section, we cover our suggested method for coloring a single variable: the pruning method, which is applicable both for categorical and numerical variables.

Our proposed solution is the pruning method, which is based on the trees used to compute tree distances. We describe the method in relation to numerical variables, but it can also be extended to categorical variables. The pruning method can only be used on variables that have associated trees. As

described in Chapter II, a tree comprises several nodes, which contain several observations following If-Then conditions. The deviance of the observations in each node is smaller than the deviance of its parent node. Therefore, values of the variable of observations in the nodes can indicate how to “cut” values of a numerical variable. The leaves of a tree are natural candidates. But in order not to select too many values, it is possible to set a threshold for the number of observations in a node. Only nodes with large numbers of observations in them are selected as indicators, which reduces the potential number of splits.

The pruning method steps are as follows:

1. Select the lowest nodes in each branch of the variable’s associated tree that contain more observations than a threshold value.

2. Calculate the mean of the variable’s values for each selected node, add these to the list of the minimum and maximum values of the variable, and order the values. For each ordered pair of means in this set, we call the smaller of the two L and the larger H.

3. For each pair of ordered nodes, compute the median value among observations that are both larger than L and also smaller than H. Each node’s mean is an estimator of the values inside the nodes. We are interested in the midpoints between the nodes because they represent the split of the tree that creates the two nodes, which in turn determine the dissimilarities. The median values are the proposed cutting values for the variable. There could be other possibilities for calculating the cutting values, such as the mean or a simple average of the cutting values.

4. Assign a different color to each interval defined by the values.

How can one decide on an appropriate threshold? One suggestion is to start with the number of groups that the analyst wants to split the data into. The number of original classes (if it is known) is a possible good choice, but there could be others. Assuming that the size of the groups should be roughly the same, the threshold is the number of observations divided by the number of groups, plus a small value (for example, 5% of the number of observations). Because we seek to find the largest number of nodes in each branch that have fewer observations than the threshold, we add an additional

buffer of a small value for the sake of sensitivity. If the data splits in a reasonable way, the result is roughly the same number of cuts as the number of groups that were chosen.

The pruning method has several advantages. The main advantage is the fact that this method is based on the trees used to compute the trees distances. Another advantage of using the pruning method is the uneven ranges it creates. The trees can cut the variables unevenly because the data's distribution does not have to be uniform. The distances measured using the trees are linearly correlated not with the variables, but with the splits of the trees. Therefore, coloring by the nodes properties using the pruning method creates uneven assignments to the colors, but relates better to the mapping than even assignments and can help to explain uneven distributions of the data.

The last advantage that we mention is the fact that the pruning method is fast. It is based on analyzing one tree, which is much smaller than the whole data set. The heaviest computational step is computing the median, and this could be replaced by faster methods, such as the simple average of the cutting values.

The major disadvantage of the pruning method is the need for the variable to have an associated tree. Not every important variable has an associated tree, which contains more than a root node. The pruning method exploits the properties of the trees, and therefore cannot work on variables with only a root node.

Another disadvantage is that the method assumes that the tree and the data set behave "nicely." When using a threshold to control the number of groups, we assume the groups will split mostly evenly. When using the median for finding the cutting values, we assume the data set distributes well between the mean values of the nodes, and so forth.

The last disadvantage of the pruning method is the inability to identify discontinuity in the data. There could be a situation where the values of a specific variable are distributed only in specific regions. The pruning method

calculates the ranges between the different cuts. It also includes values that invalid in the data, which could lead to misleading interpretation of the relationships among the variables.

The Iris data set is a good example for the pruning method. Figure 5, shown previously in the chapter, is colored using the pruning method, where the selected variable is the petal length. The threshold is set in order to create three groups, which correspond to the number of the classes of the data. The value of the threshold is 58. The reader can observe that the method splits the variable into the different clusters of the map. The splits are not equal in their range; one of them has a width of 0.9 cm, while the others are more similar to each other with widths of 2.9 cm and 2.1 cm.

D. CONCLUSIONS

A well-colored map of a data set can lead to meaningful insights about the data. In this chapter, we covered several aspects of coloring the maps created by tree distances. We started with an example of the insights an analyst can create of the Splice data set using tree distances color-coded mapping. We then described several techniques for choosing a coloring scheme that would produce insights. We reviewed several methods that produce candidates for important variables to color the mapping with respect to the variables. The methods included Kruskal and Wish's (1978) regression methods, and we proposed two new methods, which are unique to tree distances mapping—the maximum deviance ratio and the purity methods. We reviewed each method, discussed their advantages and disadvantages, and analyzed examples of several of them. We continued with a discussion of how to color the map according to a single variable. Finally, we proposed the pruning method as a tree-based method for assigning colors for ranges of the variable.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. STRONG DEPENDENCE AMONG VARIABLES

In this chapter, we discuss the issue of data sets containing sets of strongly dependent variables, which can bias the results of tree distances.

In most situations, variables in data sets exhibit dependence. Often the dependence is between measurements of related attributes (e.g., height, weight, and body fat percent), all of which might be important for the task at hand. Indeed, the tree distances exploit such dependence. However, just as often, in large data sets, variables exhibit strong dependence by virtue of how they are constructed (e.g., one variable for temperature measured in Fahrenheit and another variable with the same temperatures measured in Celsius). Dependence among these variables in no way sheds insight into the nature of the data and should not influence analytical results. We can define this type of dependence as constructed dependence. We expect to see more of constructed dependence in the big-data era, where a large number of data sets are created by merging and combining different data sets.

In the first section of the chapter, we describe how such constructed dependence can cause problems in interpreting visualizations based on tree distances. We illustrate these issues with the Splice data. The second section of the chapter modifies the tree distance algorithm to address this problem. We describe the modification and its advantages and disadvantages, and finally we give examples of experiments of our proposed modified algorithm applied to several data sets.

A. THE ISSUE OF STRONG DEPENDENCE

In this section, we discuss the problems associated with strong dependence and how to account for them.

1. Theoretical Analysis

Tree distances measure dissimilarities by exploiting dependencies among variables in a data set. Problems occur when the dependence among

the variables is a result of constructed dependence. This phenomenon could happen, for example, when

1. The variable is recorded twice.
2. The variable is recorded in two different units of measurements, such as temperature in Celsius and in Fahrenheit.
3. One numeric variable is a monotonic function of another variable in the data set, such as including population size and the log of population size in the same data set.
4. One of the variables is a summary of the other, such as temperature in Fahrenheit and a categorical variable of two levels, “Cold” and “Hot,” with respect to some threshold.
5. A categorical variable is constructed with levels that are collapsed versions of another categorical variable. This happens, for example, when several levels of a categorical variable are combined into “Other” and both the new variable and the old variable are retained.

There are two consequences of using tree distances with data containing a set of such variables. First, because of the strong dependence, the trees for the relevant variables contain only the constructed dependent variables. This is because they “explain” one another perfectly and there is no need for other variables. This “blocks” the tree from representing the relationships between the variable and other variables, and thus we lose information about the structure of the data. Second, the strong dependence results in a large deviance reduction. The tree distance variants d2, d3, and d4, take into account the deviance reduction. The tree distances d2 and d4 weight the contribution of each variable by its deviance reduction ratio. In addition, d3 and d4 use deviance reduction within each tree to compute inter-leaf distances. Especially with d2 and d4, variables with strong constructed dependence make the major contribution to the dissimilarities. This results in high bias for the dependent-redundant variables, which eliminates the other variables.

If the dependence between variables is “natural,” the consequences could be a virtue—a true representation of the data. But if it is because of constructed dependence, we lose important information about the data, and the results could be highly biased.

2. Case Study: Splice Data Set

We review in detail an example of the issue of strong dependence using the Splice data set, which was described in Chapter II. As illustrated in the next subsection (Section A.2.a), the Splice data does not contain strongly dependent variables, so in Section A.2.b, we illustrate the effects of adding artificial strongly dependent variables to the Splice data set for comparison.

a. Splice Data Mapped Using d1 and d4

If we calculate the tree distances for the Splice data, we get the reduction in the deviance ratio (DevRat) per variable plot, as shown in Figure 18.

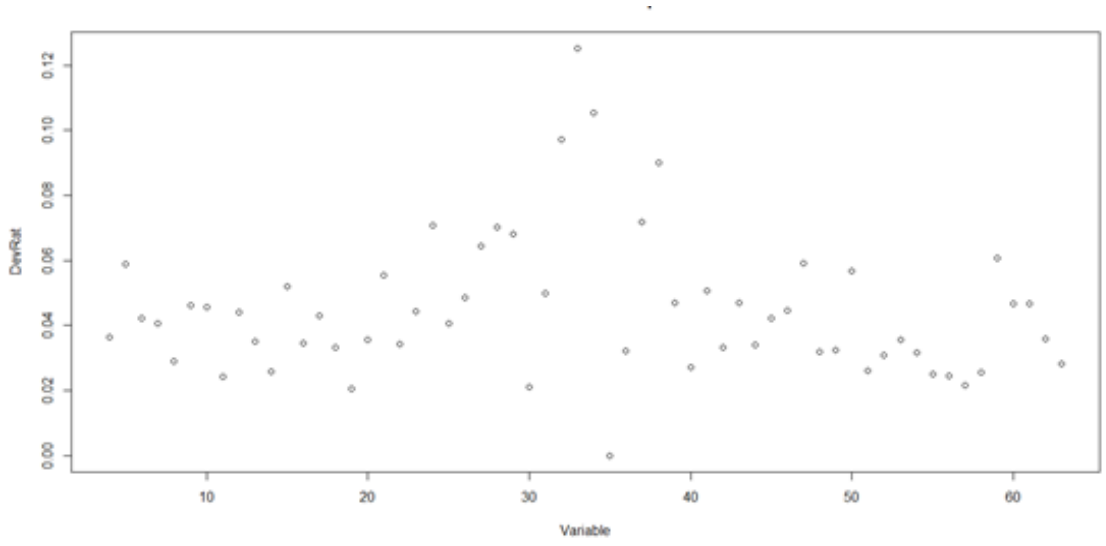


Figure 18. Reduction in R^2 analog (deviance ratio-DevRat) per variable for the Splice data

We can see that none of the variables has a R^2 analog, the deviance reduction ratio, larger than 0.15. We focus our analysis on d1 and d4 distances as the two extreme tree distances variants with respect to the R^2 analog. The d1 is not affected by the R^2 analog, and d4 is a weighted contribution of each tree's R^2 analog. The plots in Figures 19 and 20 are 3D mappings of the Splice data visualized using d1 and d4. The colors correspond to the three different levels of class, "EI," "IE," and "N."

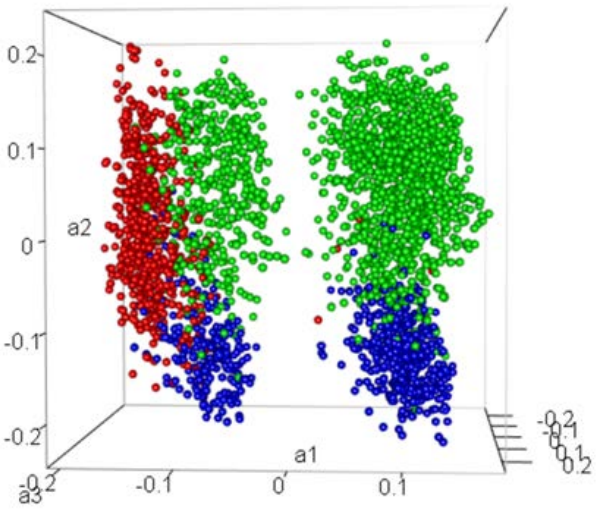


Figure 19. Splice data mapped using d1 colored by Splice class levels

Legend “EI” – red, “IE” – blue, and “N” – green

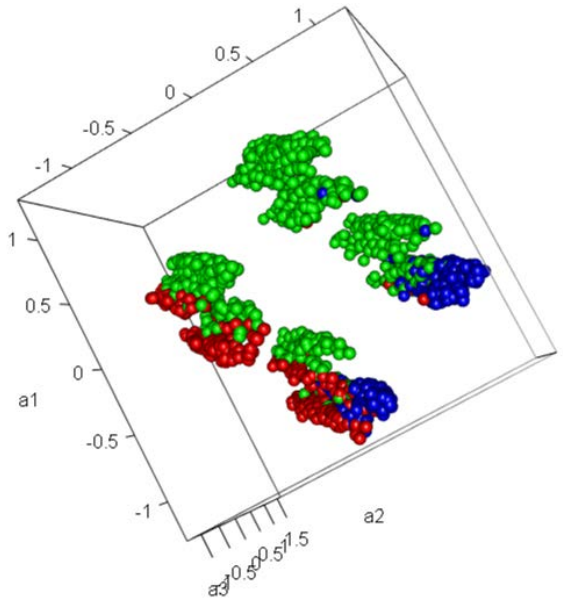


Figure 20. Splice data mapped using d4 colored by Splice class levels

Legend “EI” – red, “IE” – blue, and “N” – green

Examining the mappings in Figures 19 and 20, one can see two large groups divided into smaller groups. We can see that it is possible to identify spatially the different classes from the mapping based on either d1 or d4 dissimilarities.

Tree distances split the data by the important variables and their relationships and spreads the non-important variables' values almost randomly through the lower dimension mapping. For example, if we color the mappings of Figures 19 and 20 by variable V4, a non-important variable, we get Figures 21 and 22 for d1 and d4, respectively.

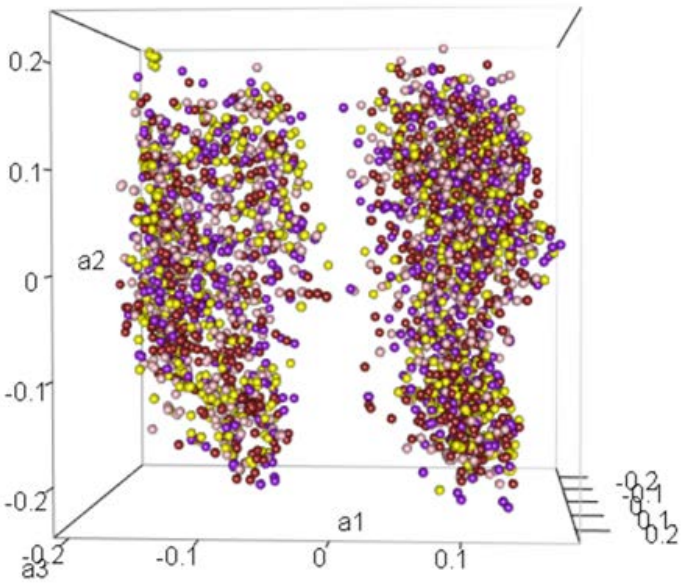


Figure 21. Splice data mapped by d1 colored by V4

Legend "C" – brown, "A" – purple, "G" – pink, and "T" – yellow

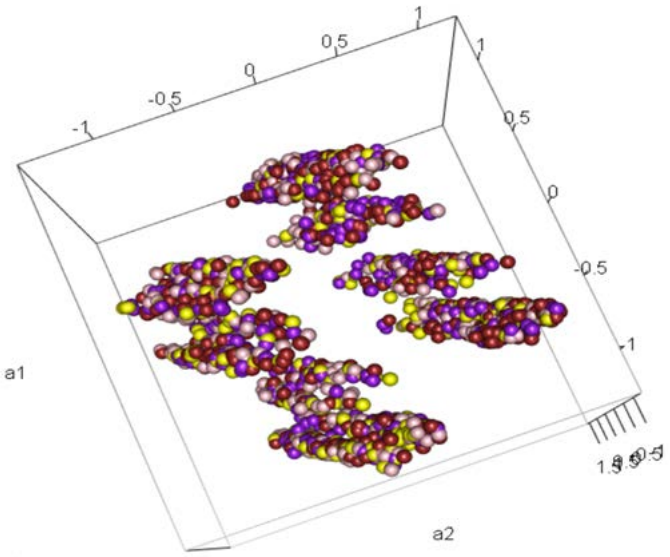


Figure 22. Splice data mapped by d4 colored by V4

Legend "C" – brown, "A" – purple, "G" – pink, and "T" – yellow

These plots suggest that the tree distance algorithm does not consider V4 as an important variable. The colors corresponding to different levels of V4 are spread almost randomly over the mapping in the different groups.

b. Splice Data With Constructed Dependence

In this section, we examine the impact of constructing variables that are strongly dependent on existing variables in the data set. We start by adding a copy of the variable V4 to the Splice data set. With the addition of "V0," an exact copy of V4, to the data, we get the deviance ratio plot shown in Figure 23.

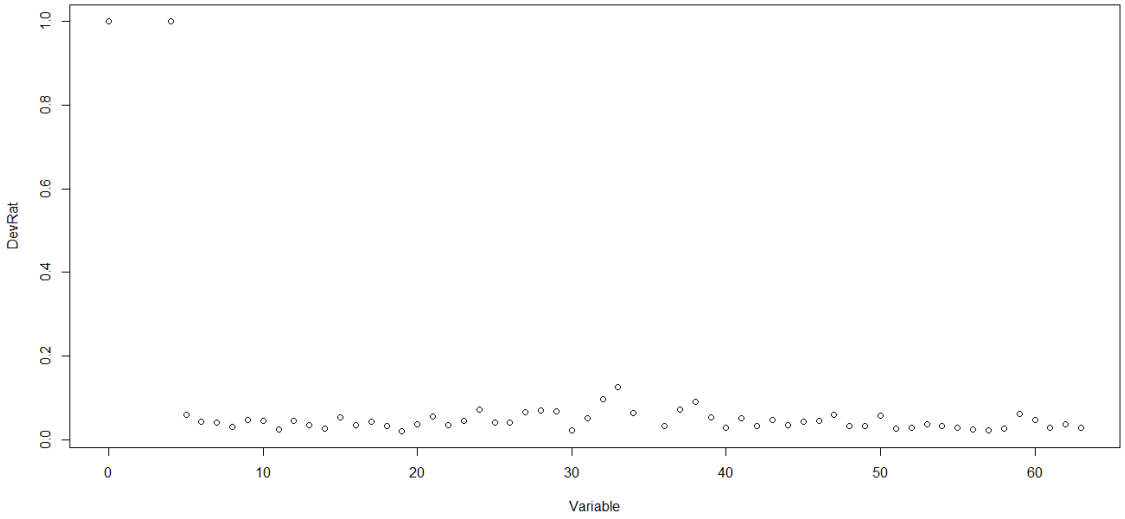


Figure 23. Reduction in deviance ratio per variable for the Splice data with additional V0 variable, a constructed dependence variable to V4

The deviance reduction ratios of the artificial variable V0 and V4 are very high compared to the deviance reduction ratios of the rest of the variables. Both are equal to 1, indicating that the leaves for both trees are pure and the trees perfectly predict their corresponding variables.

We see in Figure 24 that the tree created for V4 is based only on the V0 variable and has pure leaves. Similarly, the tree for the V0 variable is only split on V4 and also has pure leaves.

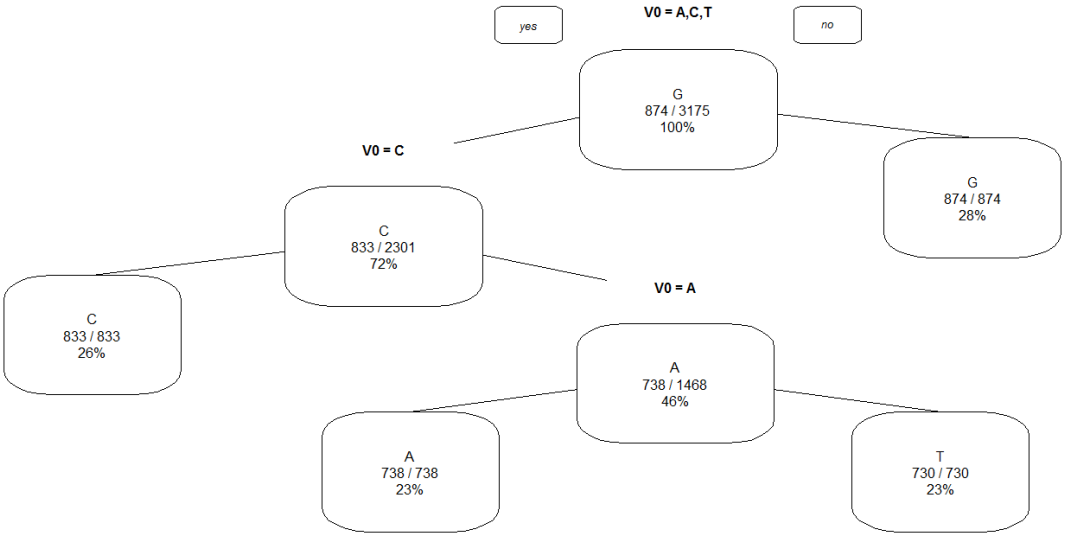


Figure 24. The V4 associated tree

V4-associated tree based only on the V0 variable

Each node representation consists of the most common level of the node, the ratio between the most common level of the node and the number of the observation in the node, and the percent representation of that ratio.

With the added artificial variable, the mapping based on d1 does not seem to change as shown in Figure 25, but the mapping based on d4 changes considerably as shown in Figure 26.

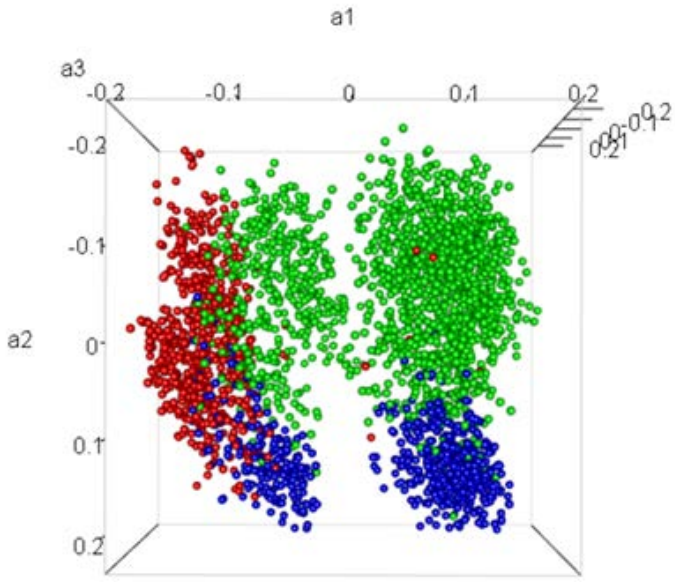


Figure 25. Splice with additional correlated variable V0 mapping based on d1 colored by Splice class levels

Legend “EI” – red, “IE” – blue, and “N” – green. The map is similar to the map of the original Splice data mapped by d1.

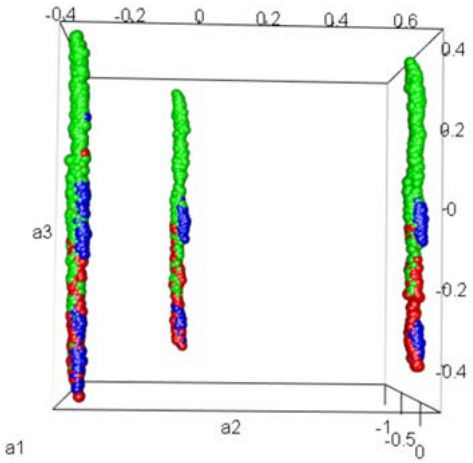


Figure 26. Splice with additional correlated variable V0 mapped with d4 colored by Splice class levels

Legend “EI” – red, “IE” – blue and “N” – green. The map is different from the map of the original Splice data mapped by d4.

There are three very distinct columns in Figure 26. The content of the columns is ordered internally according to the Splice levels. The greens are

higher on the a3 axis and the blues and reds are lower on the a3 axis. If we color the data by V4, we can see in Figure 27 that with the addition of the artificial variable V0, V4 is now the dominant variable in the data.

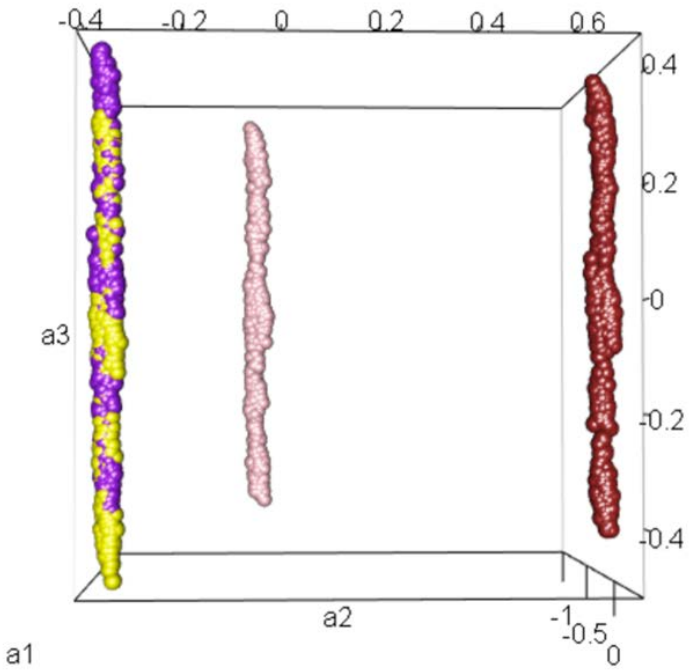


Figure 27. Splice with additional correlated variable V0 mapping based on d4 colored by V4 values

Legend "C" – brown, "A" – purple, "G" – pink, and "T" – yellow. V4 is the dominant variable of the visualization.

It is clear that the mapping is influenced severely by the dependence between V4 and the artificial variable.

3. Discussion

The fact that the d4 mapping is affected by introducing the artificial high-correlated variable, and the d1 mapping is not, means that at least in the Splice data, the weighting of each variable's contribution in d4 by the deviance causes the more severe problem. The fact that including V0 causes the V4 tree to depend only on V0, blocking V4's dependence on other variables, does not seem to be a problem when all variables are weighted equally as they are for d1. Even when choosing to experiment with more

important variables such as V33, the mapping based on d1 remains largely unaffected. Our hypothesis is that because there are many highly dependent variables within the Splice data and the data structure is revealed by other variables, “losing” information from the two variables corresponding to the trees that the algorithm removed variables does not result in a visible change in d1 dissimilarities.

The Splice data set is not unique in having these issues. We experiment with adding correlated variables to all other data sets we examine in this thesis: Iris, Credit, and Seeds. In Iris, adding a copied variable does not result in a large change in the mapping for all of the distances. Our hypothesis is because the initial deviance reduction ratio of all the variables is very close to 1 (it is always 0.62 or above), adding another high deviance reduction ratio tree does not significantly change the computation.

In Credit, there is a severe change for the d4 mapping, just as in Splice. Credit also has some change in the d1 mapping. Our hypothesis is the reason that d1 mapping changes for Credit and not for Splice is because there are not a lot of variables in Credit compared to Splice. Credit also has a constructed strong dependence in it without adding a correlated artificial variable. In Section 4.B, we discuss Credit maps in more detail.

4. How Does the Deviance Change as a Factor of the Amount of Correlation?

Most of the time, highly dependent variables are not perfect copies of each other. We want to understand the impact of non-identical but strongly dependent variables on the reduced deviance ratio in order to understand the impact on the mapping.

For Splice, we create another column that is highly dependent on V4. We copy V4 where a certain percentage of the values are randomly permuted. Permutation reduces the dependence between V4 and the constructed variable. While doing so, we keep the marginal distributions of V4 and the constructed variables the same. The percentage permuted is varied from 0 to 100%. After permuting, we grow the trees as usual and measure the reduction in the deviance ratio of the permuted variable. Figure 28 is a graph

of the reduction of the deviance ratio for the tree corresponding to the permuted variable versus the percent permuted.

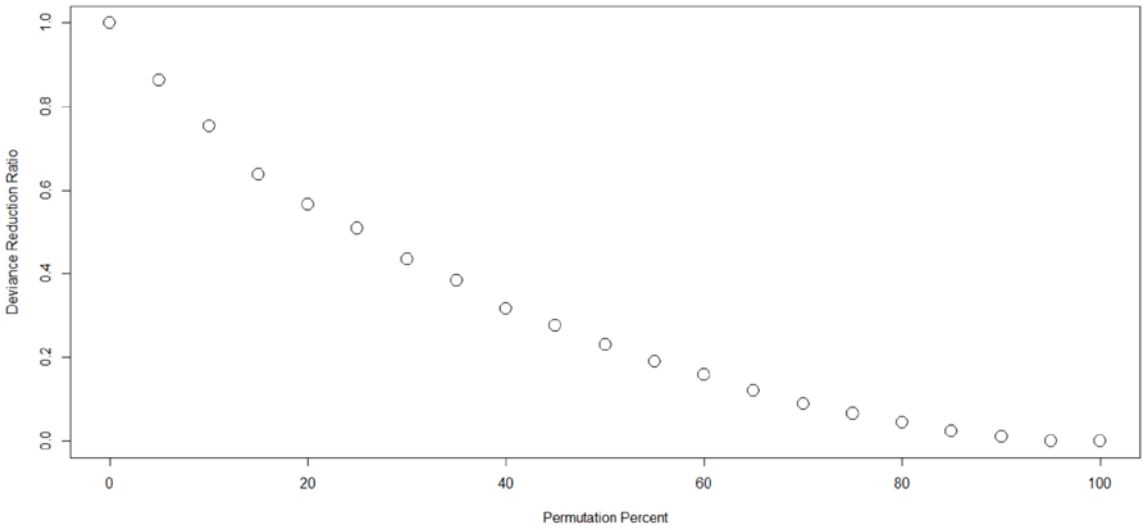


Figure 28. Reduction in R^2 analogy (Deviance Reduction Dev-Rat) for V4 as a function of the percent of permutation

In this example, the dependence must be quite strong before the R^2 analog suffers a severe impact. For example, 20% permuted results in a roughly 40% reduction of the deviance ratio. The graph in Figure 11 remains similar if we choose other, more important variables to permute. Permuting variables from all the other data sets we examined in this thesis results in the same behavior and similar numbers (between 35%–45% reduction) for the R^2 analog.

B. A PROPOSED SOLUTION

In this section, we discuss our proposed solution for the strong dependence issue. Our proposed solution is to replace the trees that split on only one variable and have an R^2 analog above a certain fixed threshold. The algorithm tries to grow a replacement tree based on all the variables except the one with the high correlation. If there is another highly correlated variable, the algorithm continues to replace trees until a tree without these properties is

found or no trees are found, in which case the variable does not have an associated tree.

1. Advantages and Disadvantages

The advantages of the proposed algorithm are:

1. There is no need for a separate procedure in order to find pairs of strongly dependent variables.
2. The solution can identify dependence between any combinations of variables—numerical and numerical, numerical and categorical, categorical and ordinal, and so on. If a variable has a high dependence with another variable, the tree prefers choosing that variable before the others, no matter the type.
3. The solution does not omit the variables' associated trees completely, but instead grows trees referring to the other variables, which could be important in revealing additional connections in the data.
4. If there are no strongly depended variables in the data set, the proposed solution and the current solution produce the exact same results.

There are several disadvantages of the proposed solution. First, the proposed algorithm cannot distinguish between constructed strong dependence and other forms of strong dependence. The proposed algorithm discards pairs of strongly dependent variables in the same way for all forms of dependence.

Second, if there are strongly dependent variables, the computation time of the algorithm increases because of the need to grow additional trees. We can calculate the new running time as follows: The algorithm for calculating distances for visualization has two stages, growing trees and calculating distances. The current algorithm grows *p* trees, where *p* is the number of variables in the data. There are *n* observations in the data. The runtime of growing one tree is $O(np^2)$ (Su 2006), so the total time of this part is $O(np^3)$. Finding the correct leaf for an observation in a tree is a constant time operation, and comparing *n* observations to each other is $O(n^2)$ in order to create the inter-point distances. Therefore, the total runtime of the current tree distance algorithm is $O(np^3 + n^2)$.

For the proposed solution in the worst case, the algorithm grows *p* trees. The runtime of growing the trees is $O(n\,\mathrm{p}^4)$. The calculation of the distances' runtime does not change. The runtime of the whole algorithm is $O(np^4 + n^2)$ compared to the $O(np^3 + n^2)$ of the current algorithm. Most of the time, the runtime of the algorithm is equal to the runtime of the current algorithm, and it increases only in cases of high dependence.

Third, the proposed solution addresses only strong dependence between two variables. It does not deal with a situation where a variable is strongly dependent on a linear combination of other variables, such as if one of the variables is the sum of two or more variables.

Fourth, the ideal threshold for the deviance ratio could be different for different data sets. In particular, this could happen if the threshold is set higher than the R^2 analog of a natural strong dependence variable in the data set.

2. Proposed Solution Experiments

In order to test the proposed algorithm, we run it with a comparison of the current algorithm on two data sets: the Credit data set and the artificial Splice data set with the extra copy of V4 variable. The Credit data set is a good example of a data set with natural strong dependence variables.

In all the experiments in this chapter, the chosen threshold for the deviance reduction ratio is 0.9. We choose 0.9 based on Figure 28 and the experiments resulting in permuting variables. Many data sets, such as Iris, have a deviance reduction ratio higher than 0.85 for non-constructed variables. However, the trees are a combination of several variables. A deviance reduction ratio of 0.9 and higher, in a tree with only one variable as the predictor variable, is a good indication of a constructed dependence. We conduct experiments on all the data sets that were used in this thesis with different thresholds, and our conclusion is that 0.9 balances the need to identify as much constructed dependence as possible (reduce the false-negative) without damaging the connections between true dependent variables (increase the false-positive).

3. Credit Data Set Experiment

Plotting the deviance ratio per variable for the original distances of the credit data in Figure 29 shows that some of the variables have high correlation. V4, V5, and V10 have a deviance reduction ratio larger than 0.97. Mapping d1 into three-dimensional Euclidean space gives Figure 30.

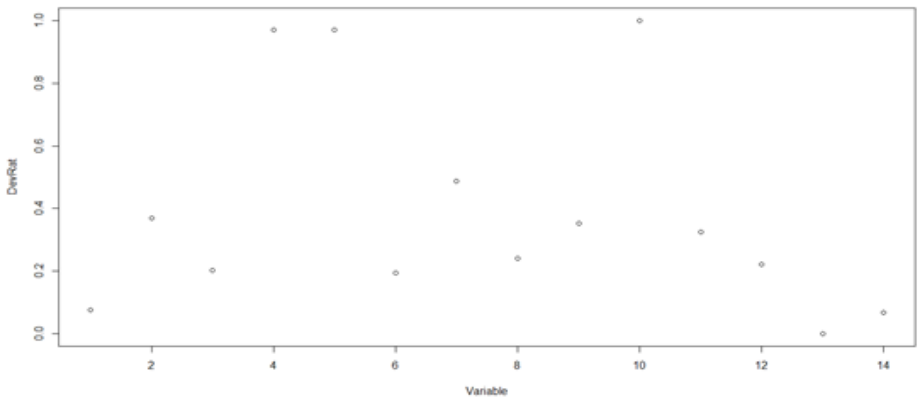


Figure 29. R^2 analog (DevRat) per variable of Credit data set using the current tree distance algorithm

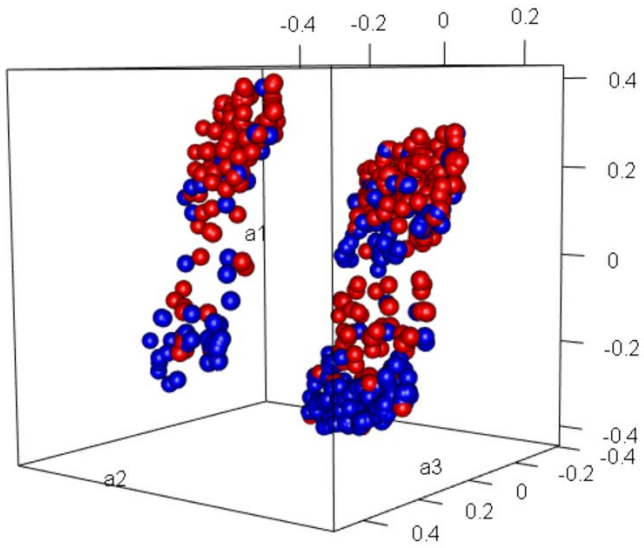


Figure 30. Mapping Credit using d1 of the current solution, colored by Credit class

Legend "+" – Blue and "-" – Red

The reader can see that the two levels of Credit class are spread among four clusters. It is arguable that in some of the clusters, there is some separation of the Credit class, but the separation is not optimal. Color-coding the map by some of the high deviance variables demonstrates the bias of the current tree distance algorithm, as shown in Figures 31 and 32.

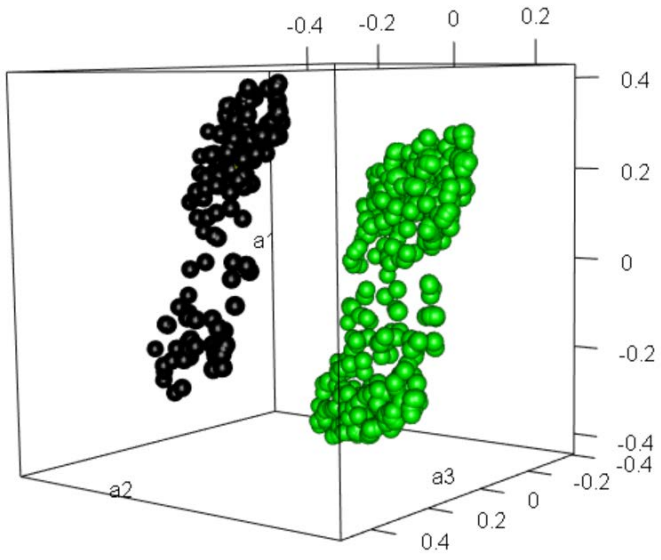


Figure 31. Mapping Credit using d1 of the current solution colored by V5

Legend “g” – green, “p” – black, and “gg” – yellow

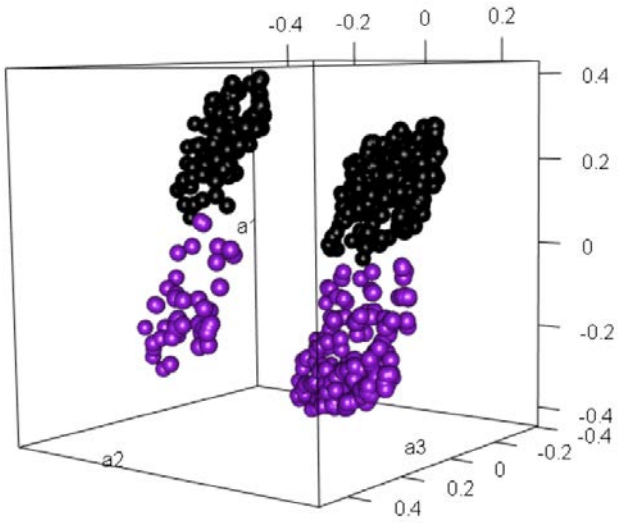


Figure 32. Mapping Credit using d1 of the current solution colored by V10

Legend "t" – purple and "f" – black

It is clear by viewing Figures 31 and 32 color-coded, respectively, by V5 and V10 that the mapping is split perfectly by V5 and V10 (and also V4 and V11, which are the strongly dependent variables of V5 and V10, respectively). We can conclude that tree distances d1 splits the Credit data by the highly correlated variables but does not perform well in separating the Credit class.

We now apply our proposed algorithm to the Credit data. The deviance reduction ratio plot in Figure 33 shows that there is no tree with a deviance reduction ratio higher than 0.45 for the new algorithm. The algorithm omits the strongly dependent trees of the variables V4, V5, and V10. For the variables V4 and V5 the algorithm has not found replacement trees, while it replaced the tree for V10 with a tree built on several variables: V6, V9, V14, and V15. The R^2 analog of the new tree is 0.24 instead of the original 1 R^2 analog's value.

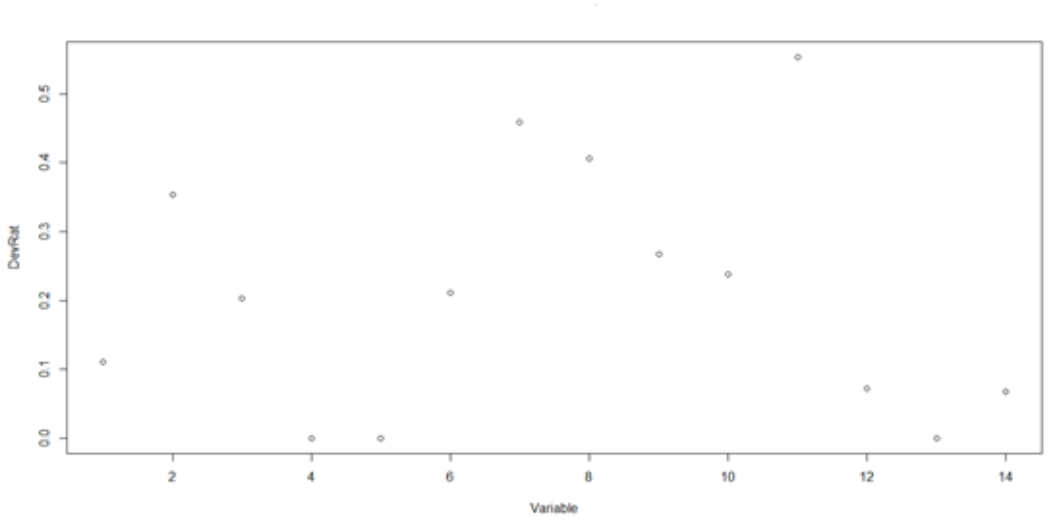


Figure 33. R^2 analog (DevRat) per variable of Credit data using the proposed algorithm

Figure 34 shows the 3D mapping based on d1 created by the proposed algorithm. The new mapping splits the data better than the original algorithm. At the negative a1 values, most of the observations are red from type "-," where at the positive a1 values, there are more blue observations from type "+."

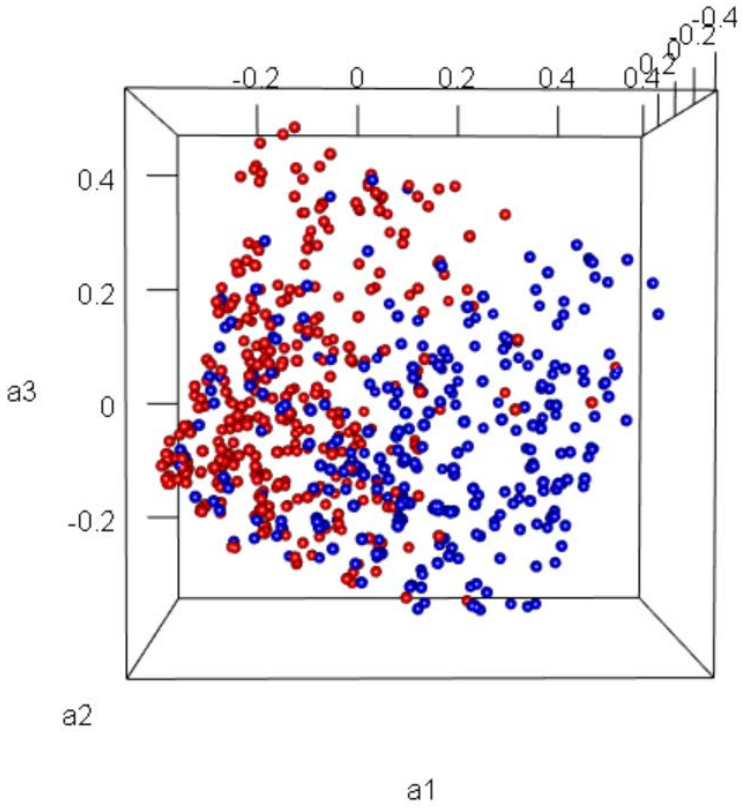


Figure 34. Mapping Credit using d1 of the proposed solution colored by Credit class

Legend "+" – Blue and "-" – Red

If we color-code the map by V5 as shown in Figure 35, there is no visible relationship between V5 and the observations' locations. If we color-code the map by V10 as shown in Figure 36, we find a relationship between the "t" value and the "+" class. This relationship seems to be a legitimate relationship in the data because the proposed algorithm does not completely remove the V10 tree, but replaces it with another one. Our proposed algorithm is successful in finding the hidden structure of the Credit data.

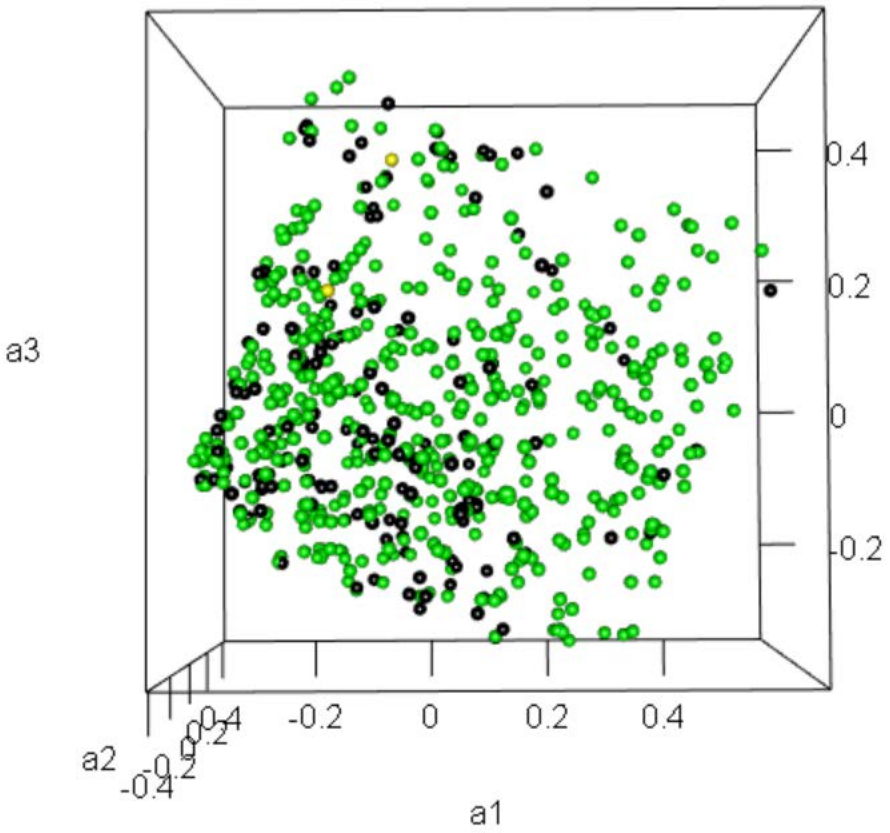


Figure 35. Mapping Credit using d1 of the proposed solution colored by V5

Legend “g” – green, “p” – black, and “gg” – yellow

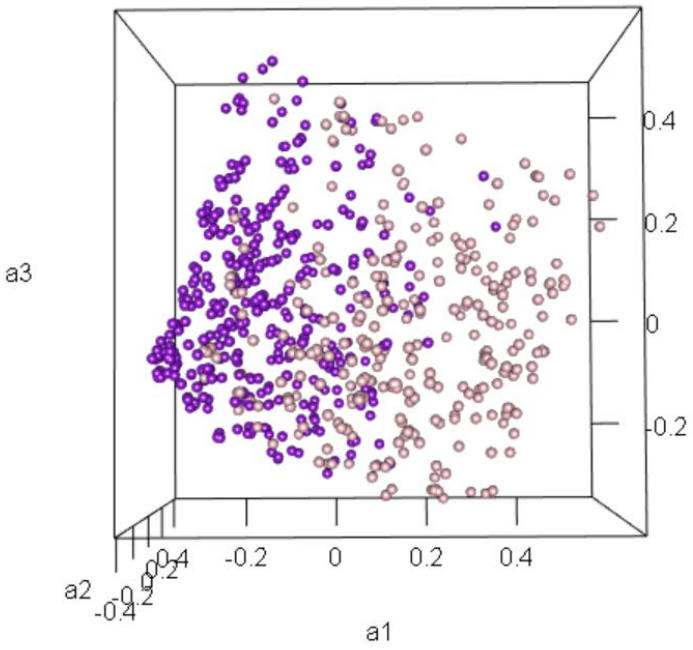


Figure 36. Mapping Credit using d1 of the proposed solution colored by V10

Legend “t” – purple and “f” – pink

4. Artificial Splice Data Set Experiment

Earlier in this chapter we introduced the Artificial Splice data set, which is the Splice data set with an additional column that is equal to the V4 variable. We demonstrated how the additional variable completely changes the mapping of d4 using the current algorithm, where the mapping is biased for V4. In this experiment, we run the proposed algorithm on the Artificial Splice data set. In Figure 37, the deviance reduction ratio of the proposed algorithm is plotted, and it is different than the one created by the current algorithm.

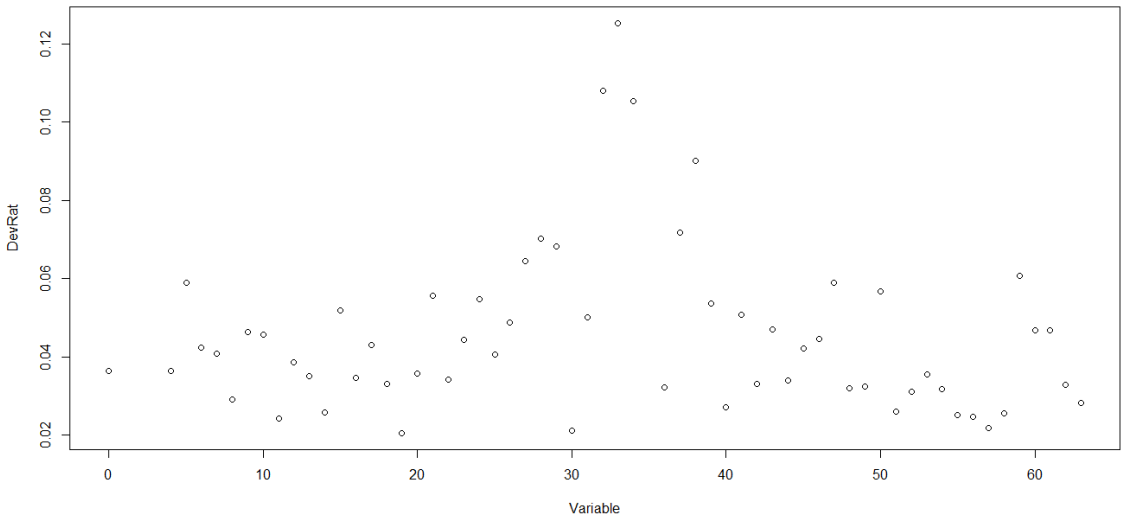


Figure 37. R^2 analog (DevRat) per variable of Artificial Splice Data using the proposed algorithm

The deviance reduction ratio plot does not contain any ratio above 0.15, while in the current algorithm there are two variables with a deviance reduction ratio of 1: V4 and its copy, V10. Mapping the distances measured by d4 of the proposed solution creates the images in Figure 38 and Figure 39, color-coded by Splice class and V4, respectively.

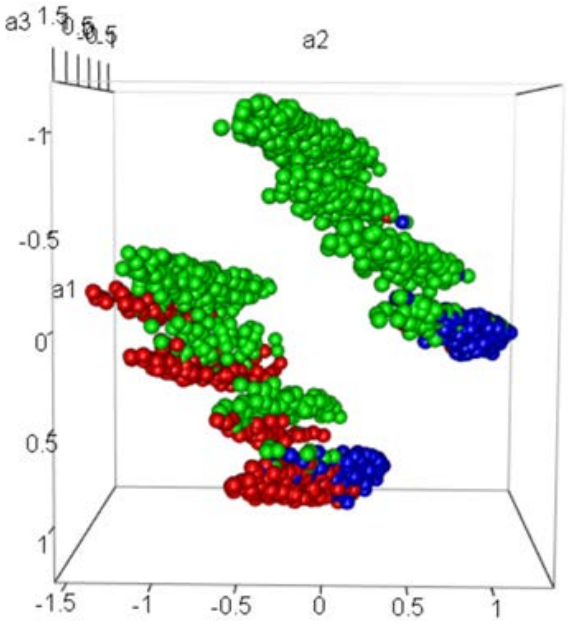


Figure 38. Mapping Artificial Splice data using d4 of the proposed solution, colored by Splice class

Legend “EI” – red, “IE” – blue, and “N” – green

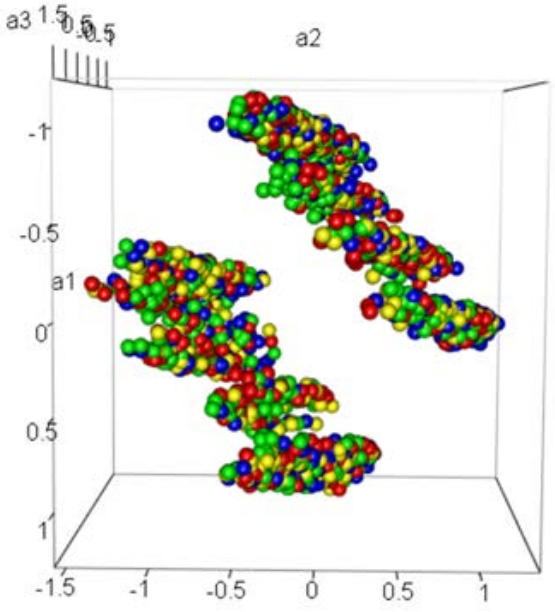


Figure 39. Mapping Artificial Splice data using d4 of the proposed solution, colored by V4

Legend “C” – blue, “A” – red, “G” – green, and “T” – yellow

It is clear that the proposed algorithm does not split the Splice data by V4, which indicates that it was not influenced by the strong dependence. Comparing the map of the proposed algorithm to the map of the current algorithm of the original Splice data, we can see that they are very similar to one another and the insights that can be gained from them are the same.

In summary, we can conclude that the proposed algorithm can treat a lot of the disadvantages of the problems associated with highly dependent variables while preserving the characteristics of the original mapping.

C. SUMMARY

In this chapter we reviewed the phenomena associated with strong dependence among variables. We discussed why the existence of constructed strong dependence in a data set can bias the tree distances. A few real data examples of the biased mapping were given.

We described a proposed solution to the problem and discussed its strengths and weaknesses. At the end, we demonstrated how the proposed solution deals with the problem while having negligible impact on the mapping.

THIS PAGE INTENTIONALLY LEFT BLANK

V. DISCUSSION

In this chapter, we discuss several attributes and characteristics of the tree distance visualization method. Some of these are also attributes and characteristics of tree distances, which are easy to detect and understand using visualization. Section A of this chapter introduces two prominent patterns that occur often: the “collapsing” tendency and the “snake” shape. We discuss in Section A the collapsing tendency, its implications, and benefits for an analyst who explores the data set in search of clusters. We discuss the reasons for the snake shape and the insights that can be gained from it. Section B discusses the different properties of the variants, d1–d4, and their influence on the mappings.

In this chapter, all the figures created by using tree distance visualization were created without the addition of artificial noise.

A. CLUSTER PROPERTIES OF THE TREE DISTANCE ALGORITHM

In this section, we discuss several properties of the tree distances that make them a good tool for analyzing data from a clustering perspective. Subsection A.1 discusses the tendency of observations to collapse to a single point and includes a discussion about dimensionality reduction and outlier treatment. This property enables an analyst to capture the high-level structure of the clusters’ relationships in the data set while ignoring unnecessary information about the variability of the data for that level of analysis. Subsection A.2 discusses the nearly-equal distances phenomenon an analyst should be aware of because of its influence on the order and configuration of the observations in the mapping. Finally, Subsection A.3 discusses the snake shape phenomenon that occurs in numerical data sets, which leads to a quick understanding of numerical data sets that do not consist of clear clusters.

Tree distance visualization is a suitable tool for a clustering analysis task. The main reason for this is the way the trees partition the data into large groups using leaves. Distances are measured in respect to the relationships between the leaves of the tree, not the specific observations. Therefore, the

tree distance visualization shows the connections between the different groups of the data, which may be identified as the connections between clusters or sub-clusters of the data.

We use the Iris data set, the Seeds data set, and two artificial data sets in order to discuss these properties. The Iris and Seeds data sets are described in Chapter II. The two artificial data sets are created in order to emphasize the collapsing tendency and the equal distance property. The two artificial data sets are described as follows: The first, or the original artificial data set, is a two-dimensional data set that consists of two clusters, each of 30 observations. The observations for each cluster are sampled from a multivariate normal distribution. The blue cluster's observations values are sampled from a bivariate Normal distribution with means 3, variance 1, and correlation 0, while the red cluster's observations values are sampled from the bivariate Normal distribution with means 20, variance 2, and correlation 0. Figure 40 shows the data set plot in the original space. The second artificial data set, or the new artificial data set, consists of another cluster, the green cluster, sampled from the bivariate normal distribution with means 10, variance 0.5, and correlation 0. Figure 45 shows the new artificial data set.

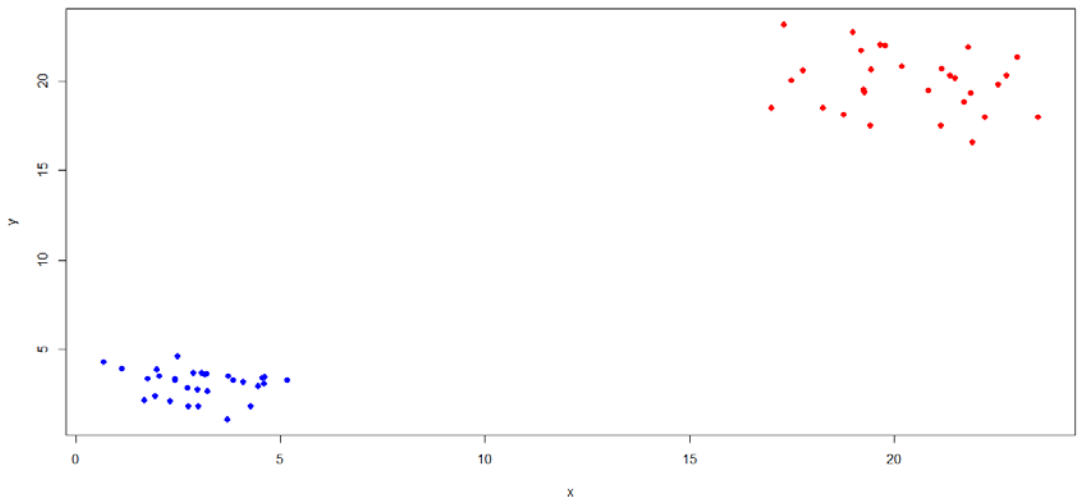


Figure 40. The artificial data set colored by the type

1. Observations' Collapsing Tendency

In this section, we discuss the tendency of similar observations to collapse into a single point in the mapping of the tree distance visualization in certain scenarios with certain data structures. This tendency results in possible dimensionality and variability reduction and outliers' removal from the mapping. All these phenomena can help the analyst identify the cluster structure of the data set.

By definition, the dissimilarity between two observations that fall in the same leaves for all the trees is zero. The leaves represent different partitions of the data, each by its associated variable. If the original data consists of groups of observations that share similar properties and differ in their properties from other groups of observations, the trees grow leaves according to the groups. Therefore, similar observations fall into the same leaves, and their distance from one another is zero. This means that although there is variability among these observations in several dimensions in the original data set, the distances measured in the mapping of the tree distance visualization are all zero. Another consequence of observations falling in the same leaf is that their distance from other observations that do not fall in the same leaf is equal for all the observations in the leaf. This happens because they all share the same leaf, and the dissimilarities are measured with respect to the leaves' relationships. The distance is measured by the leaves' relationship, and all the observations with zero dissimilarities are mapped, or "collapse," to the same location in the new lower-dimensional Euclidean space. The stress is minimized for configuration where all the points with zero dissimilarities among them are mapped to a single point in the new space.

We demonstrate this tendency by examining the original artificial data set and the Iris data set. For the artificial data set, the observations are sampled from multivariate normal distributions, and therefore they are each different, even in the same group. For example, the ranges of observations for the blue group are 0.67–5.15 and 1.1–4.62 for the x and y axes, respectively. There is dependence among the axes of the artificial data set in the original space. Because of the combination of the two clusters in one data set, high

values of the x variable correspond with high values of the y variable in the gross level. The trees grown to compute tree distances on the artificial data set are shown in Figure 41.

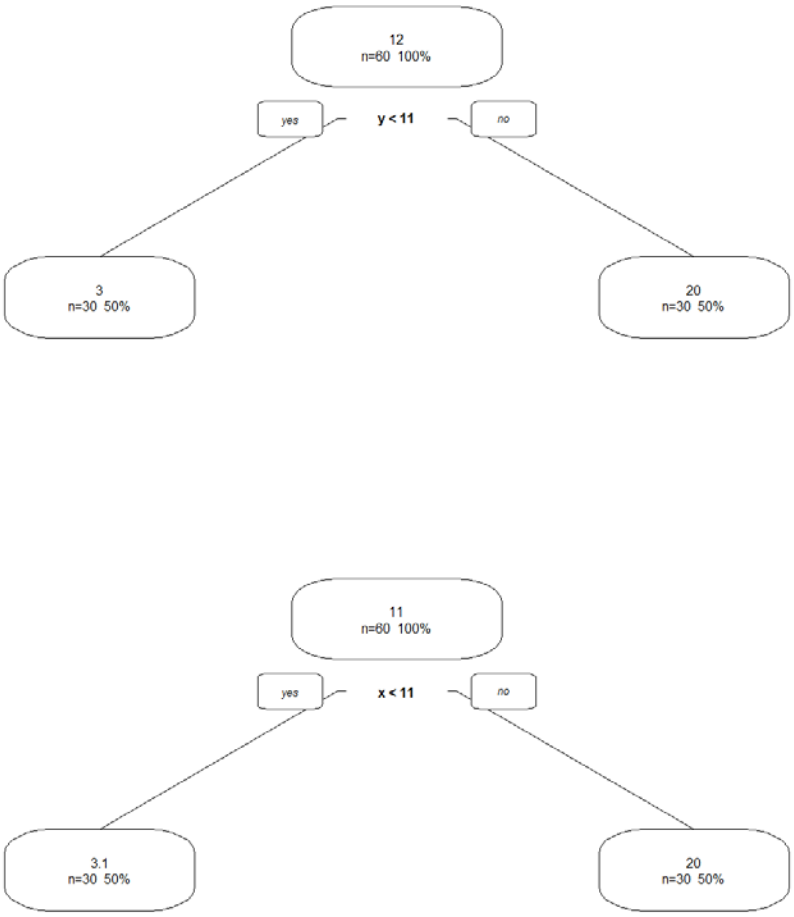


Figure 41. The trees created to compute tree distances algorithm on the artificial data set

Each tree partitions the space into two regions with each partition splitting the data set into the same subsets. When measuring the distance using the tree distance algorithm, all the blue observations fall in the same leaves (the leaves to the left in Figure 2) and all the red observations fall in the opposing leaves. Therefore, for all the different variants, d1–d4, the

dissimilarities among the observations that share the same group are zero, and their dissimilarities from the observations in the different group are constant but different for each tree distances variant. The visualization of the artificial data set using d1 is shown in Figure 42. The figure consists of only two distinct points, one for the red points and one for the blue points. Each of the points consists of 30 observations; each of the 30 has zero distance to any of the others.

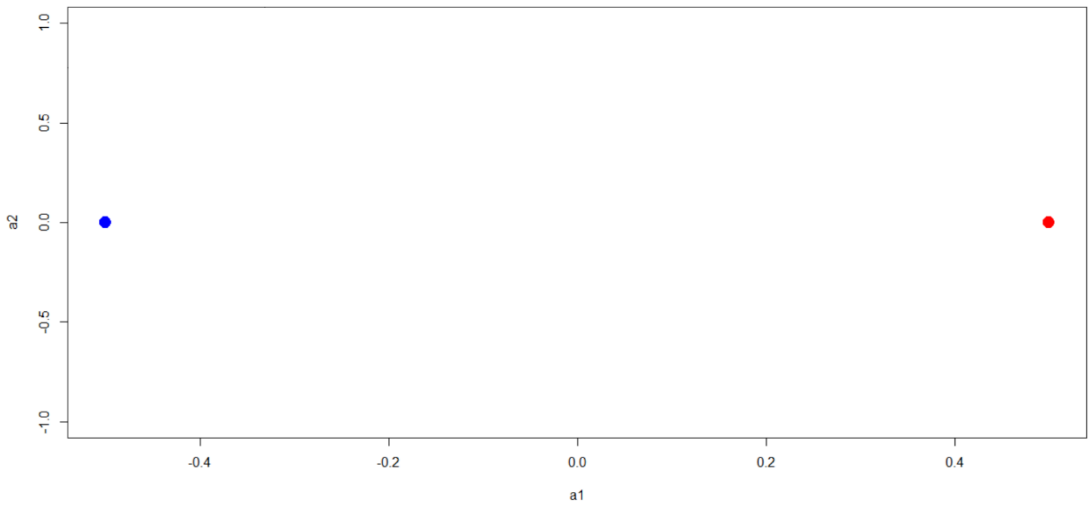


Figure 42. Artificial data set mapping using d1

An analyst examining Figure 39 can understand that the data consists of two separated clusters without getting into the variability details of the clusters. Observations in the artificial data sets lay in the two-dimensional Euclidean space. The configuration of the data using tree distance visualization can be considered one-dimensional mapping. All the “a2” values of the new configuration equal zero, and there are only non-zero values in one dimension, the “a1” dimension. The tree distance visualization reduces the dimensionality of the artificial data set from two to one dimension. This tendency does not depend on the number of original dimensions. If the artificial data set’s observations were sampled from a higher dimensional multivariate normal distribution (e.g., 20-dimensional) with sufficient separation for the trees to capture, the tree distance visualization would be

the same as Figure 42. The analyst's insights about the data are the same, regardless of the original dimensions—there are still two clusters in the data. This means that if the structure of the clusters in the data allows, the tree distance visualization reduces the dimensionality of the original data set.

We can also demonstrate the collapsing property on a non-artificial data set, the Iris data set. Figure 43 is a matrix of all pairs of scatter plots of the data set plotted one against the other, colored by the Iris class.

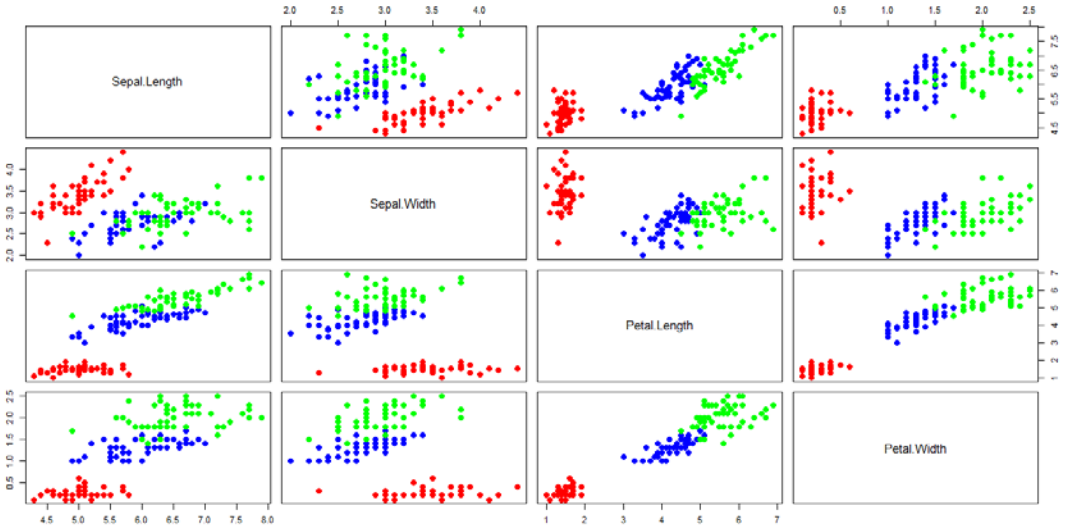


Figure 43. Iris data set, colored by the Iris class.

Legend: Setosa – red, Versicolor – blue, and Virginica – green

Clearly, the Setosa observations are separated from those of the other classes. For example, only the Setosa observations have Petal Length smaller than two. Figure 44 shows the mapping of the Iris data using d1. The Setosa observations do not collapse to a single point in space, but instead to three points. Each of them consists of more than 10 distinct observations. The partitioning of the space is not perfect for the Setosa class, but it is close to perfect, where all the observations for that class have been collapsed to one out of three distinct and close points.

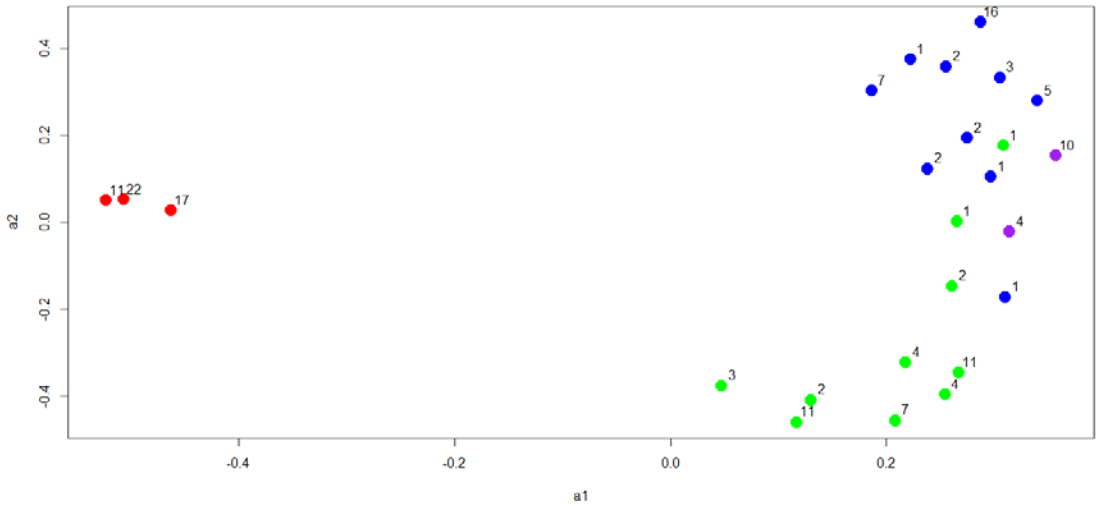


Figure 44. Iris data mapping using d1

Legend: Setosa – red, Versicolor – blue, Virginica – green, and "Mix classes of Versicolor and Virginica" – purple

The number of observations in each point is attached to the point.

For the Versicolor and the Virginica classes, the observations are more spread, although they still collapsed to several points in space. It is visible that there are quite separate Versicolor and Virginica regions in space. There are observations lying between the Virginica and Versicolor regions. These observations do not fall exclusively in leaves shared with one of the classes, but with both. The distance between two points in space is a function of the number of leaves they share. Therefore, the points between the regions represent observations that fall in some leaves with Versicolor observations and some with Virginica observations. Some of the observations have the same dissimilarity from the cluster's observations, but they differ from each other by the specific leaves they share with the cluster. Therefore, the dissimilarities between them are not equal to zero, and they are mapped to different locations with similar distance to the cluster's observations. In summary, an analyst can deduce that there is not a clear separation between the Versicolor and the Virginica classes.

The collapsing tendency can position outliers with other observations. If outliers are present in the data, they belong to leaves containing other observations. Therefore, the mapping of the data typically does not explicitly

show outliers and allows the analyst to focus on the relationships between the different clusters.

2. Nearly-Equal Distances

The clusters in tree distance visualization mappings tend to be distant one from another by a nearly equal distance. This property can deform the order of clusters in a data set's mapping, and it helps explain the snake shape phenomenon. For d1, the distance between two observations is proportional to the number of leaves they do not share with each other. This implies that for a set of clusters that do not share any leaves with each other, the dissimilarities measured among them are equal for all pairs of clusters. The reason is the number of leaves they do not share with each other is the maximum number of leaves, which is proportional to the number of trees or variables. This phenomenon occurs regardless of the original configuration and relationships between the observations. Figure 45 shows the new artificial data set, which is the original data set with an additional green cluster between the original two clusters. The blue observations are closer to the green observations than to the red observations in both dimensions. Figure 46 shows the trees created to compute the tree distances on the new artificial data set.

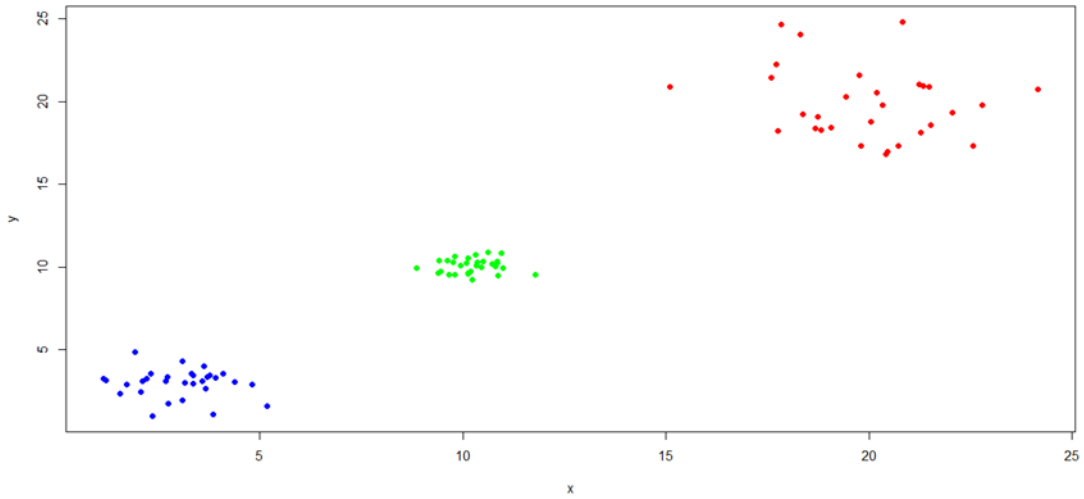


Figure 45. The new artificial data set consisting of an additional green cluster

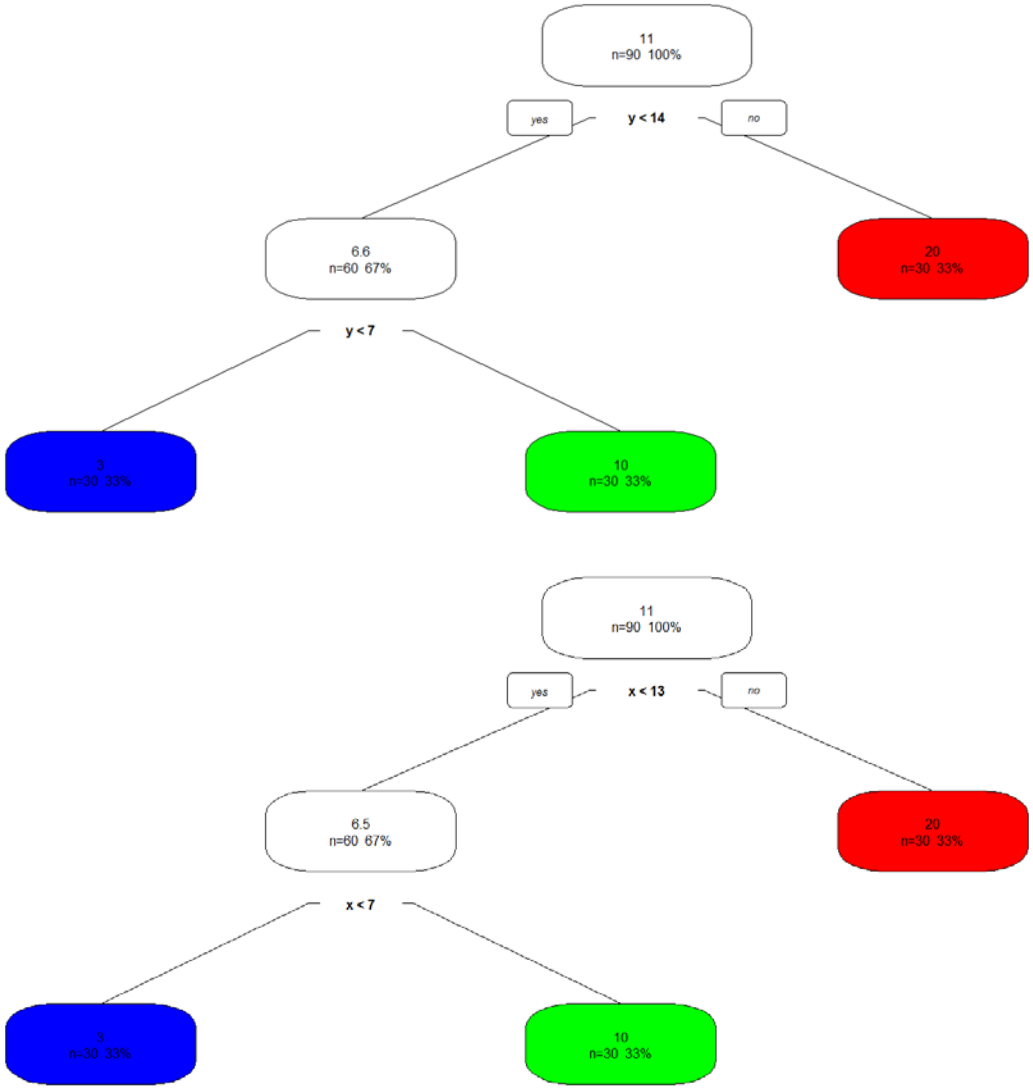


Figure 46. The trees created for tree distances on the new artificial data set; each leaf is colored by the corresponding cluster

All of the observations are split among the three leaves according to their cluster because the trees are able to differentiate between the clusters. Figure 47 shows the visualization of the new artificial data set using d1.

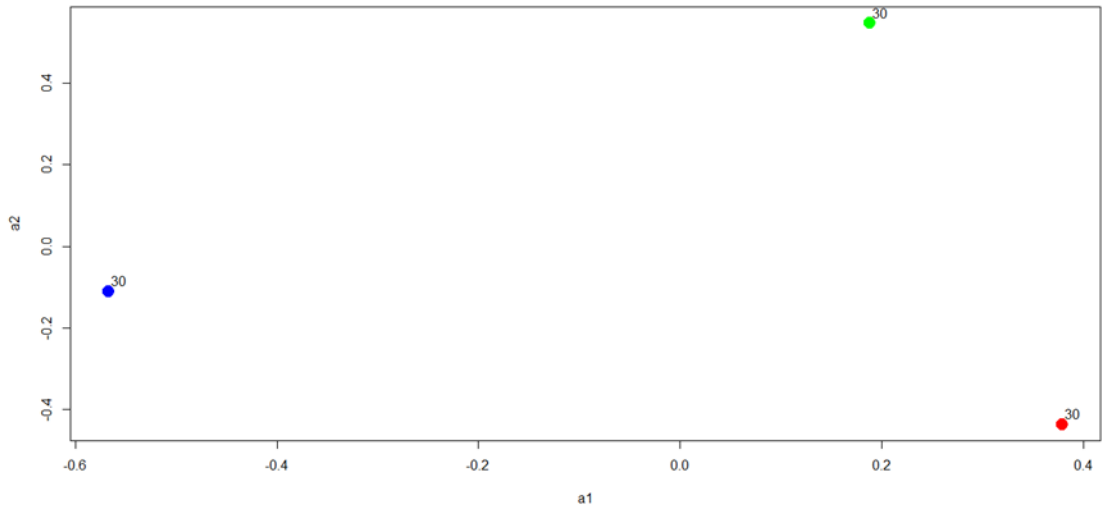


Figure 47. The new artificial data set mapping using d1
The number of observations in each point is attached to the point.

The new mapping consists of three distinct points that represent the three different clusters of the data set. The distances among the three points are equal to each other. This is because they represent the maximum separation between points in this configuration; they are different in both of the trees. Therefore, the original ordering of the data is not preserved. The blue cluster is as close to the red cluster in Figure 47 as it is to the green cluster. Another example of the equal distances property can be seen in the Iris data. For the irises' petal length, the Setosa are closer to the Versicolor than to the "Virginica," as can be observed in Figure 45. In the mapping in the new low-dimensional Euclidean space presented in Figure 47, it can be seen that the average distances between the different clusters are equal.

An analyst who researches the mapping of the new artificial data set in Figure 47 can determine that there are three different groups in the dataset. The researcher cannot deduce the ordering of the clusters in the original space, but she or he can deduce that there are three different groups, no matter the dimensionality of the original data set.

The phenomenon of equal distances happens for d1 because the distance measured between two observations that do not share the same leaf

in a specific tree is one, regardless of the deviance reduction ratio and the relationship among the leaves. For the rest of the tree distances variants, d2–d4, the dissimilarities among the different clusters can be different as a result of the different measurements. However, for some of the data sets we examined, the configuration is not different for the different variants of the tree distances, and therefore there the phenomenon of almost equal distances occurs also for those variants in these specific scenarios. We expand the discussion of the similarities and differences between the different tree distances variants in Section B.

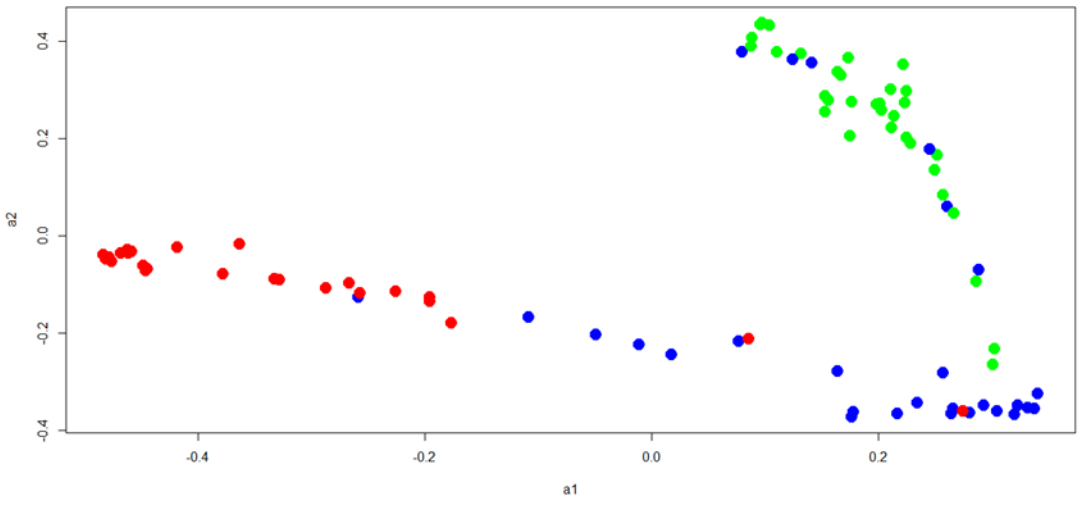
3. "Snake" Shape Mapping

Some of the numeric data set mappings using tree distances have a "snake" shape, which is a not simple cluster mapping. This snake shape is similar to what Hastie et al. (2009, 595) call a "star" shape for proximity plots of random forests (Brieman 2001). In the tree distance visualization mapping, the snake shape represents the change in numeric variables. Because the tree distances exploit dependencies in the data set, the snake shape represents the change in more than one numeric variable. The snake shape phenomenon occurs mostly when there is a monotonic relationship among variables for which the observations are not separated into clusters.

Our hypothesis for the cause of the snake shape phenomenon is as follows: The associated tree for a variable partitions the continuous variables into small pieces that represent different partitions of the data. Because there can be several variables in a data set, there is a reasonable possibility that the partitioning of the variables do not exactly fit one another. Therefore, there are small values of dissimilarities between nearby points because once in a while they fall in different leaves. These small dissimilarities create an order in the dissimilarities' set because they occur between close points in the original data set's variables. When CMDS minimizes the stress, a configuration where these points are adjacent to each other will have a low stress compares to a configuration where they are separated. Observations that are far from one another tend to have the maximum distance between them, as discussed in Subsection A.2. Therefore, the mapping tends to have a spherical or a

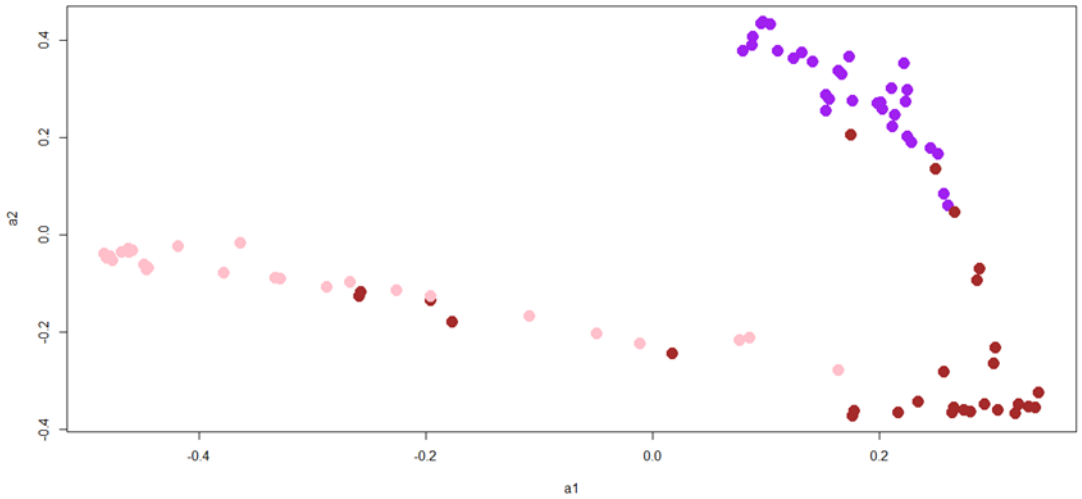
horseshoe-like shape because this shape needed to maintain equal distances. The shape is not a perfect horseshoe, but has more sharp edges. As mentioned before, the maximum dissimilarity between two observations in the configuration depends on the number of trees in the data because the largest dissimilarity occurs often for observations that fall in different leaves across all the trees. For observations having small dissimilarities to a pair of observations with the maximum dissimilarity between them, there is a combinatorial number of possible positions in the configuration. The reason is that there is a combinatorial number of possibilities for the relationship among the leaves. Therefore, in lower-dimensional Euclidean space, there is a problem fitting the observations between pairs of observations with the maximum dissimilarity. This, we assume, forces the CMDS to position the observations not in a perfect spherical shape, but instead to relax the stress by creating sharp edges.

Figure 48 is the mapping of the Seeds data set using d1, with part (a) colored by the Seeds class, part (b) colored by V1, and part (c) colored by V2. The variables and cuts were chosen by the maximum deviance ratio and pruning methods (see Chapter III).



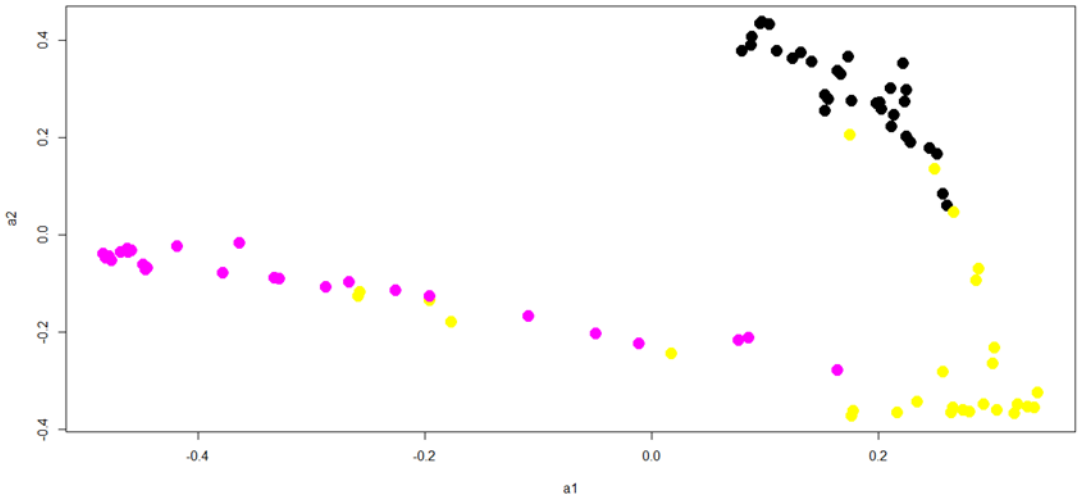
(a)

Legend “1” – blue, “2” – red, and “3” green



(b)

Legend (10.6-12.8] – purple, (12.8-15.6] – brown, and (15.6, 21.2] – pink



(c)

Figure 48. Seeds data set mapping using d1 colored by (a) Seed class, (b) V1, and (c) V2.

Legend (12.4-13.7] – black, (13.7-14.9] – yellow, and (14.9, 17.3] – magenta

The values of both V1 and V2 increase generally in a clockwise direction in the configuration when starting at the top observations (high values of a2). The Seeds classes are generally aligned with the cuts of both of

the variables. Figure 49 shows the monotonic dependence between V1 and V2, which contributes to the snake shape of the mapping.

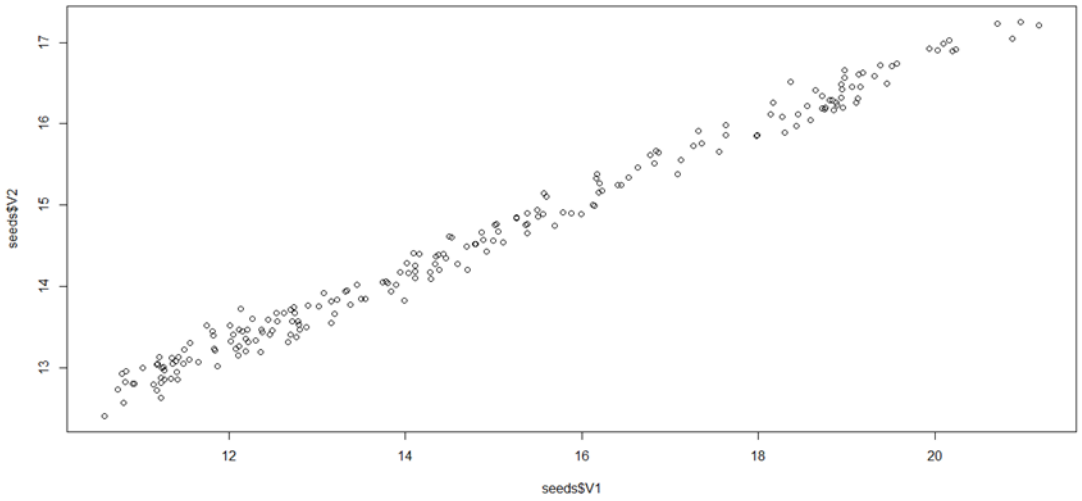


Figure 49. The monotonic increasing dependence between V1 and V2 in the Seeds data set

B. THE VARIANTS' INFLUENCE ON THE MAPPING

Buttrey and Whitaker (2015a) compare the performances of the different variants of tree distances. Their conclusion is that there is no clear best performer for all the data sets, but d2 in general performs well using the partitioning around medoids (PAM) algorithm (Kaufman and Rousseeuw 1990), for clustering. Visualization can help explain some of the phenomena of the tree distances. In Subsection C.1, we use the Splice data as an example to discuss the differences in the mappings for the different variants. Our main conclusion is that for the Splice data set, all the variants except d1 emphasize the role of a certain variable in expense of other variables. In Subsection C.2, we discuss why the mappings are similar for different variants in some of the data sets despite the differences in the dissimilarity calculations.

1. Splice Mappings Using the Different Variants

In this subsection, we demonstrate the influence of the variant selection on the Splice data set, which has been described in Chapter II. We use Splice's results in order to discuss some of the differences in the different variants. We also discuss Buttrey and Whitaker's (2015a) findings on the best performance of d4 in the clustering task of the Splice data set.

Figure 50 shows Splice mapping using d1 into three dimensions and colored by the Splice class's levels. It is clear that except for a few observations, there are clear partitions in the mapping by the Splice class. The partitions are clear and pure (see definition in Chapter III), but there is no space between the different groups. They are visible only by coloring the mapping.

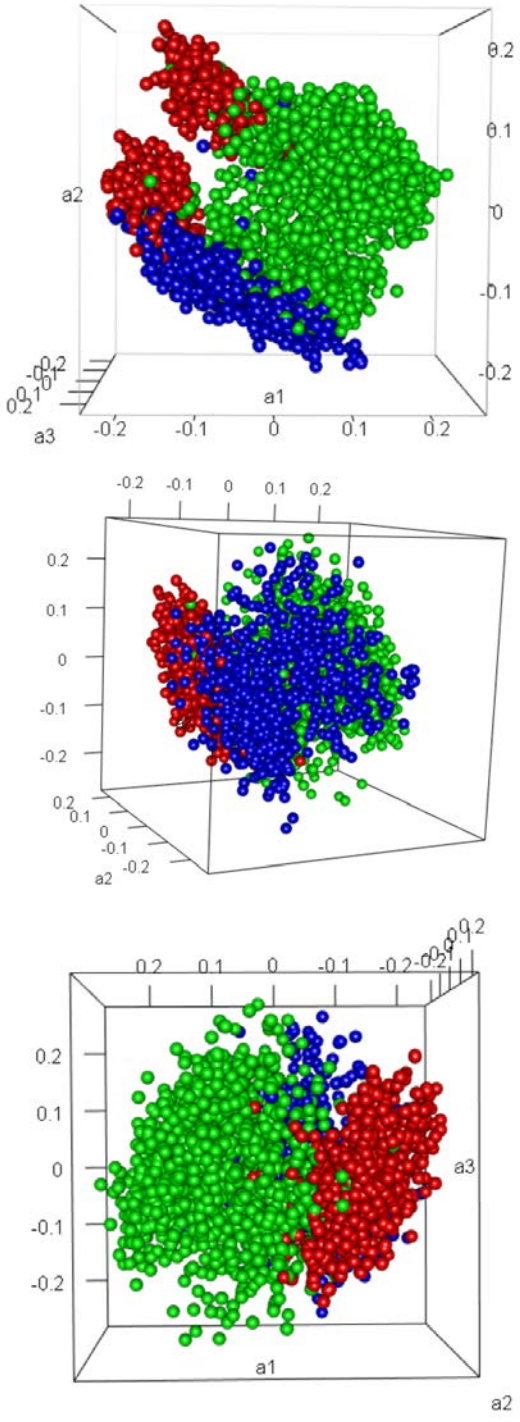


Figure 50. Splice data set mapping using d1 colored by Splice class

Legend “EI” – red, “IE” – blue, and “N” – green

Buttrey and Whitaker (2015a) compare the clustering performance of different clustering algorithms on the different variants. Figure 51 shows the

Splice mapping using d1, now colored by the clusters assigned by the PAM algorithm on the original dissimilarities. It is important to notice that the mapping is in three dimensions and the original dissimilarities are in a much higher dimensional space (the dimension is given by the sum of the number of leaves across all the trees for the data). Therefore, there is a difference in the distances in the mapping and the original dissimilarities that the algorithms use (which is expressed by the stress value). This can result in a less straight-forward behavior of the clustering algorithms' results in the mappings than desired.

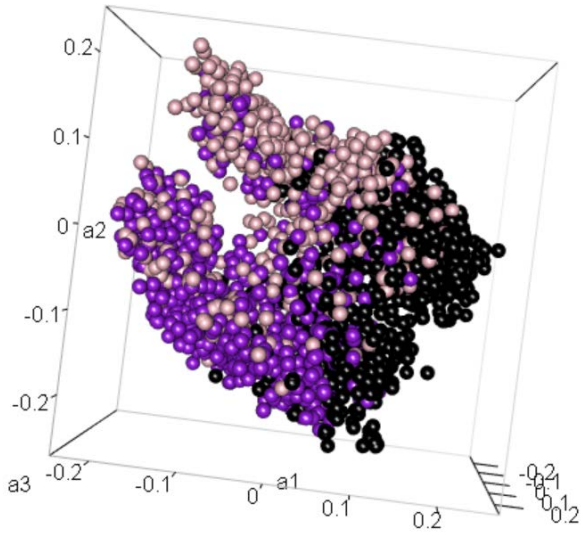


Figure 51. Splice data set mapping using d1 colored by the PAM algorithm clusters results

By comparing Figures 50 and 51, it is seen that the clusters created by the PAM algorithm do not fit the Splice class. The d1 variant is the poorest variant using the PAM algorithm. Figures 50 and 51's differences shed some light on the reasons for the poor clustering results.

Figure 52 shows the Splice mapping using d2 colored by the Splice class. The figure is similar to Figure 16 of the d1 mapping. The d2 variant is different from d1 in the weighting of each tree contribution to the dissimilarities

calculation (see Chapter II). The weighting is determined by the R^2 analog. Figure 18 in Chapter III shows the R^2 analog for the Splice data set. There is no visible difference between the mappings of d1 and d2, despite the different weighting.

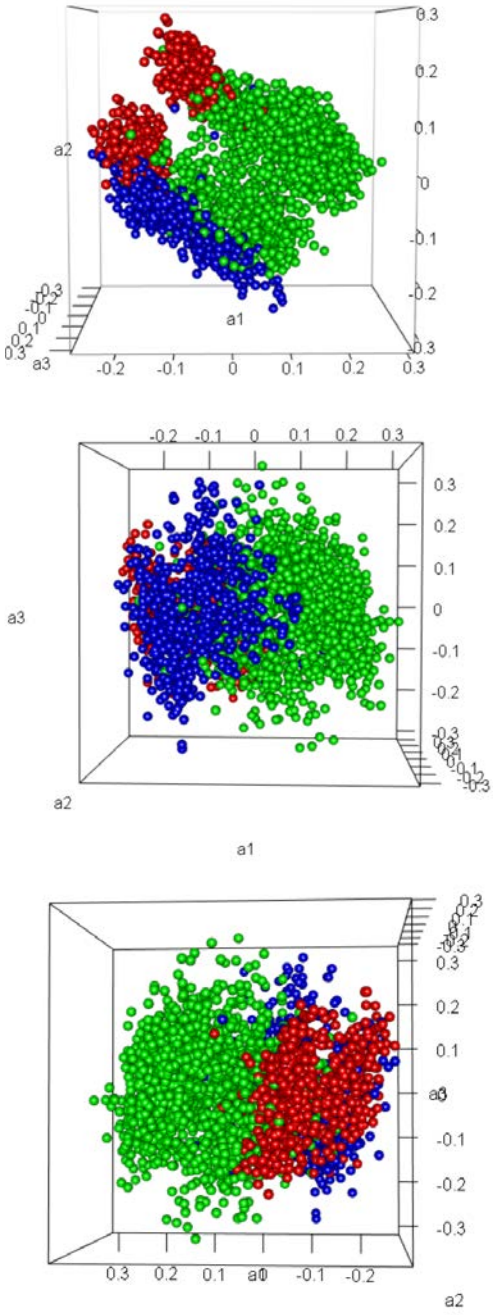


Figure 52. Splice data set mapping using d2 colored by Splice class

Legend "EI" – red, "IE" – blue, and "N" – green

Figure 53 shows the Splice mapping using d3. Figure 53 is similar to Figures 50 and 52 with one noticeable difference. The data forms two observable clusters in Figure 53. These clusters exist also in Figures 50 and 53, but in those figures, they are much less separated. Coloring the mapping by the V35 variable of the Splice data set produces Figure 54. It is clear that the separation of the two clusters is driven by the values of V35. One cluster consists only of the “T” level’s observations of V35, while the other clusters consist of all the rest of the levels. It is also clear that the separation is not related directly to the Splice class levels, except for the fact that one of the clusters has many more “IE” observations.

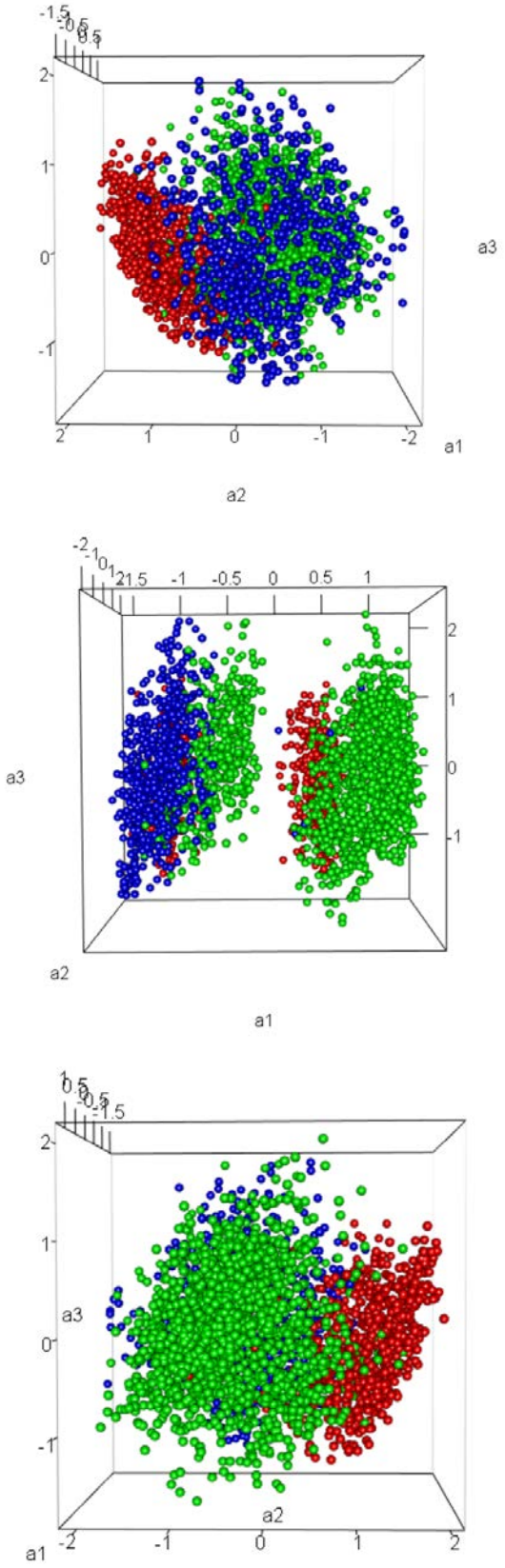


Figure 53. Splice data set mapping using d3 colored by Splice class

Legend “EI” – red, “IE” – blue, and “N” – green

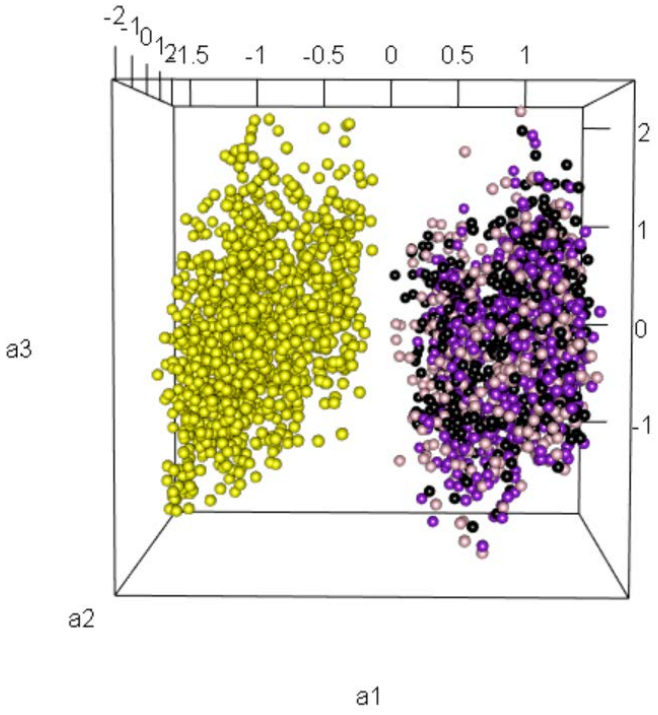


Figure 54. Splice data set mapping using d3 colored by V35

Legend "T" – yellow, "C" – black, "G" – pink, and "A" – purple

The reason for the separation is the difference in the calculation of the dissimilarity's value, $d^t(i, j)$, for each tree. $d^t(i, j)$ for d3 is calculated based on the relationship between the leaves $L_t(\mathrm{i})$ and $L_t(\mathrm{j})$ in the tree. The farther they are, meaning the larger the deviance reduction ratio of their shared parent, the larger the difference between them. From Figure 51, it seems that level "T" of the V35 variable's observations fall far from the other levels. It is important to notice that the effect of d3 is also a result of the goodness of the tree, its R^2 analog. The difference between the leaves calculated by $d^t(i, j)$ is calculated using the deviance reduction ratio inside the tree. The larger the deviance is reduced for the whole tree (larger R^2 analog), the higher the possibility of a large deviance change for specific nodes in the tree.

Figure 55 shows the d3 mapping colored by the PAM algorithm's clusters. It seems that the PAM algorithm splits the data roughly by the spatial properties of the mapping. Comparing Figures 53 and 55, an analyst can

identify better correlation between the PAM's results and the Splice class than for d1, which corresponds to the results of Buttrey and Whitaker (2015a).

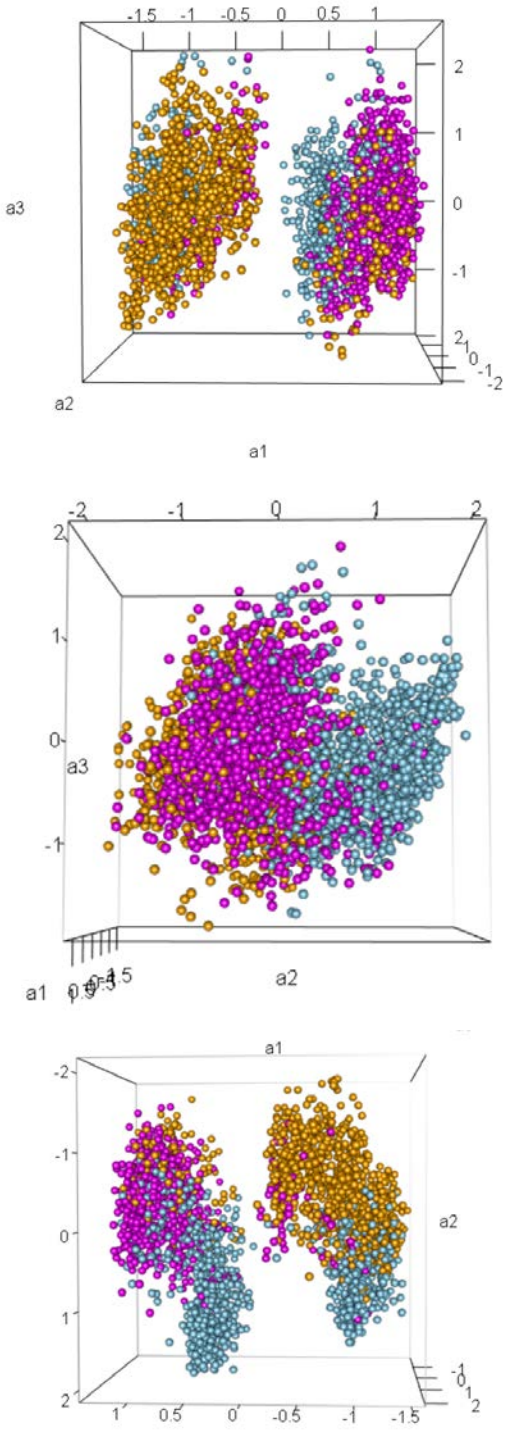


Figure 55. Splice data set mapping using d3 colored by the PAM algorithm clusters results

Figure 56 shows the Splice data set mapping using d4 colored by the Splice class. The mapping is split into eight columns, which are grouped roughly into four clusters. The order of the observations is similar to the order in the rest of the mappings. Figures 57 and 58 show the d4 mapping colored by V35 and V32 variables, respectively.

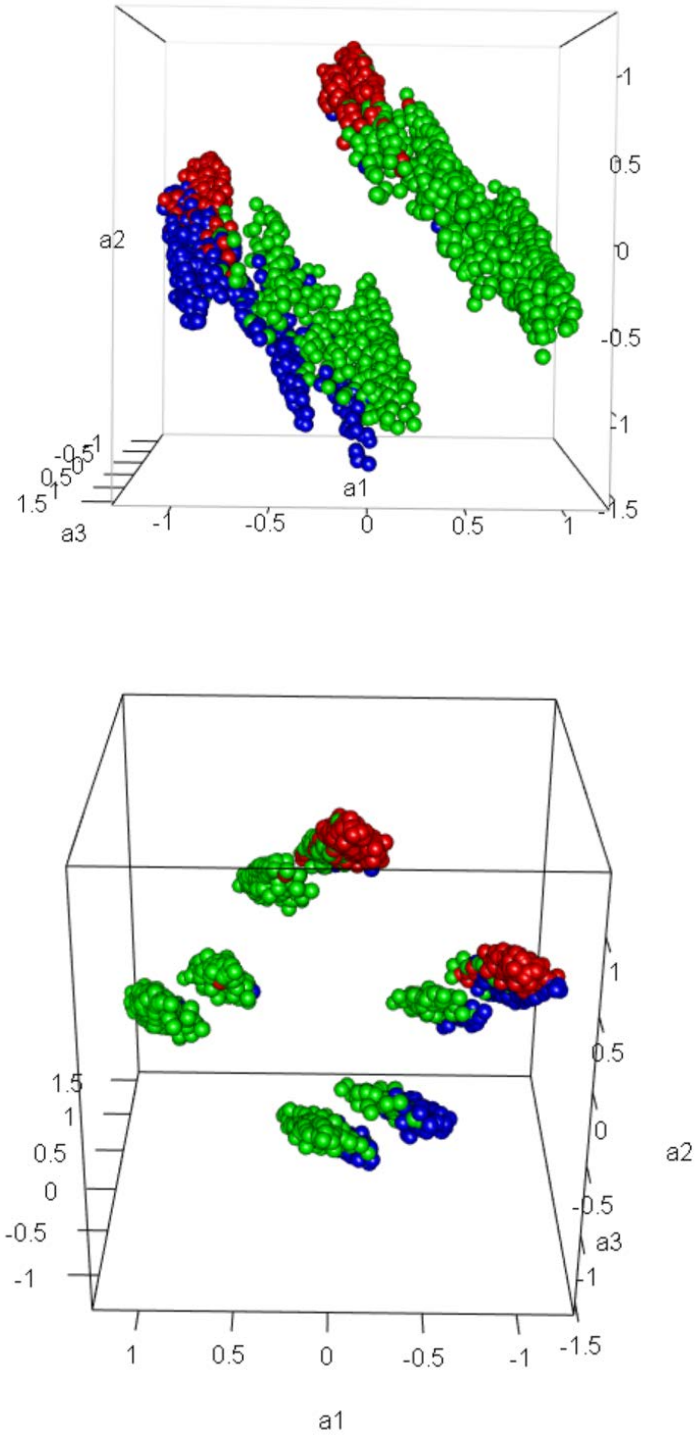


Figure 56. Splice data set mapping using d4 colored by Splice class

Legend “EI” – red, “IE” – blue, and “N” – green

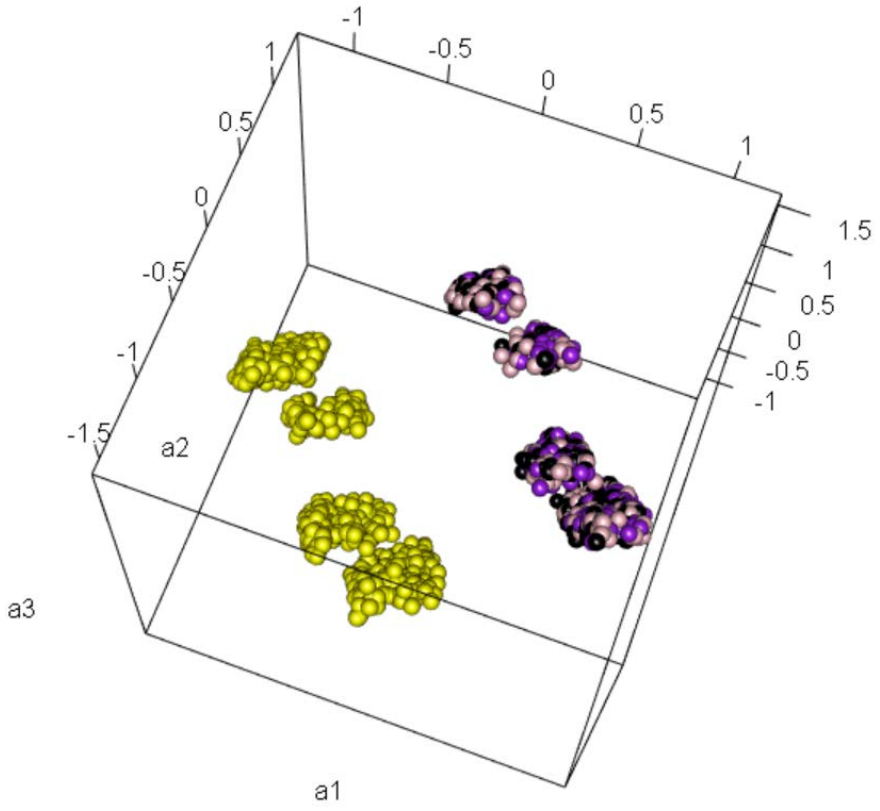


Figure 57. Splice data set mapping using d4 colored by V35

Legend “T” – yellow, “C” – black, “G” – pink, and “A” – purple

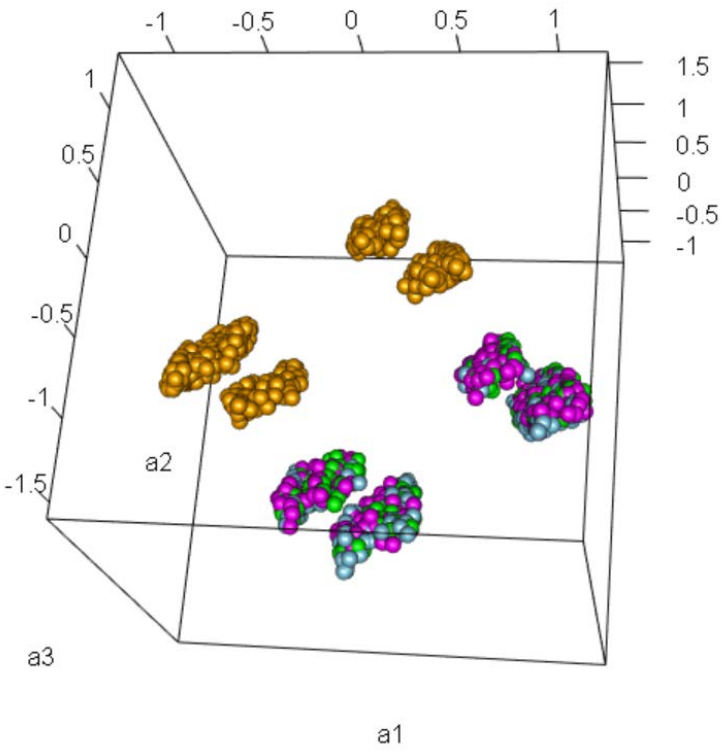


Figure 58. Splice data set mapping using d4 colored by V32

Legend “T” – green, “C” – sky blue, “G” – magenta, and “A” – orange

V35 and V32 variables form the cluster split in the mapping. The dissimilarities for d4 are calculated the same way as for d3, and the different weight of the trees is calculates similarly to d2. The effect of this combination is that the mapping is driven by variables that have the largest deviance change. There is a compound effect in which the change in the deviance is large for those variables that have a large deviance reduction ratio.

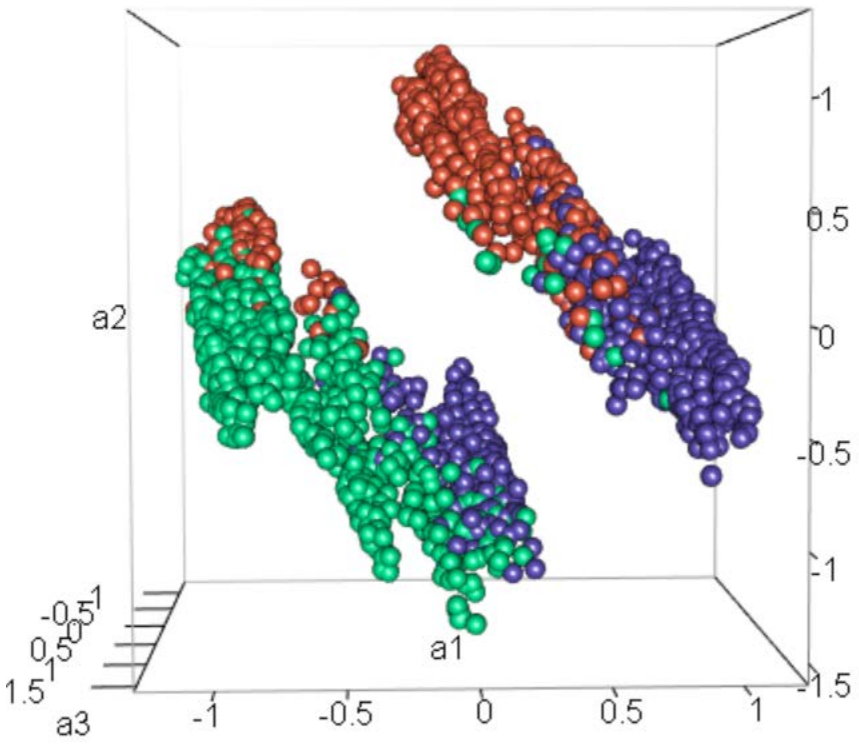


Figure 59. Splice data set mapping using d4 colored by PAM clustering algorithm's results

Figure 59 shows the d4 mapping colored by the clusters identified by the PAM algorithm. It is clear that the clusters are not created directly as a factor of the columns. This may be the result of the fact that the mapping is a three-dimensional representation of much higher-dimensional data. Comparing Figure 59 and Figure 56 shows that the PAM's clusters are similar to the Splice clusters, although not equal. This comparison is a visual representation of the fact that the d4 has the best performance for the PAM clustering task for Buttrey and Whitaker (2015a).

2. Similar Mapping for Different Variants of the Tree Distances

Although the different variants of tree distances calculate the dissimilarities between the observations differently, there are data sets in which some of their mappings, if not all of them, are very similar one to another.

The cases in which the mappings are similar to one another depends on the calculation of the dissimilarities. The d2's dissimilarities are calculated using the R^2 analog, d3's dissimilarities are measured using the leaves' deviance difference, and d4 combines both calculations. If the R^2 analog, the deviance reduction ratio, is similar among all the trees in the data set, the weights of the different trees are similar to each other. In this case, the mappings using d1 and d2 are similar to one another, and the mappings using d3 and d4 are similar to one another but not necessarily to d1 or d2. If the trees have similar structure, the differences in the leaves' deviance can be similar among all the trees. In such a case, d1's and d3's mappings are similar, and d2's mapping is similar to d4's. If both of the conditions apply, d1 and d4 are also similar, and therefore all the mappings are similar.

Figure 60 shows the mappings of the Seeds data set using the four variants. It is clear that d1's and d2's mappings are very similar to one another, as are d3's and d4's mappings. The d1's and d3's mappings and therefore also d2's and d4's mappings are different. The R^2 analog range for the trees created to compute the tree distances is 0.878–0.968 except for one tree, which has a low R^2 analog. Because the R^2 analog is similar for most of the trees, the similarities between the mappings are created. The trees of the Seeds data do not have a similar structure. One of the trees consists of 25 nodes, while another consists of only nine nodes. Therefore, d1's mapping is different from d3's, and d2's mapping is different from d4's.

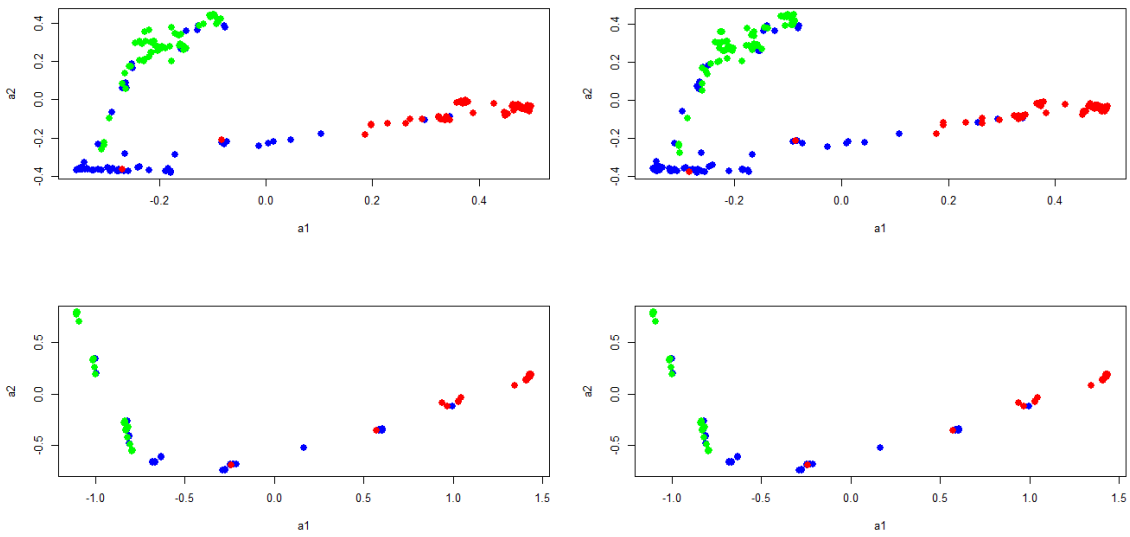


Figure 60. Seeds data set mapping using the four variants

Each plot is a mapping of the data using a different variant, d1–d4, to be viewed in a clockwise direction starting at the top left.

Legend: “1” – blue, “2” – red, and “3” – green

VI. CONCLUSIONS

In this thesis, we presented a framework for visualizing multivariate mixed-type data based on the tree distances of Buttrey (2006) and Buttrey and Whitaker (2015a, 2015b). Our tree distance visualization process includes methods for choosing which variables to use in coloring and how to assign colors to the values of numeric variables and categorical variables with many levels.

We explained why coloring the mappings correctly is an important goal for a visualization algorithm, and we demonstrated this using the coloring of the Splice data set. We discussed the disadvantages of Kruskal and Wish's (1978) regression method for finding the appropriate variables by which to color the mappings, which led us to develop two methods: the purity method and the maximum deviance ratio method. We also introduced our pruning method for choosing values of a variable for coloring.

We illustrated how to identify strong dependence among variables that can bias tree distances and their visualization. In addition, we proposed a modification of the original tree distances algorithm to mitigate the effects of such dependence. We note that this modification has already been implemented in the R package treeClust (Buttrey 2015).

We discussed several issues of the tree distance visualization method. We explained why it is a visualization technique that suits the clustering problem. We described some properties of the method, such as the collapsing tendency of similar observations, which is their tendency to group in one point in the mapping, as well as the equal distance property, which helps the analyst understand the relationships inside the data regardless of the scale of the variables. We also discussed the different mappings generated for the same data set by the different variants of the tree distances.

There are several paths for future work on tree distance visualization. First, it is not clear which dimension of the lower-dimensional Euclidean space the dissimilarities should be mapped into. All of the examples in this thesis are in two- or three-dimensional Euclidean space because an analyst cannot

easily examine higher dimensions. As we discuss in Section A.1, the tree distances can reduce the dimensionality of the data if the data's high-dimensional clusters can be identified by the trees. There could be a situation in which the original dissimilarities cannot be mapped into two or three dimensions without compromising the stress. For example, four clusters, each separated from the others and therefore having the same dissimilarity from one another, cannot necessarily be mapped into two-dimensional Euclidean space and keep the property of equal distances. In these cases, the stress, which is a measurement of the difference between the original dissimilarities and the new distances, is not equal to zero. The stress increases or does not decrease as the dimension of the mapped space reduces (Kruskal and Wish 1978). Kruskal and Wish (1978) suggest two techniques for identifying the dimension to map into using the stress obtained for different dimensions. Further research can explore the application of these two techniques or others for tree distance visualization.

Second, there could be a situation in which a mapping created by the tree distance visualization has too much variability to see important patterns because most but not all observations fall in the same leaves. In Section A, we described how tree distances reduce the variability of observations. Applying the tree distance visualization to the new mapping can reduce the variability even more. Preliminary experiments show that the variability is indeed reduced for a couple of data sets if the process of tree distance visualization is applied iteratively. Applying the tree distance algorithm repeatedly tends to revel a representation of the highest level in the hierarchy of clusters in the data set. Further research is required to determine if repeated visualization can reveal more information about the data, perhaps by reducing the dimensionality to two or three dimensions.

Third, there are other mapping techniques for dissimilarities mapping techniques that could be used instead of the CMDS. The most promising candidate is the t-SNE (van der Maaten and Hinton 2008). The t-SNE is suited for high-dimensional data which consists of several classes, which correspond to the tree distances results.

LIST OF REFERENCES

Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45: 5–32. Accessed August 14, 2015, http://link.springer.com/article/10.1023/A:1010933404324.

Breiman, Leo, and Adele Cutler. 2003. *Manual on Setting Up, Using, and Understanding Random Forests*, ver. 4.0. Accessed September 12, 2015. https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf.

Buttrey, Samuel E. 2006. A Scale-Independent Clustering Method with Automatic Variable Selection Based on Trees. Presented at the Joint Statistical Meetings, Seattle, WA, 2006.

Buttrey, Samuel E. 2015. treeClust: Create a measure of inter-point dissimilarity useful for clustering mixed data, and, optionally, perform clustering. R package version 1.1-1.

Buttrey, Samuel E., and Lyn. R. Whitaker. 2015a. “A Scale-Independent, Noise-Resistant Dissimilarity for Tree-Based Clustering of Mixed Data” (submitted). Naval Postgraduate School, Monterey, CA.

Buttrey, S. E. and Lyn R. Whitaker. 2015b. “treeClust: An R Package for Tree-Based Clustering Dissimilarities.” Volume 8 (To appear in *The R Journal*).

Charytanowicz, Malgorzata, Jerzy Niewczas, Piotr Kulczycki, Piotr Kowalski, Szymon Lukasik, and Slawomir Zak. 2010. “A Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images.” In *Information Technologies in Biomedicine*, edited by Ewa Pietka and Jacek Kawa, 15–24. Berlin Heidelberg, Germany: Springer-Verlag.

Fisher, Ronald A. 1936. “The use of Multiple Measurements in Taxonomic Problems.” *Annals of Eugenics* 7: 179–188.

Fodor, Imola K. 2002. *A Survey of Dimension Reduction Techniques*. UCRL-ID-148494. Oak Ridge, TN: U.S. Department of Energy. https://e-reports-ext.llnl.gov/pdf/240921.pdf.

Friedman, Jerome H., and John W. Tukey. 1974. “A Projection Pursuit Algorithm for Exploratory Data Analysis.” In *The Collected Works of John W. Tukey: Graphics 1965–1985*, vol. 5, 881–889. Boca Raton, FL: CRC Press.

Gower, J. C. 1966. “Some Distance Properties of Latent Root and Vector Methods used in Multivariate Analysis.” *Biometrika* 53: 325–338.

Gower, J. C. “A General Coefficient of Similarity and Some of Its Properties.” *Biometrics* 27, no. 4 (1971): 857–71. Accessed September 12, 2015. http://www.jstor.org/stable/2528823?seq=1#page_scan_tab_contents.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer-Verlag.

Inselberg, Alfred, and Bernard Dimsdale. 1990. “Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry.” In Proceedings of the First IEEE Conference on Visualization, 361–378. Los Alamitos, CA: IEEE Computer Science Press.

Johansson, Sara, Mikael Jern, and Jimmy Johansson. 2008. “Interactive Quantification of Categorical Variables in Mixed Data Sets.” In 12th International Conference on Information Visualisation, 3–10. . Los Alamitos, CA: IEEE Computer Science Press.

Jolliffe. I. T. 1986. *Principal Component Analysis*. New York, NY: Springer-Verlag.

Kagie, Martijn, Michiel Van Wezel, and Patrick J. F. Groenen. 2007. “Online Shopping Using a Two Dimensional Product Map.” In E-Commerce and Web Technologies, 9th International Conference, EC-Web 2008, Proceedings, edited by Guiseppe Psaila and Roland Wagner, 89–98. Berlin Heidelberg, Germany: Springer-Verlag.

Kaufman, Leonard, and Peter J. 1990. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley-Interscience.

Kruskal, Joseph B., and Myron Wish. 1978. *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications.

Liaw, Andy, and Matthew Wiener. 2002. “Classification and Regression by randomForest.” *R News* 2(3): 18–22.

Lichman, Moshe. 2013. “UCI Machine Learning Repository.” Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml

Noordewier, Michiel O., Geoffrey G. Towell, and Jude W. Shavlik. 1990. “Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences.” In Advances in Neural Information Processing Systems 3, edited by R. P. Lippmann, J. E. Moody, and D. S. Touretzky, 530–536. Denver, CO: Morgan Kaufmann.

Meyer, David, Achim Zeileis, and Kurt Hornik. 2006. “The Strucplot Framework: Visualizing Multi-way Contingency Tables with Vcd.” *Journal of Statistical Software* 17, no. 3. Accessed September 12, 2015. http://www.jstatsoft.org/v17/i03/.

R Development Core Team 2014. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Su, Jiang, and Harry Zhang. 2006. “A Fast Decision Tree Learning Algorithm.” *Proceedings of the National Conference on Artificial Intelligence* 21, no. 1 Accessed September 13, 2015. http://www.aaai.org/Papers/AAAI/2006/AAAI06-080.pdf.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. “Visualizing Data using t-SNE.” *Journal of Machine Learning Research* 9: 2579–2605.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California