Machine Learning Methods based on Unimodality Testing

A Dissertation

submitted to the designated
by the Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Paraskevi Chasani

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina
School of Engineering
Ioannina 2025

Advisory Committee:

- **Aristidis Likas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Konstantinos Blekas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina

Examining Committee:

- **Aristidis Likas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Konstantinos Blekas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Georgios Stamou**, Professor, School of Electrical and Computer Engineering, National Technical University of Athens
- **Grigorios Tsoumakas**, Professor, Department of Informatics, Aristotle University of Thessaloniki
- **Konstantinos Vlachos**, Assistant Professor, Department of Computer Science and Engineering, University of Ioannina
- **John Pavlopoulos**, Assistant Professor, Department of Informatics, Athens University of Economics and Business

Dedication

I dedicate this thesis to my family for their continuous support and to my greatest supporter, Fotis, for his trust and belief in me.

Acknowledgements

I would like to express my sincere gratitude and appreciation to all those who have contributed to the completion of this thesis.

I would like to extend my deepest gratitude to my thesis supervisor Prof. Aristidis Likas, for the unwavering support, insightful guidance, and continuous encouragement he has provided throughout my PhD journey. His commitment to excellence and his dedication to my development as a researcher have been invaluable, shaping both my academic and personal growth over these years. Our collaboration has been a truly enriching experience, and the knowledge and skills I have gained under his mentorship will continue to influence my career for years to come.

I am also grateful to the advisory committee members Prof. Konstantinos Blekas and Prof. Christophoros Nikou for their support and guidance throughout this research. I also express my sincere thanks to the evaluation committee members for dedicating their time and expertise to evaluate this thesis.

I would like to extend my sincere thanks to my colleagues Georgios Vardakas and Ioannis Papakostas for fostering a welcoming and supportive atmosphere in our office. The insightful discussions we shared have greatly enriched my experience. It has been a privilege to work alongside such dedicated individuals, and i will always treasure the moments we've shared together.

A big thanks goes to my parents, Sotiris and Ntina, and my sister, Katerina, for their unwavering support and belief in me. Their love, patience, and encouragement have been a constant source of strength throughout this journey.

Finally, a very special thanks goes to Fotis for standing by my side and offering endless understanding during the completion of this thesis. He has been my greatest supporter, and his presence made this journey truly enjoyable.

Table of Contents

List of Figures iv

List of Tables ix

List of Algorithms xi

Abstract xii

Εκτεταμένη Περίληψη xiv

1 Introduction 1

1.1 Statistics Basics . . . 3

1.1.1 Statistical Tests . . . 6

1.1.2 Statistical Data Modeling . . . 10

1.2 Unimodality . . . 15

1.2.1 Unimodality Definition . . . 16

1.2.2 Assessing Unimodality . . . 17

1.2.3 Significance of Unimodality Tests . . . 24

1.3 Mode Estimation . . . 27

1.3.1 Mixture models for Density Estimation and Clustering . . . 28

1.3.2 Nonparametric Methods for Density and Mode Estimation . . . 29

1.4 Decision Trees . . . 34

1.4.1 Supervised Decision Trees . . . 37

1.4.2 Unsupervised Decision Trees . . . 38

1.5 Thesis Contribution . . . 41

2 The UU-test for Statistical Modeling of Unimodal Data 46

2.1 Introduction . . . 46

2.2 Notations and Definitions 47
2.3 UU-test Description 50
2.3.1 Consistent Subsets 51
2.3.2 Sufficient Subsets 54
2.3.3 Uniformity Test 60
2.3.4 Computational Complexity 60
2.4 Modeling Unimodal Data 62
2.5 Experimental Results 64
2.5.1 Evaluating UU-test Decisions 64
2.5.2 Uniform Mixture Modeling of Unimodal Data 69
2.6 Unimodality in Multiple Dimensions 72
2.6.1 UU-test for Clustering 74
2.7 Summary 74

3 Statistical Modeling of Univariate Unimodal Data using Π-sigmoid Mixture Models **77**
3.1 Introduction 77
3.2 Statistical Modeling using the Π-Sigmoid Distribution 79
3.2.1 The Π-Sigmoid Distribution 79
3.2.2 The Π-Sigmoid Mixture Model (ΠsMM) 79
3.3 Method Description 81
3.3.1 Assessing the Unimodality of a Probability Distribution 81
3.3.2 Unimodal ΠsMM Training 82
3.4 Experimental Results 85
3.4.1 Synthetic Datasets 86
3.4.2 Real Datasets 86
3.5 Summary 88

4 Statistical Modeling of Univariate Multimodal Data **90**
4.1 Introduction 90
4.2 Detecting Valleys in Data Density 92
4.2.1 Multimodality Degree 96
4.3 The Unimodal Mixture Model (UDMM) 97
4.3.1 The UniSplit Algorithm 98
4.3.2 Merging Adjacent Intervals 100

4.3.3 Computational Complexity . . . 102
4.3.4 UDMM formulation . . . 102
4.4 Experimental Results . . . 103
4.4.1 Modeling Multimodal Data with UDMM . . . 103
4.4.2 Multimodal Data Splitting . . . 106
4.4.3 Image Segmentation . . . 108
4.4.4 UDMM Naive Bayes for Classification . . . 111
4.4.5 Examples with Noise and Outliers . . . 112
4.4.6 Impact of the Statistical Significance Level . . . 114
4.5 Summary . . . 114

5 Unsupervised Decision Trees for Axis Unimodal Clustering 116
5.1 Introduction . . . 116
5.2 Notations and Definitions . . . 118
5.2.1 Dip-Test for Unimodality . . . 119
5.2.2 UU-Test for Unimodality . . . 119
5.2.3 Axis Unimodal Dataset . . . 121
5.2.4 Node Splitting . . . 122
5.3 Axis Unimodal Clustering with a Decision Tree Model . . . 122
5.3.1 Criterion 1 . . . 123
5.3.2 Criterion 2 . . . 126
5.3.3 Decision Tree Construction . . . 127
5.3.4 An Illustrative Example . . . 129
5.4 Experimental Results . . . 131
5.4.1 Evaluating DTAUC Performance . . . 131
5.4.2 Comparing Criterion 1 with Criterion 2 . . . 139
5.5 Summary . . . 141

6 Conclusions and Future Work 143

Bibliography 148

List of Figures

1.1 Histogram and pdf curve of a Gaussian ($\mu = 0, \sigma^2 = 1$) where μ is the mean value and σ^2 is the variation. 4
1.2 Cdf (left) and ecdf (right) of a Gaussian ($\mu = 0, \sigma^2 = 1$). 5
1.3 A convex (left) and a concave (right) function. 5
1.4 Visualization of the gcm function (red dotted line), lcm function (green dotted line), gcm points (red stars) and lcm points (green circles) of four different ecdfs (blue lines). 6
1.5 Illustration of the Kolmogorov–Smirnov statistic. Red line is the ecdf of two gaussian distributions, blue line is the cdf of a gaussian (left) and a uniform (right), and the length of the black arrow is the KS statistic. 9
1.6 A Gaussian model (red curve) fits in several datasets. 13
1.7 A uniform model (red curve) fits in several datasets. 13
1.8 A Gaussian Mixture of three Gaussian distributions. 14
1.9 Histogram plots of unimodal distributions (top row) and corresponding cdf plots (bottom row). 16
1.10 Histogram plots of multimodal distributions (top row) and corresponding cdf plots (bottom row). 18
1.11 Histograms of a distribution function with different number of bins. . . 18
1.12 Graphical example of the taut string metaphor used in describing the algorithm for the dip-test. The red line is the string, the bottom blue line is $F - d$ and the upper blue line is $F + d$. Note that the string forms the gcm on (x_1, x_L) for $F - d$, and the lcm on (x_U, x_n) for $F + d$. The bottom plot depicts the minimum value of d. If we would decrease d even further, the string would get bent out of its unimodal shape at around $x \approx 0.5$ [1]. 22

1.13 Folding mechanism for univariate distribution. Initial (left) and folded (right) distribution are provided [2]. 23
1.14 Folding mechanism in dimension 2. Initial (left) and folded (right) distribution are provided [2]. 24
1.15 Impact of the pivot location. Initial (left) and folded (right) distribution are provided [2]. 24
1.16 Histogram and ecdf of univariate data. The detected modes are also presented [3]. 32
1.17 Initialization of FTC algorithm (left) and final segmentation after FTC algorithm (right) [4]. 33
1.18 The ecdf of a unimodal dataset and its cdf approximation provided by the UU-test (left). The histogram plot of the unimodal dataset, along with statistical model fits using a UMM (middle) and a UΠsMM (right), are provided. 43
1.19 Histogram plot of a multimodal dataset along with the UDMM pdf. . . 44
1.20 DTAUC method: 2D plot of a dataset split into three axis unimodal clusters (left), and the corresponding binary decision tree (right). . . . 45

2.1 (a) Gcm function and gcm points of an ecdf. (b) Lcm function and lcm points of an ecdf. 49
2.2 Gcm/Lcm function and gcm/lcm points of a unimodal ecdf. AB, BC and CD correspond to the convex, intermediate and concave part, respectively. 49
2.3 Example of multimodal dataset with consistent GL subsets that are not sufficient. 53
2.4 Example of unimodal dataset with consistent GL subsets that are sufficient. 54
2.5 Example of multimodal dataset where consistent subsets are not sufficient. 58
2.6 Example of unimodal dataset. A non-uniform interval exists between lcm points A and B. Forward search method fixes the non-uniformity problem by considering the extended interval between A and C. 58
2.7 Example of multimodal dataset where the UU function is recursively applied on the intermediate part. 59
2.8 Example of unimodal dataset where the UU function is recursively applied on the intermediate part. 61

2.9 Unimodal datasets sampled from (a) a truncated ($x < 0$) Gaussian, (b) a truncated ($x > 0$) Gaussian, (c) a Gaussian, (d) two highly overlapping Gaussians. (Left) Histogram and UMM pdf (solid line). (Right) Points of S, ecdf (solid line) and UMM cdf (dashed line). 63
2.10 Histogram (a) and ecdf (b) of a dataset X sampled from the Gaussian distribution. Histogram (c) and ecdf (d) of a dataset sampled from the UMM obtained by applying UU-test on the Gaussian dataset X of (a) and (b). 64
2.11 Top row: histogram and ecdf of a bimodal dataset generated by two Gaussians. A and B are middle lcm and gcm points respectively. Bottom row: histogram and ecdf of the dataset after adding uniform noise between the Gaussians. The middle lcm/gcm points A and B have been eliminated, however, UU-test still decides multimodality. 67
2.12 Top row: histogram and ecdf of a dataset generated by a single Gaussian. The gcm points between A and B are illustrated. Bottom row: histogram and ecdf of the dataset after adding Student's t distributed noise (outliers) on the left. Two new gcm points (C and D) have been generated, however, UU-test still decides unimodality. 68
2.13 Histogram and ecdf of feature 14 of House dataset for which dip-test decides unimodality and UU-test decides multimodality. 68
2.14 Examples of statistical model fitting on several datasets using Gaussian (left figures), Uniform (middle figures) and UMM (right figures). 71
2.15 Top row: 2-d plot, histogram and ecdf of feature 1 of a 2-d dataset sampled from three Gaussians. Cut point cp_1 is also presented. Middle row: 2-d plot, histogram and ecdf of feature 1 corresponding to the right bimodal subset obtained from the first split. Cut point cp_2 is also presented. Bottom row: 2-d plot, histogram and ecdf of feature 1 corresponding to the original dataset along with the two cutpoints. . . 75
2.16 Histogram and ecdf of feature 3 of Iris dataset [5] along with the computed cut point. 76

3.1 Statistical model fitting on data sampled from asymmetric triangular distribution using Gaussian (left figure), uniform (middle figure) and UMM (right figure). 78

3.2 Two shapes of the Π-sigmoid distribution by varying the λ parameter. (a) $\lambda = 0.5$. (b) $\lambda = 55$. 80
3.3 (a) A multimodal ΠsMM pdf with red stars indicating the second formed peak. (b) UΠsMM pdf with the multimodality issue being fixed. 83
3.4 Statistical model fitting on two synthetic datasets using UΠsMM (left plot), UMM (middle plot) and Gaussian model (right plot). 87

4.1 Histogram: gcm (AB part) and lcm (CD part) correspond to increasing and decreasing parts, respectively. Ecdf: AB, BC and CD correspond to the convex, intermediate and concave part, respectively. 94
4.2 Histogram and ecdf of a bimodal dataset. The non-uniform and unimodal $X(x_A, x_B)$ indicates a density valley between A and B. MD is a point close to the valley. vp is the valley point. (a) A, B are gcm points on increasing parts of successive modes. (b) A, B are lcm points on decreasing parts of successive modes. 94
4.3 Histogram and ecdf plot of a multimodal dataset with its best splitting intervals, processed recursively until a non-uniform and unimodal interval containing a single valley point is detected. 95
4.4 Histogram and ecdf plots of bimodal datasets with varying peak distances and valley depths. The black segments on the pdfs correspond to the horizontal distances (d_1' and d_2') between the two peaks, while on the ecdfs correspond to the max distances (d_1 and d_2) of MD from line segment AB. 96
4.5 (a) Bimodal dataset with two computed valley points by UniSplit. (b) Omitting vp_1 leads to a multimodal set $X_1 \cup X_2$, thus vp_1 is necessary. (c) Merging X_2 and X_3 (omitting vp_2) leads to a unimodal set, thus vp_2 can be deleted. vp_1 is the final valley point. 101
4.6 Examples of statistical model fitting results on several datasets using GMM and UDMM. 106
4.7 Initial images (first column). Segmented images obtained by the compared methods (second - seventh column). For rgb images the ground truth value of colors ($k^\star$) is illustrated, while the estimated number of colors (k) is provided for both rgb and grayscale images. 110
4.8 Data generated by three uniform rectangles assigned to two classes. . . 112

4.9 Histogram and ecdf plots of a trimodal dataset before and after adding noise/outliers. The original valley points (dotted vertical lines) are almost identical to the final valley points (solid vertical lines). 113

5.1 Histogram and ecdf plots of unimodal and multimodal univariate datasets. The p-values provided by the dip-test are also presented. (a) Unimodal dataset. (b) Borderline case of unimodal dataset (with two close peaks). (c) Multimodal dataset (with two peaks). (d) Multimodal dataset (with three peaks). 120

5.2 Histogram of a bimodal dataset along with its split threshold (star) computed using criterion 1. (a) The split threshold was computed without utilizing the separation criterion. (b) The split threshold was computed taking into account the separation criterion. 124

5.3 Histogram plots of synthetic datasets along with the best split thresholds found using criterion 1 (stars) and criterion 2 (circles). 126

5.4 Stepwise partitioning of a 2-D dataset (X) into axis unimodal rectangular regions using criterion 1. (a) 2-D plot of the original dataset X, with histogram plots of each feature, the obtained split points, and the resulting 2-D plot illustrating X split (by sp_2) into two clusters (X_L, X_R). (b) 2-D plot of X_L, with histogram plots of each feature, the obtained split point, and the resulting 2-D plot illustrating X_L split (by sp_3) into two clusters (X_{LL}, X_{LR}). (c) 2-D plots of X_{LL} and X_{LR}, along with the unimodal histogram plots of each feature. (d) 2-D plot of X_R, along with the unimodal histogram plots of each feature. (e) Final 2-D plot of X, illustrating the final split points (sp_2, sp_3) that partition X into three axis unimodal clusters (X_{LL}, X_{LR}, X_R). 133

5.5 Binary decision tree constructed for the two-dimensional dataset of Figure 5.4a. 133

5.6 2-D plots of (a) Synthetic I and (b) WingNut. The ground truth partition and the partitions obtained by DTAUC (using either criterion 1 or 2), ICOT and ExShallow are provided. 137

5.7 2-D plots of synthetic datasets (Synthetic II – Synthetic VIII) illustrating the ground truth partition and the partitions obtained by DTAUC_c1 and DTAUC_c2. 140

List of Tables

2.1 Summary of notation. 50
2.2 Accuracy of UU-test and dip-test on deciding unimodality (U) or multimodality (M). 66
2.3 Dip-test and UU-test unimodality (U) or multimodality (M) decisions on features of real datasets. 69
2.4 Types and parameters of distributions and size of training and test set of the datasets used for UMM evaluation. 70
2.5 Statistical model evaluation using the test set log-likelihood (the higher the better). Bold values indicate the best model in each row. 70
2.6 Statistical model evaluation using the two-sample KS test (the lower the better). Bold values indicate the best model in each row. 72

3.1 Synthetic dataset characteristics. 87
3.2 Statistical model evaluation using the test set log-likelihood (the higher the better). Bold values indicate the best model in each row. Initial and final number of components are also provided. 88

4.1 Characteristics of synthetic and real datasets. 104
4.2 Statistical model evaluation using the two-sample KS test (the lower the better). Bold values indicate the best model in each row. The ground truth number of components ($k^\star$) (in case of synthetic datasets) and the average estimated number of components (k) are also provided. . . 105
4.3 Characteristics of synthetic datasets. 107
4.4 Partition evaluation of multimodal datasets. The average and standard deviation for NMI, the ground truth number of modes ($k^\star$) and the average number of detected modes (k) are provided. 107

4.5 Image segmentation results: i) Estimated number of colors (k), ii) NMI values with respect to a ground truth solution obtained by applying k-means with the ground truth number of colors ($k^\star$). 109
4.6 Accuracy results on synthetic and real datasets. Bold numbers indicate the best average performance for each dataset. 111

5.1 Stepwise partitioning of the two-dimensional dataset of Figure 5.4a using criterion 1. 131
5.2 Parameters of synthetic and real datasets used in the experiments. . . . 134
5.3 Partition results on synthetic data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported. . . . 135
5.4 Partition results on real data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported. 136
5.5 Parameters of synthetic datasets used in the experiments comparing the two proposed criteria of the DTAUC method. 138
5.6 Partition results of DTAUC method using criterion and criterion 2 on synthetic data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported. 138

List of Algorithms

1.1 Fine to Coarse (FTC) Segmentation Algorithm 33
2.1 $S = UUtest(X)$ 51
2.2 $(S'_G, P'_I, S'_L, success) = UU(S_G, P_I, S_L)$ 52
2.3 (P',success) = sufficient(P) 56
2.4 (P'_F,success) = Forward_search(P_F, e_L) 57
2.5 (P'_B,success) = Backward_search(P_B, e_R) 57
3.1 new_model = fix_multimodality($model$) 82
3.2 new_model = Merge($[a_k, b_k], K$) 84
3.3 new_model = UIIsMM(X) 85
4.1 vp = find_vp(X) // X is multimodal 100
4.2 vp_list = UniSplit(X, vp_list) 101
5.1 $(sp^\star, q^\star)$ = best_split_point_c1(S, α) 125
5.2 $(sp^\star, d^\star)$ = best_split_point_c2(S, α) 127
5.3 $(j^\star, sp^\star)$ = best_split_c1(X, α) 128
5.4 $(j^\star, sp^\star)$ = best_split_c2(X, α) 129
5.5 DTAUC(X, α) 130

Abstract

Paraskevi Chasani, Ph.D., Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2025.
Machine Learning Methods based on Unimodality Testing.
Advisor: Aristidis Likas, Professor.

Recognizing unimodal data distributions is of great significance in statistics, machine learning and data science. The characteristic property of a unimodal distribution is that data values are gathered around a single value (peak), which is the mode of the distribution. Due to this property, data can be characterized as homogeneous, forming a single and coherent group. Well-known distributions, such as Gaussian, Student's t and Gamma are typical examples of unimodal distributions. Also, the uniform distribution is considered as an extreme unimodal case. Unimodality tests have been proposed to decide on the unimodality of a set of data values, thus providing useful knowledge about the structure of the data.

This thesis concerns the development and implementation of machine learning methods based on the notion of unimodality, focusing on four main axes: i) the creation of a new unimodality test for deciding data unimodality, ii) the analysis of key characteristics of data density, such as modes and valleys, which leads to the discovery of innovative properties explored in detail, iii) the development of statistical models, specifically mixture models, for modeling univariate unimodal and multimodal (multiple peaks) data, and iv) the development of partitioning methods for multidimensional data into clusters that are unimodal along each axis, achieved through the unsupervised construction of axis-aligned binary decision trees.

We begin, by proposing a new unimodality test called Unimodal Uniform test (UU-test) to decide if a dataset has been generated by a unimodal distribution or not. The method utilizes the empirical distribution function (ecdf) and attempts to obtain a unimodal piecewise linear approximation of the ecdf under the constraint

that the data corresponding to each linear segment follow the uniform distribution. Compared to other unimodality tests, it also produces a generative model of the unimodal data in the form of a mixture of uniform distributions (UMM). Thus, it can be used for statistical data modeling of unimodal distributions with arbitrary shape. Next, we improve UMM performance by substituting the uniform distribution with a more flexible and differential one, called Π-sigmoid. The Π-sigmoid distribution, defined as the difference of two translated logistic sigmoids, can approximate a wide range of distributions. We employ and train a mixture model of Π-sigmoids, called UΠsMM, initialized using the output of the UU-test. Additionally, we introduce a mechanism to maintain the unimodality of the model during training via the Expectation-Maximization (EM) algorithm. UΠsMM achieves an accurate fit while often requiring fewer components than UMM.

Afterward, we address the problem of modeling univariate multimodal data, with two main contributions. First, we introduce properties of critical points of the data ecdf that provide indications on the existence of density valleys. Using these properties, we propose UniSplit, an algorithm that detects valley points and partitions the dataset into unimodal subsets, automatically estimating their number. Second, we propose a statistical model, the Unimodal Mixture Model (UDMM), which models each unimodal subset with a UMM. A key strength of UDMM is its flexibility and independence from specific parametric assumptions, making it well-suited for datasets generated by sources of different probability density (e.g., one Gaussian and one uniform). Another important property is that the number of components is automatically estimated, therefore, a major issue in mixture modeling is addressed.

Finally, we focus on developing an unsupervised method for clustering multidimensional data using decision trees. We introduce the concept of axis unimodal clusters, i.e., clusters where all features are unimodal as decided by a unimodality test. We present a method that constructs binary decision trees, providing axis-aligned partitions of the data and offering interpretable clustering solutions. Two criteria are proposed to identify the best split pair (feature and threshold) at each node, aiming to improve the unimodality of the partition after each split. Compared to other unsupervised decision tree methods, this approach has several advantages: it is simple, avoids preprocessing steps and does not employ computationally expensive optimization methods or difficult to tune hyperparameters, such as number of clusters or maximum tree depth.

Εκτεταμενη Περιληψη

Παρασκευή Χασάνη, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2025.
Μέθοδοι Μηχανικής Μάθησης βασισμένες σε Έλεγχο Μονοτροπικότητας.
Επιβλέπων: Αριστείδης Λύκας, Καθηγητής.

Η αναγνώριση μονοτροπικών (unimodal) κατανομών διαδραματίζει σημαντικό ρόλο στη στατιστική, τη μηχανική μάθηση και την ανάλυση δεδομένων. Η χαρακτηριστική ιδιότητα των μονοτροπικών κατανομών είναι ότι τα δεδομένα βρίσκονται πολύ κοντά σε μία τιμή, η οποία είναι η κορυφή (mode/peak) της κατανομής. Εξαιτίας αυτής της ιδιότητας, τα δεδομένα χαρακτηρίζονται ως ομοιογενή, σχηματίζοντας μία συνεκτική ομάδα. Γνωστές κατανομές, όπως οι: Κανονική (Gaussian), Student's t και Γάμμα είναι παραδείγματα μονοτροπικών κατανομών. Επίσης, η Ομοιόμορφη (uniform) κατανομή είναι μια ακραία περίπτωση μονοτροπικής κατανομής. Τα τελευταία χρόνια έχουν προταθεί τεστ μονοτροπικότητας (unimodality tests) που αποφασίζουν τη μονοτροπικότητα ενός συνόλου δεδομένων, παρέχοντας χρήσιμη γνώση για τη δομή των δεδομένων.

Η παρούσα διατριβή επικεντρώνεται στην ανάπτυξη και εφαρμογή μεθόδων μηχανικής μάθησης βασισμένες στην έννοια της μονοτροπικότητας, εστιάζοντας σε τέσσερις βασικούς θεματικούς άξονες: α) τη δημιουργία ενός νέου τεστ μονοτροπικότητας για να αποφασίζουμε σχετικά με τη μονοτροπικότητα των δεδομένων, β) την ανάλυση χαρακτηριστικών της πυκνότητας των δεδομένων, όπως είναι οι κορυφές και οι κοιλάδες (valleys), που οδηγεί στην ανακάλυψη καινοτόμων ιδιοτήτων που θα εξερευνηθούν ενδελεχώς, γ) την ανάπτυξη στατιστικών μοντέλων, συγκεκριμένα μεικτών μοντέλων (mixture models), για μοντελοποίηση μονοδιάστατων μονοτροπικών και πολυτροπικών (πολλαπλές κορυφές) (multimodal) δεδομένων και δ) την ανάπτυξη μεθόδων διαμέρισης πολυδιάστατων δεδομένων σε ομάδες (clusters), ώστε

να είναι μονοτροπικά σε κάθε άξονα, η οποία πραγματοποιήθηκε με την κατασκευή χωρίς επίβλεψη παράλληλων με τους άξονες δυαδικών δέντρων απόφασης.

Αρχικά, προτείνουμε ένα νέο τεστ μονοτροπικότητας που λέγεται Μονοτροπικό-Ομοιόμορφο τεστ (UU-τεστ) για να αποφασίζουμε εάν ένα σύνολο δεδομένων έχει παραχθεί ή όχι από μονοτροπική κατανομή. Η μέθοδος αυτή χρησιμοποιεί την εμπειρική συνάρτηση κατανομής (ecdf) και προσπαθεί να κατασκευάσει μια μονοτροπική κατά τμήματα γραμμική προσέγγιση (piecewise linear approximation) αυτής υπό τον περιορισμό ότι τα δεδομένα που αντιστοιχούν σε κάθε γραμμικό κομμάτι να ακολουθούν ομοιόμορφη κατανομή. Συγκριτικά με άλλα τεστ μονοτροπικότητας, παράγει επίσης ένα μοντέλο για μονοτροπικά δεδομένα που έχει τη μορφή μεικτών ομοιόμορφων κατανομών (UMM). Επομένως, μπορεί να χρησιμοποιηθεί για στατιστική μοντελοποίηση μονοτροπικών κατανομών οποιασδήποτε μορφής. Ακολούθως, βελτιώνουμε την επίδοση του ομοιόμορφου μεικτού μοντέλου αντικαθιστώντας την ομοιόμορφη κατανομή με μια πιο ευέλικτη, που ονομάζεται Π-σιγμοειδή. Η Π-σιγμοειδής κατανομή ορίζεται ως η διαφορά δύο μετατοπισμένων σιγμοειδών και μπορεί να προσεγγίσει ένα ευρύ φάσμα κατανομών. Εκπαιδεύουμε ένα μεικτό μοντέλο Π-σιγμοειδών, που ονομάζεται UΠsMM και αρχικοποιείται χρησιμοποιώντας το αποτέλεσμα του UU-τεστ. Επιπροσθέτως, προτείνουμε ένα μηχανισμό για να διατηρείται η μονοτροπικότητα του μοντέλου κατά τη διάρκεια της εκπαίδευσης με τον αλγόριθμο EM. Το UΠsMM βελτιώνει την ακρίβεια της μοντελοποίησης, ενώ συχνά απαιτεί λιγότερες συνιστώσες (components) σε σχέση με το ομοιόμορφο μεικτό μοντέλο.

Στη συνέχεια, ασχολούμαστε με το πρόβλημα της μοντελοποίησης μονοδιάστατων πολυτροπικών δεδομένων κάνοντας δύο βασικές συνεισφορές. Αρχικά, προτείνουμε ιδιότητες κρίσιμων σημείων της εμπειρικής συνάρτησης κατανομής των δεδομένων, οι οποίες παρέχουν ενδείξεις για την ύπαρξη κοιλάδων στην πυκνότητα των δεδομένων. Χρησιμοποιώντας αυτές τις ιδιότητες, προτείνουμε τον UniSplit, έναν αλγόριθμο που εντοπίζει κοιλάδες και διαμερίζει το σύνολο δεδομένων σε μονοτροπικά υποσύνολα, εκτιμώντας αυτόματα τον αριθμό τους. Ακολούθως, προτείνουμε ένα στατιστικό μοντέλο, το μονοτροπικό μεικτό μοντέλο (UDMM), το οποίο μοντελοποιεί κάθε μονοτροπικό υποσύνολο με ένα ομοιόμορφο μεικτό μοντέλο. Βασικό πλεονέκτημα του μονοτροπικού μεικτού μοντέλου είναι η ευελιξία και η ανεξαρτησία του από συγκεκριμένες παραμετρικές υποθέσεις, καθιστώντας το κατάλληλο για σύνολα δεδομένων που προέρχονται από πηγές διαφορετικής πυκνότητας πι-

θανότητας (π.χ., μία κανονική και μία ομοιόμορφη). Επιπλέον, ο αριθμός των συνιστωσών υπολογίζεται αυτόματα, αντιμετωπίζοντας έτσι, ένα σημαντικό πρόβλημα των μεικτών μοντέλων.

Τέλος, εστιάζουμε στην ανάπτυξη μια μεθόδου χωρίς επίβλεψη (unsupervised) για ομαδοποίηση (clustering) πολυδιάστατων δεδομένων χρησιμοποιώντας δέντρα απόφασης σε ομάδες μονοτροπικές σε κάθε άξονα (axis unimodal), δηλαδή ομάδες όπου όλα τα χαρακτηριστικά τους είναι μονοτροπικά, σύμφωνα με τις αποφάσεις ενός τεστ μονοτροπικότητας. Αυτή η μέθοδος κατασκευάζει δυαδικά δέντρα απόφασης, παρέχοντας διαμερίσεις των δεδομένων παράλληλες με τους άξονες και προσφέροντας ερμηνεύσιμες λύσεις ομαδοποίησης. Δύο κριτήρια προτείνονται για να εντοπίσουμε το καλύτερο ζεύγος διάσπασης (χαρακτηριστικό και τιμή) σε κάθε κόμβο του δέντρου, στοχεύοντας στην βελτίωση της μονοτροπικότητας της διαμέρισης μετά από κάθε διάσπαση. Συγκριτικά με άλλες μεθόδους δέντρων απόφασης χωρίς επίβλεψη, αυτή η προσέγγιση έχει αρκετά πλεονεκτήματα: είναι απλή, αποφεύγει βήματα προεπεξεργασίας και δεν χρησιμοποιεί ακριβές υπολογιστικά μεθόδους βελτιστοποίησης ή πολλές υπερπαραμέτρους, όπως είναι ο αριθμός των ομάδων και το μέγιστο βάθος του δέντρου.

Chapter 1

Introduction

1.1 Statistics Basics

1.2 Unimodality

1.3 Mode Estimation

1.4 Decision Trees

1.5 Thesis Contribution

As the amount of available data is permanently growing, there is increasing interest in methods capable of extracting valuable knowledge from this data. In the fields of Data Analysis, Data Mining and Machine Learning specific algorithms are applied to prepare data for the purpose of either prediction or description [6]. Prediction involves finding patterns that can assist in forecasting the behavior of a phenomenon (or some entities) (e.g., feed-forward artificial neural networks). Description involves finding useful explanatory patterns that can be presented to a user in a digestible, understandable form (e.g., decision trees).

Machine learning is the area of artificial intelligence that attempts to provide machines with the ability to learn from examples [7, 8]. It aims to develop algorithms that can infer patterns, make predictions, or discover structures in data, often without explicit programming for specific tasks. The two main areas of machine learning are *supervised* and *unsupervised learning*. In supervised learning, models are trained using labeled data, where the input-output relationships are explicitly provided. This

approach is widely applied in classification and regression problems, such as image recognition or predicting housing prices. On the other hand, unsupervised learning deals with unlabeled data, aiming to uncover hidden patterns or structures within the dataset. Techniques such as clustering and dimensionality reduction fall under this category, with applications in customer segmentation, anomaly detection, and more.

But how do we learn a system using the available observations? A common approach is to consider a parametric function (*model*) that is used to describe the process that generates the observed data and then estimate the corresponding parameters.

An important issue to note is that in order for a model to be accurate, it needs to make certain assumptions for the mechanism that generates the observations. Accurate modeling of these mechanisms is essential for several reasons. First, it allows for better generalization - enabling models to perform well on unseen data. Second, it facilitates interpretability, helping domain experts gain insights into the data's structure and behavior. Finally, a well-crafted model can serve as a foundation for tasks like prediction, simulation, and decision-making. For these reasons, *statistical modeling* has become a key component of machine learning research and practice. However, models that make many assumptions have generally poor performance, since too many assumptions are unlikely to be realistic.

One powerful approach to data modeling involves the use of *mixture models*. Mixture models assume that the data is generated by a combination of underlying probability distributions, each representing a distinct cluster or component within the data. These models are highly flexible and can approximate complex distributions, making them a valuable tool for both density estimation and clustering tasks.

In many practical situations, assuming a specific model to characterize the density function of a given dataset may not be feasible, especially when we have no prior knowledge about how the data was generated. In such cases, we must rely on non-parametric statistical methods.

When examining the histogram or kernel density estimate of a dataset, one important characteristic to consider is *unimodality*, i.e., a property of data distributions where the values form a single peak or mode[1]. Understanding whether a dataset is unimodal has important implications for modeling and clustering. For example, many statistical models, including mixture models, rely on assumptions about the unimodality or multimodality of data. Moreover, identifying transitions between uni-

[1]Formal definitions are provided in the next sections.

modal and multimodal regions can help detect structural changes in the data, such as boundaries between clusters. To determine whether a given dataset follows a unimodal distribution, we use statistical tools, called *unimodality tests*. These tests are vital in many applications, as they guide decisions about how to partition data and design models.

In this thesis, the notion of unimodality serves as the foundation for the development of novel machine learning methods and models. A new unimodality test is introduced, providing a robust tool for determining whether data follows a unimodal distribution. Additionally, key characteristics of data density, such as modes and valleys, are thoroughly examined, leading to the discovery of innovative properties that are explored in detail. These insights pave the way for the creation of statistical models, particularly mixture models, which advance clustering techniques and enhance our understanding of data-driven mechanisms. The following sections delve into these problems, providing an in-depth analysis and a review of the related work. Afterward, we present the main contributions of the thesis.

1.1 Statistics Basics

Some basic definitions are provided to make more clear the rest of the thesis. First, the definitions of a probability density function (pdf), cumulative distribution function (cdf) and empirical distribution function (ecdf) are provided, while the greatest convex minorant function and the least concave majorant function are explained, since they play essential role to construct our methods.

The probability density function (pdf) of a continuous 1-d random variable X with support S is an integrable function $f(x)$ satisfying the following:

1. $f(x)$ is positive everywhere in the support S, that is, $f(x) > 0$, for all x in S.
2. The area under the curve $f(x)$ in the support S is 1, that is: $\int_S f(x)\,dx = 1$.
3. If $f(x)$ is the pdf of x, then the probability that x belongs to A, where A is some interval, is given by the integral of $f(x)$ over the interval, that is: $P(X \in A) = \int_A f(x)\,dx$.

More specifically, if $A = [a, b]$, the integral will be $P(a \leq X \leq b) = \int_a^b f(x)\,dx$.

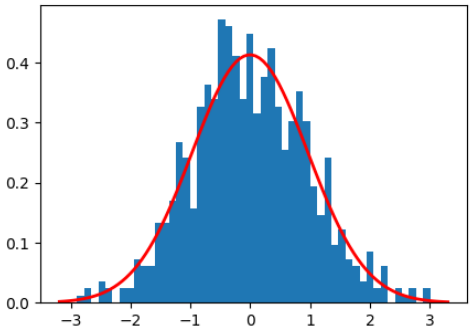


Figure 1.1: Histogram and pdf curve of a Gaussian ($\mu = 0, \sigma^2 = 1$) where μ is the mean value and σ^2 is the variation.

Fig. 1.1 illustrates the pdf and histogram of a Gaussian ($\mu = 0, \sigma^2 = 1$), where μ is the mean value and σ^2 is the variation.

The cumulative distribution function (cdf) of a real-valued random variable X, or just distribution function of X, evaluated at x, is the probability that X will take a value less than or equal to x. The cumulative distribution function of a real-valued random variable X is the function given by: $F_X(x) = P(X \leq x)$, where the right-hand side represents the probability that the random variable X takes on a value less than or equal to x. The probability that X lies in the semi-closed interval $(a, b]$, where $a < b$, is therefore $P(a < X \leq b) = F_X(b) - F_X(a)$.

An empirical distribution function (ecdf) is the distribution function associated with the empirical measure of a sample of N data points. It is a step function that jumps up by $1/N$ at each of the N data points. Its value at any specified value of the measured variable is the fraction of observations of the measured variable that are less than or equal to the specified value. In other words:

Given N ordered points $x_1, x_2, ..., x_N$, the ecdf is defined as:

$$F_X(x) = \frac{\text{number of elements in the sample} \leq x}{N} = \frac{1}{N} \sum_{i=1}^{N} I_{(-\infty,x)}(x_i) \tag{1.1}$$

$I_{(-\infty,x)}(x_i)$ is the indicator function: $I_{(-\infty,x)}(x_i) = \begin{cases} 1, & \text{if } x_i \leq x \\ 0, & \text{otherwise} \end{cases}$

In Fig. 1.2 the cdf and ecdf of a Gaussian distribution are presented. As the number of points increases, the distributions become more similar.

A significant tool in our methodology is the computation of the greatest convex minorant and the lowest concave majorant functions. We provide below the definitions

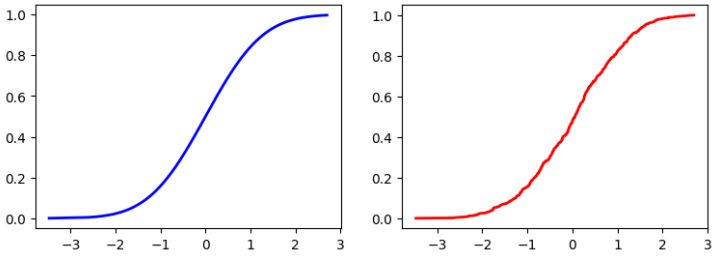


Figure 1.2: Cdf (left) and ecdf (right) of a Gaussian ($\mu = 0, \sigma^2 = 1$).

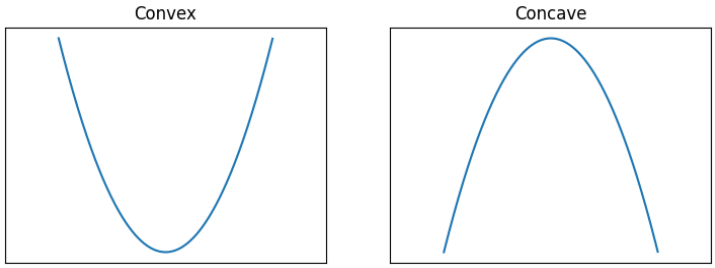


Figure 1.3: A convex (left) and a concave (right) function.

of the two functions.

The greatest convex minorant (gcm) of a function F in $(-\infty, a]$ is $supG(x)$ for $x \leq a$, where the sup is taken over all functions G that are convex in $(-\infty, a]$ and nowhere greater than F. Similarly, the least concave majorant (lcm) of a function F in $[a, \infty)$ is defined as $infL(x)$ for $x \geq a$, where the inf is taken over all functions L that are concave in $[a, \infty)$ and nowhere less than F. In Fig. 1.3 a convex and a concave function are illustrated.

The gcm/lcm points of the ecdf of four distributions are illustrated in Fig. 1.4. The red dotted line illustrates the gcm of the ecdf F (blue line), while the green dotted line illustrates the lcm of F. The red stars on the lower line are the gcm's points of contact with F, while the green circles on the upper line are the lcm's points. The former are called gcm points and the latter lcm points. These points are of great importance in approximating F, since they can be characterized as a lower and upper limit of F. In the top row of Fig. 1.4 only gcm (left plot) or lcm (right plot) points of the ecdf exist, while in the plots of bottom row both gcm and lcm points exist.

Let $X = \{x_1, ..., x_N\}$, $x_i \in \mathbb{R}$ and $x_i < x_{i+1}$ an ordered 1-d dataset of distinct real numbers. For an interval [a,b], we define $X(a, b) = \{a \leq x_i \leq b, x_i \in X\}$ the subset of X whose elements belong to that interval. This notation will be used throughout the rest of the thesis.

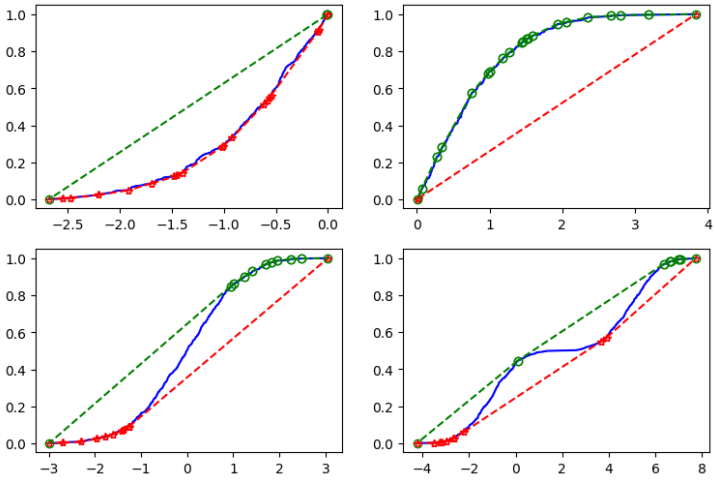


Figure 1.4: Visualization of the gcm function (red dotted line), lcm function (green dotted line), gcm points (red stars) and lcm points (green circles) of four different ecdfs (blue lines).

1.1.1 Statistical Tests

Statistical tests are used in the field of statistics to prove various properties of a dataset [9]. They provide a mechanism for making quantitative decisions about a process of interest. In other words, we use statistical tests to decide whether a pattern we observe is due to chance or due to the program or intervention effects. Research often uses them to determine if there is a relationship between an intervention and an outcome as well as to quantify the strength of that relationship.

At first, a test statistic (a quantity derived from the sample) is computed and is considered as a numerical summary of a dataset that reduces the data to one value. The intent is to determine whether there is enough evidence to "reject" a conjecture or hypothesis about the process. The conjecture is called the null hypothesis. Not rejecting may be a good result if we want to continue to act as if we "believe" the null hypothesis is true. Or it may be a disappointing result, possibly indicating we may not yet have enough data to "prove" something by rejecting the null hypothesis.

The test statistic is used in testing the statistical hypothesis and is selected or defined in such a way as to quantify, using observed data, behaviors that would distinguish the null from the alternative hypothesis. A common format for a hypothesis test is:

H_0: A statement of the null hypothesis, e.g., two population means are equal.

H_a: A statement of the alternative hypothesis, e.g., two population means are not

equal.

Test Statistic: The test statistic is based on the specific hypothesis test.

Significance Level: The significance level, α, defines the sensitivity of the test. A value of $\alpha = 0.05$ means that we inadvertently reject the null hypothesis 5% of the time when it is in fact true. This is also called the type I error. The choice of α is somewhat arbitrary, although in practice values of 0.1, 0.05, and 0.01 are commonly used.

Common test statistics are one-sample, two-sample and paired tests [10]. One-sample tests are appropriate when a sample is being compared to the population from a hypothesis. The population characteristics are known from theory or are calculated from the population. Two-sample tests are appropriate for comparing two samples, typically experimental and control samples from a scientifically controlled experiment. Paired tests are appropriate for comparing two samples where it is impossible to control important variables. Rather than comparing two sets, members are paired between samples so the difference between the members becomes the sample. Typically, the mean of the differences is then compared to zero. The common example scenario for when a paired difference test is appropriate is when a single set of test subjects has something applied to them and the test is intended to check for an effect. For example, if we compare the weight of every person in a group of people before they went on a diet with their weight after they completed the diet program.

According to the null and alternative hypothesis, there are two kinds of tests: the two-sided and one-sided tests (or two-tailed and one-tailed tests) [11]. A two-tailed test is appropriate if the estimated value may be more than or less than the reference value, for example, whether a test taker may score above or below the historical average. A one-tailed test is appropriate if the estimated value may depart from the reference value in only one direction, for example, whether a machine produces more than one-percent defective products.

A measure for evaluating the result of a test of hypothesis is critical values. Critical values for a test of hypothesis depend upon a test statistic, which is specific to the type of test. They are essentially cut-off values that define regions where the test statistic is unlikely to lie; for example, a region where the critical value is exceeded with probability α if the null hypothesis is true. The null hypothesis is rejected if the test statistic lies within this region which is often referred to as the rejection region(s).

Another quantitative measure for reporting the result of a test of hypothesis is the

p-value. The p-value is the probability of the test statistic being at least as extreme as the one observed given that the null hypothesis is true. A low p-value is an indication that the null hypothesis is false. The benefit of using p-value is that it calculates a probability estimate, we can test at any desired level of significance by comparing this probability directly with the significance level. It is good practice to decide in advance of the test how small a p-value is required to reject the test. This is exactly analogous to choosing a significance level, α, for test. For example, we decide either to reject the null hypothesis if the test statistic exceeds the critical value (for $\alpha = 0.05$) or analogously to reject the null hypothesis if the p-value is smaller than 0.05.

Known statistical tests are: Z-test, T-test and Chi-square [11]. In Z-test, the sample is assumed to be normally distributed and a z-score is calculated with population parameters, such as "population mean" and "population standard deviation". It is used to validate the hypothesis that the sample drawn belongs to the same population. A T-test is used to compare the mean of two given samples. Like a Z-test, a T-test also assumes a normal distribution of the sample. A T-test is used when the population parameters (mean and standard deviation) are not known. Chi-square test is used to compare categorical variables. There are two types of Chi-square tests: (1) Goodness of fit test, which determines if a sample matches the population and (2) a Chi-square fit test for two independent variables is used to compare two variables in a contingency table to check if the data fits.

Gaussianity (Normality) Tests

There is a large number of tests for determining if a dataset is well-modeled by a normal distribution and computing how likely it is for a random variable underlying the dataset to be normally distributed. In Statistics, these tests are called Gaussianity or Normality tests [9]. An informal and simple approach to testing normality is to compare a histogram of the sample data to a normal probability curve. The empirical distribution of the data (the histogram) should be bell-shaped and resemble the normal distribution. This might be difficult to observe if the sample is small. Another test of normality is Shapiro-Wilk test, which tests the null hypothesis that a sample $\{x_1, ..., x_n\}$ comes from a normally distributed population.

There are more general tests, which check if a dataset is well-modeled not only by a normal distribution but also by other distributions. For example, the Anderson–Darling test is a statistical test of whether a given sample of data is drawn from a given

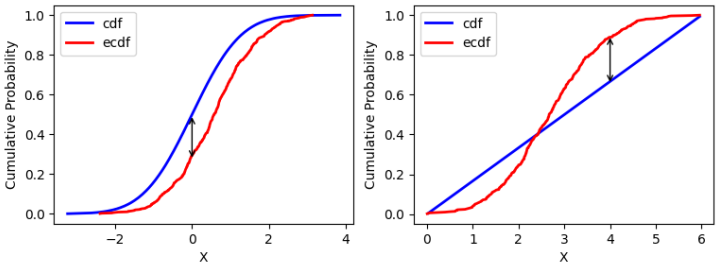


Figure 1.5: Illustration of the Kolmogorov–Smirnov statistic. Red line is the ecdf of two gaussian distributions, blue line is the cdf of a gaussian (left) and a uniform (right), and the length of the black arrow is the KS statistic.

probability distribution. It works well for distributions such as: normal, exponential, extreme-value, Weibull, Gamma, Logistic, Cauchy, etc. Kolmogorov-Smirnov test (KS test) is another statistical test which determines if a dataset comes from a given population. In our methods we use KS test, so we provide more details below.

Kolmogorov-Smirnov Test

The Kolmogorov–Smirnov (KS test) [12] is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample KS test) or to compare two samples (two-sample KS test).

The KS statistic quantifies a distance between the ecdf of the sample and the cdf of the reference distribution or between the ecdfs of two samples. The null distribution of this statistic is calculated under the null hypothesis that the sample is drawn from the reference distribution (in the one-sample case) or that the samples are drawn from the same distribution (in the two-sample case). The KS test can be modified to serve as a goodness of fit test. The goodness of fit of a statistical model describes how well it fits a set of observations. In the special case of testing for normality of the distribution, samples are standardized and compared with a standard normal distribution. The KS test is defined as:

H_0: The data follow a specified distribution.

H_a: The data do not follow the specified distribution.

The KS test statistic is defined as: $D_N = \sup_x |F_N(x) - F(x)|$, where $\sup_x$ is the supremum of the set of distances, $F_N(x)$ is the ecdf and $F(x)$ is the cdf (specified distribution).

The left plot of Fig. 1.5 illustrates the cdf and ecdf of two different gaussian distributions. The maximum distance between these two curves is indicated by the black arrow (KS statistic).

The KS test decides to reject the null hypothesis by comparing the p-value with the significance level α, not by comparing the test statistic with the critical value. Since the critical value is approximate, comparing the statistic with the critical value occasionally leads to a different conclusion than comparing p-value with α.

As we mentioned, p-value is the probability of observing a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. Small values of p cast doubt on the validity of the null hypothesis. Therefore, if p-value $\leq \alpha$, the KS test rejects the null hypothesis, otherwise, it accepts. KS test computes the critical value using an approximate formula or by interpolation in a table. The formula and table cover the range $0.005 \leq \alpha \leq 0.1$ for one-sided tests.

For testing uniformity, we use one-sample KS test with reference distribution being the uniform distribution. KS test decides if the ecdf's segments come from a uniform population (are uniformly distributed).

Similar to the left plot, the right plot of Fig. 1.5 illustrates the cdf of a uniform distribution and the ecdf of a gaussian distribution along with the KS statistic (black arrow).

An attractive feature of this test is that the distribution of the KS test statistic itself does not depend on the underlying cumulative distribution function being tested. Another advantage is that it is an exact test (e.g. the chi-square goodness-of-fit test depends on an adequate sample size for the approximations to be valid). However, KS test tends to be more sensitive near the center of the distribution than at the tails, which may affect the final decision.

1.1.2 Statistical Data Modeling

A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of similar data from a larger population [10]. A statistical model represents, often in considerably idealized form, the data-generating process. The assumptions embodied by a statistical model describe a set of probability distributions, some of which are assumed to adequately approximate the distribution from which a dataset is sampled. In simple terms, statistical modeling is a simplified,

mathematically-formalized way to approximate reality (i.e., what generates our data) and optionally to make predictions from this approximation. The statistical model is the mathematical equation that is used. It should summarize the data as closely as possible (be "a good fit") but also be as simple as possible. We cannot measure a population, so the best we can do is generalize from a sample to a population using a representative summary, i.e., a statistical model. Fitting a model to data means choosing the statistical model that predicts values as close as possible to the ones observed in our population. We need to find the values for the parameters in the model that are most appropriate to predicting the data. More details are provided below regarding the widely used Gaussian model, the uniform model and the Gaussian mixture model.

The Gaussian Model

The Gaussian model [9] is a widely used statistical model which works well as a good fit of many sets of data. It is often the case that we don't know the parameters of the Gaussian distribution, but instead want to estimate them. That is having a sample $\{x_1, ..., x_n\}$ from a Gaussian $N(\mu, \sigma^2)$ population we would like to learn the approximate values of parameters μ and σ^2. The standard approach to this problem is the Maximum Likelihood Estimation (MLE) method. The principle is called the maximum likelihood principle because, given a set of data, the probability of the data, regarded as a function of the parameters, is called a likelihood function. MLE requires maximization of the log-likelihood function. For the Gaussian model the maximum likelihood estimates are:

$$\hat{\mu} = \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{1.2}$$

Modeling our data with a Gaussian model needs estimation of the parameters of the Gaussian distribution. Estimator $\hat{\mu}$ is called the sample mean, since it is the arithmetic mean of all observations. The estimator $\hat{\sigma^2}$ is called the sample variance, since it is the variance of the sample $\{x_1, ..., x_n\}$. In practice, another estimator is often used instead of the $\hat{\sigma^2}$. This estimator is denoted s^2, and is also called the sample variance, which represents a certain ambiguity in terminology; its square root s is called the sample standard deviation. The estimator s^2 differs from $\hat{\sigma^2}$ by having

$(n-1)$ instead of n in the denominator.

$$s^2 = \frac{n}{n-1}\hat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{1.3}$$

The Gaussian model is very popular and fits well for several real datasets. A significant disadvantage, though, is its failure in asymmetric distributions. Fig. 1.6 illustrates six examples of a Gaussian Model fitting in samples of six distributions. In the top row, samples from Gaussian, Student's t and Gamma distributions are illustrated, while in the bottom row, the samples are generated by a triangular, asymmetric triangular and a mixture of a uniform and Gaussian distribution, respectively. For symmetric distributions, such as the Gaussian or triangular, the Gaussian model provides accurate fit. However, for asymmetric distributions (e.g., Gamma, asymmetric triangular, and the mixture of uniform and Gaussian), it fails to fit effectively.

The Uniform Model

Another model which can model our dataset is the uniform model. Similar to the Gaussian Model, the parameters of the Uniform model (a and b) are estimated through the maximum likelihood method. In Fig. 1.7, we work with the same distributions as in Fig. 1.6. It is evident that the uniform model lacks the flexibility needed to fit these samples accurately.

The Gaussian Mixture Model

A mixture distribution [7, 13] is the probability distribution of a random variable that is derived from a collection of other hidden random variables as follows: first, a random variable is selected by chance from the collection according to given probabilities of selection, and then the value of the selected random variable is realized. The underlying random variables may be random real numbers, or they may be random vectors (each having the same dimension), in which case the mixture distribution is a multivariate distribution.

In cases where each of the underlying random variables are continuous, the outcome variable will also be continuous, and its probability density function is sometimes referred to as a mixture density. The cdf (and the pdf) can be expressed as a convex combination (i.e., a weighted sum, with non-negative weights that sum to 1) of other distribution functions and density functions. The individual distributions

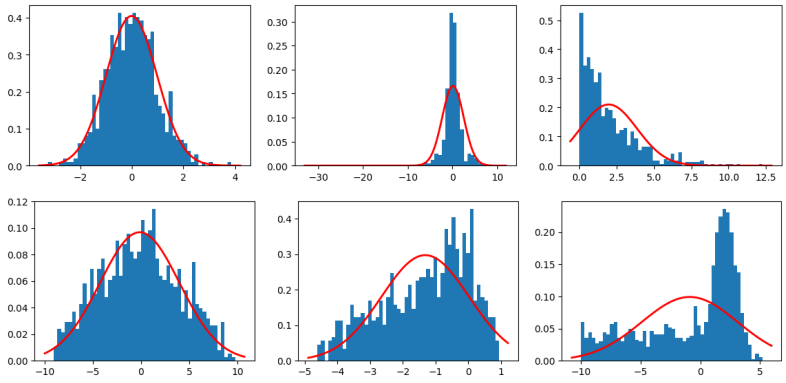


Figure 1.6: A Gaussian model (red curve) fits in several datasets.

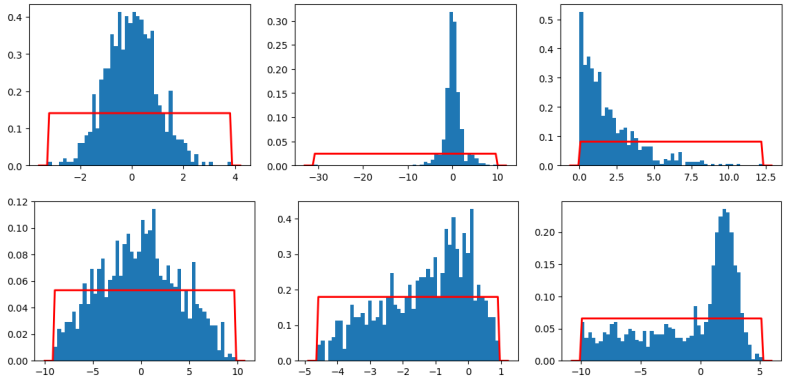


Figure 1.7: A uniform model (red curve) fits in several datasets.

that are combined to form the mixture distribution are called the mixture components, and the probabilities (or weights) associated with each component are called the mixture weights.

Given a finite set of pdfs $p_1(x), ..., p_K(x)$, or corresponding cdfs $P_1(x), ..., P_K(x)$ and weights $w_1, ..., w_K$ such that $w_i \geq 0$ and $\sum_{i=1}^{K} w_i = 1$, the mixture distribution can be represented by writing either the density (f), or the distribution function (F), as a sum (which in both cases is a convex combination):

$$f(x) = \sum_{i=1}^{K} w_i p_i(x), \quad F(x) = \sum_{i=1}^{K} w_i P_i(x) \tag{1.4}$$

Mixture distributions arise in many contexts in the literature and arise naturally when a statistical population contains two or more subpopulations. It is frequently the case that data is not explained by a single underlying distribution. Typically, this is because there are multiple phenomena occurring in the dataset, each with their own underlying distribution. If we want to try to recover the underlying distributions, we need to have a model which has multiple components. An example could be sensor

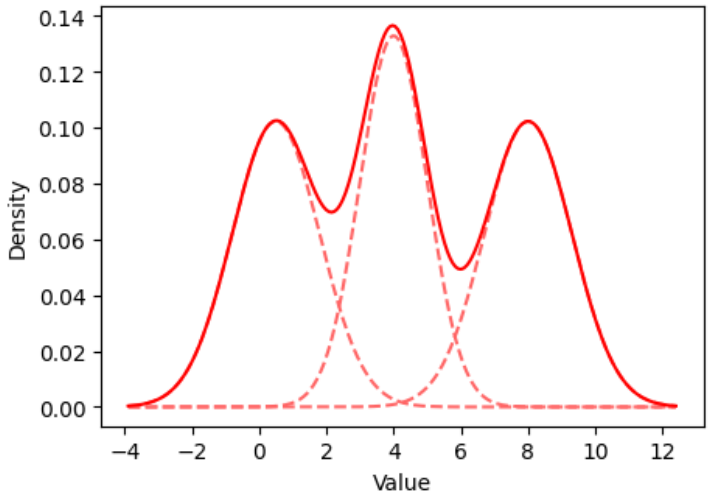


Figure 1.8: A Gaussian Mixture of three Gaussian distributions.

readings where the majority of the time a sensor shows no signal, but sometimes it detects some phenomena. Modeling both phenomena as a single distribution would be inaccurate because the readings would come from two distinct phenomena.

In such cases we utilize mixture models for modeling tasks, which rely on the assumption that the data has been generated by sampling from a set of component distributions. A common type of mixture model, called Gaussian Mixture Model (GMM) [13], is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Fig. 1.8 illustrates a GMM of three Gaussian distributions.

Given a statistical model for the data, it is necessary to estimate the parameters of that model. A standard approach used for this task is MLE (similarly to the Gaussian model). To begin, consider a set of n points $X = \{x_1, ..., x_n\}$ that are generated from a one-dimensional Gaussian distribution. Assuming that the points are generated independently, the probability of these points is just the product of their individual probability densities. Using the above Equation we can write this probability density as follows:

$$p(X|\Theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{1.5}$$

Since this probability would be a very small number, we typically will work with the log probability:

$$\log p(X|\Theta) = -\sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} - 0.5n \log 2\pi - nlog\sigma \tag{1.6}$$

In order to estimate the parameters μ and σ the MLE approach is followed. In general, we do not know which points were generated by which distribution. Thus, we cannot directly calculate the probability of each data point, and hence, it would seem that we cannot use the maximum likelihood principle to estimate parameters. The solution to this problem is the Expectation - Maximization (EM) algorithm. Briefly, given a guess for the parameter values, the EM algorithm calculates the probability that each point belongs to each distribution and then uses these probabilities to compute a new estimate for the parameters. This iteration continues until convergence. Thus, we still employ maximum likelihood estimation, but via an iterative search. Overall, mixture models are flexible in treating data of different characteristics; however a major problem is the specification of the number of mixture components, which should be specified by the user.

1.2 Unimodality

In mathematics, science, and engineering, one often encounters data that can be modeled as a sample from some unknown underlying distribution. From this limited data sample, we wish to learn about the various characteristics of the underlying distribution, which then gives information about the dynamics of the system that is being examined. One such characteristic is whether the underlying distribution is *unimodal* or *multimodal*: whether it has one or several maxima [14], where the probability density is (locally) maximal. Such maxima are called *modes*, and can manifest themselves in samples from the distribution in the form of clusters: intervals where a large number of data points are concentrated. This can often be observed when a histogram is drawn from the data, where a large concentration of data points appears as a hill near the mode, i.e. the data has a "grouping behavior". In particular, a multimodal underlying distribution can imply that there are two or more separate groups present in the data, which behave differently from one another. For example, suppose one observes two modes in the distribution of the incubation times of a disease. This could point towards there being two different strains of this disease that behave differently, where on average, one strain has a larger incubation time compared to the other strain. It could also point towards a certain subgroup of patients being more resistant to the disease than others, and several other hypotheses could be formulated. In contrast,

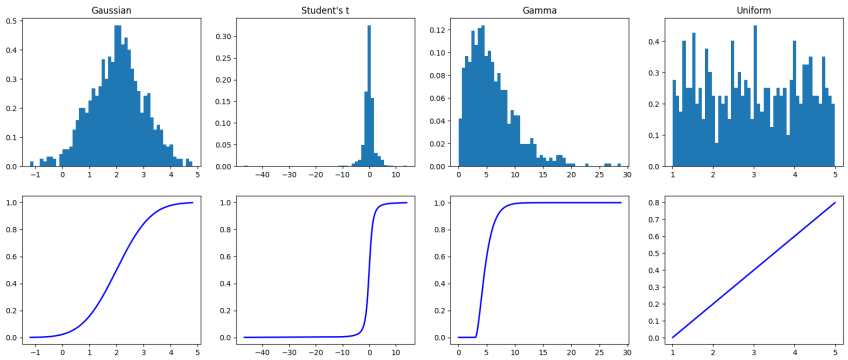


Figure 1.9: Histogram plots of unimodal distributions (top row) and corresponding cdf plots (bottom row).

a histogram with a single mode highlights the most common value; for example, the heights of adult males are often grouped around a single peak. Either way, the observation of unimodality/multimodality leads to several novel research questions, which contributes to a greater understanding of the dynamics at play.

1.2.1 Unimodality Definition

In what concerns the unimodality [15] of a distribution there are two definition options. The first relies on the probability density function (pdf). A pdf is unimodal, if it has a single mode; a region where the density becomes maximum, while non-increasing density is observed when moving away from the mode. In other words, a function $f(x)$ is a unimodal function if for some value m , it is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$. In that case, the maximum value of $f(x)$ is $f(m)$ and there are no other local maxima. The most widely used unimodal distribution function is the Gaussian. Other examples of unimodal functions are Student's t, Gamma, Chi-square, triangular, Cauchy and exponential and etc.

The second definition option relies on the cumulative distribution function (cdf). A cdf is unimodal if there exist two points x_l and x_u such that the function can be divided into three parts: a) a convex part $(-\infty, x_l)$, b) a constant part $[x_l, x_u]$ and c) a concave part (x_u, ∞). It is worth mentioning that it is possible for either the first two parts or the last two parts to be missing. Fig. 1.9 illustrates the histogram (top row) and cdf (bottom row) plots of four unimodal functions. In the cases of Gaussian and Student's t, we clearly see first a convex part, after a linear, and last a concave part

on the cdf plots. In the case of Gamma function, the convex part is small enough, but it still remains unimodal according to unimodality's definition. The single modes in the corresponding histogram plots of these functions are evident. We should clear that the uniform distribution is an extreme single mode case where the mode covers all the region with non-zero density. The cdf plot only contains the linear part of unimodality's definition as it is shown in Fig. 1.9.

On the other hand, a non-unimodal distribution is called multimodal with two or more modes. A common case is when a distribution has only two modes; these appear as distinct peaks (local maxima) in the pdf plot. The distribution with exactly two modes is called bimodal, while the distribution with exactly three modes is called trimodal. A bimodal distribution most commonly arises as a mixture of two different unimodal distributions (i.e. distributions having only one mode). For example, a mixture of two Gaussian distributions with the same variance, but different means is a bimodal distribution.

In Fig. 1.10, we present the histogram and cdf plots of four multimodal distributions. In the first two columns, the histogram plots of two Gaussians (showing the presence of two modes) are illustrated along with the corresponding cdf plots. In the cdf plots, we observe two convex parts separated by a concave region (or equivalently, two concave parts separated by a convex region). This pattern is more pronounced in the case of the two widely-separated Gaussians (second column). The existence of a second convex (or concave) part violates the definition of unimodality, indicating that the functions in these plots are multimodal (specifically, bimodal). This phenomenon becomes even more pronounced in the third (mixture of three Gaussians) and fourth (mixture of three Gaussians and a uniform distribution) columns. Three and four modes are evident in the histogram plots, respectively, while the presence of multiple alternating convex and concave regions is clearly visible in the corresponding cdf plots.

1.2.2 Assessing Unimodality

Visual Inspection

We could consider "visual inspection" as a method in order to recognize unimodal distributions, however this methodology is not always sufficient to test for multimodality. In everyday life, when one suspects a distribution might be multimodal,

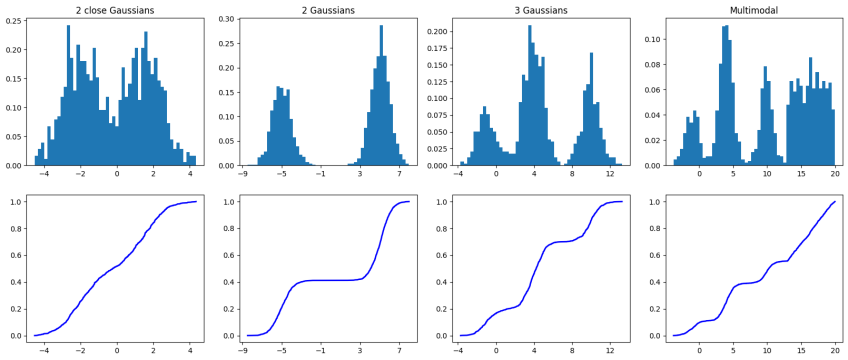


Figure 1.10: Histogram plots of multimodal distributions (top row) and corresponding cdf plots (bottom row).

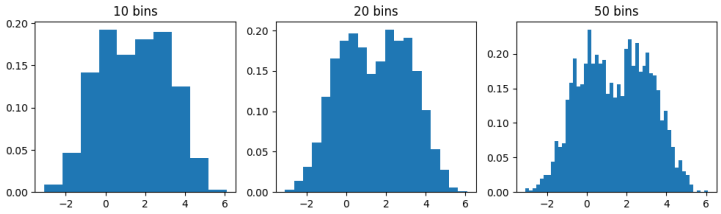


Figure 1.11: Histograms of a distribution function with different number of bins.

one might simply draw a histogram (or kernel density plot), and visually inspect the maxima of this histogram. If several disjoint intervals contain a large concentration of data points, one might conclude the distribution is multimodal. Conversely, if only one maximum is shown (or the histogram does not show clear maxima at all), one might conclude the distribution is unimodal.

While this approach can certainly give correct results, it has several disadvantages. Firstly, one has to properly choose the bin size (or kernel width): if the bin size is chosen to be too large, several present modes can be combined into a seemingly single mode. Conversely, if the bin size is chosen to be too small, extraneous maxima that do not correspond to modes in the distribution can be introduced, that appear due to randomness in the finite sample. Choosing a "correct" bin size is often a somewhat subjective matter.

In Fig. 1.11 we see the histograms of the same pdf for three different numbers of bins (10, 20, 50). This number affects significantly the plot and if we depend on the number of bins, we may end up in different conclusions.

Secondly, even for a well-chosen bin size, it can still be subjective whether an observed maximum is an extraneous maximum or a mode that truly appears in the underlying distribution: a small hump in the histogram can represent a weak mode

in the distribution, but can also be a point where several data points are concentrated purely by chance. Finally, conclusions drawn from this method are highly qualitative in nature, since no quantitative confidence level can be assigned. In many occasions, this makes it insufficiently reliable to use in science and industry.

Although the definition of unimodality in the case of pdf functions is sufficiently comprehensible and simple, the shape of a histogram - in which we visualize a pdf - varies depending on the number of bins. In contrast, a cdf plot is independent of any parameters, even though defining unimodality for cdf functions may be a little tricky. Thus, the cdf has the clear advantage over the pdf, as a more stable and handier tool, and it is primarily utilized in the methods proposed in this thesis.

Moreover, since the underlying distribution function is not known, and we work with sample observations, ecdf is favored over cdf in this thesis. The ecdf is useful, since it approximates the true cdf well if the sample size (the number of data) is large, and knowing the distribution is helpful for statistical inference. Also, a plot of the ecdf can be visually compared to known cdfs of frequently used distributions to check if the data came from one of those common distributions.

Unimodality Tests

A reliable way to assess data unimodality are statistical tests, called unimodality tests, that are used for discovering the presence of more than one mode in a distribution [15]. In other words, they are used to decide whether a set of data points has been generated by a probability distribution with a single mode (peak). The unimodality property is directly related to the grouping behavior of points, i.e. whether data are 'gathered' or not.

The most typical example of unimodality is normality (or Gaussianity), which can be tested using several well-known tests, for example the Anderson-Darling test [16] and Shapiro-Wilk test [17]. For this reason, in several data analysis methods, the normality test has been used to check the grouping behavior of data. One test, suggested in [18] uses the likelihood ratio for a two-component normal mixture against the normal null hypothesis. A related test [19] divides the sample into two subsets to maximize the likelihood ratio that the two subsets are sampled from normal with different means, against the null hypothesis that the means are equal. It is obvious that the employment of normality tests to check unimodality relies on a crude assumption, since there are many datasets whose density (e.g. histogram) has

a single peak (i.e. they are unimodal) but its shape does not resemble the shape of the normal distribution. It is obvious that in such cases a normality test will fail. Therefore, the development of general unimodality tests offers great advantage compared to normality tests, since it allows to test the "gathering property" of data without focusing on a particular functional form (e.g. Gaussian, Student's t, uniform, etc.).

Many statistical tests to assess unimodality (or multimodality) have been proposed in the literature, for example a nice survey is presented in [1] (one may also refer to [20] for earlier contributions). The approaches commonly test unimodality versus bimodality, but they can be adapted to test unimodality versus k modes.

In [21] a test is suggested for multimodality, called k-critical windows. The idea is the following: The smallest window width is used, such that the resulting kernel density estimate is unimodal, as a test statistic for unimodality. A sample from a density with more than k modes will require more smoothing to exhibit k or less modes in the density estimate compared to a sample from a density with exactly k modes. The significance level of the test statistic is evaluated by empirically sampling from a rescaled version of the unimodal density estimate. It applies kernel density estimation with Gaussian kernel and relies on the kernel bandwidth to decide on unimodality. Note that kernel bandwidth is related to the amount of smoothing. If high bandwidth (i.e. large smoothing) is needed to obtain a unimodal estimate, this is an indication of multimodality. The above idea is well-studied and several weaknesses have been identified [22]. Another 1-d unimodality test is the excess mass test [23] that measures the excess mass of the modes, i.e. the amount of density (as estimated by a histogram) that is above a specific level L. If this excess mass is distributed in several regions, then this is an indication of multimodality. Even if the test is theoretically designed to tackle multivariate densities, general effective algorithms are not available [24].

The RUNT test [25] and the MAP test [26] constitute attempts to address the unimodality issue in multiple dimensions. RUNT test is based on single linkage clustering, while MAP test uses minimum trees with additional constraints, thus both approaches are computationally expensive.

The *Hartigans' dip-test* [27] is a notable test statistic which decides on the unimodality of a real-valued dataset. It takes as input an 1-d dataset, examines the underlying ecdf of the set of numbers and decides whether it contains a single or

more than one mode (peak). Specifically, it computes the dip statistic as the maximum difference between the ecdf, and the unimodal distribution function that minimizes that maximum difference. The uniform distribution is the asymptotically least favorable unimodal distribution, and the distribution of the test statistic is determined asymptotically and empirically when sampling from the Uniform.

Given a set of real numbers $X = \{x_1, ..., x_n\}$ the dip-test computes the dip value $dip(X)$, which is the departure from unimodality of the ecdf. For bounded input functions F, G, let $\rho(F, G) = \max_x |F(x) - G(x)|$, and let U be the class of all unimodal distributions. Then the dip statistic of a distribution function F is given by: $dip(F) = \min_{G \in U} \rho(F, G)$.

In other words, the dip statistic computes the minimum among the maximum deviations observed between the cdf F and the cdfs from the class of unimodal distributions. A nice property of dip is that, if X is a sample distribution of n observations from F, then $\lim_{n\to\infty} dip(F_n) = dip(F)$. It is argued that the class of Uniform distributions U is being used for the null hypothesis, since its dip values are stochastically larger than other unimodal distributions, such as those having exponentially decreasing tails. Dip-test also has the benefit of not requiring a kernel width.

Given a dataset $X = \{x_1, ..., x_n\}, x_i \in \mathbb{R}$ the dip statistic is computed as follows:

i. Begin with $x_L = x_1, x_U = x_n, D = 0$.

ii. Compute the gcm G and lcm L for F in $[x_L, x_U]$; suppose the points of contact with F are respectively $g_1, g_2, ..., g_k$ and $l_1, l_2, ..., l_m$.

iii. Suppose $d = \sup |G(g_i) - L(g_i)| > \sup |G(l_i) - L(l_i)|$ and that the sup occurs at $l_j \leq g_i \leq l_{j+1}$. Define $x_L^0 = g_i, x_U^0 = l_{j+1}$.

iv. Suppose $d = \sup |G(l_i) - L(l_i)| > \sup |G(g_i) - L(g_i)|$ and that the sup occurs at $g_i \leq l_j \leq g_{i+1}$. Define $x_L^0 = g_i, x_U^0 = l_j$.

v. If $d \leq D$, stop and set $D(F) = D$.

vi. If $d > D$, set $D = \sup\{D, \sup_{x_L \leq x \leq x_L^0} |G(x) - F(x)|, \sup_{x_U^0 \leq x \leq x_U} |L(x) - F(x)|\}$

vii. Set $x_U = x_U^0, x_L = x_L^0$ and return to ii.

A graphical example is given in Fig. 1.12 [1].

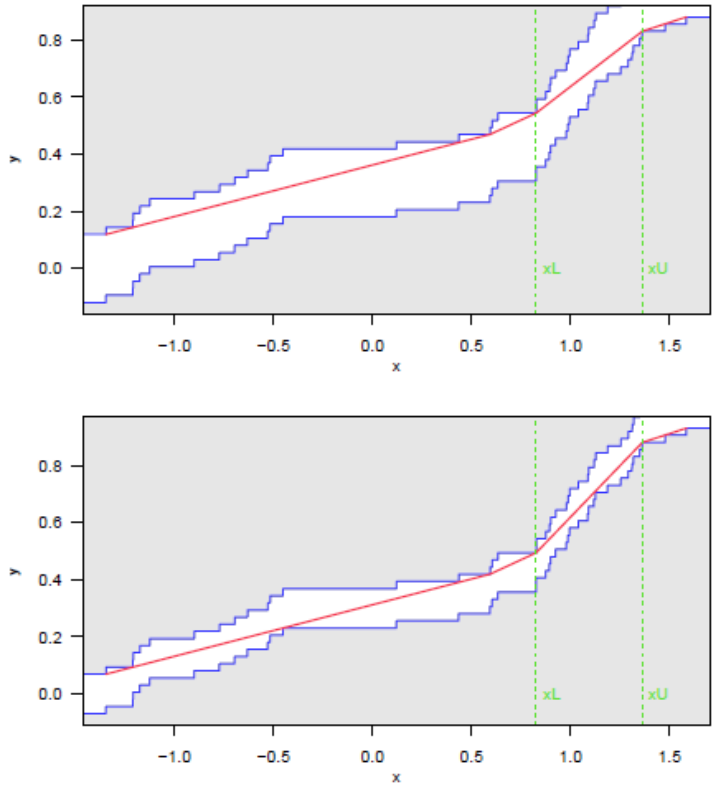


Figure 1.12: Graphical example of the taut string metaphor used in describing the algorithm for the dip-test. The red line is the string, the bottom blue line is $F - d$ and the upper blue line is $F + d$. Note that the string forms the gcm on (x_1, x_L) for $F - d$, and the lcm on (x_U, x_n) for $F + d$. The bottom plot depicts the minimum value of d. If we would decrease d even further, the string would get bent out of its unimodal shape at around $x \approx 0.5$ [1].

Dip-test examines the $n(n-1)/2$ possible modal intervals $[x_L, x_U]$ between the sorted n individual observations. For all these combinations it computes in $O(n)$ time the respective gcm and the lcm curves in $(\min_x X_n, x_L)$ and $(x_U, \max_x X_n)$, respectively. Fortunately, for a given X_n, the complexity of one dip computation is $O(n)$. The dip-test returns not only the dip value, but also the statistical significance of the computed dip value, i.e. a p-value. The computation of the p-value for a unimodality test uses bootstrap samples and expresses the probability of $dip(X_n)$ being less than the dip value of a cdf U_n^r of n observations sampled from the $U[0,1]$ Uniform distribution:

$$P = \#[dip(X) \leq dip(U_n^r)]/b, \quad r = 1, ..., b \tag{1.7}$$

It should be stressed that for each value of n, the bootstrap samples U_n^r do not

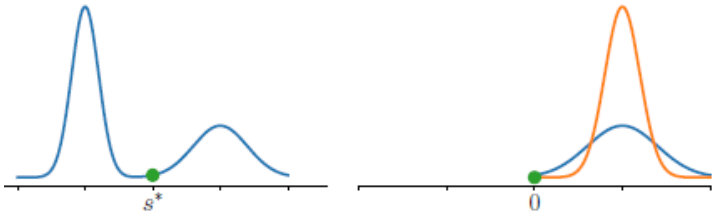


Figure 1.13: Folding mechanism for univariate distribution. Initial (left) and folded (right) distribution are provided [2].

depend on the dataset X, therefore they can be computed only once, along with the corresponding values $dip(U_n^r)$. The null and alternative hypothesis, H_0 and H_a, are given below:

H_0: X_n is unimodal $\quad$ H_a: X_n is multimodal

H_0 is accepted at significance level α if p-value $> \alpha$, otherwise H_0 is rejected in favor of the alternative hypothesis H_a, which suggests multimodality.

In [2] a multivariate unimodality test is proposed, called *folding test*, which makes no distribution assumption and utilizes only a p-value. Given a multidimensional dataset of numerical attributes, the authors wonder about the "grouping behavior" of the data points. In [2] it is argued that in unimodal cases, the data points make a single peak in the histogram. It is obvious that if this fact is known, we will not proceed in a clustering method, since our data points make exactly one group (cluster).

The approach of the folding test relies on a folding technique and is the following: (1) fold up the distribution (left plot of Fig. 1.13) with respect to a pivot $s^\star$, (2) compute the variance of the folded distribution and (3) compare it with the initial variance. The main idea is that the resulting density (right plot of Fig. 1.13) of the folded distribution will have a far lower variance in multimodal distributions, while this phenomenon will not appear in unimodal cases (i.e. not with the same amplitude).

The folding step is performed with the transformation $X \mapsto |X - s^\star|$ and the folding ratio is computed as:

$$\phi(X) = \frac{Var|X - s^\star|}{VarX} \tag{1.8}$$

In higher dimensions the absolute value is replaced by the Euclidean norm $Var\|X - s^\star\|$ where X is a random vector of $\mathbb{R}^d$ and $s^\star \in \mathbb{R}^d$ is the pivot. The variance

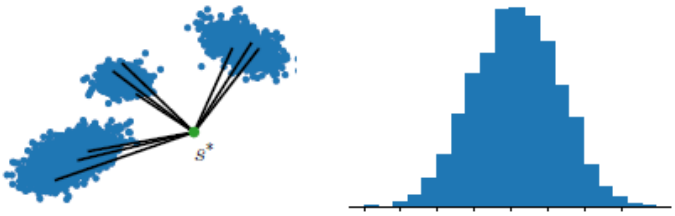


Figure 1.14: Folding mechanism in dimension 2. Initial (left) and folded (right) distribution are provided [2].

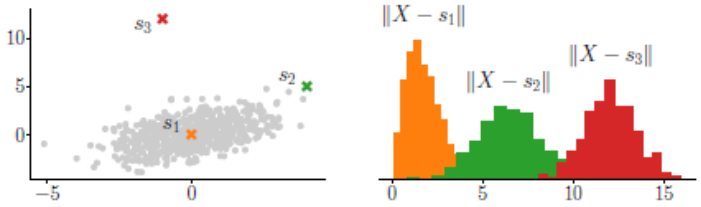


Figure 1.15: Impact of the pivot location. Initial (left) and folded (right) distribution are provided [2].

is replaced by $E[\|X - E[X]\|^2]$. In the unimodal case, this expected value will be much lower than in the multimodal case. Finally, the folding ratio is generalized through:

$$\phi(X) = \frac{Var\|X - s^\star\|}{E[\|X - E[X]\|^2]} \tag{1.9}$$

Fig. 1.14 gives an empirical example of the folding mechanism in two dimensions.

To this end, more details should be given about pivot. According to [2], the right pivot should be found, so the variance can be significantly reduced through the folding process. Thus, the pivot should reduce the variance the most (if such a pivot exists). It is mentioned that the best pivot $s^\star$ is likely to be close to the mode in the unimodal case, while is likely to stand "between" the modes in the multimodal case.

The best pivot $s^\star$ is found as: $s^\star = \arg\min_{s \in \mathbb{R}^d} Var\|X - s\|$. The level of confidence of the test is computed with a p-value. The lower p-value is, the more significant the decision will be. A unimodal example in $\mathbb{R}^2$ is given in Fig. 1.15.

1.2.3 Significance of Unimodality Tests

The concept of unimodality and the application of unimodality tests have been utilized across various machine learning domains, including clustering, density estimation,

and feature selection. In data analysis it is of great importance to discover information about the structure in data. There are significant topics, such as clustering, which are only appropriate when cluster structure is present. In [28] it is presented in a nice way why a unimodality test is so important in cluster analysis. First, the meaning of clusterability needs to be mentioned. Clusterability [28] depends on the presence of inherent structure and aims to quantify the degree of cluster structure. This analysis should precede the application of clustering algorithms, as the success of any clustering algorithm depends on the presence of underlying cluster structure. If such a structure exists, the next step of choosing a clustering algorithm will follow. In other cases, the results of any clustering technique become arbitrary and potentially misleading, so clustering should possibly not be applied.

For concreteness, consider a dataset randomly generated from a single Gaussian distribution. Because the data contains only one cluster, it makes no sense to divide the dataset into clusters. Most clustering algorithms (e.g. k-means with $k \geq 2$) would find multiple clusters in the data, even though no multi-cluster structure is present. Before the clustering algorithm, we could have checked with a unimodality test, if the dataset makes one single and coherent cluster or not. In other words, we would know that the data are homogeneous, so clustering would not be suitable in this case.

On the other hand, if a dataset contains multiple clusters, then there should be some separation between the clusters. For example, consider a dataset randomly generated from two Gaussian distributions with the two means being extremely different. There are two peaks in the Gaussians' histogram, so there are two clusters in the dataset and a unimodality test will decide multimodality. Thus, running a unimodality test before the clustering algorithm ensures us that the cluster structure is present in the dataset. For multidimensional data, these unimodality methods face a significant limitation, making their performance unpredictable for real-world datasets, which often have multiple or high dimensions. Their application typically requires reducing the data to a single dimension beforehand.

Exploitation of Unimodality Tests in Various Scientific Fields

Although unimodality tests are primarily applied to 1-dimensional data, the dip-test has also been utilized for multidimensional datasets. In such cases, dataset unimodality can be evaluated by conducting multiple 1-dimensional tests. Below, we present several multidimensional methods that leverage the dip test to assess data unimodal-

ity.

The dip-means algorithm [29] integrates the dip statistic with k-means to evaluate cluster homogeneity using the dip-dist criterion, which relies on Hartigan's dip-test. Instead of testing unimodality in the original data space, the dip-dist criterion assesses the density distribution of pairwise distances among data objects. Each object acts as a "viewer", determining unimodality or multimodality based on the dip test applied to its pairwise distances. If the majority of viewers detect multimodality, the cluster is deemed multimodal; otherwise, it is considered unimodal. When k-means produces a non-unimodal cluster, dip-means reruns the algorithm with an additional cluster, refining the clustering process.

In [30], Hartigan's dip-test is utilized in the DipTransformation, a nearly parameter-free method that enhances dataset structure, improving k-means clustering and other clustering techniques. By leveraging the dip statistic as a measure of a dimension's structure and relevance, the method scales more relevant dimensions to increase their impact on clustering. This deterministic algorithm requires no distance calculations or distributional assumptions, making it a versatile and effective tool for improving clustering performance.

In [31], the dip statistic is applied in projection pursuit, a method for identifying low-dimensional projections of high-dimensional data that are "interesting". This approach is critical for exploratory data analysis, visualization, and addressing the "curse of dimensionality" in machine learning. Unlike traditional projection indices that focus on non-Gaussianity, the dip measures distance from unimodality, offering a more generalized criterion. Efficient algorithms are introduced to maximize the dip for detecting multimodal data projections and extended for finding higher-dimensional projections through two strategies: iterative orthogonal searches and recursive procedures that remove interesting structures to achieve unimodality. According to their experiments, they demonstrate the dip's robustness, effectively identifying informative directions even in high-dimensional spaces with minimal preprocessing.

Unimodality and unimodality tests, apart from their applications in machine learning, have been utilized across various scientific fields, including ecology [32, 33], biology [34, 35], and economics [36, 37, 38]. In ecology, unimodality is central to the humped-back model (HBM), which explains plant species richness as peaking at intermediate productivity due to a balance between abiotic stress in unproductive ecosystems and competitive exclusion in highly productive ones. In [32] regression

models are employed to confirm a significant unimodal relationship between plant richness and productivity using a negative binomial generalized linear model (GLM). In biology, unimodality tests like Hartigans' dip-test [27] and Silverman's test [21] are applied in cytometry for identifying unimodal cell populations. These tests support automated gating algorithms by distinguishing between unimodal and bimodal density distributions with low error rates, providing an objective alternative to manual gating [34].

1.3 Mode Estimation

The concept of *mode* is central to statistical analysis, particularly when dealing with the distribution of data. In its simplest form, the mode refers to the most frequently occurring value in a dataset. However, when applied to continuous data, this traditional definition becomes less useful, as each data point is typically unique. For continuous random variables, the mode is better understood as the value that maximizes the pdf.

The importance of mode increases when dealing with multimodal distributions, which often indicate the presence of multiple subpopulations within the data. In such cases, both the mean and median may fail to provide an accurate measure of central tendency. Even the global mode may not effectively represent a central value for the entire distribution. This highlights the importance of conducting a "modal analysis" to fully understand the data's structure. Key steps in this analysis include *identifying all modes* or *estimating valleys*, and, in the case of multimodal distributions, modeling each subpopulation using a mixture density approach.

Modes and valley estimation are particularly useful in scenarios such as image segmentation or cluster analysis. In these applications, modes are often detected by analyzing histograms or density estimates of the data. Valley points, which correspond to local minima between modes, can serve as natural dividing points that separate the different modes or clusters. Identifying these valleys accurately is important for defining mixture models, where the goal is to model the data as a combination of multiple distributions.

The process of mode detection, however, is not without challenges. In real-world data, noise and skewness can distort the clear identification of modes and valleys. Furthermore, determining the number of modes is often not straightforward, as it may

depend on the underlying structure of the data. As a result, several methods have been developed to address these challenges, including mixture modeling and unimodality-based methods aiming to identify peaks (modes) and valleys (split points) in a data density.

1.3.1 Mixture models for Density Estimation and Clustering

As mentioned in Section 1.1.2, mixture models are essential for gaining meaningful insights into the underlying distribution. GMMs are popular models, using the Gaussian distribution to model each mixture component. In this context, the density is estimated using a parametric model, and the modes are found by identifying the local maxima of the estimated density. The EM algorithm is commonly used to estimate the parameters of these distributions, allowing for efficient mode detection. A key limitation of parametric approaches (such as GMM) is their reliance on the assumption that the data follows a known distribution. In many real-world cases, especially in image processing or other high-dimensional data, this assumption may not hold. For instance, histograms of natural image data may not follow a Gaussian mixture, which can lead to poor performance when using methods like GMM for mode estimation.

A recent extension is proposed in [39], where GMMs carry out density estimation not on the original data but on appropriately transformed data in case of bounded variables. The basic idea is to use an invertible function to map a bounded variable to an unbounded support, estimate the density of the transformed variable, and then back-transform to the original scale. For particular applications, mixtures of distributions other than Gaussian have been explored for clustering. For example, in [40] a two-way mixture model of Poisson distributions is proposed for document classification and word clustering, while in [41] a mixture of Mises-Fisher distributions is used to cluster data on a unit sphere.

Many clustering algorithms assume that multiple modes indicate multiple clusters, while unimodality is a sign for a single cluster. Several classical clustering algorithms are based on partitioning the space around a pre-fixed number of central points (these are usually called partitioning methods, and include k-means clustering, for instance). In the recent times, however, there is a growing body of researchers that advocate that "density needs to be incorporated in the clustering procedures" [42].

In this spirit, mixture models have been successfully used for data clustering [43], where data points are assigned to clusters based on the component distribution that most probably generated them. The clustering procedure involves fitting a mixture model, often using the EM algorithm, and assigning each data point to the component with the highest posterior probability. Another density-based approach is clustering based on high density regions [44]. In the last approach (also characterized as modal clustering [45, 46]) the clusters are taken as the "domains of attraction" of the density modes.

1.3.2 Nonparametric Methods for Density and Mode Estimation

Nonparametric methods do not assume a specific underlying distribution. These methods estimate the data density directly from the data points and are more flexible in capturing the true structure of the data.

Kernel Density Estimation

Kernel Density Estimation (KDE) is a non-parametric method used to estimate the pdf of a dataset. By smoothing data points with a chosen kernel function, KDE provides a continuous estimate of the underlying distribution. For example, in case a Gaussian kernel is used, each data point contributes a bell-shaped curve to the overall estimate. In contrast to mixture models, the number of kernels is equal to the number of data points. Furthermore, the kernel function, the bandwidth of the kernel and other hyperparameters have to be chosen by the user. A small bandwidth may overfit the data, capturing spurious fluctuations and leading to too many detected peaks (modes), while a large bandwidth may smooth out important features of the data, resulting in an underestimation of the number of modes.

Mean Shift and Medoidshift

One widely used nonparametric method is mean shift [47, 48], a "mode seeking" clustering algorithm based on the idea of associating each data point to a mode of the underlying probability density function, which is modeled using kernel density estimation. The general idea is to shift each data point until it reaches its nearest peak of the data density, thus a cluster is formed around each peak. While mean shift is

effective in many cases, it often detects too many peaks in noisy data, leading to over-segmentation. The challenge here is distinguishing between true modes and spurious peaks, which are often caused by noise. Another major difficulty is that it includes a critical user defined hyperparameter which is the bandwidth of kernel function used in kernel density estimation. Medoidshift [49] follows a similar approach to mean shift also requiring the bandwidth of kernel function. Many attempts for a bandwidth selection have been made, however they do not always work successfully [50].

Density Peaks

Another popular mode seeking algorithm is density peaks [51], which detects clusters based on two simple and intuitive assumptions: cluster centers are usually in dense areas and are surrounded by points with lower density. The algorithm first calculates the local density around each point and then calculates the distance (called delta) of each point to its nearest point with higher density. The cluster centers are selected so that they have a high value of both delta and density. After that, the remaining points are allocated to the clusters by merging with the nearest higher density point. Similar to mean shift, the method requires the specification of several hyperparameters [52].

Modal Clustering Methods

Two recent methods have been proposed in [53] and [42] for modal clustering. In [53] the goal is to associate each data point with a local maximum, or mode. The method relies on specifying kernel density functions (specifically Gaussian kernels) and estimating each of them. Modal clustering is applied to the dataset upon mixture density estimation. The exact mechanism is an EM-type nonparametric algorithm called Modal EM (MEM) [53] that allows finding "hilltops" of the given density. The suggested algorithm is then extended to hierarchical clustering by recursively locating modes of kernel density estimators with increasing bandwidths. An extended version of MEM algorithm is proposed in [54] to deal with any parsimonious component-covariance matrix decomposition. Furthermore, a fast implementation of the algorithm is discussed that allows to perform the M-step simultaneously for all data points. Once the modes of the underlying density are estimated, a modal clustering partition can be obtained by associating each observation to the pertaining mode.

In [42] the "modclust" methodology is presented which combines modal clustering

with mixture modeling. Specifically, it applies the modal clustering methodology to a density estimate obtained by fitting a mixture model to the data. By applying the EM algorithm to find the maximum likelihood estimates of the parameters and mixing weights, and the Bayesian Information Criterion (BIC) [7] to select the number of components, a density in the form of a GMM is fitted to the data. Then, the mean shift algorithm is used on the Gaussian mixture density to find the domains of attraction of the estimated density modes. The methods proposed in [53] and [42], as any other mode-seeking procedure, relies on the quality of the underlying density estimate. Clearly, if the parameters of the mixture model are not well estimated, several issues could arise. A review on non-parametric modal clustering is given in [46].

Unimodality-Based Methods

SkinnyDip [3] is a clustering method inspired by Hartigans' dip-test of unimodality. SkinnyDip offers a compelling set of features: it is highly resistant to noise, nearly parameter-free, and fully deterministic. Unlike traditional methods, SkinnyDip avoids multivariate distance calculations, instead employing insightful recursion through "dips" into univariate projections of the data. It can identify various cluster shapes and densities, provided that each cluster exhibits a unimodal distribution. In [3] a dip-based heuristic solution (called UniDip) is proposed for univariate clustering. It operates recursively and mirrors the mechanism used by the dip-test to compute its statistic, isolating one mode at a time from the sample.

Let a sorted univariate sample $\{x_1, ..., x_n\}$ of size n. To be more clear, we also present a histogram plot of the sample, as illustrated in Fig. 1.16 [3]. At first the dip-test is applied on the initial sorted sample. The resulting p-value indicates, based on a significance level α, that the distribution has at least two modes. Additionally, the dip-test identifies the modal interval $[x_L, x_U]$ for this sample, which in this context, represents the interval for ecdf's prescribed maximum constant slope. In this example, the initial modal interval encompasses modes C, D, and E (the gray region in Fig. 1.16). These modes are grouped into one interval because of their proximity in the ecdf, allowing for a minimal dip. A recursion into this interval results in extracting the individual modes C, D, and E, yielding three distinct modal intervals $[x_{L_C}, x_{U_C}], [x_{L_D}, x_{U_D}], [x_{L_E}, x_{U_E}]$.

Since there is at least one more mode, the search continues. The next mode must be outside the gray interval $[x_L, x_U]$. If a mode exists to the right, then dipping over

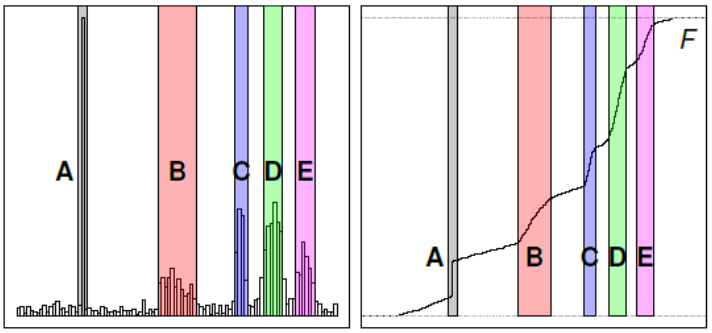


Figure 1.16: Histogram and ecdf of univariate data. The detected modes are also presented [3].

$[x_{L_E}, x_n]$ (including mode E and everything to its right) will produce a significant result. Choosing this interval is crucial. If the search is focused over $[x_U, x_n]$ (everything right of the gray region), and this region contained only uniform noise, the dip test would yield a non-significant (unimodal) result. The same result would occur if a single "cluster" (e.g., another Gaussian) was present. By including mode E, if the dip test indicates unimodality, it is evident that the single mode is mode E, with nothing notable to its right. Conversely, a multimodal result implies something of interest to the right. Similarly, if a mode exists to the left of x_L, dipping over $[x_1, x_{U_C}]$ (including mode C) would yield a significant result. In the above example, the right part $[x_{L_E}, x_n]$ is unimodal, so the search is done right of x_U. The left part $[x_1, x_{U_C}]$ is multimodal, prompting further recursion into the region $[x_1, x_L]$. This recursion identifies intervals $[x_{L_A}, x_{U_A}]$ and $[x_{L_B}, x_{U_B}]$ for modes A and B. Overall, UniDip leverages the dip test's capabilities to 1) make a binary decision (unimodal or not) and 2) determine the primary modal interval.

TailoredDip [55] is an enhancement of UniDip, designed to address a specific limitation of UniDip: its tendency to overly classify the tails of distributions as outliers. TailoredDip improves the identification of these tails. This is achieved by examining the spaces between clusters for additional structures after the standard UniDip algorithm has completed. Specifically, the area between two clusters is mirrored and the dip p-value is calculated. If this suggests multimodal structures, the corresponding modes are identified and those points are assigned to the most suitable neighboring cluster. Additionally, if outlier detection is not required, a strategy is employed to assign points to either the left or right cluster: rather than using the midpoint between neighboring clusters as the decision boundary, the point where the ecdf intersects

Algorithm 1.1 Fine to Coarse (FTC) Segmentation Algorithm

1) Initialize $S = \{s_0, ..., s_n\}$ as the finest segmentation of the histogram, i.e., the list of all the local minima, plus the endpoints $s_0 = 1$ and $s_n = L$.
2) Repeat:

   Choose i randomly in $[1, length(S) - 1]$. If the pair of segments on both sides of s_i can be merged into a single interval $[s_{i-1}, s_{i+1}]$ following the unimodal hypothesis, group them. Update S.

   Stop when no more pair of successive intervals in S follows the unimodal hypothesis. $length(S)$ has decreased by one with each merging. Now $length(S) = l_0$.
3) For j from 3 to l_0, repeat step 2 with the unions of j segments.

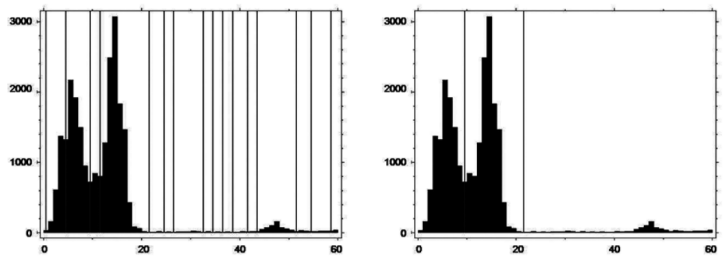


Figure 1.17: Initialization of FTC algorithm (left) and final segmentation after FTC algorithm (right) [4].

with the line connecting the right boundary of the left cluster and the left boundary of the right cluster is chosen. This approach provides a more accurate handling of different tails.

Histograms have long been used in image and data analysis for two main reasons: they offer a compact representation of large datasets, and they often allow us to infer global properties of the data through their behavior. One of the key features of a 1-d histogram is the list of its modes, which are the intervals where data is concentrated. For instance, a histogram of hues or intensities of an image composed of different regions will show several peaks, each corresponding to a different region. Proper image segmentation can then be achieved by identifying suitable thresholds that separate these modes in the histogram. However, quantifying the "data concentration" within an interval, and thus separating the modes, is not always straightforward.

In [4] a method (called Fine to Coarse (FTC) Segmentation Algorithm) is proposed to segment a 1-d histogram without a priori assumptions about the underlying

density function. It is based on the automatic detection of unimodal intervals in the histogram, which allow the histogram segmentation. A density function f is considered unimodal on an interval $[a, b]$ if it increases on some $[a, c]$ and decreases on $[c, d]$. It is therefore useful to segment a histogram by identifying segments where it is likely to represent a unimodal distribution. On such intervals, the histogram is denoted as "statistically unimodal". Clearly, such segmentation is generally not unique. Specifically, the segmentation defined by all the local minima of the histogram exhibits this property. Yet, minor variations due to the sampling process should not be mistaken for modes. To achieve a "minimal" division of the histogram, these fluctuations should be neglected. This leads to two criteria for acceptable segmentation: 1) each segment of the histogram should be "statistically unimodal" and 2) there is no union of consecutive segments on which the histogram is "statistically unimodal".

Algorithm 1.1 describes the FTC algorithm. Let h be a histogram on L bins $\{1, ..., L\}$. Starting from the segmentation defined by all the local minima of h, merge recursively the consecutive intervals until both properties are satisfied. The necessity of step 3) comes from the fact that a union of j successive segments can follow the unimodal hypothesis whereas no more union of k successive segments for $k < j$ does.

Fig. 1.17 [4] illustrates the initial and final segmentation of a dataset providing the corresponding histogram plots along with the detected splits. The initialization of the algorithm (all the local minima of the histogram) is provided in the left plot, where the histogram presents small oscillations, creating several local minima. The final segmentation after FTC algorithm is given in the right plot. Three modes are detected in this histogram with one of them being very small.

1.4 Decision Trees

Decision trees are widely used models in machine learning and data analysis, recognized for their intuitive, rule-based structure. They are constructed by iteratively splitting a dataset into subsets based on feature values, with the goal of creating increasingly homogeneous groups in relation to a specific target, outcome, or inherent pattern in the data. This process yields a tree-like structure that visually represents the decision-making process: each path in the tree corresponds to a series of feature-based decisions leading to a specific outcome.

Decision trees can be applied to both classification tasks, where the goal is to group data into discrete classes, and regression tasks, where the objective is to predict a continuous outcome. By tracing each decision path from the top to the final nodes, a decision tree model can either assign a label (in classification) or estimate a value (in regression). Additionally, decision trees have been proposed for clustering in an unsupervised framework.

A decision tree consists of the following components:

- **Root Node**: This is the topmost node in the tree, representing the entire dataset. It is the starting point for the decision-making process, where the initial data partition occurs.
- **Internal Nodes**: These nodes represent decision points where the data is split based on a specific feature's value. Each internal node performs a test on a feature, branching data into subgroups depending on the outcome of the test. This recursive partitioning continues through successive layers of internal nodes.
- **Leaf Nodes**: The terminal nodes in the tree are the leaf nodes, which correspond to the final output or decision reached after following a path through the tree. For classification, a leaf node contains a class label, while for regression, it contains a continuous predicted value. In unsupervised tasks, these nodes contain cluster labels.
- **Edges**: These are the connectors between nodes, representing the outcomes of each decision. An edge indicates which path to follow based on the result of a feature test, guiding the flow of data through the tree from root to leaves.

To prevent overfitting and ensure a manageable tree size, decision trees employ stopping criteria that terminate further splitting when certain conditions are met:

- **Maximum Depth**: Limits the depth of the tree. Once a specified maximum depth is reached, no further splits are allowed, and nodes at this level become leaf nodes.
- **Minimum Data Points per Leaf**: Specifies the minimum number of data points required to create a leaf node. If a split results in a subset with fewer data points than this threshold, the split is not made, reducing the likelihood of forming nodes based on insufficient data.

- **Impurity Threshold**: Some algorithms define a minimum reduction in their splitting criterion that must be achieved by a split. If a split does not produce the minimum required improvement, the split is not made.
- **Early Stopping with Validation Data**: In some implementations, decision trees are trained with a separate validation dataset. The tree stops growing when its performance on the validation set begins to degrade, preventing overfitting to the training data.

One of the major advantages of decision trees is their interpretability [56]. Unlike many other machine learning models, decision trees are easy to visualize and understand, making them particularly valuable in domains where model transparency is critical. They provide a clear, intuitive structure that allows users to trace the decision-making process step-by-step, from the root node to the leaf nodes. In a typical decision tree, the decision rule of an internal node involves simple thresholding on a feature value, thus it is straightforward to interpret. Typical decision trees (often called axis-aligned trees) partition the data space into hyperrectangular regions. The property of interpretability is especially useful in fields like healthcare, finance, and law, where decisions need to be justified. Additionally, decision trees can handle both categorical and continuous features without requiring feature scaling, which simplifies the preprocessing of data. They also perform well with non-linear relationships between features and the target variable, as they can create complex, hierarchical decision boundaries that adapt to the data.

However, decision trees also have notable limitations. One of the most significant drawbacks is their tendency to overfit the training data, especially when the tree is allowed to grow deep without constraint. Overfitting occurs when the tree learns the noise and fine details of the training data, which can hurt its ability to generalize to new, unseen data. This is addressed through pruning methods, which remove branches that add little predictive power, and by limiting tree depth or the minimum number of data points required for a split. Another limitation is their instability: small changes in the data can lead to large changes in the structure of the tree. This sensitivity to noise makes decision trees less reliable when the data is prone to fluctuations. Additionally, decision trees have a bias towards features with many values. When a feature has many possible values, it is more likely to be selected for a split, even if it does not provide the most meaningful or useful information. This bias

can lead to suboptimal tree structures if the model is not regularized appropriately.

1.4.1 Supervised Decision Trees

In supervised learning, decision trees are used with labeled data, where the objective is to create a model that can predict a specific outcome based on input features. Supervised decision trees are commonly applied to both classification and regression tasks. For classification tasks, the decision tree algorithm aims to split data in a way that maximizes class purity at each node. The tree continues to grow until a stopping criterion is met, with each path ultimately leading to a specific class label. These labels are determined by the majority class within each leaf node after the final split. For regression, the target variable is continuous, and the tree seeks to partition data in a way that minimizes prediction errors, typically based on metrics like variance reduction. Each leaf node in a regression tree contains a predicted value, which is usually the average of the target variable within that subset.

At each internal node, the decision tree algorithm selects a feature and corresponding threshold to partition the data into two or more subsets. The choice of feature and threshold is based on a criterion that seeks to maximize the separation of the target variable within each subset. Commonly used criteria include:

- **Information Gain**: Often used for classification, information gain measures the reduction in entropy (a measure of disorder) achieved by splitting the data on a given feature. Features that produce greater reductions in entropy are favored, as they contribute more to creating homogenous subsets.
- **Gini Impurity**: Another common metric for classification, Gini impurity calculates the likelihood of incorrectly classifying a randomly chosen element from the dataset if it were labeled according to the distribution of labels within the subset. Lower Gini values indicate more homogenous groups.
- **Variance Reduction**: For regression tasks, variance reduction is used to assess the quality of splits. It measures the reduction in variance within each subset after the split. Splits that result in lower variance within subsets are preferred as they improve the model's ability to make accurate predictions on continuous data.

The recursive partitioning process continues through each level of the tree until one of the stopping criteria is met, ensuring the tree does not become overly complex and prone to overfitting.

Popular decision tree algorithms include CART (Classification and Regression Trees) [57], ID3 (Iterative Dichotomiser 3) [58], C4.5 [59], CHAID (Chi-squared Automatic Interaction Detector) [60] and M5 [61]. CART is widely used due to its simplicity and ability to handle both classification and regression tasks, using criteria like Gini impurity or mean squared error for splitting. ID3, one of the earliest algorithms, uses information gain based on entropy to make splits, though it may favor features with many values. C4.5 improves on ID3 by using gain ratio to reduce bias, incorporating pruning to prevent overfitting, and supporting both discrete and continuous features. CHAID uses the statistical χ^2 test to determine the best split during the tree-growing process. The M5 algorithm extends decision trees to regression by fitting linear models at the leaf nodes. These algorithms differ in their splitting criteria, pruning techniques, and handling of missing data, with the choice depending on the specific task and dataset.

1.4.2 Unsupervised Decision Trees

Unsupervised decision trees are decision tree models used for tasks where labels are not available, focusing instead on discovering inherent structures or patterns within the data. While commonly applied for clustering, outlier detection, and data partitioning, unsupervised decision trees differ from supervised ones in their objectives, learning methods, and evaluation criteria.

Clustering methods aim at partitioning a set of points into groups, the clusters, such that data within the same group share common characteristics and differ from data in other groups. Most of the popular clustering algorithms, such as k-means, do not directly provide any explanation of the clustering result. To overcome this limitation, some research works propose the use of decision tree models for clustering in order to achieve explainability. Tree-based clustering methods return unsupervised binary trees that provide an interpretation of the data partitioning.

While in the supervised case, the construction of decision trees is relatively straightforward due to the presence of target information, this task becomes more challenging in the unsupervised case (e.g., clustering) where only data points are available. The

difficulty arises for two reasons:

- Definition of splitting criterion. Metrics like information gain or Gini index, which are commonly used to guide the splitting process in supervised learning, cannot be applied in unsupervised learning, since no data labels are available.
- Specification of hyperparameters (e.g., number of clusters), since cross-validation cannot be applied.

Despite the apparent difficulties, several methods have been proposed to build decision trees for clustering. The category of indirect methods typically follows a two-step procedure: first, they obtain cluster labels using a clustering algorithm, such as k-means, and then they apply a supervised decision tree algorithm to build a decision tree that interprets the resulting clusters. For example, in [62], labels obtained from k-means are used as a preliminary step in tree construction. Similarly, in [63], the centroids derived from k-means are also involved in splitting procedures. Indirect methods heavily rely on the clustering result of their first stage. Moreover, fitting the cluster labels with an axis-aligned decision tree may be problematic since the clusters are typically of spherical or ellipsoidal shape. It is also assumed that the number of clusters is given by the user.

Direct methods integrate decision tree construction and partitioning into clusters. Many of them follow the typical top-down splitting procedure used in the supervised case but exploit unsupervised splitting criteria, e.g., compactness of the resulting subsets. Some direct unsupervised methods are described below.

In [64] a top-down tree induction framework with applicability to clustering (Predictive Clustering Trees) as well as to supervised learning tasks is proposed. It works similarly to a standard decision tree with the main difference being that the variance function and the prototype function, used to compute a label for each leaf, are treated as parameters that must be instantiated according to the specific learning task. The splitting criterion is based on the maximum separation (inter-cluster distances) between two clusters, while after the construction of the tree, a pruning step is applied using a validation set.

In [65] four measures for selecting the most appropriate split feature and two algorithms for partitioning the data at each decision node are proposed. The split thresholds are computed either by detecting the top $k-1$ valley points of the histogram along a specific feature or by considering the inhomogeneity (information content)

of the data with respect to some feature. Distance-related measures and histogram-based measures are proposed for selecting an appropriate split feature. For example, the deviation of a feature histogram from the uniform distribution is considered (although it depends on the bin size).

In [66] an unsupervised method is proposed, called Clustering using Unsupervised Binary Trees (CUBT), which achieves clustering through binary trees. This method involves a three-stage procedure: maximal tree construction, pruning, and joining. First, a maximal tree is grown by applying recursive binary splits to reduce the heterogeneity of the data (based on the input's covariance matrices) within the new subsamples. Next, tree pruning is applied using a criterion of minimal dissimilarity. Finally, similar clusters (leaves of the tree) are joined, even if they do not necessarily share the same direct ascendant. Although CUBT constructs clusters directly using trees, it relies on several parameters throughout the three-stage process, while post hoc methods are required to combine leaves into unified clusters, which adds to the complexity and parameter dependency of the approach.

An alternative method for constructing decision trees for clustering is proposed in [67]. At first, noisy data points (uniformly distributed) are added to the original data space. Then, a standard (supervised) decision tree is constructed by classifying both the original data points and the noisy data points under the assumption that the original data points and the noisy data points belong to two different classes. A modified purity criterion is used to evaluate each split, in a way that dense regions (original data) as well as sparse regions (noisy data) are identified. However, this method requires additional preprocessing through the introduction of synthetic data in order to create the binary classification setting.

In contrast to axis-aligned trees, oblique trees allow test conditions that involve multiple features simultaneously, enabling oblique splits across the feature space. In [68] oblique trees for clustering are proposed, where each split is a hyperplane defined by a small number of features. Although oblique trees can produce more compact trees, finding the optimal test condition for a given node can be computationally expensive, while they may not always be interpretable [69].

An interesting direct approach [70] exploits the method of Optimal Classification Trees (OCT) [71], which are built in a single step by solving a mixed-integer optimization problem. Specifically, in [70] the Interpretable Clustering via Optimal Trees (ICOT) algorithm is presented, where two cluster validation criteria, the Silhouette

Metric [72] and the Dunn Index [73] are chosen as objective functions. The ICOT algorithm begins with the initialization of a tree, which serves as the starting point. Two options are provided for a tree initialization: either a greedy tree is constructed or the k-means is used as a warm-start algorithm to partition the data into clusters and then OCT is used to generate a tree that separates these clusters. Next, ICOT runs a local search procedure until the objective value (Silhouette Metric or Dunn Index) reaches an optimum value. This process is repeated from many different starting trees, generating many candidate clustering trees. The final tree is chosen as the one with the highest cluster quality score across all candidate trees and is returned as the output of the algorithm. ICOT is able to handle both numerical and categorical features as well as mixed-type features efficiently, by introducing an appropriate distance metric. Although it performs well on very small datasets and trees, it is slower compared to other methods. In addition, there exist hyperparameters that have to be tuned by the user, such as the maximum depth of the tree and the minimum number of observations in each cluster.

Unsupervised decision trees offer several advantages: they are highly interpretable, provide a hierarchical clustering structure, and are capable of handling large and high-dimensional datasets with ease. Their axis-aligned splits simplify implementation and make them computationally efficient compared to more complex clustering algorithms. However, these trees also have limitations. The axis-aligned splits can sometimes yield overly simplistic clusters that do not capture complex patterns as well as other methods might do. Additionally, unsupervised decision trees may be sensitive to the choice of splitting criteria and stopping rules, potentially leading to variability in the clusters they generate.

1.5 Thesis Contribution

In this thesis, we develop machine learning methods based on unimodality, mainly focusing on four different axes: i) creating a unimodality test for deciding data unimodality, ii) splitting multimodal data into unimodal subsets by detecting appropriate valley points, iii) building statistical models of univariate unimodal and multimodal data and iv) constructing (unsupervised) binary decision trees for clustering based on axis unimodal partitions. These problems are not independent from each other, while

the common key among them is the notion of unimodality. Next, we summarize the contribution of this thesis.

In Chapter 2 we present a new method for deciding on dataset unimodality, called UU-test (Unimodal Uniform test) [74]. The method takes as input a 1-d dataset and works with the ecdf of the dataset. It attempts to approximate the ecdf by constructing a cdf that is piecewise linear, unimodal and models the data sufficiently. The latter is ensured by applying uniformity tests on the data subsets corresponding to the linear segments. Unimodality is ensured by first computing the set (GL) of gcm and lcm points of the ecdf graph and then determining consistent subsets of GL, i.e. subsets where all gcm points lie before the lcm points. In the case where a cdf is found with the above two properties, then UU-test decides unimodality. The left plot of Fig. 1.18 illustrates the ecdf (blue solid line) of a unimodal dataset along with its piecewise linear cdf approximation (red dotted line) provided by UU-test. A unique feature of the method is that it also provides a statistical model of a unimodal dataset in the form of a uniform mixture model (UMM). In the middle plot of Fig. 1.18, the histogram of the previously mentioned dataset, along with the pdf of the statistical model UMM (red line) provided by the UU-test, are shown. Experimental results are presented in order to assess the ability of UU-test to decide on unimodality and perform comparisons with the well-known dip-test approach. In addition, in the case of unimodal datasets we evaluate the uniform mixture models provided by the proposed method using the test set log-likelihood and the two-sample Kolmogorov-Smirnov test.

Chapter 3 introduces a statistical model (called UΠsMM) that effectively models univariate unimodal data [75]. It is based on a Π-sigmoid mixture model (ΠsMM), where each component is a Π-sigmoid distribution. The Π-sigmoid distribution is defined as the difference of two translated sigmoid functions. This distribution is flexible enough to approximate data distributions ranging from Gaussian to uniform depending on the slope of the sigmoids. Therefore, in this chapter, instead of using a mixture of uniform distributions, we train a mixture of Π-sigmoid distributions, called Π-sigmoid Mixture Model (ΠsMM). This model is initialized from the UMM provided by the UU-test and subsequently trained through EM algorithm to maximize the likelihood of the dataset. A notable difficulty on this training task is that since the data has been characterized as unimodal, training of the ΠsMM should ensure that its density also remains unimodal. Therefore, during training, we check

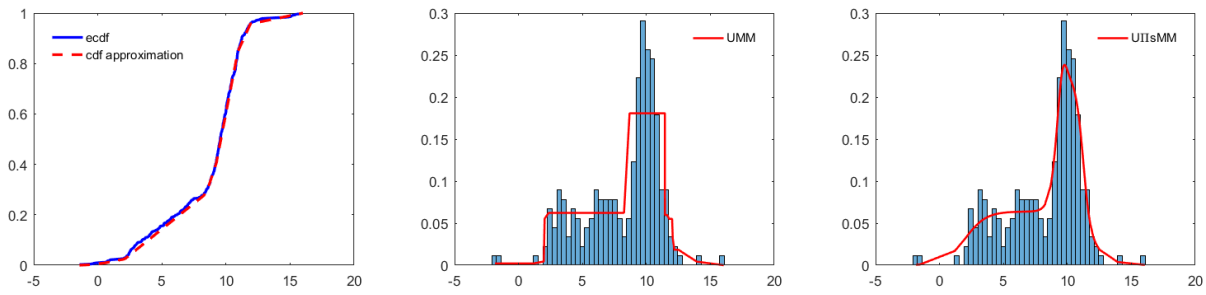


Figure 1.18: The ecdf of a unimodal dataset and its cdf approximation provided by the UU-test (left). The histogram plot of the unimodal dataset, along with statistical model fits using a UMM (middle) and a UΠsMM (right), are provided.

whether the model remains unimodal and in case of multimodality, we follow an appropriate strategy that gradually reduces the number of components, to ensure the model's unimodality. A benefit from this strategy is that as the initial number of components decreases, a simpler ΠsMM model is obtained with better generalization ability. Numerical experiments are presented that compare UΠsMM with UMM and the typical Gaussian model showing that UΠsMM provides a more accurate fit in several datasets, while it achieves a lower number of components than UMM. The right plot of Fig. 1.18 illustrates the pdf of the solution provided by the UΠsMM for the previously mentioned dataset. It is evident that the UΠsMM fit is more accurate than that of the UMM.

Chapter 4 focuses on partitioning and statistical modeling of univariate datasets (including multimodal datasets) [76]. The proposed method relies on the notion of unimodality and partitions the dataset into unimodal subsets through a novel approach for determining valley points in the probability density. We have introduced properties of critical points (gcm/lcm points) of the data ecdf that provide indications on the existence of density valleys. Those critical points are exploited in the proposed algorithm, called UniSplit. UniSplit is non-parametric and automatically estimates the number of unimodal subsets. In contrast to other approaches, it requires only a statistical significance threshold as input and no other user specified hyperparameters. Based on the splitting result, we introduce and construct a Unimodal Mixture Model (UDMM), where each mixture component constitutes a statistical model of the corresponding unimodal subset in the form of a Uniform Mixture Model (UMM). Fig. 1.19 illustrates the histogram and pdf plot of the solution provided by the UDMM for a multimodal dataset with three modes. The number of UDMM components is auto-

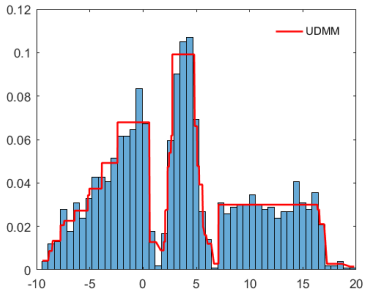


Figure 1.19: Histogram plot of a multimodal dataset along with the UDMM pdf.

matically obtained by the proposed UniSplit method, which constitutes a significant advantage over other models (e.g. Gaussian Mixture Model). In addition UDMM is very flexible and does not assume any specific parametric form for the unimodal mixture components. We present extensive experimental results aiming at evaluating in various tasks involving synthetic and real datasets, both the effectiveness of the splitting procedure as well as the performance of the constructed unimodal mixture model.

In Chapter 5 we propose the Decision Trees for Axis Unimodal Clustering (DTAUC) method for constructing unsupervised binary trees for clustering based on axis unimodal partitions [77]. This method identifies multimodal features of the data utilizing two proposed criteria. In criterion 1, the dip-test for unimodality is employed to detect the best split feature among the multimodal features based on a novel greedy approach. In this approach two new values are defined which measure the quality of each candidate split point of multimodal features. Using these values, the best split feature and the best split point are identified. Criterion 2 is based on the multimodality degree of the feature with the best split point estimated more directly using the UU-test for unimodality. Based on the detected split (calculated using either criterion 1 or criterion 2), each node of the tree is split into two subnodes, until all features are unimodal. In case all features of the data at a node are unimodal, the node is considered as leaf and the splitting procedure stops. The DTAUC method relies on the idea of unimodality, which is a novel technique for constructing unsupervised trees for clustering. This approach is simple and since it provides axis-aligned partitions of the data, it also offers interpretable clustering solutions. In addition the method requires no training, while it demonstrates the significant advantage that (apart from the typical statistical significance level) it does not include user specified

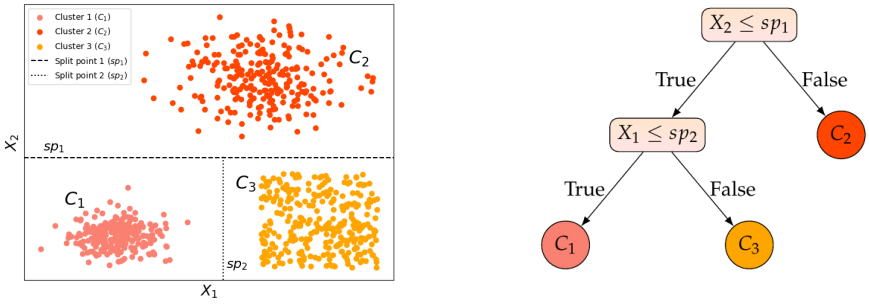


Figure 1.20: DTAUC method: 2D plot of a dataset split into three axis unimodal clusters (left), and the corresponding binary decision tree (right).

hyperparameters such as for example the maximum depth of the tree, or post hoc techniques, such as a pruning step. Our experimental evaluation reveals that DTAUC demonstrates good (and often superior) performance compared to other unsupervised decision tree methods using synthetic and real datasets. Fig. 1.20 illustrates the split result of a 2D dataset (with three axis unimodal clusters and two axis-aligned splits) using either criterion 1 or criterion 2, along with the constructed decision tree.

Finally, Chapter 6 summarizes this dissertation and draws directions for future research work.

Chapter 2

The UU-test for Statistical Modeling of Unimodal Data

2.1 Introduction

2.2 Notations and Definitions

2.3 UU-test Description

2.4 Modeling Unimodal Data

2.5 Experimental Results

2.6 Unimodality in Multiple Dimensions

2.7 Summary

2.1 Introduction

Gaining knowledge of data distributions is a significant topic in data analysis. As mentioned in Chapter 1, Section 1.2, it is of considerable importance to understand the grouping behavior of points, i.e., whether the data is unimodal or not. Although a great deal of research work focused on Gaussianity (or normality) tests (see Chapter 1, Section 1.1.1), few methods have been proposed for the more general problem of deciding distribution unimodality.

In this chapter we propose the *UU-test* (Unimodal Uniform test) method for modeling one dimensional data generated by unimodal distributions [74]. It works with

the empirical distribution function (ecdf) of the data, assuming the data distribution is continuous. It is important to note that UU-test does not make use of any parameters. In addition, it relies on well-known uniformity tests (e.g. Kolmogorov-Smirnov [12]), thus it does not require the computation of bootstrap samples (like Hartigans' dip-test [27]), a fact that saves computational time. Note also that all other tests focus on the decision on distribution unimodality and do not address the problem of statistical modeling of unimodal data. On the contrary our approach, in the case of unimodality, provides also a statistical model of the data in the form of a *Uniform Mixture Model* (UMM).

The proposed UU-test approach exhibits analogy to the dip-test methodology, i.e. it is applied on 1-d datasets and works with the ecdf of the dataset. However, instead of computing the distance of the ecdf from the family of unimodal distributions (dip-test), it attempts to define a unimodal distribution whose cdf sufficiently approximates the ecdf, i.e. the obtained distribution is both unimodal and a good statistical model of the dataset. In this way, in the case where unimodality is detected, we also obtain a generative model of the dataset in the form of a mixture of uniform distributions. Therefore, the method has a clear advantage over the dip-test.

The rest of this chapter is organized as follows. In Section 2.2 we provide the necessary definitions and notations, while in Section 2.3 we present the proposed UU-test method and attempt to explain the method using several illustrative examples. In Section 2.4, we present the statistical model of unimodal data in the form of a uniform mixture model provided by UU-test. Experimental results are provided in Section 2.5 aiming at evaluating both the decisions of the method as well as the performance of the constructed uniform mixture model. Section 2.6 refers to unimodality in multiple dimensions while appropriate cut points are suggested by UU-test in order to split multimodal datasets into unimodal subsets. Finally, Section 2.7 summarizes the chapter.

2.2 Notations and Definitions

In this section we provide the main definitions needed to present and clarify our method. Let $X = \{x_1, ..., x_N\}$, $x_i \in \mathbb{R}$ and $x_i < x_{i+1}$ an ordered 1-d dataset of distinct real numbers. Let a subset $S = \{s_1, ..., s_L\}$ of $X (s_i \in X)$ with $s_i \neq s_j$, $s_1 = x_1$, $s_L = x_N$.

We define the *piecewise linear cdf* $PL_S(x)$ obtained by 'drawing' the line segments from $(s_i, F_X(s_i))$ to $(s_j, F_X(s_j))$. Also, we assume that $PL_S(x) = 0$ if $x < s_1$ and $PL_S(x) = 1$ if $x \geq s_L$.

It is important to note that using a piecewise linear cdf $PL_S(x)$ as data model, we make the assumption that the subset $X(s_i, s_{i+1})$ of data points in each interval $[s_i, s_{i+1}]$ is uniformly distributed. Thus $PL_S(x)$ is actually the cdf of a uniform mixture model (UMM).

In UU-test, we aim to approximate the ecdf $F_X(x)$ using a $PL_S(x)$ that is unimodal. In order for the $PL_S(x)$ to be a good approximation of the ecdf, it should be *sufficient* in the sense defined as follows:

Let a subset $S = \{s_1, ..., s_L\} \subseteq X$ with $s_i \neq s_j$, $s_1 = x_1$, $s_L = x_N$. Subset S will be called *sufficient* if the cdf $PL_S(x)$ is a good statistical model of X. Since $PL_S(x)$ models the data in each interval using the uniform distribution, in order for $PL_S(x)$ to be a good statistical model of X, for each i the subset $X(s_i, s_{i+1})$ should follow the uniform distribution as decided by a uniformity test. Thus in the case where $PL_S(x)$ is sufficient, the corresponding uniform mixture model fits well to the data.

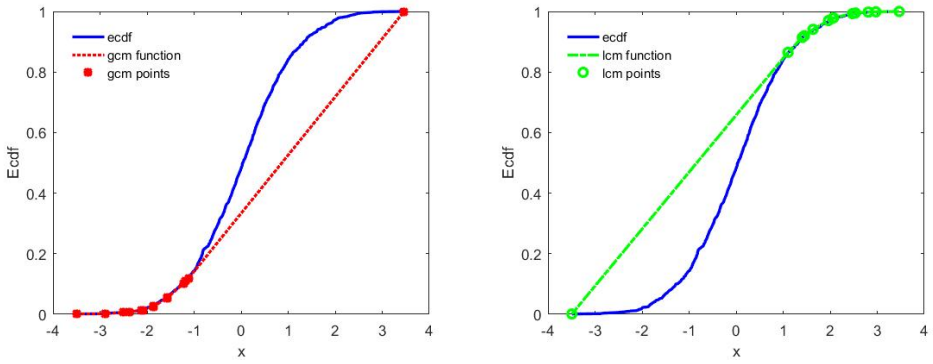
If $PL_S(x)$ is both unimodal and sufficient then we consider that the dataset X is unimodal and $PL_S(x)$ provides a good statistical model of X. Thus, *the UU-test method searches for a subset S of X, such that the cdf $PL_S(x)$ is unimodal and sufficient*.

In order to address the unimodality issue of $PL_S(x)$ we confine our search to the gcm and lcm points of the ecdf, exploiting the idea used in the dip-test method [27] for computing the dip statistic.

Fig. 2.1a presents an ecdf plot, along with the gcm function $G_X(x)$ and the set of gcm points G, while in Fig. 2.1b an ecdf plot along with the lcm function $L_X(x)$ and the corresponding set of lcm points L are illustrated.

Given the sets of gcm (G) and lcm points (L) of $F_X(x)$, we define as GL the ordered set of points obtained from the union of G and L: $GL = \{v_1, ..., v_M\}$, where $v_1 = x_1$, $v_M = x_N$, $v_i < v_j$ if $i < j$ and either $v_i \in G$ or $v_i \in L$. Note that $v_1 = x_1$ and $v_M = x_N$ belong to both G and L. We also define as $maxG = max(v_i | v_i \in G - \{x_N\})$ and $minL = min(v_i | v_i \in L - \{x_1\})$, the maximum value of G and the minimum value of L respectively, excluding the maximum and minimum elements of X.

Let S be a subset of GL that i) includes v_1 and v_M and ii) has the property that $maxG < minL$. Based on the definition of unimodality for cdf, it is straightforward to observe (see Fig. 2.2) that $PL_S(x)$ *is unimodal* and we will call the set S with the



(a) (b)

Figure 2.1: (a) Gcm function and gcm points of an ecdf. (b) Lcm function and lcm points of an ecdf.

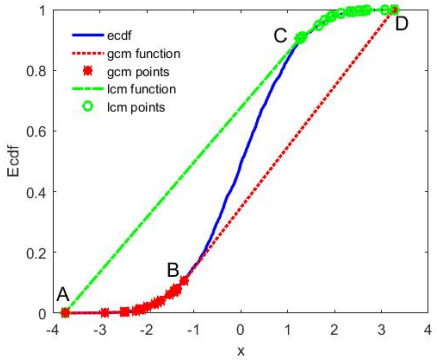


Figure 2.2: Gcm/Lcm function and gcm/lcm points of a unimodal ecdf. AB, BC and CD correspond to the convex, intermediate and concave part, respectively.

above two properties as *consistent*. It should be stressed that this definition includes the cases where either the gcm or the lcm part is missing.

A remarkable implication of consistency is that, *since* $PL_S(x)$ *is unimodal*, *the set* S *can be decomposed into three subsets* namely:

- S_G with the elements of S less than or equal to $maxG$ (convex part, $PL_{S_G}(x)$ is convex)
- $P_I = \{maxG, minL\}$ (intermediate linear part, $PL_{P_I}(x)$ is linear)
- S_L with the elements of S greater than or equal to $minL$ (concave part, $PL_{S_L}(x)$ is concave).

Fig. 2.2 presents a *unimodal* ecdf and the gcm/lcm (GL) points. The three sets S_G, P_I and S_L correspond to segments AB, BC and CD, respectively. Table 2.1 summarizes

Table 2.1: Summary of notation.

Notation	Explanation
$X = \{x_1, \ldots, x_N\}$	A set of distinct N points in $\mathbb{R}$.
$X(a, b)$	The set $\{a \leq x_i \leq b, x_i \in X\}$ for an interval [a,b].
$F_X(x)$	Ecdf of X.
S	Subset of X.
$PL_S(x)$	Piecewise linear cdf of $F_X(x)$.
$G = \{g_1, ..., g_{P_G}\}$	The set of gcm points of $F_X(x)$, where $g_1 = x_1, g_{P_G} = x_N$.
$G_X(x)$	The gcm function of $F_X(x)$.
$L = \{l_1, ..., l_{P_L}\}$	The set of lcm points of $F_X(x)$, where $l_1 = x_1, l_{P_L} = x_N$.
$L_X(x)$	The lcm function of $F_X(x)$.
GL	The set of ordered points of $G \cup L$.
$maxG$	$maxG = max(v_i \| v_i \in G - \{x_N\})$.
$minL$	$minL = min(v_i \| v_i \in L - \{x_1\})$.
sufficient(S)	True, if $X(s_i, s_{i+1})$ is uniform for each i.
consistent(S)	True, if $S \subseteq GL$ and S includes x_1 and x_N and $maxG < minL$.

the necessary notations and definitions for this chapter.

2.3 UU-test Description

As mentioned in the previous section, UU-test aims at finding a subset S of dataset X, such that the corresponding cdf $PL_S(x)$ is unimodal and sufficient. The latter means that the data in each interval $[s_i, s_{i+1}]$ are well-fitted by the uniform distribution. It should be noted that exhaustive search could have been used to determine an appropriate subset S, but it is computationally prohibitive. Alternatively, search techniques based on generate-and-test could also have been used.

In the UU-test method, search is restricted to subsets S of $GL = \{v_1, \ldots, v_M\}$, instead of examining the whole dataset X. We make the search even more focused, by looking for subsets of GL that are consistent, since consistency implies unimodality. Thus, *we search for a subset S of GL that is consistent and sufficient*. If such a set S is found, then $PL_S(x)$ defines a unimodal distribution that sufficiently models the

Algorithm 2.1 $S = UUtest(X)$

$E = (x_i, F_X(x_i)) \leftarrow ecdf(X)$

$S_G \leftarrow \emptyset$, $S_L \leftarrow \emptyset$, success $\leftarrow$ true, $P_I \leftarrow \{x_{min}, x_{max}\}$

$(S'_G, P'_I, S'_L, success) \leftarrow UU(S_G, P_I, S_L)$

return $S \leftarrow S'_G \cup S'_L$

dataset X. In this case UU-test decides unimodality and outputs the corresponding statistical model.

Given a 1-d dataset $X = \{x_1, \ldots, x_N\}$, function $S = UUtest(X)$ (Algorithm 2.1) takes X as input and outputs a non-empty set S (that is consistent and sufficient) in the case of unimodality and the empty set $S = \emptyset$ in the case of multimodality. It first computes the ecdf of the dataset (set E) and then calls function UU (Algorithm 2.2) where most of the work takes place.

UU function takes three sets as input, namely S_G (convex part), P_I (intermediate part) and S_L (concave part) and, if successful, it returns (possibly) updated versions of the three sets, otherwise it returns empty sets. Initially S_G and S_L are empty, while $P_I = \{x_1, x_N\}$, i.e., $X(P_I) = X(x_1, x_N) = X$. UU function operates on the data in the intermediate part $X(P_I)$. At first it checks for early success, this means that we test the uniformity of $X(P_I)$. If this happens, the function terminates successfully.

2.3.1 Consistent Subsets

If $X(P_I)$ is not uniform, we compute the corresponding set GL (union of gcm and lcm points) of $X(P_I)$ and determine the set C containing the consistent subsets GL_C of GL. Two cases are considered:

- either $C = \{GL\}$, i.e. GL is itself consistent, $maxG < minL$
- or $C = \{GL_1, GL_2\}$

In the latter case, the first consistent subset (GL_1) is obtained by removing all gcm points that lie after the first lcm point. Similarly, the second consistent subset (GL_2) is obtained by removing all lcm points that lie before the last gcm point.

Next we examine each set $GL_C \in C$. Since GL_C is consistent, it is decomposed into three sets corresponding to the convex (P'_G), intermediate (P'_I) and concave (P'_L) part. Then we try to determine a sufficient subset S'_G of P'_G as well as a sufficient subset S'_L

Algorithm 2.2 $(S'_G, P'_I, S'_L, success) = UU(S_G, P_I, S_L)$

if $check\_uniformity(X(P_I)) = true$ **then**
  return $(S_G, P_I, S_L, true)$
end if
$E_I = \{(x_i, y_i) \in E / x_i \in X(P_I)\}$
$GL \leftarrow$ **compute** gcm & lcm points of E_I
determine set C of consistent subsets of GL
for all consistent subsets $GL_C \in C$ **do**
  $(P'_G, P'_I, P'_L) \leftarrow$decompose$(GL_C)$
  $(S'_G, success) \leftarrow$sufficient$(P'_G)$
  if success=false **then**
    continue
  end if
  $(S'_L, success) \leftarrow$sufficient$(P'_L)$
  if success=false **then**
    continue
  end if
  $S'_G \leftarrow S'_G \cup S_G$
  $S'_L \leftarrow S'_L \cup S_L$
  $(S''_G, P''_I, S''_L, success) \leftarrow UU(S'_G, P'_I, S'_L)$
  if success=true **then**
    return $(S''_G, P''_I, S''_L, true)$
  end if
end for
return $(\emptyset, \emptyset, \emptyset,$false$)$

of P'_L. In the case of failure, the second consistent subset GL_C is examined (if it exists). In the case of success (i.e. both sufficient sets S'_G and S'_L have been found), the sets S'_G and S'_L are updated, and the UU function is called recursively in order to examine the intermediate part P'_I. The recursion ends either if P'_I cannot be decomposed into a sufficient gcm part (S''_G) and a sufficient lcm part (S''_L) (unsuccessful termination) or when $X(P'_I)$ is found uniform (successful termination). If UU is successfully applied on $X(P'_I)$ providing the sets S''_G, P''_I, S''_L, then $S'' = S''_G \cup P''_I \cup S''_L$ is the final solution for X. If the UU function fails on $X(P'_I)$, then the calling function $UU(X(P_I))$ also

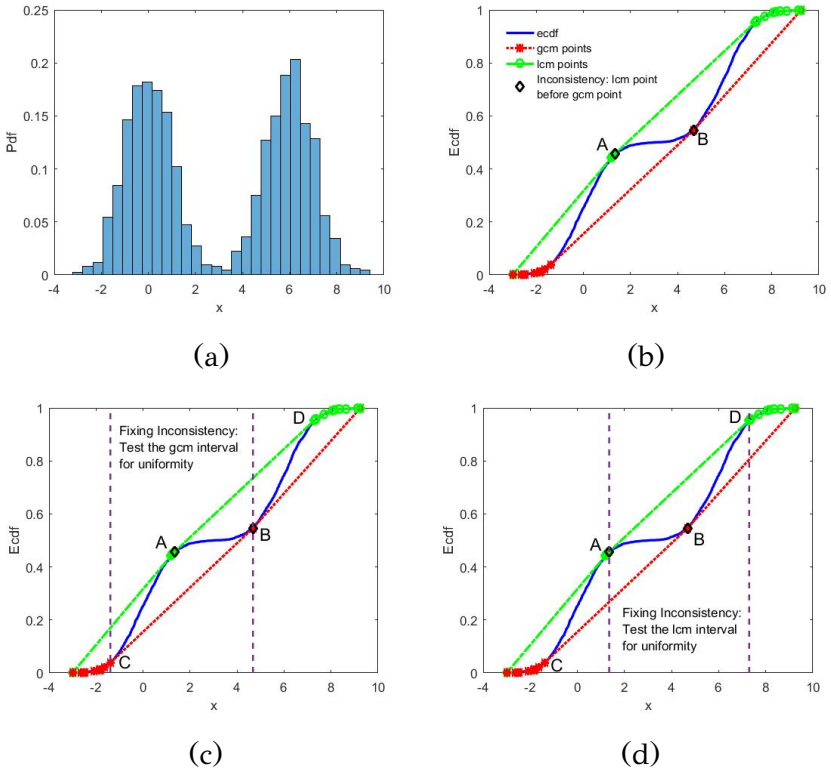


Figure 2.3: Example of multimodal dataset with consistent GL subsets that are not sufficient.

fails for the specific consistent subset GL_C. In this case the second GL_C subset (if it exists) should be examined.

Fig. 2.3 concerns a multimodal dataset. The histogram and the ecdf are presented in Fig. 2.3a and Fig. 2.3b respectively. In Fig. 2.3b the GL points are also presented. It can be observed that there exist lcm points (e.g. A) that lie before a gcm point (B). Therefore GL is inconsistent. In Fig. 2.3c we consider the consistent subset of GL that is obtained by omitting the lcm points (e.g. A) that lie between gcm points (B) and (C). Another consistent subset of GL can be obtained by omitting the gcm point (B) that lies between lcm points A and D. This case is shown in Fig. 2.3d. In UU function, both consistent subsets are checked for sufficiency and they fail in this test. Thus the dataset is characterized as multimodal.

Fig. 2.4 concerns a unimodal dataset. The histogram and the ecdf are presented in Fig. 2.4a and Fig. 2.4b respectively. In Fig. 2.4b the GL points are also presented. It can be observed that there exists a lcm point (A) that lies before gcm points (e.g. B). Therefore GL is inconsistent. In Fig. 2.4c we consider the consistent subset of GL that is obtained by omitting the lcm point (A) that lies between gcm points (B) and

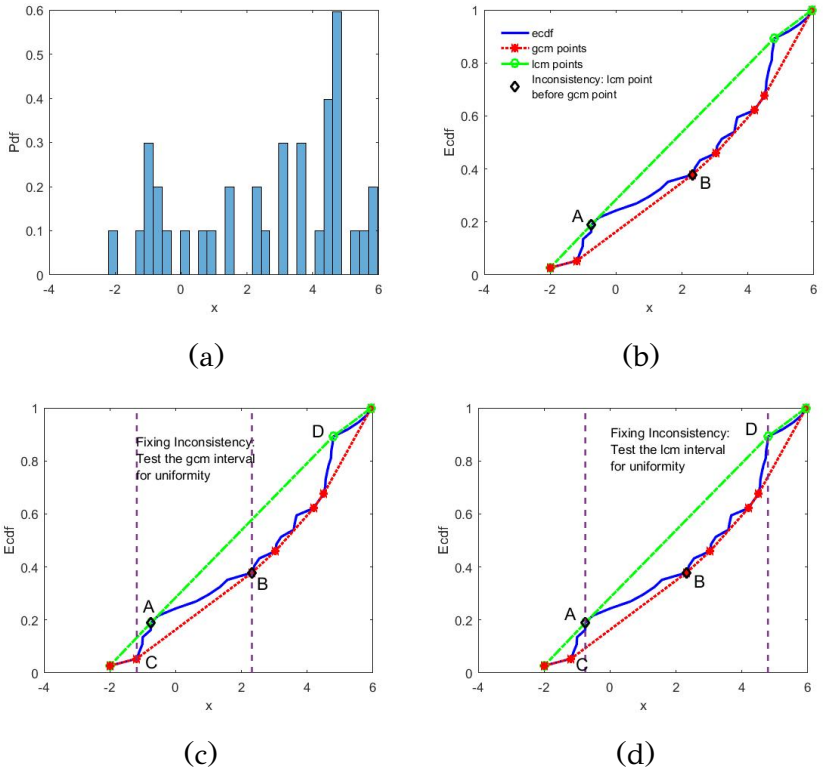


Figure 2.4: Example of unimodal dataset with consistent GL subsets that are sufficient.

(C). Another consistent subset of GL can be obtained by omitting the gcm points (e.g. B) that lie between lcm points A and D. This case is shown in Fig. 2.4d. In contrast to the case of Fig. 2.3, both consistent subsets are sufficient. In UU function, the first consistent subset is checked for sufficiency and succeeds in this test. Thus the dataset is characterized as unimodal.

2.3.2 Sufficient Subsets

Let $GL_C = \{s_1, \ldots, s_K\}$ be a consistent subset of GL. Note that $s_1 = x_1$ and $s_K = x_N$. Since GL_C is consistent, $PL_{GL_C}(x)$ is unimodal. In the general case, GL_C contains both gcm and lcm points. Thus, there exists a single index c such that all elements s_i for $i = 1, \ldots, c$ are gcm points and all elements s_j (for $j = c+1, \ldots, K$) are lcm points. Thus we can write that: $GL_C = P_G \cup P_I \cup P_L$, where $P_G = \{s_1, \ldots, s_c\}$ (gcm elements), $P_I = \{s_c, s_{c+1}\}$ and $P_L = \{s_{c+1}, \ldots, s_K\}$ (lcm elements).

Moreover, *every subset of GL_C that includes the points s_1, s_c, s_{c+1}, s_K is also consistent* (i.e. cdf PL_S is unimodal). Therefore it can be decomposed into three parts. The convex part which is a subset of P_G having s_1 and s_c as the first and last element. The concave part which is a subset of P_L having s_{c+1} and s_K as the first and last

element. The intermediate part is always the two-element set $P_I = \{s_c, s_{c+1}\}$. Thus our objective is to find a subset S of GL_C that is also sufficient.

In order to determine a sufficient subset S'_G of $P_G = \{s_1, \ldots, s_c\}$ (convex part) we work as follows: We first test whether the subset $X(s_1, s_c)$ succeeds in the uniformity test. If this is the case, we have successfully determined a sufficient set $S'_G = \{s_1, s_c\}$. However, if it fails, we continually test the successive subsets $X(s_i, s_{i+1})$ $(i = 1, \ldots, c-1)$ using the uniformity test. If the test succeeds, the points s_i, s_{i+1} are saved in S'_G. If all subsets $X(s_i, s_{i+1})$ are uniform, then P_G is sufficient ($S'_G = P_G$). However, it is possible for some subset to fail in the uniformity test. Let $X(s_i, s_{i+1})$ the first non-uniform data subset that is encountered, thus currently $S'_G = \{s_1, \ldots, s_i\}$. We make two attempts to fix this problem.

The first attempt is the Forward search method that searches for uniform supersets of $X(s_i, s_{i+1})$ by *moving the right interval endpoint*, i.e. $X(s_i, s_j)$, $j > i+1$. This method works in increasing order of s_i and successively tests whether the sets $X(s_i, s_{i+2})$, $X(s_i, s_{i+3})$, $X(s_i, s_{i+4})$ etc. are uniform. If a set $X(s_i, s_j)$, $j > i+1$ is found uniform, the element s_j is added in S'_G and we continue by testing the next subset $X(s_j, s_{j+1})$ for uniformity.

If the Forward search fails, the Backward search method is called that searches for uniform supersets of $X(s_i, s_{i+1})$ by *moving the left interval endpoint*, i.e. $X(s_m, s_{i+1})$, $m < i$. This method searches backwards and tests successively if the sets $X(s_{i-1}, s_{i+1})$, $X(s_{i-2}, s_{i+1})$, $X(s_{i-3}, s_{i+1})$ etc. are uniform. If such a set is found, the non-uniformity problem is fixed. More specifically, if a set $X(s_m, s_{i+1})$, $m < i$ is found uniform, the elements $s_{m+1}, \ldots, s_i$ are removed from S'_G and we continue by testing if the next subset $X(s_{i+1}, s_{i+2})$ succeeds in the uniformity test.

In order to determine a sufficient subset S'_L (concave part) we work in a similar way with the S'_G set. Algorithm 2.3 describes the overall method of determining a sufficient subset of a convex or concave set. We denote $e_n = \text{next}(e, P)$ the next element of e in set P and $e_p = \text{prev}(e, P)$ the previous element of e in set P. Algorithm 2.4 describes the Forward search method, while Algorithm 2.5 describes the steps of the Backward search method.

Fig. 2.5 presents an example of a multimodal dataset which exhibits non-uniformity in the interval between two successive lcm points. Fig. 2.5a presents the histogram of the dataset and Fig. 2.5b the ecdf of the dataset along with the GL points. It can be observed that the part of the ecdf between lcm points A and B is not linear, i.e.

Algorithm 2.3 (P',success) = sufficient(P)

$e_1 \leftarrow \min(P)$
$P' \leftarrow \{e_1\}$, success $\leftarrow$ true
while $e_L \leftarrow \max(P') \neq max(P)$ **do**
  if $e_R \leftarrow$ next(e_L, P) not exist **then**
    return $(\emptyset, false)$
  end if
  if check_uniformity(X(e_L, e_R))=true **then**
    $P' \leftarrow P' \cup \{e_R\}$
  else
    (P'_F,success) $\leftarrow$ Forward_search(P, e_L)
    if success=true **then**
      $P' \leftarrow P' \cup P'_F$
    else
      (P'_B,success)$\leftarrow$Backward_search(P', e_R)
      if success=false **then**
        return $(\emptyset, false)$
      end if
      $P' \leftarrow P'_B$
    end if
  end if
end while
return (P',success)

the subset is not uniform. In Fig. 2.5c and Fig. 2.5d we zoom into the concave (lcm) part of the ecdf where the nonlinearity (i.e. non-uniformity) of the ecdf is made more clear. In such a case we attempt to fix this issue by using the Forward and Backward search algorithms, however in this example both attempts fail.

Fig. 2.6 concerns an example of a unimodal dataset that includes a data subset in the concave part that is not uniform. However, in contrast to the case of Fig. 2.5, the Forward search algorithm manages to fix this problem. Fig. 2.6a presents the histogram of the dataset and Fig. 2.6b the ecdf of the dataset along with the GL points. It can be observed that the ecdf segment between successive lcm points A and B is not linear. In Fig. 2.6c and Fig. 2.6d we zoom into the lcm part of the dataset.

Algorithm 2.4 (P'_F,success) = Forward_search(P_F, e_L)

$P_F \leftarrow P_F - \{\text{next}(e_L, P_F)\}$, $e_R \leftarrow$next(e_L, P_F)
while e_R exist **do**
  if check_uniformity(X(e_L, e_R))=true **then**
    $P'_F \leftarrow \{e_R\}$,
    return (P'_F, true)
  end if
  $e_R \leftarrow \text{next}(e_R, P_F)$
end while
return (∅,false)

Algorithm 2.5 (P'_B,success) = Backward_search(P_B, e_R)

$P'_B \leftarrow P_B - \{\text{maximum element of } P_B\}$
$e_L \leftarrow \max(P'_B)$
while e_L exist **do**
  if check_uniformity(X(e_L, e_R))=true **then**
    $P'_B \leftarrow P'_B \cup \{e_R\}$,
    return (P'_B, true)
  end if
  $P'_B \leftarrow P'_B - \{e_L\}$, $e_L \leftarrow \text{prev}(e_L, P'_B)$
end while
return (∅,false)

Fig. 2.6c presents the histogram and Fig. 2.6d the ecdf of this subset, where subset $X(A, B)$ between points A and B is characterized non-uniform. Using the Forward search algorithm, the superset $X(A, C)$ is found uniform, thus the non-uniformity issue is fixed.

Two examples of the recursive application of UU function are presented in Fig. 2.7 and Fig. 2.8. Fig. 2.7 concerns a multimodal dataset whose histogram is shown in Fig. 2.7a. In Fig. 2.7b the ecdf is presented along with the gcm and lcm points (GL points). It can be observed that the intermediate part $X(A, B)$ (between points A and B) of the ecdf is not linear (uniform). Fig. 2.7c and Fig. 2.7d focus on the intermediate part presenting the histogram of this subset and the ecdf respectively. In Fig. 2.7d it is clear that the intermediate part is not uniform. For this reason the UU function is recursively applied on subset $X(A, B)$. Fig. 2.7d presents the GL points of $X(A, B)$. It

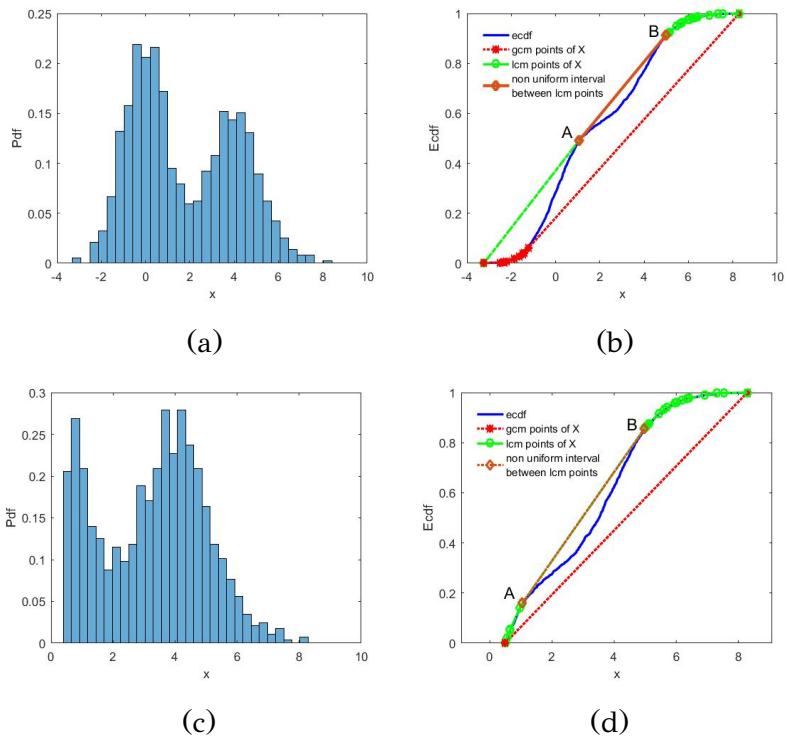


Figure 2.5: Example of multimodal dataset where consistent subsets are not sufficient.

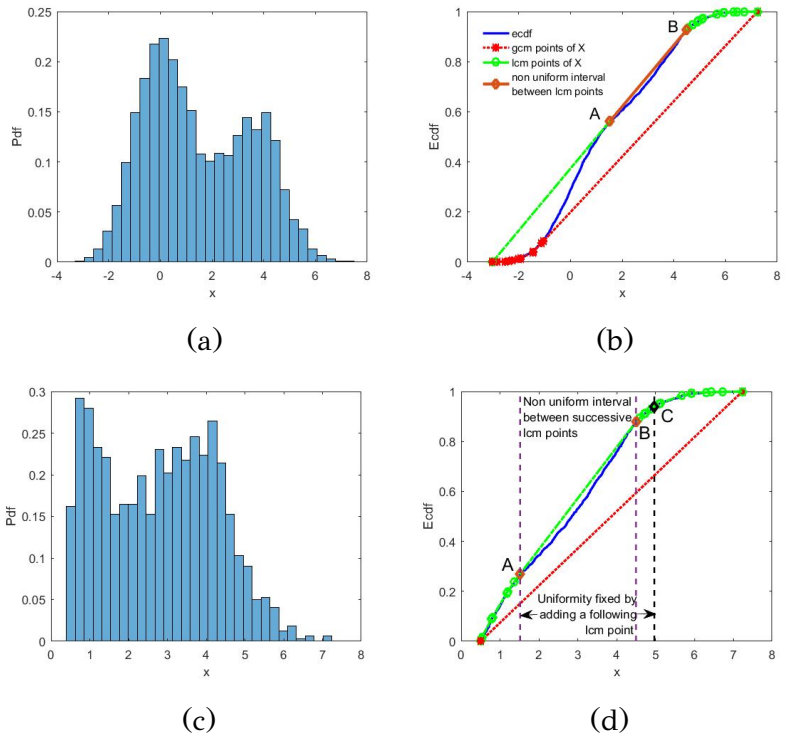


Figure 2.6: Example of unimodal dataset. A non-uniform interval exists between lcm points A and B. Forward search method fixes the non-uniformity problem by considering the extended interval between A and C.

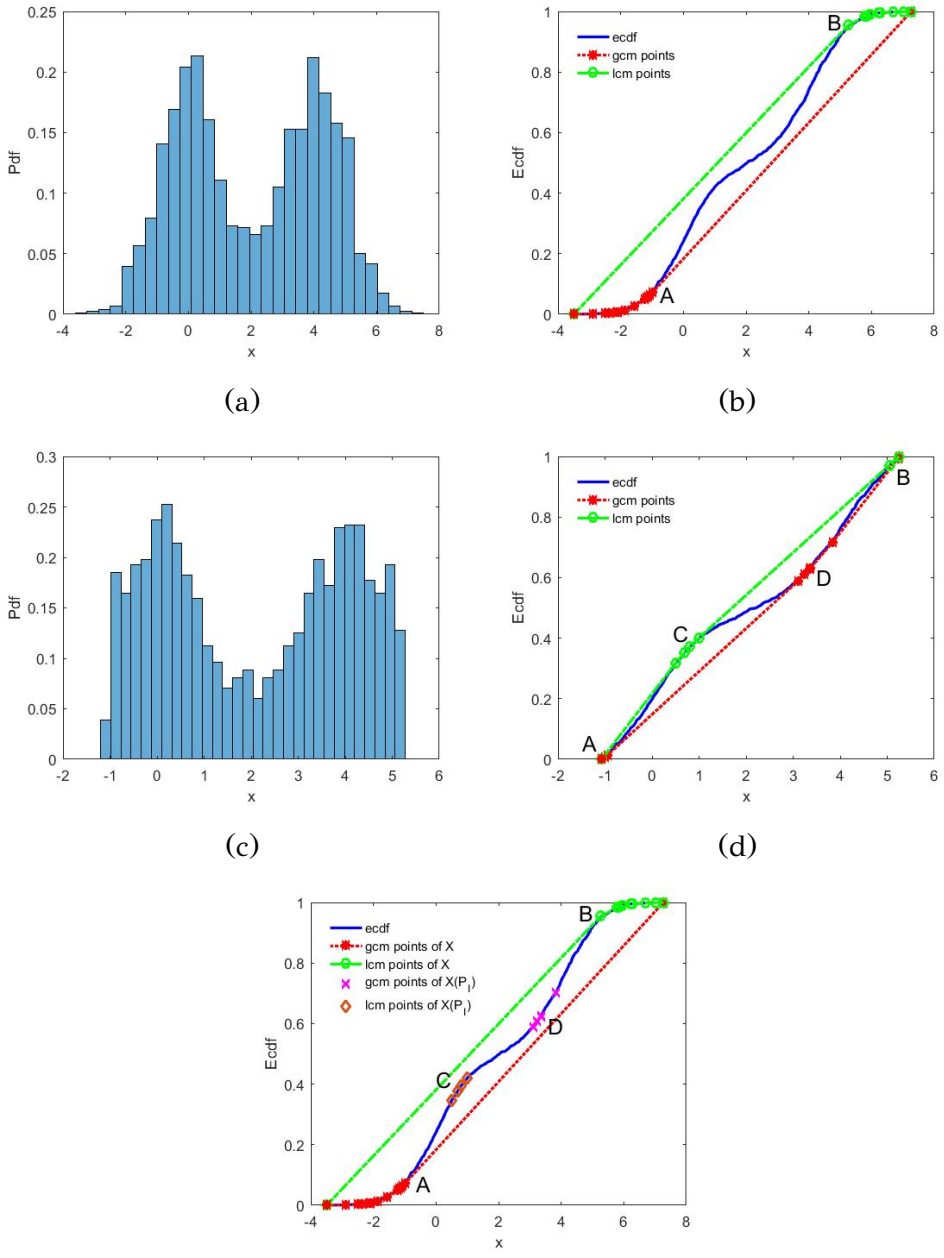


Figure 2.7: Example of multimodal dataset where the UU function is recursively applied on the intermediate part.

can be observed that there exist lcm points among gcm points, UU function cannot fix this inconsistency, thus the whole dataset is characterized as multimodal. In Fig. 2.7e the initial ecdf is presented along with both the GL points of the initial ecdf and the GL points of the ecdf of the intermediate part. It is clear that there exist gcm points that lie among lcm points and this observation leads to decide multimodality.

Fig. 2.8 concerns a unimodal dataset whose histogram is shown in Fig. 2.8a. In

Fig. 2.8b the ecdf is presented along with the gcm and lcm points (GL points). It can be observed that the intermediate part of the ecdf is not linear (uniform). Fig. 2.8c and Fig. 2.8d focus on the intermediate part $X(A, B)$ presenting the histogram of this subset and the ecdf respectively. In Fig. 2.8d it is clear that the intermediate part is not uniform and UU function is recursively applied on this subset. As shown in Fig. 2.8d, the intermediate part is unimodal and the whole dataset is characterized as unimodal. In Fig. 2.8e the initial ecdf is presented along with both the GL points of the initial ecdf and the GL points of the ecdf of the intermediate part. It can be observed that all gcm points precede the lcm points and this is an indication of unimodality, provided that the sufficiency criterion is also met.

2.3.3 Uniformity Test

A very common operation in the UU-test method, is to decide whether a subset is sufficiently modeled by the uniform distribution. For this reason a uniformity test is needed. In our implementation we use the Kolmogorov-Smirnov test (KS test) as a uniformity test. KS test first computes the KS statistic, which is the distance between the ecdf of the dataset and the cdf of the uniform distribution. Next a p-value is determined and compared with a user-defined significance level α (we use $\alpha = 0.01$ in our experiments). Therefore, if p-value $\leq a$, the KS test will reject uniformity.

There are two interesting features of KS test. First, the distribution of the KS test statistic itself does not depend on the underlying cumulative distribution function being tested and second, it is an exact test. Moreover, it is straightforward to determine the corresponding p-value, while the dip-test employs bootstrapping to compute the p-value.

However, KS test exhibits a peculiarity that may affect the result. It tends to be more sensitive near the center of the distribution than at the tails. In several experiments with large unimodal datasets, the KS test fails to early accept the uniformity of the intermediate part requiring the additional iterations. However, the final unimodality decision is not affected.

2.3.4 Computational Complexity

The computational complexity of UU-test mainly depends on cost of computing the gcm/lcm points of the ecdf which is O(n) using the isotonic regression method for

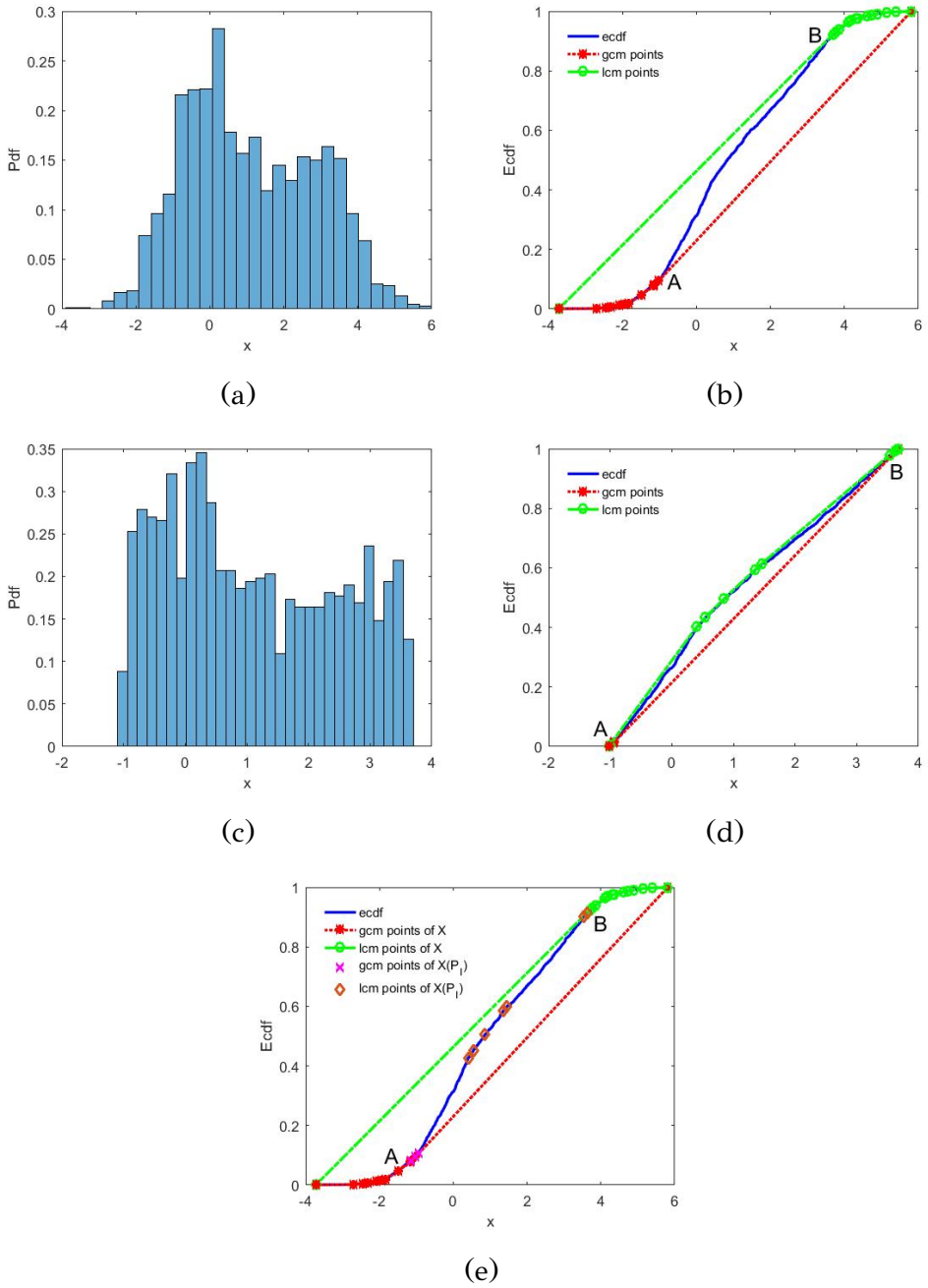


Figure 2.8: Example of unimodal dataset where the UU function is recursively applied on the intermediate part.

gcm points and antitonic regression for lcm points [78]. Gcm/lcm points can also be determined in O(nlogn) from the convex hull of the ecdf plot. If the data are not sorted an additional O(nlogn) complexity should be considered. It should be stressed that UU-test relies on KS-test, thus it does not require extra computations on bootstrap samples to obtain the p-value.

2.4 Modeling Unimodal Data

As already mentioned, in contrast to other unimodality tests, UU-test also achieves to model adequately a unimodal dataset X. In unimodal cases, the UU-test directly provides a statistical model, through the final set of S points it returns. The cdf of the statistical model is $PL_S(x)$, which is both unimodal and sufficient approximation of the ecdf. Since the cdf model is piecewise linear, it defines a *Uniform Mixture Model (UMM)* in which each component is the uniform distribution [79, 80, 81]. More specifically, if the set provided by UU-test is $S = \{s_1, \ldots, s_{M+1}\}$, then a UMM with M components is defined, where each component i is uniformly distributed in the range $[s_i, s_{i+1}]$, $(i = 1, ..., M)$. If N is the size of X and N_i is the number of data points in each interval $[s_i, s_{i+1})$, then the UMM pdf is defined as follows:

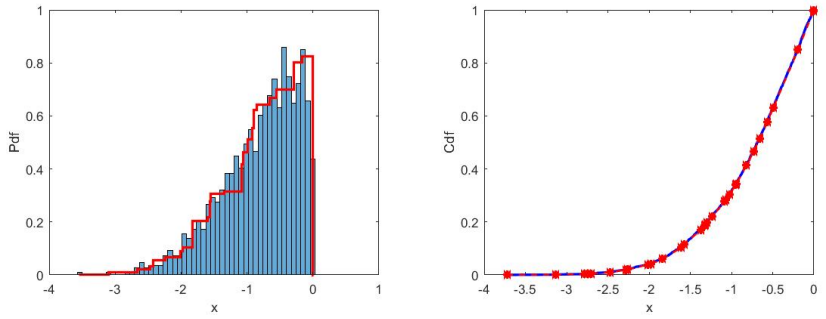
$$p(x) = \sum_{i=1}^{M} \pi_i \frac{x - s_i}{s_{i+1} - s_i} I(x \in [s_i, s_{i+1})), \quad \pi_i = N_i/N \tag{2.1}$$

The corresponding cdf $F(x)$ of the UMM is:

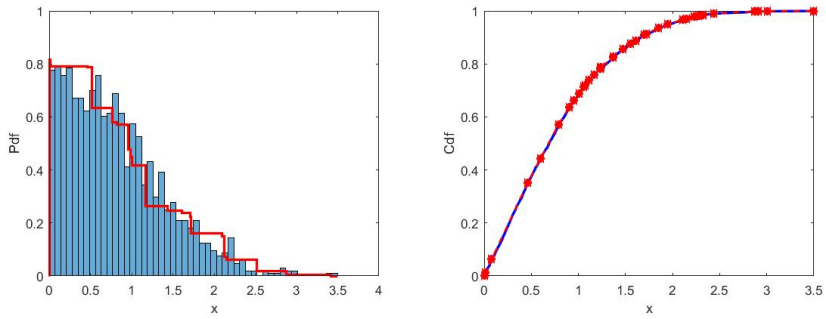
$$F(x) = \sum_{j=1}^{i-1} \pi_j + \pi_i \frac{x - s_i}{s_{i+1} - s_i}, \quad s_i \leq x \leq s_{i+1} \tag{2.2}$$

and it is expected to be close to the ecdf. In Figure 2.9 the UMMs obtained by applying the UU-test on four unimodal datasets are presented both in terms of UMM pdf and of UMM cdf. Left subfigures present the histogram and the UMM pdf (solid line), while right subfigures present the points of set S, the ecdf (solid line) and the UMM cdf (dashed line).

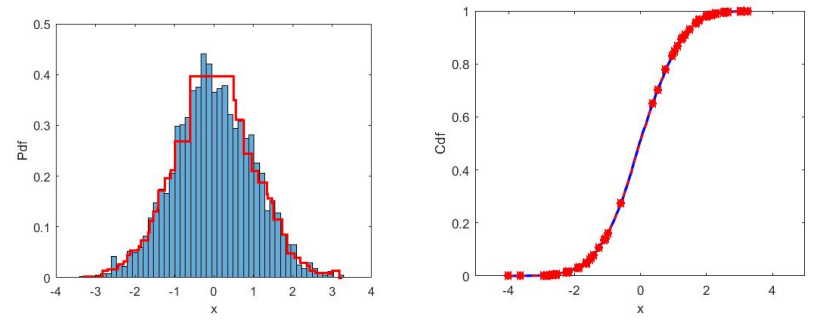
The UMM provided by the UU-test can also be used to *generate synthetic data samples* following the same unimodal distribution as the original dataset using the typical approach for sampling from a mixture model. Fig. 2.10 refers to a dataset with 2000 points generated by a Gaussian distribution. The histogram and the ecdf of the dataset are presented in Fig. 2.10a and Fig. 2.10b respectively. The UU-test is applied to this dataset and a UMM model is obtained. Fig. 2.10c and Fig. 2.10d present the pdf and ecdf of a dataset of the same size that is generated using the UMM model. It is obvious that both histograms and ecdfs are almost identical.



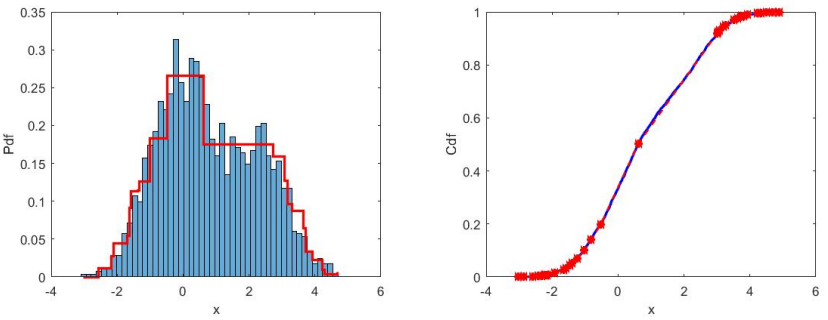
(a)



(b)



(c)



(d)

Figure 2.9: Unimodal datasets sampled from (a) a truncated ($x < 0$) Gaussian, (b) a truncated ($x > 0$) Gaussian, (c) a Gaussian, (d) two highly overlapping Gaussians. (Left) Histogram and UMM pdf (solid line). (Right) Points of S, ecdf (solid line) and UMM cdf (dashed line).

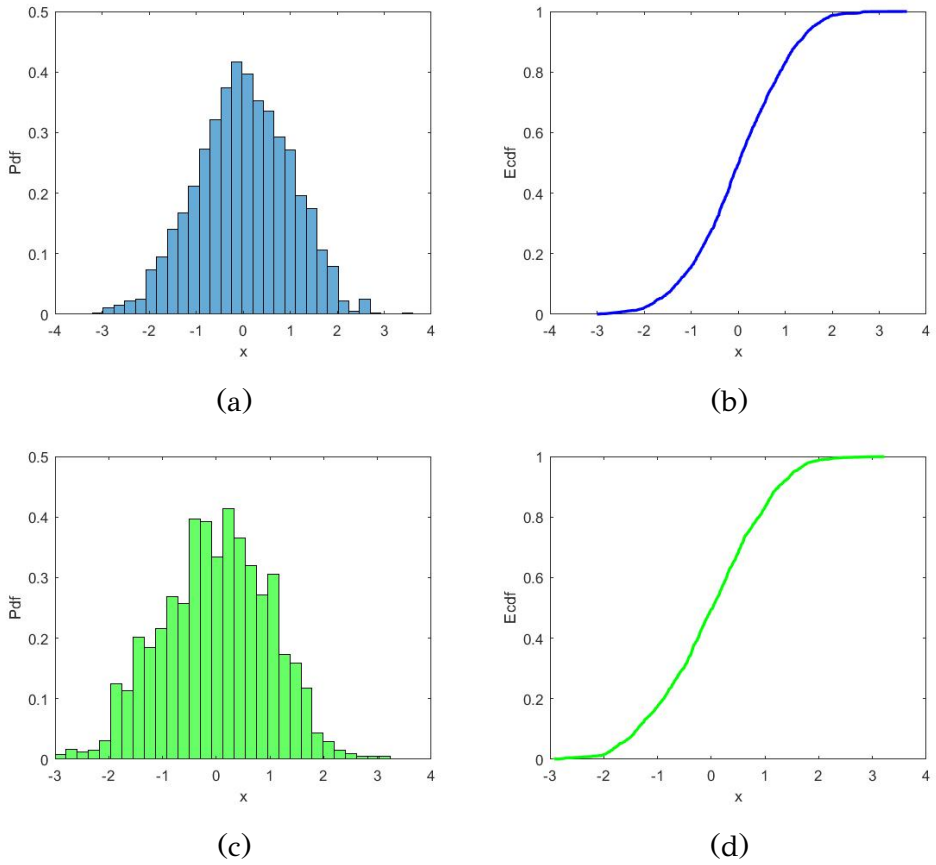


Figure 2.10: Histogram (a) and ecdf (b) of a dataset X sampled from the Gaussian distribution. Histogram (c) and ecdf (d) of a dataset sampled from the UMM obtained by applying UU-test on the Gaussian dataset X of (a) and (b).

2.5 Experimental Results

To assess the effectiveness of the UU-test, we conducted two series of experiments. In the first series, we compared the decisions of UU-test to those of the dip-test using several unimodal and multimodal synthetic and real datasets. In the second series of experiments, our aim was to evaluate the Uniform Mixture Model provided by UU-test as a tool for statistical modeling of unimodal data.

2.5.1 Evaluating UU-test Decisions

This part of our experimental study aims to assess the performance of UU-test in deciding on the unimodality of a dataset. At first, we generated synthetic unimodal and multimodal datasets and computed the decisions of dip-test and UU-test. In addition, we provide two synthetic examples illustrating the influence of noise and

outliers on gcm/lcm points and UU-test decision. Finally, we present the results from the application of dip-test and UU-test on the features of several real datasets.

Synthetic Datasets

At this part, we generated datasets from 15 unimodal (U) and multimodal (M) distributions as presented in Table 2.2. The multimodal distributions were mixtures of two or three Gaussians. The parameters and the dataset sizes (N) are shown in the second column of Table 2.2. For each distribution 50 datasets were generated and both UU-test and dip-test were applied on each dataset. Thus, in total 750 synthetic datasets were generated.

We compared the results of UU-test and dip-test using the same significance level ($\alpha = 0.01$). For each distribution, the percentage of 50 datasets for which each test provides correct decision is presented (third and fourth column) as well as the percentage of 50 datasets for which the two tests provided the same decision (fifth column). It can be observed that UU-test provides in most cases (for 741 out of 750 datasets) correct unimodality decisions that are in agreement with those of the dip-test.

Examples with noise and outliers

As it can be expected, noise and outliers affect the existence and position of gcm/lcm points. In Fig. 2.11 we present a bimodal dataset generated from two Gaussians which is distorted by adding uniform noise between the two Gaussians. It can be observed that the lcm/gcm points (A/B) in the middle of the ecdf (Fig. 2.11b) have been eliminated once the noise has been added (Fig. 2.11d). Nevertheless, the application of UU-test on the noisy dataset provides the correct decision, i.e. that the dataset remains multimodal.

In Fig. 2.12 we present a unimodal dataset generated from a single Gaussian, which distorted by the addition of outliers (left tail) generated from a Student's t distribution. It can be observed that the original gcm points (between A and B) (Fig. 2.12b) neither change or move, however, due to the addition of outliers on the left, two new gcm points (C and D) are generated (Fig. 2.12d). As with the previous example, the addition of outliers does not modify the result of the UU-test, which decides that the distorted dataset remains unimodal.

It should be noted that, in case we wish to explicitly deal with noise and outliers,

Table 2.2: Accuracy of UU-test and dip-test on deciding unimodality (U) or multimodality (M).

Distribution	Parameters	Dip-test (%)	UU-test (%)	Agreement of two tests (%)
Gaussian(μ, σ^2) (U)	$\mu = 0, \quad \sigma = 1, \quad N = 2000$	100	100	100
Student's t(ν) (U) ν: degrees of freedom	$\nu = 4, \quad N = 2000$	100	100	100
Gamma(k, θ) (U) k: shape, θ: scale	$k = 1, \quad \theta = 2, \quad N = 2000$	100	100	100
Exponential(λ) (U) λ: rate	$\lambda = 3, \quad N = 2000$	100	100	100
Cauchy(v) (U) v: degrees of freedom	$v = 1, \quad N = 2000$	100	100	100
Triangular (L, U, m) (U) L: Lower limit, U: Upper limit, m: mode	$L = -1, \quad U = 1, \quad m = 0,$ $N = 3700$	100	100	100
Asymmetric Triangular (U)	$L = -4, \quad U = 3, \quad m = 0,$ $N = 6500$	100	96	96
Two Gaussians (M)	$\mu_1 = 0, \quad \sigma_1 = 1, \quad N_1 = 2000$ $\mu_2 = 4, \quad \sigma_2 = 1, \quad N_2 = 2000$	100	100	100
Two Gaussians (M)	$\mu_1 = 0, \quad \sigma_1 = 1, \quad N_1 = 2000$ $\mu_2 = 4, \quad \sigma_2 = 1, \quad N_2 = 1000$	100	100	100
Two Gaussians (U)	$\mu_1 = 0, \quad \sigma_1 = 1, \quad N_1 = 1000$ $\mu_2 = 4, \quad \sigma_2 = 2, \quad N_2 = 1000$	100	100	100
Two Truncated Gaussians (U) with same mean	$\mu_1 = 0, \quad \sigma_1 = 1, \quad N_1 = 1000$ (Left part) $\mu_2 = 0, \quad \sigma_2 = 3, \quad N_2 = 1000$ (Right part)	100	94	94
Three Gaussians (M)	$\mu_1 = 0, \quad \mu_2 = 4, \quad \mu_3 = 8,$ $\sigma_1 = \sigma_2 = \sigma_3 = 1,$ $N_1 = N_2 = N_3 = 1000$	100	100	100
Three Gaussians (M)	$\mu_1 = 0, \quad \mu_2 = 4, \quad \mu_3 = 7,$ $\sigma_1 = \sigma_2 = \sigma_3 = 1,$ $N_1 = N_2 = 1000, \quad N_3 = 2000$	100	100	100
Student's t(ν) & Uniform(a, b) (U) a:minimum value b:maximum value	ν=10, $\quad a = 0, \quad b = 10, \quad N = 15000$	100	96	96
Uniform(a, b) & Gaussian(μ, σ^2) (U)	$a = -10, \quad b = 5, \quad \mu = 3, \quad \sigma = 1,$ $N = 16000$	100	96	96

we could approximate the ecdf using an appropriate regression method, and then work (e.g. compute gcm and lcm points) with the obtained regression model. Such an approach has been successfully applied in [82] where the image histogram is approximated using support vector regression and the obtained support vectors are exploited to appropriate segmentation thresholds.

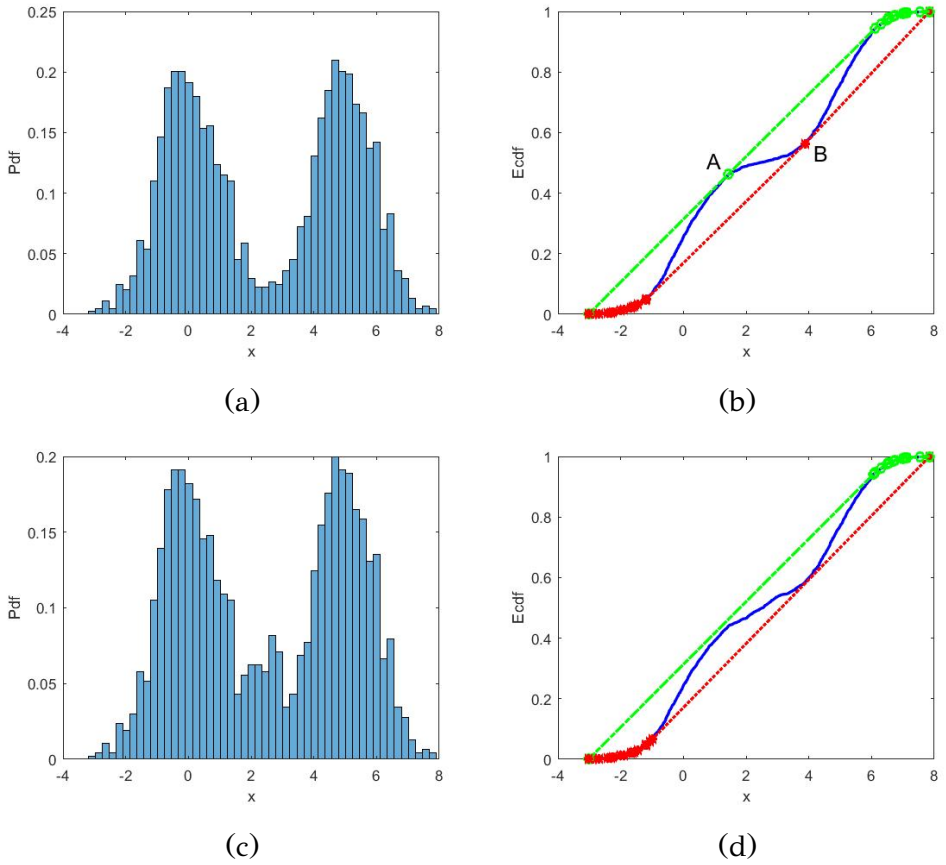


Figure 2.11: Top row: histogram and ecdf of a bimodal dataset generated by two Gaussians. A and B are middle lcm and gcm points respectively. Bottom row: histogram and ecdf of the dataset after adding uniform noise between the Gaussians. The middle lcm/gcm points A and B have been eliminated, however, UU-test still decides multimodality.

Real Datasets

We also applied dip-test and UU-test on each feature of five known real datasets, namely Iris, Banknote, Seeds, House from the UCI Machine Learning Repository [5] and Prestige [83]. Table 2.3 presents the datasets and the decision (unimodality (U) or multimodality (M)) of dip-test and UU-test on each dataset feature. Note that the ground truth decision for each feature is not available. The two tests agree on all dataset features except for feature 14 of House dataset. This feature is unimodal based on dip-test and multimodal based on UU-test. Fig. 2.13 presents the histogram and ecdf of this feature. As it can be observed, this is a borderline case.

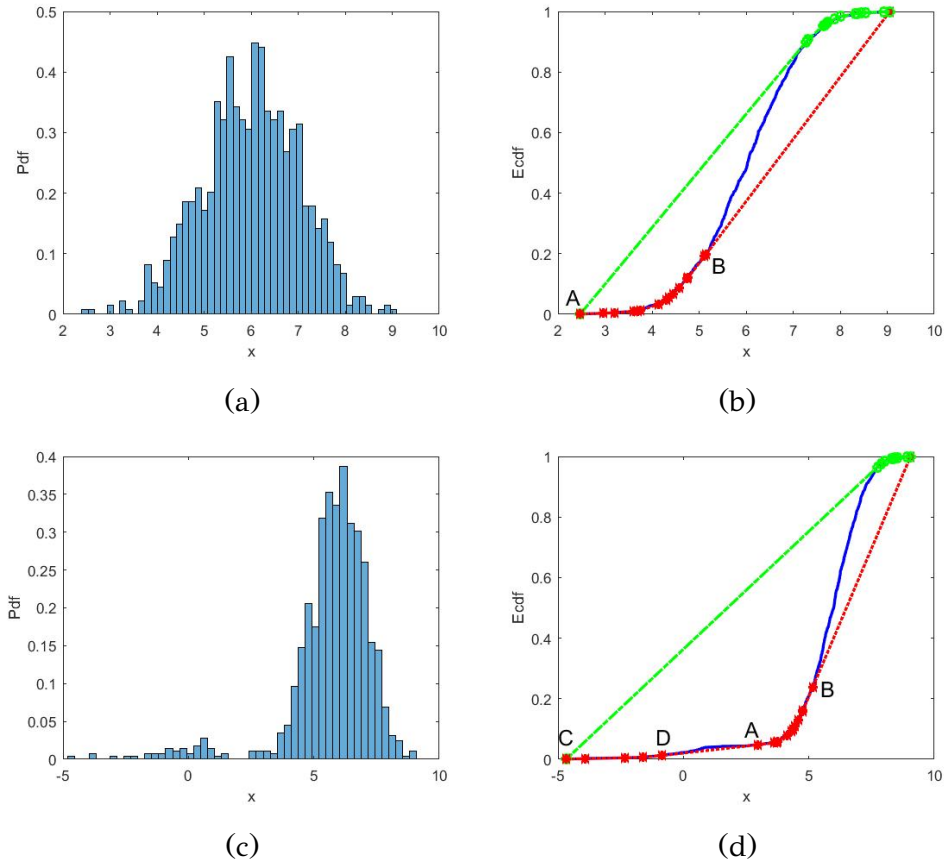


Figure 2.12: Top row: histogram and ecdf of a dataset generated by a single Gaussian. The gcm points between A and B are illustrated. Bottom row: histogram and ecdf of the dataset after adding Student's t distributed noise (outliers) on the left. Two new gcm points (C and D) have been generated, however, UU-test still decides unimodality.

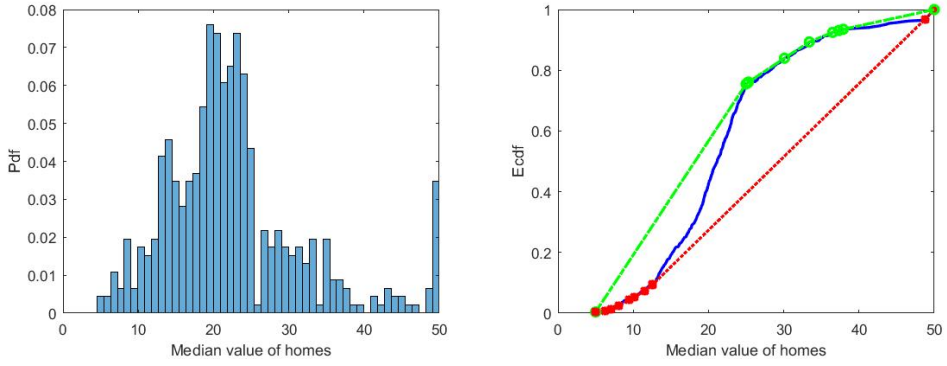


Figure 2.13: Histogram and ecdf of feature 14 of House dataset for which dip-test decides unimodality and UU-test decides multimodality.

Table 2.3: Dip-test and UU-test unimodality (U) or multimodality (M) decisions on features of real datasets.

Datasets	Features	Dip-test	UU-test	Agreement of two tests
Iris	1-2	U	U	yes
	3-4	M	M	yes
Banknote	1	U	U	yes
	2	M	M	yes
	3-4	U	U	yes
Seeds	1-7	U	U	yes
Prestige	1-4	U	U	yes
	5	M	M	yes
House	1	U	U	yes
	2-5	M	M	yes
	6-8	U	U	yes
	9-11	M	M	yes
	12-13	U	U	yes
	14	U	M	no

2.5.2 Uniform Mixture Modeling of Unimodal Data

We also conducted a series of experiments using synthetic datasets in order to evaluate the UMM provided by the UU-test. For each dataset, we also fitted a Gaussian model as well as a uniform model. In our experiments we considered a variety of unimodal distributions. Table 2.4 describes the distributions, their parameters and the size of training and test set. In Fig. 2.14, we present for each dataset the pdfs of the three fitted models: Gaussian (left figure), Uniform (middle figure) and UMM (right figure). It can be clearly observed that the UMMs provided by the UU-test constitute accurate statistical models of the datasets.

In order to measure the quality of the three statistical models, two criteria were considered. The first one is the log-likelihood on a test set and the results are presented in Table 2.5. We used 75% of the sample without replacement as a test set. The rest 25% was used as a training set to build the UMM, Gaussian and Uniform models. Then we computed the log-likelihood of each model on the test set (higher

Table 2.4: Types and parameters of distributions and size of training and test set of the datasets used for UMM evaluation.

Distribution	Parameters	Size of training set	Size of test set
Gaussian(μ, σ^2)	$\mu = 0, \quad \sigma = 1$	650	2000
Student's t(ν)	$\nu = 4$	650	2000
Gamma(k, θ)	$k = 1, \quad \theta = 2$	650	2000
Triangular (L, U, m)	$L = -1, \quad U = 1, \quad m = 0$	12500	37000
Asymmetric Triangular	$L = -4, \quad U = 3, \quad m = 0$	2150	6500
Two Gaussians	$\mu_1 = 0, \quad \sigma_1 = 1$ $\mu_2 = 3, \quad \sigma_2 = 1$	5850	17500
Student's t(ν) & Uniform(a, b)	$\nu = 10, \quad a = 0, \quad b = 10$	5000	15000
Uniform(a, b) & Gaussian(μ, σ^2)	$a = -10, \quad b = 5, \quad \mu = 3, \quad \sigma = 1$	5300	16000

Table 2.5: Statistical model evaluation using the test set log-likelihood (the higher the better). Bold values indicate the best model in each row.

Distribution	Gaussian Model	Uniform Model	UMM
Gaussian	$\mathbf{-13338}$	-17694	-14027
Student's t	-16331	-26681	$\mathbf{-16149}$
Gamma	-38283	-37044	$\mathbf{-34591}$
Triangular	-19451	-25697	$\mathbf{-18899}$
Asymmetric Triangular	-93852	-104910	$\mathbf{-89273}$
Two Gaussians	-32877	-42027	$\mathbf{-32483}$
Student's t & Uniform	-40245	-43284	$\mathbf{-36288}$
Uniform & Gaussian	-45959	-45828	$\mathbf{-39153}$

values imply better fit).

In addition, we used the two-sample Kolmogorov-Smirnov test as another criterion to evaluate and compare the three models. The two-sample KS test is a nonparametric hypothesis test that evaluates the difference between the ecdfs of two datasets, by computing the maximum absolute difference between the two ecdfs. Actually the test decides if two datasets have been generated from the same continuous distribution. In each experiment we used a dataset (test set) generated from the ground truth distribution and compared it (using the two-sample KS test) with a dataset generated from each of the three fitted models. The smaller the distance provided by the KS test, the better the fitted model. The experimental results are provided in Table 2.6.

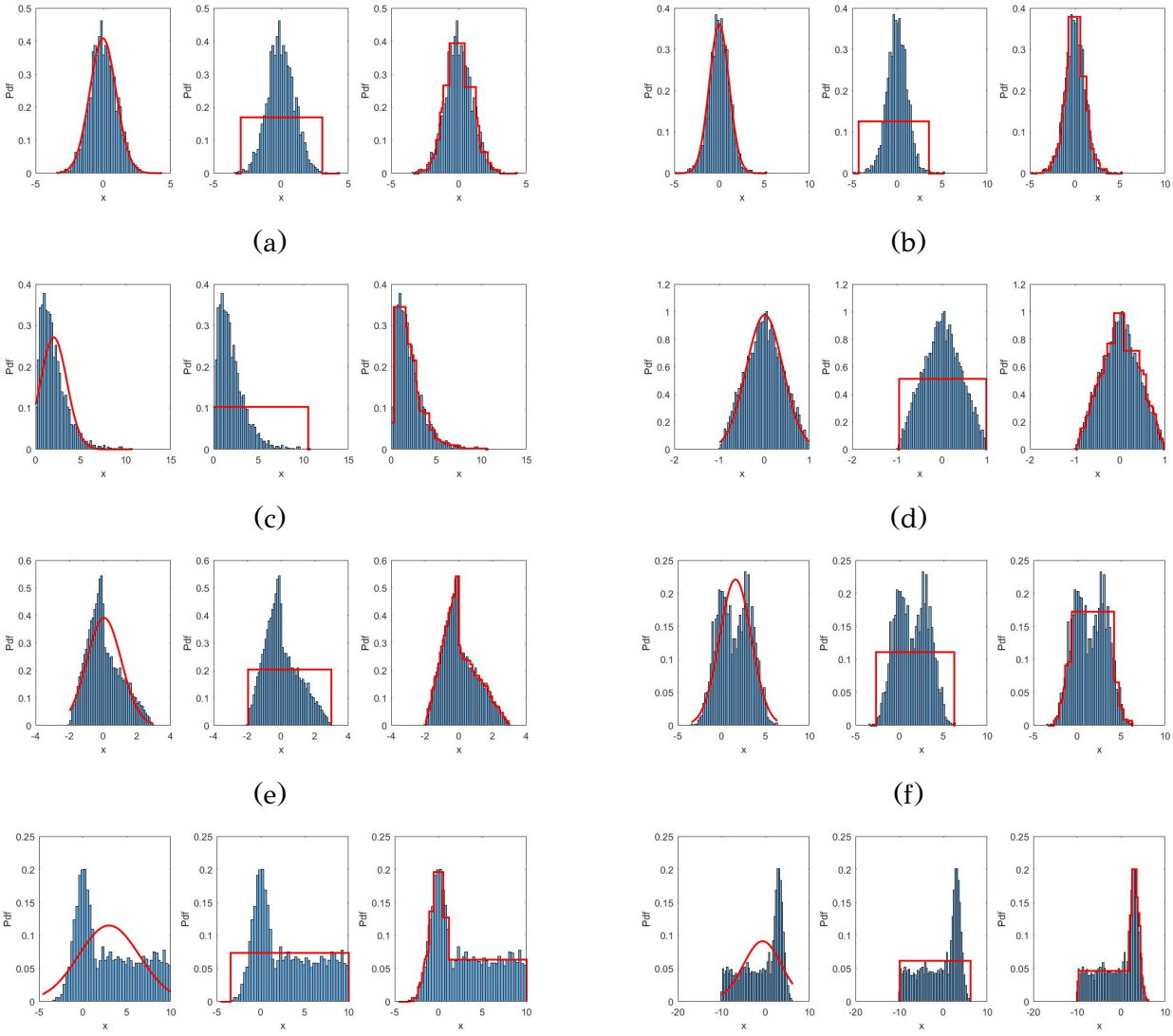


Figure 2.14: Examples of statistical model fitting on several datasets using Gaussian (left figures), Uniform (middle figures) and UMM (right figures).

The experimental results clearly indicate that the UU-test successfully models unimodal data through the UMM it provides. According to the test set likelihood criterion (Table 2.5) the Gaussian model constitutes a better solution only in the case of Gaussian distribution. According to the two-sample KS test criterion, the UMM provides much better results except for the case of Gaussian dataset (Table 2.6). In most cases the difference in performance is notable and becomes much more higher in the case of asymmetric distributions.

Table 2.6: Statistical model evaluation using the two-sample KS test (the lower the better). Bold values indicate the best model in each row.

Distribution	Gaussian Model	Uniform Model	UMM
Gaussian	**0.0133**	0.1945	0.02320
Student's t	0.0366	0.3451	**0.0186**
Gamma	0.1046	0.2350	**0.0164**
Triangular	0.0180	0.1260	**0.0062**
Asymmetric Triangular	0.0929	0.2072	**0.0050**
Two Gaussians	0.0365	0.2336	**0.0055**
Student's t & Uniform	0.1488	0.2720	**0.0065**
Uniform & Gaussian	0.2041	0.2703	**0.0048**

2.6 Unimodality in Multiple Dimensions

Tackling the unimodality issue for multidimensional datasets is not straightforward. The folding test [2] provides a direct approach to assess the 'unimodality character' (or level of unimodality) of a dataset X in multiple dimensions. As mentioned in Chapter 1, Section 1.2.2, it is based on the idea of folding up the distribution with respect to a pivot point $s^\star$, computing the variance of the folded distribution and finally computing the *folding statistic* ($\Phi(X)$) based on the ratio of the folded variance to the initial variance. Values of $\Phi(X)$ greater or equal to one indicate unimodality of X. Although $\Phi(X)$ is easy to compute, the computation of p-value relies on bootstraps sampled from the uniform distribution and is computationally heavy especially in multiple dimensions.

A major concern regarding the folding test is that it relies on the *empirical claim* that folding up a multimodal distribution leads to variance reduction. Therefore, the notion of unimodality is not explicitly involved in folding test computation. It is not difficult to specify distributions where the above claim is not valid, thus folding test fails to provide the correct decision. We next provide two 1-d characteristic examples. According to the folding test, a dataset X sampled from three Gaussians ($\mu_1 = -4, \mu_2 = 0, \mu_3 = 4$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.5$, $N_1 = N_2 = N_3 = 2000$ points) is unimodal ($\Phi(X) = 1.12$, p-value=0.009). On the contrary, for this clearly multimodal dataset, dip-test and UU-test agree that it is multimodal. Another example is a dataset generated by a Gaussian ($\mu = 0, \sigma = 0.5, N_1 = 2400$ points) and a Uniform ($\alpha = 1$,

$\beta = 4$, $N_2 = 1600$ points). For this clearly unimodal dataset, the folding test decides multimodality, since $\Phi(X) = 0.853$ and p-value=0.01. On the contrary, dip-test and UU-test correctly decide unimodality.

The most common approach to assess the unimodality character of a multidimensional dataset X is through the exploitation of 1-d unimodality tests. A characteristic example is the *dip-dist criterion* which is used in the dip-means clustering algorithm algorithm [29]. The dip-dist criterion decides on the unimodal character of X by exploiting the notion of viewer. A viewer is an arbitrary data point whose role is to suggest on the unimodality of the dataset by forming the set of its distances to all other data points and applying the unimodality test on this set of distances. The idea is that the distribution of the values in this distance vector could reveal information about the cluster structure. In presence of a homogeneous cluster, the distribution of distances is expected to be unimodal. In the case where distinct subclusters exist, the distribution of distances should exhibit distinct modes, with each mode containing the distances to the data objects of each subcluster. Considering each data point as a viewer, the result of unimodality tests on the rows of the distance matrix provide evidence on whether the dataset X contains subclusters or not.

Another way to assess the unimodality character of a multidimensional dataset is based on the assumption that, if a dataset is unimodal, then every 1-d projection of X should be unimodal. To approximately implement this idea the projection axes should be selected. The skinny-dip method [3] applies dip-test on the data axes, while the projected dip-means method [84] applies dip-test both on data axes and PCA axes.

UU-test could directly replace dip-test in the above two approaches. It should be stressed, that UU-test has particular advantages over dip-test. In the case of unimodality, it provides a statistical model in the form of UMM. This can be exploited in the naive Bayes framework [85, 86]: if all features are found unimodal, their joint density can be modeled as a product of UMMs. In another scenario, if the PCA projections [87, 7] of a multidimensional dataset are unimodal, then each PCA projection can be modeled using a UMM. Since PCA projections are independent, the density of the PCA vector of projections can be modeled as a product of UMMs.

2.6.1 UU-test for Clustering

Another useful property of UU-test (compared to dip-test) is that, in the case of multimodality, it provides information on how to cut (split) the dataset into subsets so as to finally obtain unimodal subsets. This property is particularly useful for designing incremental clustering schemes (based on cluster splitting) [88, 89, 90] since it provides information on how to split the multimodal clusters.

Two illustrative examples are provided next. Fig. 2.15a illustrates a 2-d dataset sampled from three Gaussians. It is clear that feature 1 (horizontal axis) is multimodal, while this of feature 2 (vertical axis) is unimodal. Fig. 2.15b presents the histogram of the values of multimodal feature 1. We wish to split this set of values and describe how UU-test can be used to determine effective cut points. UU-test fails to accept unimodality, due to the existence of lcm point A before gcm point B in Fig. 2.15c. Therefore, it is reasonable to assume that an effective cut point (cp_1) exists in the middle between x_A and x_B. After splitting the dataset using cp_1, we obtain a left subset that is unimodal and a right subset that is bimodal (see Fig. 2.15d, Fig. 2.15e). Focusing on the right subset, UU-test decides multimodality due to the existence of lcm point C before gcm point D in Fig. 2.15f. Therefore, the middle between x_C and x_D specifies a new cutpoint cp_2 that further splits the bimodal subset into two unimodal subsets. Fig. 2.15g, Fig. 2.15h and Fig. 2.15i illustrate the final split of the original dataset into three clusters.

Fig. 2.16 presents another application of the split method on feature 3 of Iris dataset [5]. More specifically, we see the histogram and ecdf of the bimodal feature 3. The existence of lcm point A before gcm B indicates multimodality, and the middle between A and B determines and effective cut point.

2.7 Summary

In this chapter, we have introduced UU-test (Unimodal Uniform test), which is a new method for deciding on dataset unimodality and for statistical modeling of unimodal data. The method takes as input a 1-d dataset and works with the ecdf of the dataset. It attempts to approximate the ecdf by constructing a cdf that is piecewise linear, unimodal and models the data sufficiently. The latter is ensured by applying uniformity (KS) tests on the data subsets corresponding to the linear segments. Unimodality is

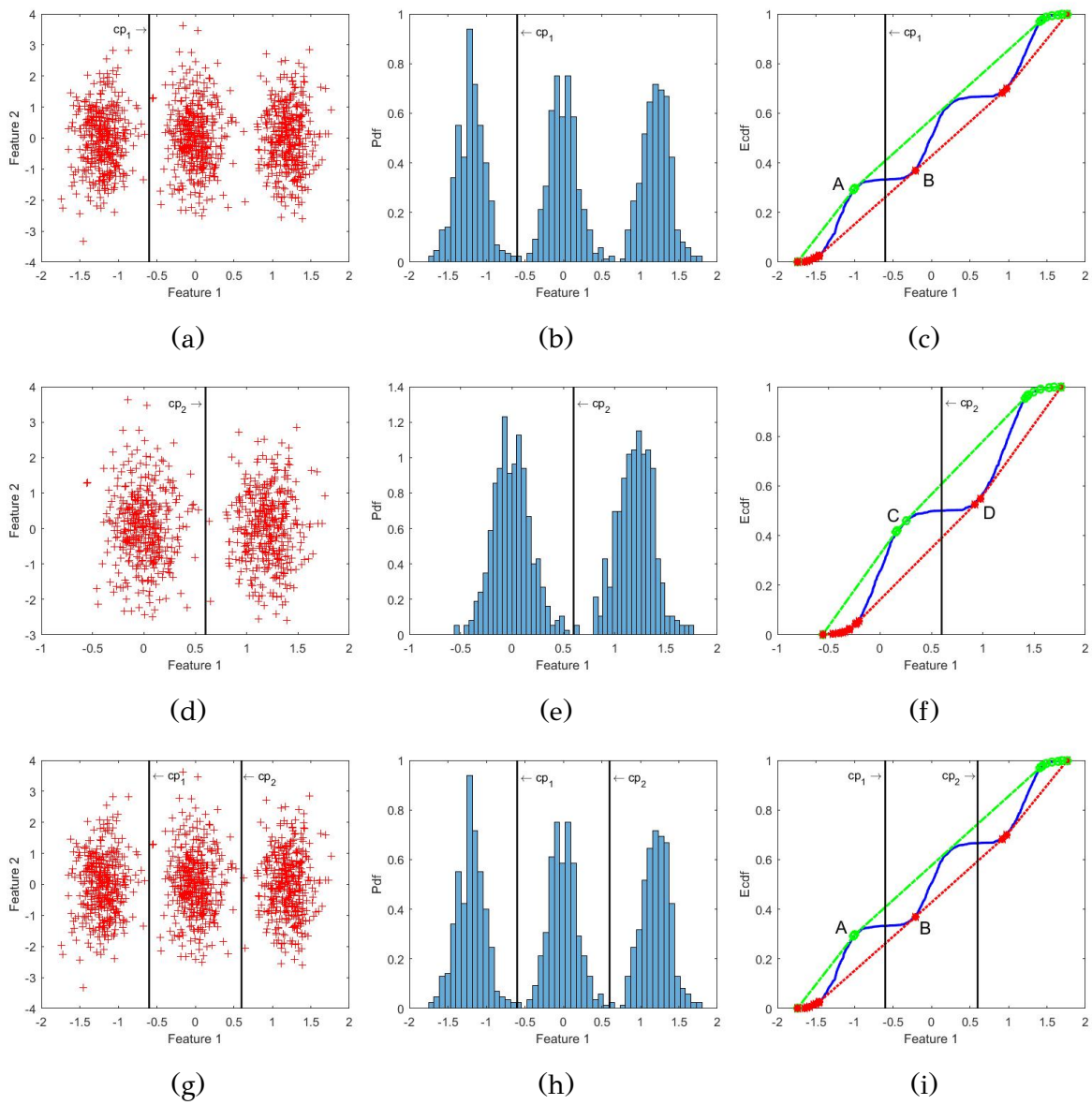


Figure 2.15: Top row: 2-d plot, histogram and ecdf of feature 1 of a 2-d dataset sampled from three Gaussians. Cut point cp_1 is also presented. Middle row: 2-d plot, histogram and ecdf of feature 1 corresponding to the right bimodal subset obtained from the first split. Cut point cp_2 is also presented. Bottom row: 2-d plot, histogram and ecdf of feature 1 corresponding to the original dataset along with the two cut-points.

ensured by first computing the set GL the gcm and lcm points of the ecdf graph and then determining consistent subsets of GL, i.e. subsets where all gcm points lie before the lcm points. In the case where a cdf is found with the above two properties (consistent and sufficient), then UU-test decides unimodality. A unique feature of the

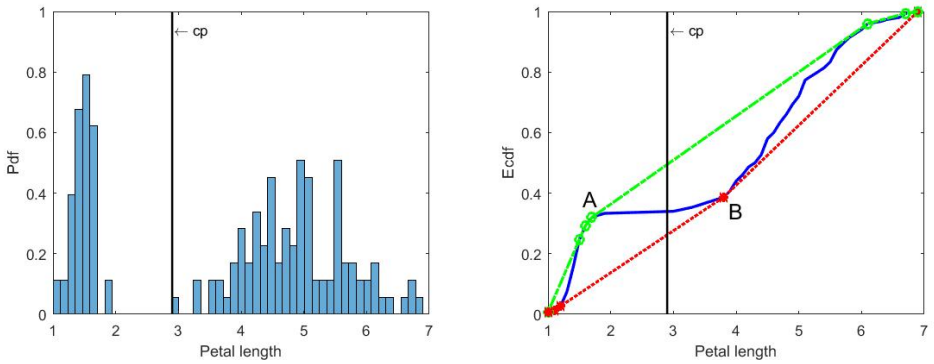


Figure 2.16: Histogram and ecdf of feature 3 of Iris dataset [5] along with the computed cut point.

method is that it also provides a statistical model of a unimodal dataset in the form of a uniform mixture model (UMM).

In our experimental evaluation we compared UU-test and dip-test in assessing dataset unimodality using synthetic and real datasets. Initially, we generated synthetic unimodal and multimodal datasets and computed the decisions made by the two tests. Results demonstrate that UU-test provides in most cases correct unimodality decisions that are in agreement with those of the dip-test. Additionally, we presented two synthetic examples to illustrate how noise and outliers influence gcm/lcm points and the UU-test's decisions. Lastly, we applied both tests to the features of several real datasets, highlighting their respective performance.

We also evaluated the Uniform Mixture Model (UMM) provided by the UU-test as a statistical model for unimodal data. Experiments with synthetic datasets showed that the UMM accurately models various unimodal distributions, outperforming Gaussian and uniform models using criteria, such as the log-likelihood and the two-sample Kolmogorov-Smirnov tests. The results confirm effectiveness of UU-test in modeling unimodal data.

Chapter 3

Statistical Modeling of Univariate Unimodal Data using Π-sigmoid Mixture Models

3.1 Introduction

3.2 Statistical Modeling using the Π-Sigmoid Distribution

3.3 Method Description

3.4 Experimental Results

3.5 Summary

3.1 Introduction

UU-test (proposed in Chapter 2) is a unimodality test which decides whether a dataset is generated by a unimodal or multimodal distribution. The unique feature of UU-test is that in case it decides that a dataset is unimodal, it also directly provides a statistical model of the data, in the form of a mixture of uniform distributions (i.e. a Uniform Mixture Model (UMM)). In Chapter 2 it is shown that this model is effective in modeling data generated from unimodal distributions of various shape. Fig. 3.1, provides an illustrative example of how a dataset sampled from an asymmetric triangular distribution is fitted by three models: Gaussian (left figure), uniform (middle figure) and UMM provided by UU-test (right figure). It is clear that UMM provides

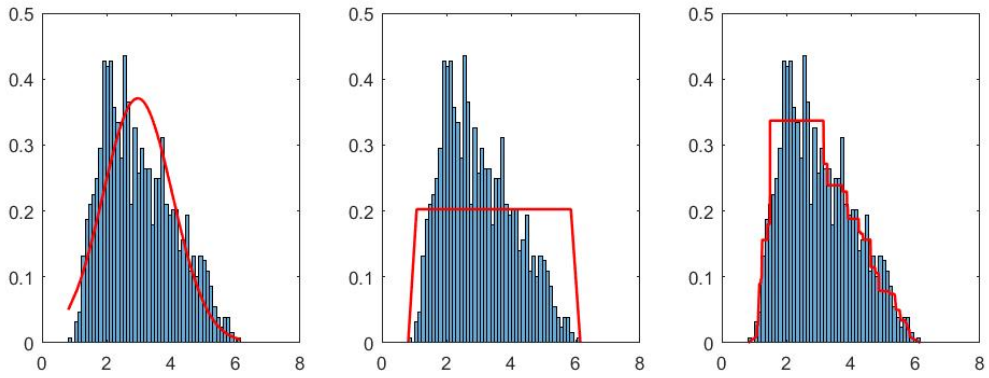


Figure 3.1: Statistical model fitting on data sampled from asymmetric triangular distribution using Gaussian (left figure), uniform (middle figure) and UMM (right figure).

a better solution, however there is still room to improve UMM performance, if we substitute the uniform distribution with a more flexible distribution.

For this reason in this chapter, we have considered the Π-sigmoid distribution [91] defined as the difference of two translated sigmoid functions. This distribution is flexible enough to approximate data distributions ranging from Gaussian to uniform depending on the slope of the sigmoids. Therefore, instead of using a mixture of uniform distributions, we consider in this chapter a mixture of Π-sigmoid distributions, called Π-sigmoid Mixture Model (ΠsMM) [75]. This model is initialized from the UMM provided by the UU-test and subsequently trained through EM algorithm to maximize the likelihood of the dataset. A notable difficulty on this training task is that since the data has been characterized as unimodal, training of the ΠsMM should ensure that its density also remains unimodal. Therefore, during training, we check whether the model remains unimodal and in case of multimodality, we follow an appropriate strategy that gradually reduces the number of components, to ensure the model's unimodality. A benefit from this strategy is that as the initial number of components decreases, a simpler ΠsMM model is obtained with better generalization ability.

The rest of this chapter is organized as follows. In Section 3.2 the Π-sigmoid distribution is described and the corresponding ΠsMM is explained. Section 3.3 presents the proposed methodology, while Section 3.4 presents experimental results on synthetic and real unimodal datasets. Finally, Section 3.5 concludes this chapter.

3.2 Statistical Modeling using the Π-Sigmoid Distribution

The Π-sigmoid, proposed in [91], is a probability density function, which has the ability to form the shape of the letter "Π" by appropriately combining two sigmoid functions. It is used to define the ΠsMM which is a mixture model with each component being a Π-sigmoid distribution. Below, we provide the necessary notations of a Π-sigmoid distribution and a ΠsMM.

3.2.1 The Π-Sigmoid Distribution

The Π-sigmoid distribution is computed as the difference between two logistic sigmoid functions with the same slope. The logistic sigmoid with slope λ is given by: $\sigma(\mathrm{x}) = \frac{1}{1+e^{-\lambda x}}$. The Π-sigmoid pdf with parameters a, b, λ (with $b > a$) is defined by subtracting two translated sigmoids:

$$\Pi s(x) = \left(\frac{1}{b-a}\right)\left[\frac{1}{1+e^{-\lambda(x-a)}} - \frac{1}{1+e^{-\lambda(x-b)}}\right], \quad b > a, \lambda > 0 \tag{3.1}$$

The parameters a, b, λ affect the behavior of the distribution, with a, b being related with the variance of the data points. Large values of λ indicate a uniform distribution, while smaller values make the function more bell-shaped. In Fig. 3.2a a small value of λ is employed ($\lambda = 0.5$), thus the curve is more bell-shaped, while in Fig. 3.2b, the distribution tends to be more uniform, since a large value of λ is used ($\lambda = 55$).

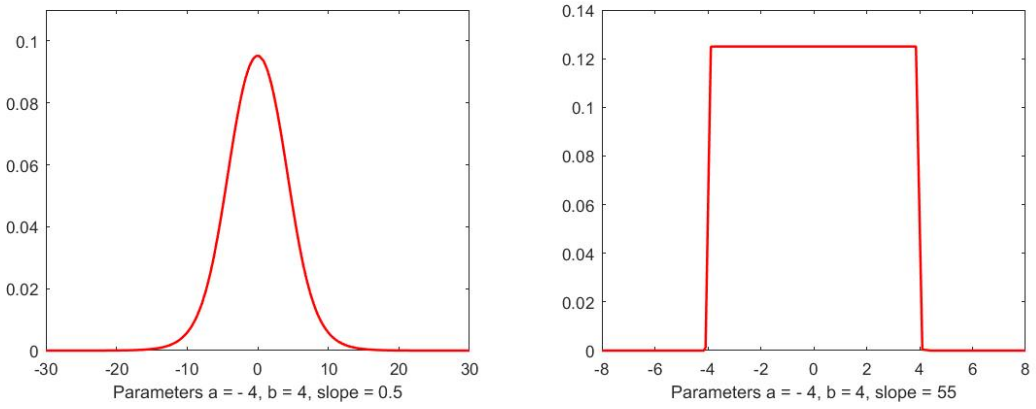
Given a dataset to be modeled by a Π-sigmoid distribution, the parameters of the distribution can be estimated by maximizing the likelihood of the dataset with respect to the parameters $a, b,$ and λ. The maximum likelihood solution cannot be obtained in closed form, however gradient-based maximization methods have been proposed.

3.2.2 The Π-Sigmoid Mixture Model (ΠsMM)

Exploiting the Π-sigmoid distribution, the Π-sigmoid Mixture Model (ΠsMM) [91] is defined as follows:

$$p(x) = \sum_{k=1}^{K} \pi_k \Pi s(x; a_k, b_k, \lambda_k) \tag{3.2}$$

where K is the number of Π-sigmoid components, a_k, b_k, λ_k are the parameters of k-th component and the mixing weights π_k satisfy the constraints: $\pi_k \geq 0$, $\sum_{k=1}^{K} \pi_k = 1$.



(a) (b)

Figure 3.2: Two shapes of the Π-sigmoid distribution by varying the λ parameter. (a) $\lambda = 0.5$. (b) $\lambda = 55$.

Given a dataset $X = \{x_1, ...x_N\}, x_i \in \mathbb{R}$ the parameters of the ΠsMM can be estimated through maximum likelihood using the EM algorithm. The EM algorithm is an iterative approach involving two steps at each iteration. The E-step computes the posterior probability that x_i has been generated by component k:

$$P(k|x_i) = \frac{\pi_k \Pi s(x_i; a_k, b_k, \lambda_k)}{\sum_{j=1}^{K} \pi_j \Pi s(x_i; a_j, b_j, \lambda_j)} \tag{3.3}$$

The M-step requires the maximization of the complete likelihood L_c with respect to the parameters of the ΠsMM model.

$$L_c = \sum_{i=1}^{N} \sum_{k=1}^{K} P(k|x_i) log[\pi_k \Pi s(x_i; a_k, b_k, \lambda_k)] \tag{3.4}$$

For the parameters π_k the update equation is the same for all mixture models: $\pi_k = \frac{1}{N} \sum_{i=1}^{N} P(k|x_i)$. The M-step does not lead to closed form update equations for the parameters a_k, b_k, λ_k of the Π-sigmoid components, though. Thus the GEM (generalised EM) algorithm [92, 93] is suggested to update the model parameters so that higher (not necessarily maximum) values of the complete likelihood are obtained. Based on [91] the following changes are made, in order to simplify the optimization: $\lambda_k = g_k^2$ and $b_k = a_k + r_k^2$. Thus the ΠsMM model is finally defined as:

$$p(x) = \sum_{k=1}^{K} \pi_k \Pi s(x; g_k, a_k, r_k, \pi_k) \tag{3.5}$$

3.3 Method Description

We propose a method that builds a statistical model of univariate unimodal data by training a Π-sigmoid mixture model under the constraint that its distribution remains unimodal. We call such a model as Unimodal-ΠsMM (UΠsMM).

3.3.1 Assessing the Unimodality of a Probability Distribution

As mentioned in Chapter 1, Section 1.2.1, a pdf is defined as unimodal, if it has a single mode; a region where the density becomes maximum, while non-increasing density is observed when moving away from the mode. A second definition option relies on the cdf: a cdf $F(x)$ is unimodal if there exist two points x_l and x_u such that $F(x)$ can be divided into three parts: a) a convex part $(-\infty, x_l)$, b) a constant part $[x_l, x_u]$ and c) a concave part (x_u, ∞). For a twice-differentiable function, it is convex if its second derivative is non negative and concave if its second derivative is non positive. The cdf of the Π-sigmoid distribution is:

$$F(x) = \frac{ln(e^{-\lambda(x-a)} + 1) - ln(e^{-\lambda(x-b)} + 1)}{\lambda(b-a)} + 1 \tag{3.6}$$

and its second derivative is:

$$F''(x) = \frac{d^2F}{dx^2} = \frac{\lambda}{b-a}\left[\frac{e^{-\lambda(x-a)}}{(e^{-\lambda(x-a)}+1)^2} - \frac{e^{-\lambda(x-b)}}{(e^{-\lambda(x-b)}+1)^2}\right]. \tag{3.7}$$

For a ΠsMM with K components (equation 3.5), $F''_{\Pi}(x) = \sum_{k=1}^{K} \pi_k F''_k(x)$ is the second derivative of its cdf $F_{\Pi}(x) = \sum_{k=1}^{K} \pi_k F_k(x)$ where $F_k(x)$ and $F''_k(x)$ are given by equation 3.6 and equation 3.7 respectively.

In order for $F_{\Pi}(x)$ to be unimodal, there must be a point x_m of our dataset X, such that $F''_{\Pi}(x) \geq 0, \forall x \leq x_m$ and $F''_{\Pi}(x) < 0, \forall x > x_m$. If such a point exists, we decide unimodality for $F_{\Pi}(x)$ and we characterize the corresponding ΠsMM as a unimodal model, since the values of $F''_{\Pi}(x)$ are first positive and then negative. Otherwise, when there is no point with the above property, $F_{\Pi}(x)$ and the corresponding ΠsMM are considered multimodal. In such a case there exist points with negative F''_{Π} value among points with positive F''_{Π} value and/or vice versa. We denote the points of X with the above property as multimodality indicators (MI). In Fig. 3.3a the points depicted as stars constitute multimodality indicators.

Algorithm 3.1 new_model = fix_multimodality($model$)

$small\_interval \leftarrow [a_k, b_k]$ with $|b_k - a_k| < 10^{-4}$
if a $small\_interval$ **exists then**
  $new\_model \leftarrow$ Merge($small\_interval, K$) // $K := K - 1$
else
  $F_{\Pi} \leftarrow$ cdf of $model$
  $MI \leftarrow$ set of multimodality indicators
  determine the $interval$ $[a_k, b_k]$ which contains the majority of $x \in MI$
  $new\_model \leftarrow$ Merge($interval, K$) // $K := K - 1$
end if
return new_model

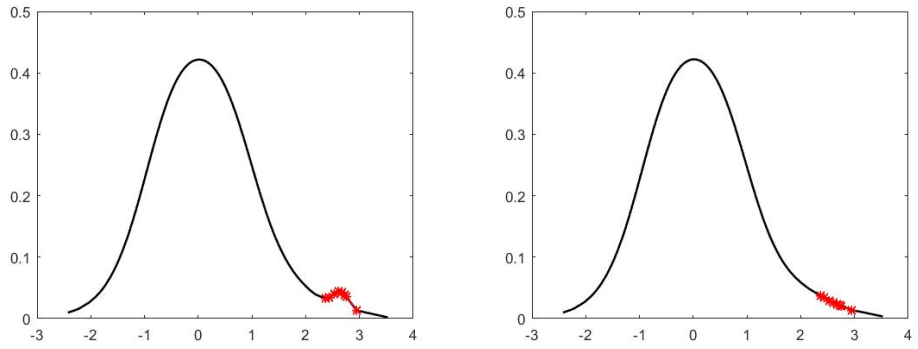
3.3.2 Unimodal ΠsMM Training

In our method we aim to build a Unimodal ΠsMM (UΠsMM) which adequately models unimodal data. Let $X = \{x_1, \ldots, x_N\}$ be our dataset. We first call UU-test with X as input. In the case that the dataset is considered unimodal, UU-test provides a data subset $S = \{s_1, \ldots, s_{K+1}\}$ that defines a unimodal UMM (see equation 2.1 in Chapter 2). Then, using the UMM information, we initialize our model. A big issue in mixture modeling is the specification of the number of components K. In our case this number is initially provided by the UMM.

To initialize our UΠsMM, we set the parameters of each component k as follows: $a_k = s_k$, $b_k = s_{k+1}$ and $r_k = \sqrt{b_k - a_k}$. As suggested in [91], we choose a small value to initialize g_k ($g_k = 1$), since it makes the distribution wider and thus it is more easy to change its shape. The prior π_k is initialized as $\pi_k = \frac{N_k}{N}$ where N_k is the number of data points in the interval $[a_k, b_k]$.

Training proceeds by applying EM iterations to update model parameters so that the data likelihood is maximized (see Section 3.2.1). However, after each EM iteration, we have to verify whether the updated model remains unimodal. To do this we first check whether there exist model components k that are restricted on very small intervals $[a_k, b_k]$. In such a case very narrow peaks appear in the model's density. We identify those peaks by setting a very small threshold (10^{-4}) to the interval width $b_k - a_k$ and we eliminate those components of the model.

Next, in order to ensure unimodality, we check the second derivative of the model's cdf to determine data points that are multimodality indicators as described previously.



(a) (b)

Figure 3.3: (a) A multimodal ΠsMM pdf with red stars indicating the second formed peak. (b) UΠsMM pdf with the multimodality issue being fixed.

We count the number of such points that fall into each interval $[a_k, b_k]$, and we eliminate the component with the maximum number of counts.

To tackle both of the above sources of multimodality, we eliminate the component by merging its corresponding interval $[a_k, b_k]$ with those of other components. This approach effectively reduces the number of components and enhances the model's generalization. Experimentally, we have concluded that the update of model's parameters after EM step may create overlapping intervals. Two intervals $[a_k, b_k]$ and $[a_l, b_l]$ are considered as overlapping when either $a_l \in [a_k, b_k]$ or $a_k \in [a_l, b_l]$. These overlapping intervals can lead to regions with increased density, which may be the source of multimodality. Therefore, for the selected interval $[a_k, b_k]$, we first check whether it overlaps with other intervals. In case this occurs, we select the closest overlapping interval as its candidate for merging. To determine the closest interval to $[a_k, b_k]$ we first compute the $d_l = min(|a_k - a_l|)$ and $d_m = min(|b_k - b_m|)$ for each $l = 1, ..., K$ and $m = 1, ..., K$ with $l \neq k$ and $m \neq k$. d_l corresponds to the minimum distance among a_k and a_l, while d_m corresponds to the minimum distance among b_k and b_m. Next we detect the closest interval to $[a_k, b_k]$ as follows: if $d_l < d_m$ then the closest interval is $[a_l, b_l]$, otherwise the closest interval is $[a_m, b_m]$. In cases where there are no overlapping intervals, we simply choose the closest interval for merging, as no overlapping intervals are present. The closest interval is detected similarly as the closest overlapping interval. We then proceed to merge these two intervals into one and appropriately adjust the priors π_k and the rest parameters (a_k, b_k, r_k, and g_k). This approach effectively reduces the number of intervals, or in other words, the number

Algorithm 3.2 $new\_model = \text{Merge}([a_k, b_k], K)$

$[a_{k'}, b_{k'}] \leftarrow$ the closest overlapping interval to $[a_k, b_k]$

if $[a_{k'}, b_{k'}] = \emptyset$ **then**

$[a_{k'}, b_{k'}] \leftarrow$ the closest interval to $[a_k, b_k]$

end if

merge $[a_k, b_k]$ and $[a_{k'}, b_{k'}]$ into a new interval $[min(a_k, a_{k'}), max(b_k, b_{k'})]$

update: $a_k = min(a_k, a_{k'}), \quad b_k = max(b_k, b_{k'}), \quad r_k = \sqrt{b_k - a_k}$

$g_k = (g_k + g_{k'})/2, \quad \pi_k = \pi_k + \pi_{k'}$

delete: $a_{k'}, b_{k'}, r_{k'}, g_{k'}, \pi_{k'}$

$K \leftarrow K - 1$

$new\_model \leftarrow \Pi\text{sMM}(x; g, a, r, \pi)$

return new_model

of components, and leads to unimodal models since the sources of multimodality are eliminated.

Fig. 3.3 presents a ΠsMM pdf which is initially multimodal and after applying the fixing method it turns into unimodal. Fig. 3.3a illustrates the initial multimodal pdf with a second low peak in range of $[2, 3]$. To address the issue of multimodality, we identify the points that form the second peak (depicted as stars) and merge overlapping intervals in this region. This merging process reduces the number of components in the model, leading to parameter adjustments that make the model less multimodal and finally unimodal. In Fig. 3.3b, it can be observed that the second peak has been eliminated.

Algorithm 3.1 describes the procedure for fixing multimodality. It takes the model as input and returns an updated version with adjusted parameters. It initially detects possible small intervals. If at least one small interval is found, the algorithm selects this as the interval to merge. Otherwise, it computes the cdf F_Π of the current model and identifies and selects the interval that includes the majority of multimodality indicators as the interval to merge. Then the Merge procedure is called (Algorithm 3.2) that returns a new model with decreased number of components K.

Algorithm 3.2 describes the process of merging two intervals. It takes as input the selected interval $[a_k, b_k]$ and merges this interval with another one $[a_{k'}, b_{k'}]$ with $k' = 1, ..., K$ and $k' \neq k$. The interval $[a_{k'}, b_{k'}]$ is detected as follows: In case $[a_k, b_k]$ overlaps with another interval, then $[a_{k'}, b_{k'}]$ will be the closest overlapping interval

Algorithm 3.3 $new\_model = \text{UΠsMM}(X)$

UMM $\leftarrow$ UU-test(X)
initialize: g, a, r, π from UMM // $K \leftarrow$ number of UMM components
$model \leftarrow \Pi\text{sMM}(x; g, a, r, \pi)$
loop until EM convergence
loop
  $new\_model \leftarrow \text{EM}(X, model)$
  if new_model is multimodal **then**
    $new\_model \leftarrow$ fix_multimodality(new_model) // $K := K - 1$
  end if
  $model \leftarrow new\_model$
end loop
return new_model

to $[a_k, b_k]$. Otherwise, $[a_{k'}, b_{k'}]$ will be simply the closest interval to $[a_k, b_k]$. The result of the merging is a new interval $[min(a_k, a_{k'}), max(b_k, b_{k'})]$. Then the slope parameter of the new component is set equal to the average of g_k and $g_{k'}$, while its prior is set equal to $\pi_k + \pi_{k'}$.

As already mentioned, after each EM step we check the unimodality of the model. If it is determined to be multimodal, we use Algorithm 3.1 and Algorithm 3.2 to resolve the multimodality issue. The above steps are repeated until EM converges to a unimodal solution. The whole method is described in Algorithm 3.3, which takes a dataset X as input and returns a fitted unimodal model (new_model).

3.4 Experimental Results

In our experimental evaluation we have compared the modeling capabilities of UΠsMM against UMM (provided by UU-test) and the single Gaussian model. A variety of unimodal distributions were used including synthetic and real datasets. In the case of real datasets, we fitted the three compared models to their unimodal features. In order to measure the performance of the three statistical models, we used the test log-likelihood criterion. We used 30% of the sample without replacement as a test set. The rest 70% was used as a training set to build the UΠsMM, UMM and Gaussian

models. Then we computed the log-likelihood of each model on the test set (higher values imply better fit).

3.4.1 Synthetic Datasets

We have generated synthetic datasets through sampling from seven unimodal distributions as detailed in Table 3.1. For each distribution, 20 datasets were generated to assess the performance of the three models. We also evaluated the behavior of K (number of components) during fitting. The initial value of K corresponds to the number of UMM components, which is also provided as input to UΠsMM. The final value of K is the reduced number of UΠsMM components. In most cases, we observed a decrease of almost 50% in the initial number of components.

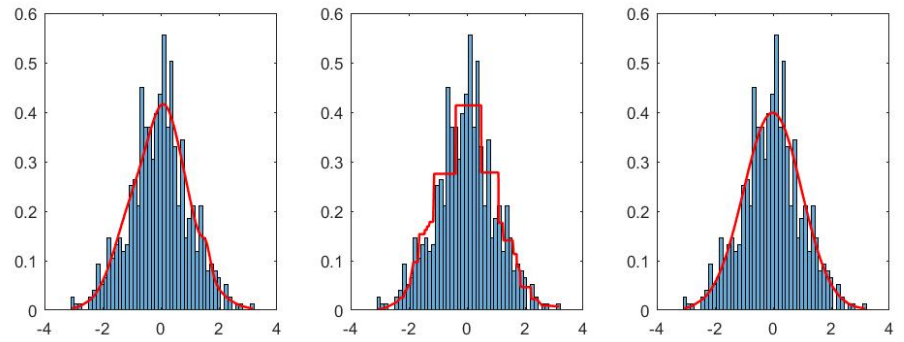
In Table 3.2 we present the average initial and final values of K for each distribution and the mean log-likelihood value of each model on the test sets. It is evident that UΠsMM achieves superior performance in most cases. For datasets generated from a Gaussian distribution, it is reasonable for the Gaussian model to fit better. However, the UΠsMM's performance is quite similar to that of the Gaussian model, since they both provide close log-likelihood results. UΠsMM has also achieved a significant decrease on the number of components (from 29 to 15), resulting in a simpler model with fewer parameters than UMM. In asymmetric triangular and mixture of Student's t and uniform distributions the component decrease is also remarkable (from 23 to 8). In Fig. 3.4, we present the histograms of two datasets sampled from a Gaussian (Fig. 3.4a) and a mixture of Student's t & Uniform (Fig. 3.4b) along with the pdf plots of the three fitted models: UΠsMM (left figure), UMM (middle figure) and Gaussian model (right figure).

3.4.2 Real Datasets

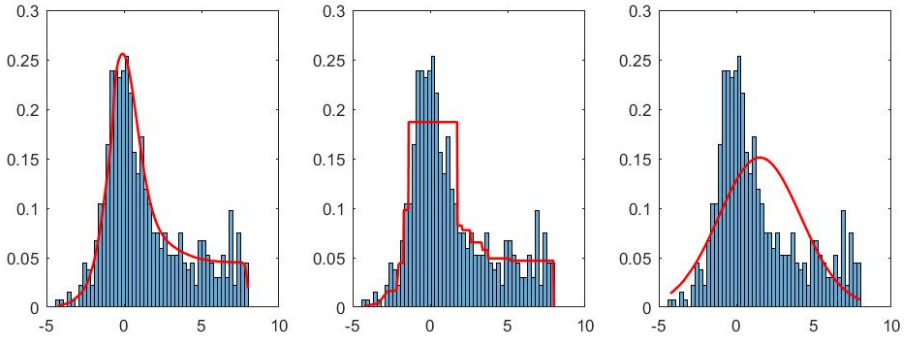
We also evaluated the three models on unimodal features of five known real datasets: Iris, Banknote and Seeds from the UCI Machine Learning Repository [5], Prestige [83], and Boston house [94]. Similar to synthetic datasets, we employed a held-out test set for each dataset to calculate the log-likelihood. The procedure was repeated 20 times and in Table 3.2 we present the average test log-likelihood results along with the mean initial and final number of components. For most features, UΠsMM and UMM provide the best fit. The Gaussian model is superior in the Iris features,

Table 3.1: Synthetic dataset characteristics.

Distribution	Parameters	Training set	Test set
Gaussian(μ, σ^2)	$\mu = 0, \quad \sigma = 1$	1400	600
Student's t(ν)	$\nu = 2$	700	300
Uniform(a, b)	$a = 0, \quad b = 3$	700	300
Triangular (L, U, m)	$L = 0, \quad U = 2, \quad m = 1$	2100	900
Asymmetric Triangular (L, U, m)	$L = -1, \quad U = 3, \quad m = 0$	1400	600
Student's t(ν) & Uniform(a, b)	$\nu = 5, \quad a = 1, \quad b = 8$	1260	540
Uniform(a, b) & Gaussian(μ, σ^2)	$a = -3, \quad b = 2, \quad \mu = 4, \quad \sigma = 1$	2100	900



(a)



(b)

Figure 3.4: Statistical model fitting on two synthetic datasets using UΠsMM (left plot), UMM (middle plot) and Gaussian model (right plot).

with UΠsMM being extremely close to its performance. We should note here that even in the cases where UMM provides a slightly better fit than UΠsMM, the latter achieves to significantly decrease the number of components K, thus constituting a simpler model providing a sufficiently accurate fit.

Table 3.2: Statistical model evaluation using the test set log-likelihood (the higher the better). Bold values indicate the best model in each row. Initial and final number of components are also provided.

Distribution	K		UΠsMM	UMM	Gaussian model
	Initial	Final			
Synthetic					
Gaussian	29	15	−857.468	−868.968	**−855.899**
Student's t	33	18	**−591.29**	−596.006	−727.374
Uniform	1	1	**−341.776**	−345.225	−384.38
Triangular	28	24	**−458.968**	−470.309	−475.614
Asymmetric Triangular	23	8	**−719.646**	−726.533	−751.14
Student's t & Uniform	23	8	**−1198.399**	−1210.339	−1295.897
Gaussian & Uniform	22	12	**−1895.798**	−1898.702	−2082.739
Real					
Iris feat.1	1	1	-54.314	-64.601	**-54.232**
Iris feat.2	1	1	-26.377	-46.032	**-26.373**
Banknote feat.1	21	2	-1001.8	**-999.51**	-1014.6
Seeds feat.1	6	4	**-3.27**	-13.6	-8.55
Seeds feat.2	1	1	**-0.47**	-8.16	-8.07
Seeds feat.5	1	1	**-2.11**	-12.83	-8.17
Prestige feat.2	6	6	-298.083	**-297.898**	-304.299
Prestige feat.3	10	2	-142.304	**-132.559**	-151.465
Prestige feat.4	1	1	**-131.209**	-139.777	-131.526
Boston house feat.6	19	5	**-151.686**	-162.967	-159.917
Boston house feat.8	17	3	-326.116	**-299.808**	-329.274

3.5 Summary

In this chapter, we have proposed a methodology to train a unimodal mixture (UΠsMM) that effectively models univariate unimodal data. UΠsMM is based on a Π-sigmoid mixture model (ΠsMM), where each component is a Π-sigmoid distribution defined as the difference of two translated logistic sigmoids. Depending on the slope value of the sigmoids, it can capture a wide range of data from Gaussian to uniform. This property makes UΠsMM a powerful tool for statistical modeling.

An important aspect in UΠsMM training is its effective initialization which is provided by the UU-test algorithm for deciding unimodality. The model is updated using the EM algorithm, but care is taken so that the unimodality constraint is not violated. If such violation occurs, an appropriate fixing procedure is applied that aims to eliminate multimodality by reducing the number of mixture components.

The modeling capabilities of UΠsMM were tested against the Uniform Mixture Model (UMM) provided by UU-test and the single Gaussian model. Our evaluation included a variety of unimodal distributions using both synthetic and real datasets. For real datasets, the three models were fitted to the unimodal features, and their performance was assessed using the test log-likelihood criterion. The results demonstrate that UΠsMM achieves superior performance in most cases. While the Gaussian model performed better on datasets generated from Gaussian distributions, UΠsMM provided comparable log-likelihood results, showcasing its flexibility and robustness. Furthermore, we observed that UΠsMM significantly reduced the number of components K initially provided by UMM. Overall, UΠsMM proves to be an efficient choice for the statistical modeling of unimodal data, constituting a simpler model with fewer parameters compared to UMM.

Chapter 4

Statistical Modeling of Univariate Multimodal Data

4.1 Introduction

4.2 Detecting Valleys in Data Density

4.3 The Unimodal Mixture Model (UDMM)

4.4 Experimental Results

4.5 Summary

4.1 Introduction

As mentioned in Chapter 1, Sections 1.1.2 and 1.3.1, mixture models (e.g. GMMs) can be used for clustering and density estimation tasks. Other methods[1] achieve clustering by focusing on the underlying density structure, detecting high-density regions (modes) and separating them based on low-density areas. While the above approaches rely on local density estimates for cluster identification, another research direction is to employ unimodality testing. Data unimodality could play a decisive role in building a successful statistical model (such as the Uniform Mixture Model (UMM) introduced in Chapter 2), estimating the number of components and partitioning a dataset into clusters[2]. In Chapter 1, Section 1.3 the significance of mode estimation and valley

[1] More details are provided in Chapter 1, Section 1.3

[2] More details are provided in Chapter 1, Section 1.2.3

detection in understanding the structure of complex datasets is also explained. Mode estimation captures the central tendencies in complex distributions (e.g., skewed or asymmetric), while valley detection reveals the boundaries between modes, facilitating a clearer distinction of clusters or regions with lower density.

This chapter addresses both the detection of valleys in univariate multimodal data and the development of a corresponding statistical model. Since a UMM can be used to model univariate unimodal data (as described in Chapter 2), we propose a more general method, which builds a statistical mixture model that models adequately *univariate multimodal data*, i.e., data generated by distributions with more than one mode (peak) [76]. This statistical model is called *Unimodal Mixture Model* (UDMM). Its mixture components correspond to arbitrary unimodal distributions and each of them is modeled using a UMM provided by UU-test algorithm. Thus UDMM is actually a hierarchical mixture model, since each component is also a uniform mixture model. We also propose a technique, called *UniSplit*, for determining valley points of univariate multimodal data achieving to split the original data into unimodal subsets. Our approach relies on the idea of unimodality. We introduce properties of critical points (gcm/lcm points) of the data empirical distribution function (ecdf) that provide indications on the existence of density valleys. These properties are exploited in the proposed UniSplit algorithm. Based on the computed valley points, the initial dataset is partitioned into unimodal subsets. Then we model each unimodal subset with a UMM and obtain the final Unimodal Mixture Model (UDMM) as a mixture of the computed UMMs. In this way the number of UDMM components is automatically determined as a result of the unimodal data splitting procedure.

The proposed approach is very flexible, since it makes no assumptions about the specific parametric of each unimodal mixture component. Therefore it can effectively model datasets generated by sources of different probability density (e.g., one Gaussian and one uniform). In addition the method requires no training, while it demonstrates the significant advantage that (apart from a typical statistical significance level of a uniformity test) it does not include user specified hyperparameters, such as for example the number of components in GMMs, the kernel bandwidth in mean shift, etc.

In a nutshell, we make the following contributions:

- A novel valley detection method (called UniSplit) for determining valley points of univariate multimodal datasets and obtaining partitions into unimodal sub-

sets is proposed.

- Since UniSplit works with the ecdf of the data, significant properties of the ecdf are presented based on critical points (called gcm and lcm) in the convex hull of the ecdf graph. Their location on the graph and the uniformity of the data in intervals between successive critical points help us to draw conclusions on the existence of valleys in those intervals.

- A statistical mixture model is proposed, where each mixture component corresponds to a unimodal distribution. UniSplit is used to split a multimodal dataset into unimodal subsets and then, the UU-test algorithm is employed to model each unimodal subset with a Uniform Mixture Model (UMM). The final model is called Unimodal Mixture Model (UDMM).

- The proposed method is flexible, requires no training, while apart from the typical statistical significance level, it does not include user specified hyperparameters. The number of components in the UDMM model is automatically determined by the UniSplit algorithm, rather than being manually defined by the user.

- Experiments are conducted using both synthetic and real datasets. Comparisons are made with other clustering algorithms to evaluate the UniSplit algorithm, while the performance of UDMM for statistical modeling is also evaluated.

The rest of this chapter is organized as follows. Section 4.2 introduces the ideas implemented in our method for detecting valley points of the data density. Section 4.3 presents the proposed UniSplit methodology and defines the Unimodal Mixture Model (UDMM). Section 4.4 presents extensive experimental results aiming at evaluating in various tasks involving synthetic and real datasets, both the effectiveness of the splitting procedure as well as the performance of the constructed unimodal mixture model. Finally, Section 4.5 provides a brief summary of the chapter.

4.2 Detecting Valleys in Data Density

The shape of the ecdf of a univariate dataset provides crucial information on the multimodality of the underlying data distribution. Gcm and lcm points constitute key

points in the ecdf plot, since their location and the uniformity of intervals defined by successive gcm/lcm points constitute significant indicators related to the existence of density valleys in those intervals. We have identified and present below three main cases for intervals $[a, b]$ defined by successive gcm or lcm points:

(a) Uniformity of $X(a, b)$ indicates no density valley in $[a, b]$.

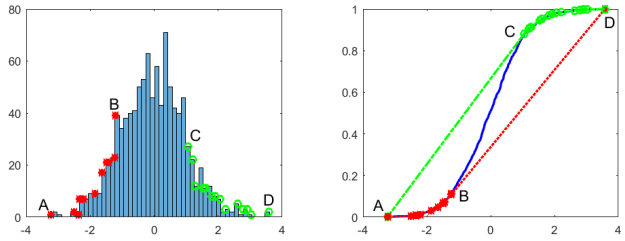
(b) If $X(a, b)$ is non-uniform and unimodal, a single density valley exists in $[a, b]$.

(c) If $X(a, b)$ is non-uniform and multimodal, multiple density valleys exist in $[a, b]$.

We clarify below in detail each of the above cases and present illustrative figures.

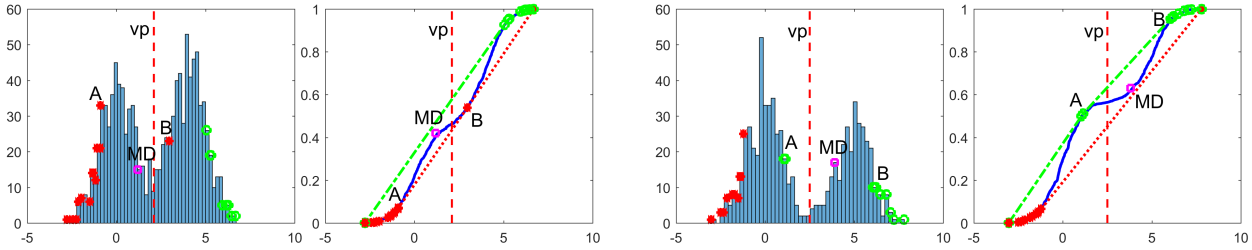
(a) *Uniform $X(a, b)$ indicates no density valley in $[a, b]$:* in case $X(a, b)$ is uniform, the corresponding ecdf segment is linear. Based on the type of a and b (gcm or lcm), they both lie on the increasing or decreasing part of the same mode on a histogram plot, respectively. Fig. 4.1 illustrates the histogram and ecdf plots of a unimodal dataset. The ecdf segments between the gcm/lcm points are linear, indicating uniformity. On the histogram plot, the gcm points (between A and B) lie on the increasing part of the mode, and the lcm points (between C and D) lie on the decreasing part, thus no density valleys are detected between gcm or between lcm points.

(b) *Non-uniform and unimodal $X(a, b)$ indicates a single density valley in $[a, b]$:* non-uniformity of $X(a, b)$ indicates the existence of non-linear ecdf segments (i.e., convex and/or concave ecdf segments) within $[a, b]$. If an interval $[a, b]$ exists, where a, b are successive gcm points and $X(a, b)$ is non-uniform and unimodal, this implies that the ecdf segment is exclusively concave. This property is ensured, since, if it were partially convex and partially concave, then $X(a, b)$ would be multimodal. Thus, a and b lie on increasing parts, while the concave segment corresponds to a decreasing part between a and b on a histogram plot, with one valley (and one peak) being detected in $[a, b]$. It is evident that a and b lie on increasing parts of successive modes. Fig. 4.2a illustrates the histogram and ecdf plots of a bimodal dataset sampled from two close Gaussians. On the ecdf plot, we can see that A, B are gcm points and $X(x_A, x_B)$ is not uniformly distributed (AB is non-linear), since the ecdf segment AB is concave (unimodal $X(x_A, x_B)$). It is evident that one density valley is formed between A and B on the histogram plot. Similarly, in case a, b are successive lcm points, the ecdf segment is convex. The lcm points a and b lie on successive decreasing parts of different modes, while the convex segment corresponds to an increasing part between a and b on a



(a)

Figure 4.1: Histogram: gcm (AB part) and lcm (CD part) correspond to increasing and decreasing parts, respectively. Ecdf: AB, BC and CD correspond to the convex, intermediate and concave part, respectively.

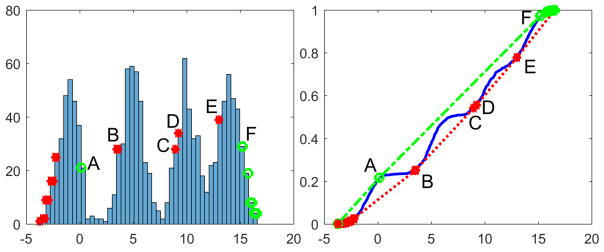


(a) (b)

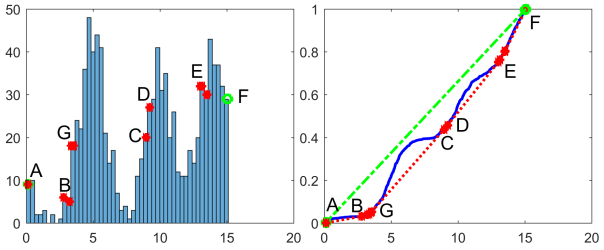
Figure 4.2: Histogram and ecdf of a bimodal dataset. The non-uniform and unimodal $X(x_A, x_B)$ indicates a density valley between A and B. MD is a point close to the valley. vp is the valley point. (a) A, B are gcm points on increasing parts of successive modes. (b) A, B are lcm points on decreasing parts of successive modes.

histogram plot. Thus one density valley (and one peak) is detected in $[a, b]$. Fig. 4.2b illustrates the histogram and ecdf plots of a bimodal dataset. On the ecdf plot, AB is convex, while on the histogram plot A and B lie on the decreasing parts of different modes with one density valley (and one peak) being formed between them.

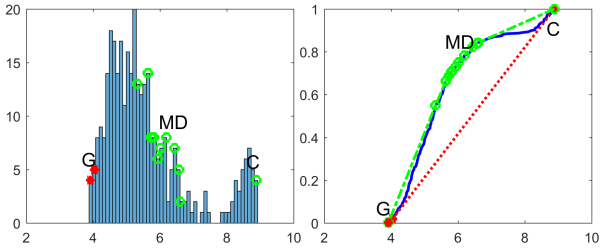
(c) *Non-uniform and multimodal $X(a, b)$ indicates multiple density valleys in $[a, b]$:* similarly to case (b), non-uniformity of $X(a, b)$ corresponds to a non-linear ecdf segment, and since $X(a, b)$ is multimodal, the corresponding ecdf is expected to include convex and concave segments. Thus, multiple increasing/decreasing parts exist on a histogram plot, i.e., multiple density valleys are formed. We should note here that all multimodal sets $X(a, b)$ are also non-uniform. We choose to refer both properties of multimodality and non-uniformity for sake of clarity. In Fig. 4.3a the histogram



(a) Multiple density valleys in non-uniform and multimodal $X(x_A, x_F)$.

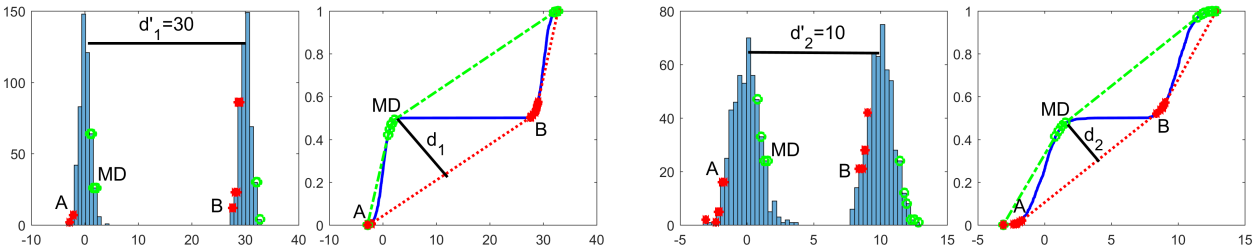


(b) Candidate splitting intervals $[x_A, x_B]$, $[x_G, x_C]$, $[x_D, x_E]$ in zoomed set $X(x_A, x_F)$. Best splitting interval: $[x_G, x_C]$.

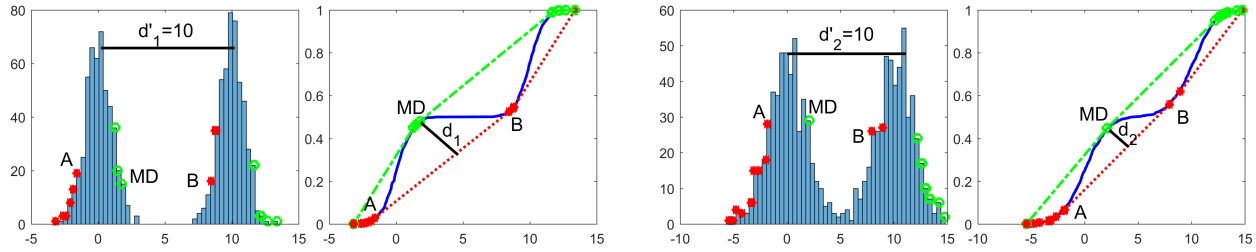


(c) Non-uniform and unimodal set $X(x_G, x_C)$ with a density valley being formed between G and C. MD point is also illustrated.

Figure 4.3: Histogram and ecdf plot of a multimodal dataset with its best splitting intervals, processed recursively until a non-uniform and unimodal interval containing a single valley point is detected.



(a) Closer peaks demonstrate a lower degree of non-uniformity.



(b) Smaller valley depth corresponds to lower degree of non-uniformity.

Figure 4.4: Histogram and ecdf plots of bimodal datasets with varying peak distances and valley depths. The black segments on the pdfs correspond to the horizontal distances (d'_1 and d'_2) between the two peaks, while on the ecdfs correspond to the max distances (d_1 and d_2) of MD from line segment AB.

and ecdf plot of a multimodal dataset are illustrated. On the ecdf plot, A and F are successive lcm points with $X(x_A, x_F)$ being non-uniform and more specifically multimodal. Multimodality is evident by the multiple peaks and density valleys on the histogram plot and the multiple convex/concave parts between A and F on the ecdf plot.

4.2.1 Multimodality Degree

To apply our splitting algorithm, we need a fast and easy way to assess of the *degree of multimodality* of a data subset. To define the multimodality degree, we consider the *distance among the peaks* and the *depth of the valley between the peaks on the data histogram*. As the distance and the depth become larger, the degree should be higher. Since multimodality implies that at least two peaks (and at least one valley) exist, it is strongly related to non-uniform intervals (considering cases (b) and (c)). Thus we measure the multimodality degree of the data in an interval $[a, b]$ as the distance from

uniformity. Let $F_U(x)$ be the cdf of the uniform distribution in $[a, b]$. Then the value $d = \max_{x \in X(a,b)} (|F(x) - F_U(x)|)$ computes the maximum distance (deviation) of the ecdf from uniformity. In plots we denote as $MD(x_{MD}, F(x_{MD}))$ the point of maximum distance.

Fig. 4.4a shows two bimodal datasets with peaks at 0 and 30 (left) and at 0 and 10 (right). Both datasets are multimodal due to non-linear ecdf segments AB. The black segments on the ecdf plots illustrate the maximum distances (d_1 and d_2) of MD from the line segment connecting $(x_A, F(x_A))$ to $(x_B, F(x_B))$, with d'_1 and d'_2 representing the distances between the peaks on the pdf plots. The peaks in the dataset on the right are closer ($d'_2 < d'_1$), which is also evident on the ecdf plots where $d_2 < d_1$. Valley depth is also related to the distance from uniformity as shown in Fig. 4.4b. Two bimodal datasets with equally spaced peaks (distance = 10) are illustrated. The left histogram shows a deeper valley than the right, with the deeper valley corresponding to a higher degree of non-uniformity on the ecdf plots ($d_1 > d_2$).

In what concerns the location of the maximum deviation (MD) point, in case there exists a single valley in $[a, b]$, the MD point will be very close to the valley point. In particular, if a, b are successive gcm points, then a sequence of increasing (containing a), decreasing (containing MD) and again increasing (containing b) parts is formed. It is clear that $[x_{MD}, b]$ defines the valley region, since a valley exists between a decreasing part and an increasing part. Similarly, if a, b are successive lcm points, the valley region will be $[a, x_{MD}]$, since a sequence of decreasing (containing a), increasing (containing MD), decreasing (containing b) parts exists. Fig. 4.2a and Fig. 4.2b illustrate these cases with histograms and ecdf plots, showing the relationship between the MD point and valley regions. In Fig. 4.2a, A and B are gcm points, thus between MD and B a valley is identified, while in Fig. 4.2b a valley is identified between A and MD, since A and B are lcm points.

4.3 The Unimodal Mixture Model (UDMM)

In this section, we propose a method that builds a statistical mixture model for modeling univariate multimodal data. In this model, each component is unimodal as determined by UU-test for unimodality. First, we present a technique, called *UniSplit*, that splits multimodal data into unimodal sets. To achieve this, we identify an interval

with high degree of multimodality and then compute an appropriate valley point inside this interval. Based on the computed valley points, we recursively partition the data until unimodal segments are obtained. Finally, we provide the formulation for the Unimodal Mixture Model (UDMM) where each component constitutes a statistical model of a unimodal subset in the form of a uniform mixture model.

4.3.1 The UniSplit Algorithm

Based on Section 4.2, a dataset is characterized as multimodal, when at least one non-uniform interval $[a, b]$ defined by successive gcm or lcm points of the ecdf exists. In that case, at least one valley is noted inside the interval. In our method, we aim to compute valley points in the density of multimodal datasets, thus we need to detect non-uniform intervals between successive gcm or lcm points in the ecdf. These intervals constitute *candidate splitting intervals*, since they contain at least one valley. To detect candidate splitting intervals, the UU-test algorithm is applied, which utilizes a uniformity test (Kolmogorov-Smirnov [12]), to decide whether a set of points follows the uniform distribution or not. As happens with every statistical test, the uniformity test requires a user-defined statistical significance level as input (we use the value equal to 0.01 in our experiments). We should note here that apart from the uniformity significance level, our approach does not include any other user specified hyperparameters.

Our method starts by calling UU-test that takes the initial dataset X as input. In case X is unimodal and since no valley points are detected in unimodal datasets, the algorithm terminates and returns the corresponding UMM. Let G and L be the ordered sets of gcm and lcm points respectively, and GL be the ordered union of them. Let also $maxG$ and $minL$ be the maximum value of G and minimum value of L, respectively. In case X is multimodal, we search for non-uniform intervals defined by successive gcm or lcm points to detect valley points (cases (b) and (c) in Section 4.2).

A special case occurs when $maxG < minL$, i.e., all gcm points precede all lcm points. If $X(maxG, minL)$ is uniform (linear ecdf) then it is ensured that no valleys exist in $[maxG, minL]$. Otherwise, we compute the gcm set G' and lcm set L' of $X(maxG, minL)$ to detect possible valley points in non-uniform intervals defined by successive gcm (or lcm) points in G' (or in L'). Thus, we augment the original GL

set with the new gcm and lcm points, $GL := G \cup G' \cup L' \cup L$.

Based on the computed GL set, UU-test detects and finally returns a set I of candidate splitting (multimodal) intervals where at least one valley exists. Next, we determine the multimodality degree of each candidate interval and select the one with the highest degree, called as the *best splitting interval*. Let $T = [a^*, b^*]$ be the best splitting interval. If $X(a^*, b^*)$ is unimodal, a single valley is formed in T (case (b)), otherwise, multiple valleys are detected (case (c)).

In case of a single valley in $T = [a^*, b^*]$, the following strategy is used to determine a point in the valley region. We compute the MD point of T and subsequently, use the x_{MD}, a^* and b^* values to compute a valley point. If a^*, b^* are gcm points, the valley point lies in the middle of x_{MD} and b^*. Since MD is on the decreasing part of the mode and b^* is on the increasing part of the next mode, a valley is formed between them, thus their middle point seems a reasonable location for the valley point. Similarly, if a^*, b^* are lcm points, a valleys exists between the decreasing part (a^*) of the mode and the increasing part (MD) of the next mode, thus we compute the valley point as the middle point between a^* and x_{MD}. In Fig. 4.2 the best splitting interval $T = [x_A, x_B]$ and the MD point of T are illustrated. In Fig. 4.2a, A and B are gcm points. On the histogram plot a valley exists between MD and B, thus the middle point (vp) between x_{MD} and x_B is considered as a reasonable location for the valley point. Similarly, in Fig. 4.2b A, B are lcm points with the average of x_A and x_{MD} denoting the vp.

In case where $X(a^*, b^*)$ is multimodal, multiple valleys exist in $T = [a^*, b^*]$. For an accurate valley point computation, we aim at detecting an interval with a single valley. Thus, we focus on the multimodal set $X(a^*, b^*)$ and work recursively, until we detect a non-uniform and unimodal interval. Such an interval will contain a single valley, thus we can follow the previously described methodology to compute a valley point. Fig. 4.3a shows the histogram and ecdf plots of a multimodal dataset with its best splitting interval being $[x_A, x_F]$, identified as non-uniform and multimodal. Focusing on $X(x_A, x_F)$ (Fig. 4.3b), three candidate intervals are identified: $[x_A, x_B]$, $[x_G, x_C]$, and $[x_D, x_E]$. Among these, $T = [x_G, x_C]$ demonstrates the highest degree of non-uniformity (largest distance of the ecdf of $X(x_G, x_C)$ from the line segment GC) and is unimodal, since a single peak is formed (histogram in Fig. 4.3c). Thus, T contains a single valley, making it the final splitting interval for valley point computation. MD on the ecdf plot (Fig. 4.3c) is a close point to the valley region and helps us compute

Algorithm 4.1 vp = find_vp(X) // X is multimodal

Compute GL set of X
$I \leftarrow$ set of candidate splitting intervals of GL
$T = [a^*, b^*] \leftarrow$ best splitting interval
if $X(a^*, b^*)$ is unimodal **then**
  $x_{MD} \leftarrow$ compute MD point of T
  if a^*, b^* gcm points **then**
    $vp \leftarrow \frac{x_{MD}+b^*}{2}$
  else
    $vp \leftarrow \frac{a^*+x_{MD}}{2}$
  end if
  return vp
else
  $vp \leftarrow$ find_vp($X(a^*, b^*)$)
end if

the valley point. Algorithm 4.1 presents the steps of computing a valley point of a univariate multimodal dataset X. It takes X as input and returns an appropriate valley point (vp).

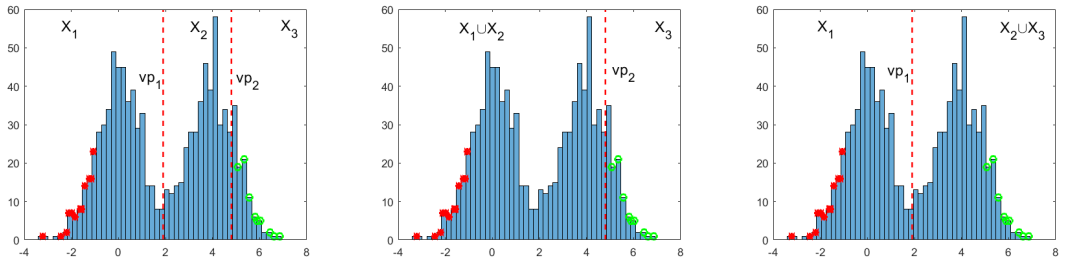
After computing a valley point vp, we split the data into two subsets: a left subset X_L (points on the left of vp) and a right subset X_R (points on the right of vp). Then, the method runs recursively on each subset, until all obtained subsets are unimodal. The whole method (UniSplit algorithm) is described in Algorithm 4.2, which takes a univariate dataset X and a list of valley points (vp_list) as input and returns an updated vp_list that partitions the data domain into adjacent unimodal intervals.

4.3.2 Merging Adjacent Intervals

It should be noted that there exist cases where *oversplitting* may occur, due to low density variations at the tails of unimodal subsets. This results in unnecessary splittings that define subsets with a small number of data points. To tackle this issue, we follow the typical merging procedure: Let our dataset X has been splitted into R adjacent unimodal subsets: $X = \{X_1, X_2, ..., X_R\}$. We iteratively merge the two first sets into one set and check its unimodality. In case it is unimodal we replace the two sets with their union, otherwise we merge the next two sets and repeat the procedure.

Algorithm 4.2 vp_list = UniSplit(X, vp_list)

result $\leftarrow$ UU-test(X)
if result = unimodal **then**
  return vp_list
end if
$vp \leftarrow$ find_vp(X)
$vp\_list \leftarrow vp\_list \cup \{vp\}$
$X_L \leftarrow X(x_1, vp), \quad X_R \leftarrow X(vp, x_N)$
$vp\_list \leftarrow$ UniSplit(X_L, vp_list)
$vp\_list \leftarrow$ UniSplit(X_R, vp_list)
return vp_list
// First call: vp_list=UniSplit($X, \emptyset$)



(a) (b) (c)

Figure 4.5: (a) Bimodal dataset with two computed valley points by UniSplit. (b) Omitting vp_1 leads to a multimodal set $X_1 \cup X_2$, thus vp_1 is necessary. (c) Merging X_2 and X_3 (omitting vp_2) leads to a unimodal set, thus vp_2 can be deleted. vp_1 is the final valley point.

The iterations stop when there is no unimodal union of successive sets. In this way a minimal unimodal partition is obtained, i.e., there is no union of successive subsets resulting in a unimodal set.

Fig. 4.5a illustrates the histogram of a bimodal dataset with a single density valley. However, two valley points (vp_1, vp_2) have been determined by UniSplit method with the resulting unimodal subsets being X_1, X_2 and X_3. In Fig. 4.5b we merge sets X_1 and X_2 (by omitting vp_1) resulting to a multimodal set $X_1 \cup X_2$. This means that vp_1 is a required split point and cannot be omitted. Next, we merge X_2 with X_3 (omitting vp_2), which results to a unimodal set $X_2 \cup X_3$ (Fig. 4.5c). In such case, we delete vp_2 and our final solution contains a single valley point (vp_1).

4.3.3 Computational Complexity

The computational complexity mainly depends on determining the gcm/lcm points of the ecdf, which can be computed in $O(n \log n)$ using the convex hull of the ecdf plot [74]. In case the data is unsorted, an additional $O(n \log n)$ is needed. Once the gcm/lcm points are computed, calculating the multimodality degree of a subset requires $O(n)$, and the valley point is computed in $O(1)$. Thus, computing the first valley point has a total complexity of $O(n \log n)$. As the method iterates through subsets after each split, with far fewer splits than n, the overall complexity remains $O(n \log n)$. Additionally, the merging procedure incurs $O(n \log n)$ complexity due to the UU-test for unimodality.

4.3.4 UDMM formulation

Based on the result of the UniSplit algorithm which splits multimodal data into unimodal subsets, a mixture model can be defined with each component modeling the unimodal data of each subset. More specifically, given a univariate dataset X, we first apply UniSplit method to obtain unimodal subsets of X. Then we employ UU-test to generate a UMM that models each unimodal set. Thus we obtain a hierarchical statistical model in the form of a *mixture of UMMs*, where each component is unimodal. We call such a model as *Unimodal Mixture Model* (UDMM). Let we split X into K unimodal subsets, i.e., $X = \{X_1, ..., X_K\}$. Thus we can build a UDMM with K components where each component j is unimodal with $j = 1, ..., K$.

Let N be the size of X and N_j be the size of X_j. For each unimodal subset X_j, UU-test provides the set $S_j = \{s_1^j, \ldots, s_{M_j+1}^j\}$. Then a UMM with M_j components is computed for X_j, where each UMM component i is uniformly distributed in the range $[s_i^j, s_{i+1}^j]$, $(i = 1, ..., M_j)$. Let N_{ij} be the the number of data points of X_j in each interval $[s_i^j, s_{i+1}^j]$. The UDMM pdf of the multimodal set X is defined as follows [74]:

$$p(x) = \sum_{j=1}^{K} w_j \sum_{i=1}^{M_j} \frac{\pi_{ij}}{s_{i+1}^j - s_i^j} I(x \in [s_i^j, s_{i+1}^j)), \quad w_j = \frac{N_j}{N}, \quad \pi_{ij} = \frac{N_{ij}}{N_j}$$

It should be noted that the computed UDMM could also be used to *generate synthetic data samples following the same multimodal distribution as the original dataset*.

4.4 Experimental Results

This section presents the experimental evaluation of our method across various tasks. At first, the modeling performance of UDMM was assessed using synthetic and real datasets. Next, its effectiveness in splitting tasks, mode estimation, and splitting quality was evaluated. UDMM's applicability to image segmentation based on pixel intensity was tested, followed by its use as a probability density model in the Naive Bayes [7] classification method, where class distributions of each feature are modeled by a UDMM. Finally, we provide examples involving noise and outliers, demonstrating the robustness of our method and discuss the impact of the statistical significance level (α) on our method.

4.4.1 Modeling Multimodal Data with UDMM

We conducted a series of experiments using synthetic and real datasets to evaluate the statistical modeling capabilities of UDMM against GMM, KDE and GMDEB[3] [39]. We have generated synthetic datasets by sampling from various univariate multimodal distributions defined as mixtures of different unimodal distributions: i) $N(\mu, \sigma, n)$: Gaussian (Normal) distribution with mean μ and standard deviation σ, ii) $U(a, b, n)$: uniform distribution between a and b, iii) $Tr(l, d, u, n)$: triangular distribution with lower limit l, mode d and upper limit u, iv) $St(\nu, l, s, n)$: Student's t distribution with ν degrees of freedom, location l and scale s, v) $C(l, s, n)$: Cauchy distribution with location l and scale s, and vi) $\Gamma(k, \theta, l, n)$: Gamma distribution with shape k, scale θ and location l. In all cases, the parameter n indicates the dataset size.

Specifically, the synthetic datasets were generated from 12 multimodal distributions (D1 - D12), as shown in Table 4.1. The size of each distribution is also presented as a multiple of m, where $m = 100$. We also evaluated the four models on 9 real datasets [97]. The size and the description of each real dataset is also provided in Table 4.1.

For each synthetic distribution, 50 datasets were generated, and the four models were fitted to each dataset. While UDMM automatically estimates the number of components, GMM requires this number as input. Two criteria were used for this task: BIC [7] and the silhouette score [72]. For BIC, we fit the dataset under consideration using several GMMs with components k ranging from 1 to 10, and the GMM corresponding to the k value yielding the lowest BIC was considered the best

[3]GMDEB is implemented using the mclustAddons package [95, 96].

Table 4.1: Characteristics of synthetic and real datasets.

Name	Parameters	
Synthetic		
D1	$N(0,1,5m) \cup N(6,1,8m)$	
D2	$N(-1,0.8,20m) \cup N(4,1.5,25m)$	
D3	$St(2,0,1,5m) \cup U(4,7,2m) \cup N(10,1,4m)$	
D4	$Tr(-5,-4,0,3m) \cup Tr(1,5,6,5m) \cup U(7,10,2m)$	
D5	$\Gamma(1,2,0,5m) \cup Tr(5,6,7,5m) \cup N(10,0.2,5m) \cup St(10,15,1,8m)$	
D6	$C(0,2,m) \cup U(50,55,3m) \cup U(100,105,3m) \cup St(1,200,1,m)$	
D7	$U(-1,1,10m) \cup U(2,7,12m)$	
D8	$St(1,-10,1,2m) \cup St(2,0,1,3m) \cup St(1,5,1,3.5m) \cup St(3,15,1,2.5m) \cup St(5,20,1,4m)$	
D9	$U(-20,-15,10m) \cup U(-10,0,25m) \cup U(1,10,30m) \cup U(12,14,20m) \cup U(20,50,15m) \cup U(55,60,5m)$	
D10	$U(-15,-7,50m) \cup N(-2,4,40m) \cup N(9,3,30m) \cup U(15,20,20m)$	
D11	$St(5,-2,1,2m) \cup N(5,0.5,2m) \cup U(7,10,2m) \cup \Gamma(2,3,12,2m) \cup U(25,30,2m) \cup Tr(40,45,50,2m) \cup Tr(55,56,60,2m)$	
D12	$St(1,-50,1,m) \cup C(0,2,m) \cup U(30,60,m)$	
Real	Size	Description
suicide	$n = 86$	Lengths of spells of psychiatric treatment undergone by control patients in a suicide study.
racial	$n = 56$	Proportion of white student enrollment in school districts in Nassau County (Long Island, New York), for the 1992-1993 school year.
acidity	$n = 155$	Acidity index measured in a sample of lakes in the Northeastern United States.
faithful eruptions	$n = 272$	Eruption duration of the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA.
faithful waiting	$n = 272$	Waiting time in between eruptions of the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA.
galaxy	$n = 82$	Velocities of distant galaxies, diverging from our own galaxy.
enzyme	$n = 245$	Distribution of enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances.
stamps	$n = 485$	Thickness measurements (in millimeters) of unwatermarked used white wove stamps of the 1872 Hidalgo stamp issue of Mexico.
geyser	$n = 272$	Interval times between the starts of the geyser eruptions on the Old Faithful Geyser.

solution for the dataset. For the silhouette score, k ranged from 2 to 10 (as silhouette does not support $k = 1$), with the k achieving the highest silhouette score selected as the best GMM solution. Since GMDEB [39] utilizes GMMs for density estimation, k is estimated using BIC, as with the typical GMM. For KDE, we used a Gaussian kernel, and considered two rules for the bandwidth estimation: Scott's rule [98] and Silverman's rule [99].

To evaluate the quality of the obtained UDMM, GMM, KDE and GMDEB solutions, we used the two-sample Kolmogorov-Smirnov (KS) test criterion. The two-sample KS test computes the maximum absolute difference between the ecdfs of two datasets. In the case of synthetic datasets, in each experiment we used a dataset generated from the ground truth distribution and compared it (using the two-sample KS test) with a dataset generated from each of the four models fitted on the generated dataset. In the case of real datasets we compared the original dataset with a dataset generated from each of the four fitted models. We repeated the above procedure 50 times and obtained the average distance (KS statistic) and the average number of components (k) for each model. The smaller the distance provided by the KS test, the better the obtained statistical model. The results are presented in Table 4.2.

The results in Table 4.2 demonstrate that UDMM effectively models univariate multimodal data. While GMM and KDE excel for datasets generated by Gaussian

Table 4.2: Statistical model evaluation using the two-sample KS test (the lower the better). Bold values indicate the best model in each row. The ground truth number of components ($k^\star$) (in case of synthetic datasets) and the average estimated number of components (k) are also provided.

	Average KS statistic						Average number of components (k)				
Name	GMM (BIC)	GMM (Sil)	KDE (Scott)	KDE (Silverman)	GMDEB	UDMM	$k^\star$	GMM (BIC)	GMM (Sil)	GMDEB	UDMM
Synthetic											
D1	0.031	0.031	0.026	**0.025**	0.053	0.032	2	**2**	**2**	2.82	**2**
D2	**0.013**	**0.013**	0.014	**0.013**	0.019	0.016	2	**2**	**2**	4.1	**2**
D3	**0.026**	0.044	0.027	0.029	0.172	0.027	3	4.96	3.94	1.78	**3.1**
D4	0.030	0.037	0.030	0.027	0.047	**0.025**	3	6.14	**3**	4.26	**3**
D5	0.021	0.022	0.030	0.032	0.099	**0.017**	4	7.26	**4**	3.18	**4**
D6	0.037	0.061	0.032	0.032	0.267	**0.031**	4	8.78	4.94	1.78	**4.1**
D7	0.021	0.042	0.025	0.025	0.022	**0.020**	2	8.66	2.04	8.16	**2**
D8	0.026	0.140	0.024	0.022	0.306	**0.018**	5	8.84	2.62	1.38	**5.06**
D9	0.016	0.079	0.009	0.010	0.017	**0.007**	6	10	2	9.9	**6**
D10	0.011	0.049	0.007	0.007	0.011	**0.006**	4	9.55	2	8.45	**4.05**
D11	0.033	0.051	0.024	0.024	0.072	**0.017**	7	8.28	6.24	3.78	**7**
D12	0.061	0.132	0.053	0.052	0.281	**0.048**	3	7.32	**3.02**	1.48	**3.02**
Real											
suicide	0.097	0.135	0.098	0.094	0.115	**0.011**		6	3	2	1
racial	0.145	0.145	0.372	0.381	0.225	**0.120**		2	2	1	1
acidity	**0.082**	**0.082**	0.100	0.100	0.165	0.090		2	2	2	1
faithful eruptions	0.070	0.070	0.080	0.080	0.157	**0.050**		2	2	2	2
faithful waiting	0.070	0.070	0.080	0.090	0.168	**0.050**		2	2	2	2
galaxy	0.121	0.121	**0.048**	0.085	0.390	0.109		3	3	1	2
enzyme	0.085	0.093	0.220	0.240	0.155	**0.044**		2	3	2	2
stamps	0.475	0.565	0.478	0.478	0.099	**0.058**		3	2	2	1
geyser	**0.051**	**0.051**	0.058	0.073	0.161	0.062		2	2	2	1

distributions (e.g., D1 and D2), UDMM's performance is comparable. In other cases, UDMM outperforms, accurately estimating the true number of components ($k^\star$), unlike GMM (BIC), which often overestimates, and GMDEB, which provides less accurate results. For real data, UDMM performs well except for acidity, galaxy, and geyser datasets, where it uses fewer components than GMM. However, it is noteworthy that UDMM achieves its performance using only a single component for the acidity and geyser datasets, while GMMs employ two components. Overall, UDMM is a successful statistical model for univariate multimodal data, correctly estimating components in synthetic datasets and providing accurate modeling solutions for real data with fewer components compared to other methods.

In Fig. 4.6, we present the histogram and pdf plots of the solutions provided by the two best-performing models, namely GMM (left plot) and UDMM (right plot), for some of the datasets from Table 4.1. For the synthetic datasets, GMMs were trained using the true number of components ($k^\star$), since the estimated number of components

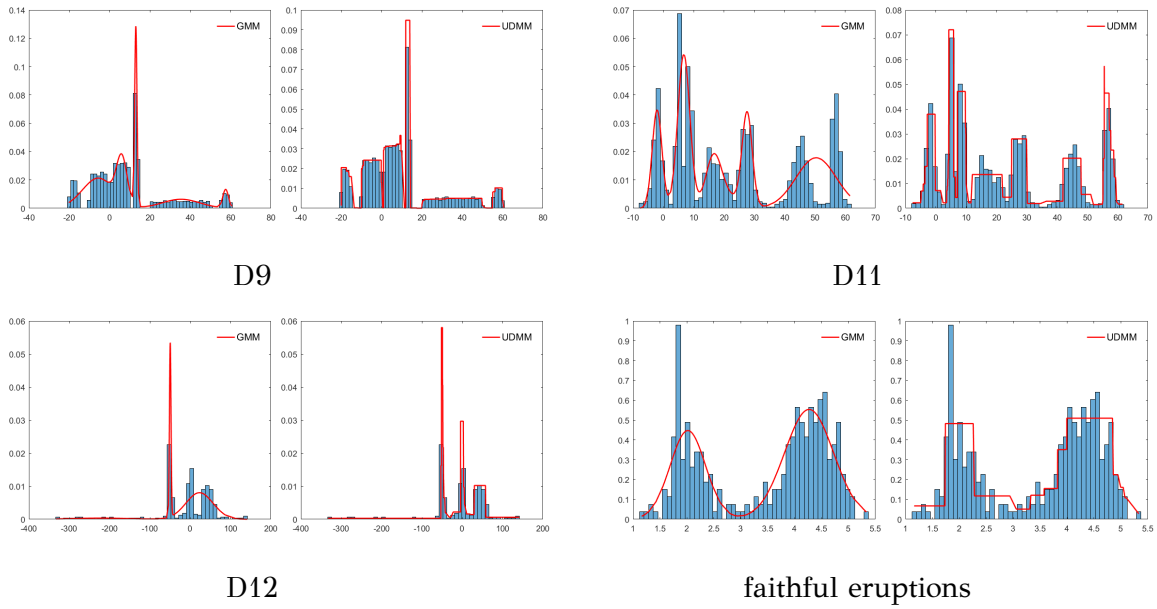


Figure 4.6: Examples of statistical model fitting results on several datasets using GMM and UDMM.

(k) for GMM (BIC) and GMM (Sil) is averaged in Table 4.2. For the real datasets, GMMs were trained using the minimum number of components provided by GMM (BIC) and GMM (Sil) in Table 4.2. It is evident that the obtained UDMMs constitute accurate statistical models for the datasets, whereas the GMMs do not always provide adequate solutions. For instance, in the plots of D11 and D12 in Fig. 4.6, although GMM uses the ground truth number of components ($k^\star = 7$ and $k^\star = 3$, respectively), it fails to accurately fit the two rightmost components. In contrast, UDMM successfully captures these components without requiring prior knowledge of $k^\star$.

4.4.2 Multimodal Data Splitting

We also assessed the performance of UniSplit on partitioning univariate synthetic data, focusing on the accurate estimation of the number of modes and the quality of data splitting. We compare UniSplit with TailoredDip[4] [55], FTC[5] [4], mean shift[6] [47, 48], modclust[7] [42] and MEM[8] [53, 54]. TailoredDip and FTC rely on exploiting unimodality as UniSplit does, while the remaining three methods (mean shift, mod-

[4]TailoredDip is implemented using the ClustPy package [100] in Python.
[5]FTC is implemented in Matlab as described in [4].
[6]For mean shift we use the sklearn package in Python.
[7]The implementation of modclust is available in http://matematicas.unex.es/ jechacon.
[8]MEM is implemented using the mclustAddons package [95, 96] in R.

Table 4.3: Characteristics of synthetic datasets.

Name	Distribution Parameters
D13	$N(0, 1.7, 700) \cup N(5, 1, 500)$
D14	$U(-1, 3, 300) \cup U(8, 10, 200)$
D15	$Tr(0.8, 1, 5, 1000) \cup Tr(3, 7.8, 8, 1000)$
D16	$N(0, 1, 1000)$ (right part) $\cup N(4, 1, 1000)$ (left part)
D17	$Tr(-3.3, 1, 2.5, 1000) \cup N(4, 1, 1000)$
D18	$U(-2, 0, 200) \cup U(1, 5, 300) \cup U(6, 7, 450)$
D19	$N(0, 1, 500) \cup N(6, 1, 80) \cup N(12, 1, 500) \cup N(18, 1, 100)$
D20	$N(0, 1, 500) \cup N(4, 1, 300) \cup N(11, 1, 500) \cup U(14, 15, 50)$
D21	$N(0, 1, 500) \cup N(4, 1, 300) \cup U(10, 11, 100) \cup U(14, 15, 50)$
D22	$N(0, 1, 500) \cup U(2.5, 4, 200) \cup U(10, 11, 100) \cup U(14, 15, 50)$

Table 4.4: Partition evaluation of multimodal datasets. The average and standard deviation for NMI, the ground truth number of modes ($k^\star$) and the average number of detected modes (k) are provided.

	Mean NMI						Average Number of Detected Modes (k)						
Distributions	UniSplit	TailoredDip	FTC	Mean shift	Modclust	MEM	$k^\star$	UniSplit	TailoredDip	FTC	Mean shift	Modclust	MEM
D13	0.78±0.02	0.71±0.07	0.77±0.03	0.71±0.07	0.75±0.03	**0.79±0.02**	2	**2**	**2**	2.05	2.46	**2**	**2**
D14	**1.00±0.00**	0.94±0.08	**1.00±0.00**	0.87±0.12	0.41±0.04	0.86±0.16	2	**2**	**2**	**2**	2.56	5.58	2.44
D15	**0.71±0.02**	0.65±0.07	0.64±0.13	0.61±0.05	0.38±0.06	0.40±0.04	2	**2**	**2**	1.97	2.83	4.74	4.29
D16	**0.73±0.03**	0.71±0.05	0.58±0.29	0.64±0.07	0.41±0.06	0.55±0.10	2	**2**	**2**	1.8	2.7	4.2	3.16
D17	**0.84±0.02**	0.69±0.11	0.78±0.03	0.75±0.09	0.82±0.02	**0.84±0.02**	2	**2**	**2**	**2**	2.54	**2**	**2**
D18	**0.99±0.01**	0.92±0.11	0.96±0.04	0.91±0.02	0.51±0.04	0.80±0.04	3	**3**	**3**	3.32	3.06	8.24	4.43
D19	0.97±0.03	0.89±0.07	0.97±0.04	**0.99±0.01**	**0.99±0.01**	**0.99±0.01**	4	3.9	3.3	3.8	4.07	**4**	**4**
D20	0.91±0.04	0.84±0.03	0.89±0.06	0.86±0.02	**0.93±0.02**	**0.93±0.02**	4	3.7	3	3	3.01	**3.96**	3.93
D21	**0.89±0.08**	0.80±0.08	0.80±0.11	0.73±0.14	0.77±0.12	0.78±0.06	4	**3.92**	3.76	3.5	3.5	4.56	5.48
D22	**0.93±0.02**	0.89±0.06	0.92±0.07	0.70±0.00	0.68±0.10	0.75±0.07	4	**4.02**	3.92	3.88	3	5.92	5.66

clust and MEM) focus on identifying modes and their corresponding clusters within a distribution, making them well-suited for modal clustering tasks.

We generated synthetic datasets by sampling from various univariate multimodal distributions (D13 - D22 in Table 4.3). For each distribution, 100 datasets were created, and the six methods were applied to cluster the generated data. Ground truth clustering information was available for each dataset, thus the methods were evaluated in terms of splitting (clustering) using the Normalized Mutual Information (NMI) score. NMI ranges from 0 to 1, with values closer to 1 indicating better clustering performance.

The parameters of each method are initialized as follows. For UniSplit and TailoredDip, the significance level is set to 0.01. TailoredDip also requires a *factor* parameter, which defines the maximum difference in sample size during the merge

test of two clusters, while FTC requires a segmentation parameter e, with large and small values resulting in coarse and finer segmentation, respectively. We have tuned both parameters taking into account the NMI value, and finally selected: $factor = 0.5$ and $e = 0$. In mean shift, the bandwidth was calculated based on distances between points, scaling it according to a quantile (0.3) of nearest neighbor distances. Finally, for GMMs employed in modclust and MEM, we use the k value (ranging from 1 to 10) that yields the lowest BIC score.

Table 4.4 provides the average and standard deviation of NMI values, along with the ground truth ($k^\star$) and estimated number of modes (k) for each method across 100 datasets generated by each distribution. UniSplit outperforms most methods in both splitting performance (NMI) and estimating k. In Gaussian mixture distributions, such as D13 and D19, UniSplit's NMI values are slightly lower than the best-performing methods but remain close, with the estimated k being closely to the ground truth. Interesting examples include D20, D21, and D22, where the number of uniform components increases. In these cases, UniSplit improves, while other methods, such as modclust and MEM, deteriorate. Overall, UniSplit shows robust performance, accurately estimating the number of modes across different multimodal distributions.

4.4.3 Image Segmentation

A widely studied statistical modeling task concerns image segmentation where the objective is to identify and differentiate various objects or regions within an image based on pixel intensities. We have applied UniSplit, TailoredDip, FTC, mean shift, modclust and MEM to solve this task and tested their performance in estimating the number of segments and their ability to accurately segment the image. To apply the methods for rgb (colored) images, each rgb image is first converted to grayscale, thus a univariate dataset is obtained containing the gray values of the pixels. Then we applied each compared method to the resulting dataset and obtained a segmentation of the image, i.e., a partition of the pixels into subsets.

We tested the performance of the six methods on rgb images, where the ground truth number of colors can be easily determined through visual inspection. Once the ground truth value of colors ($k^\star$) has been specified, we used the k-means algorithm to obtain the ground truth partition for each image, which is subsequently used to evaluate the quality of the obtained solutions using the NMI score. The parameters

Table 4.5: Image segmentation results: i) Estimated number of colors (k), ii) NMI values with respect to a ground truth solution obtained by applying k-means with the ground truth number of colors ($k^\star$).

Images	$k^\star$ / NMI	UniSplit	TailoredDip	FTC	Mean shift	Modclust	MEM
France flag	$k^\star = 3$	$k = 5$	$k = 6$	$k = 6$	$k = 4$	$k = 8$	$k = 3$
	NMI	**0.969**	0.967	0.967	0.960	0.736	0.740
Europe flag	$k^\star = 2$	$k = 2$	$k = 2$	$k = 2$	$k = 11$	$k = 2$	$k = 9$
	NMI	0.936	**0.965**	0.656	0.730	0.853	0.119
Face	$k^\star = 3$	$k = 3$	$k = 3$	$k = 3$	$k = 5$	$k = 5$	$k = 4$
	NMI	**0.998**	0.963	0.936	0.950	0.830	0.877
Flower	$k^\star = 6$	$k = 6$	$k = 6$	$k = 6$	$k = 3$	$k = 7$	$k = 6$
	NMI	**0.998**	0.996	0.992	0.770	0.851	0.995

of each method are set as they were in the previous experiments.

In Table 4.5 we present the NMI values and the obtained number of colors for each image as provided by the six methods. In the second column, we provide the ground truth value of colors ($k^\star$). In general, the differences in the highest NMI values for each image are small, indicating that some methods provide similar segmentations. An interesting case is the flag of Europe, where mean shift and MEM fail to provide correct segmentation, detecting 11 and 9 colors, respectively, instead of the correct 2, while FTC, despite its correct estimation, it does not achieve the optimal NMI value. As shown in Table 4.5, it is clear that UniSplit achieves very high NMI values (> 0.93) for all images and provides accurate or very close estimates of k compared to $k^\star$.

In the top four rows of Fig. 4.7 we present the original images in grayscale (leftmost image in each row) along with the segmented images obtained by each method. Above the original and segmented images, the ground truth value of colors ($k^\star$) and the obtained value of colors (k) by the compared methods are recorded, respectively. For most images, the methods provide similar visual results, accurately detecting the main colors of each image. It can be observed that when additional segments are detected compared to ground truth, these segments correspond to very small regions of the image that are difficult to be visually detected. For instance, UniSplit and TailoredDip detect thin line segments between the three major segments of the France flag. Similarly, in the European flag, mean shift assigns multiple colors to the stars, while MEM produces a noisy segmentation, as indicated in Table 4.5.

We also evaluated the six compared methods on grayscale images utilizing the

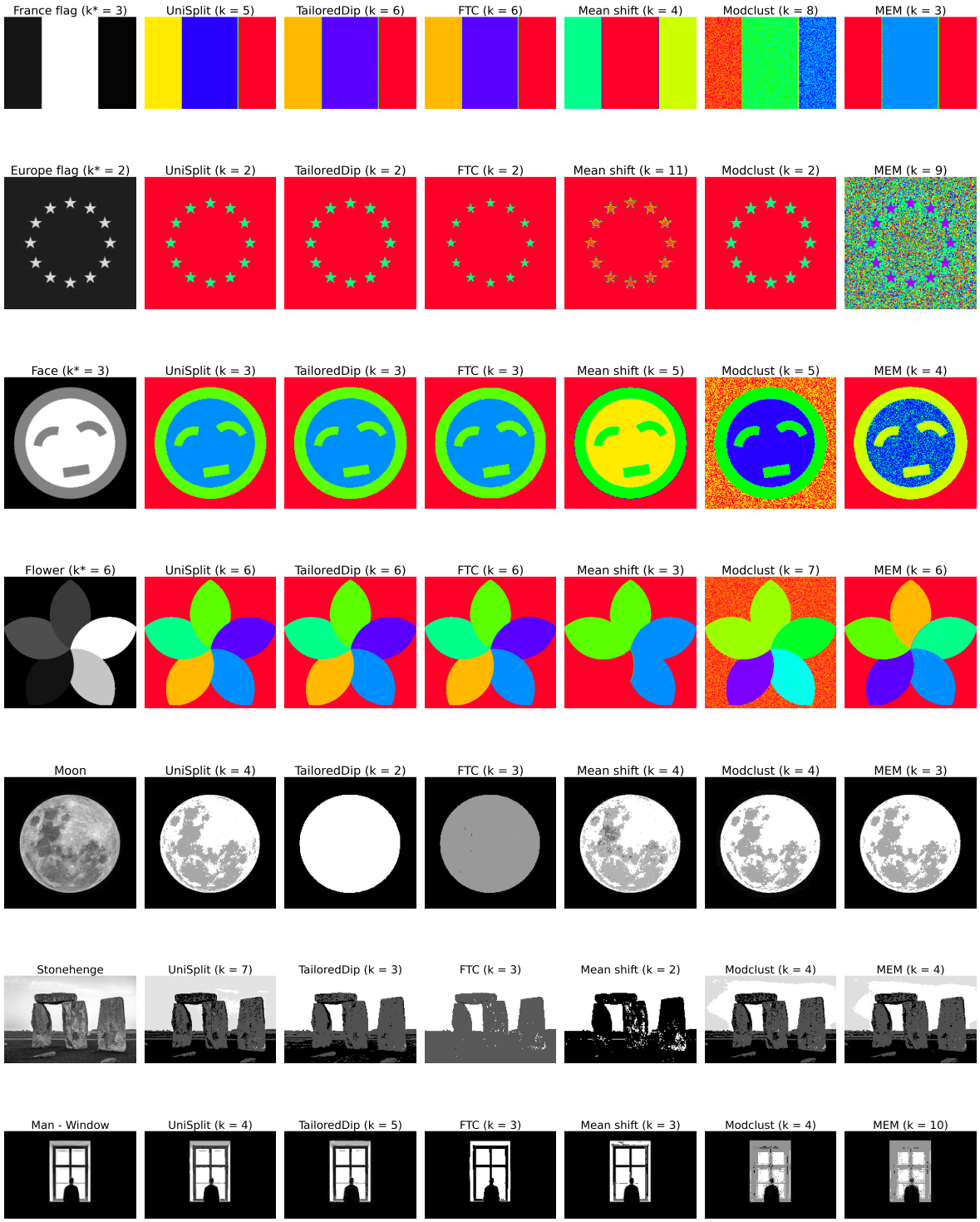


Figure 4.7: Initial images (first column). Segmented images obtained by the compared methods (second - seventh column). For rgb images the ground truth value of colors ($k^\star$) is illustrated, while the estimated number of colors (k) is provided for both rgb and grayscale images.

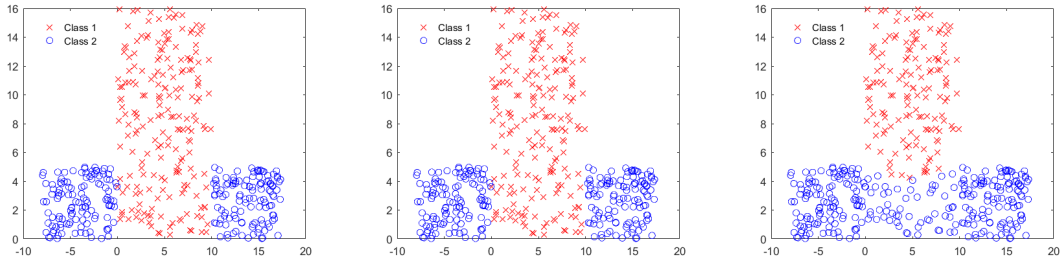
Table 4.6: Accuracy results on synthetic and real datasets. Bold numbers indicate the best average performance for each dataset.

	Parameters			Accuracy	
Datasets	n	d	K	UDMM - NB	GNB
Synthetic	400	2	2	**0.998 ± 0.01**	0.883 ± 0.03
Banknote	1371	4	2	**0.916 ± 0.02**	0.837 ± 0.04
Cardiotocography	2126	21	10	**0.702 ± 0.02**	0.637 ± 0.03
Dermatology	358	34	6	0.891 ± 0.05	**0.893 ± 0.08**
Glass	214	9	6	**0.560 ± 0.06**	0.458 ± 0.10
Image-Segmentation	210	19	7	**0.785 ± 0.09**	0.766 ± 0.11
Iris	150	4	3	0.946 ± 0.05	**0.960 ± 0.04**
Page Blocks	5473	10	5	**0.940 ± 0.01**	0.888 ± 0.02
Prestige	98	5	3	0.918 ± 0.06	**0.948 ± 0.06**
Steel Plates Faults	1941	27	7	**0.663 ± 0.02**	0.462 ± 0.02
Wall Following Robot Navigation	5456	4	4	**0.972 ± 0.01**	0.891 ± 0.01
Wall Following Robot Navigation	5456	24	4	**0.898 ± 0.03**	0.524 ± 0.01

same parameter values as those used in the previous experiment. To illustrate the segmentation result for each image, we assign to each pixel the average color value of its group, since the number of colors in these images cannot be reliably assessed through visual inspection. Therefore a ground truth partition cannot be specified, thus NMI values cannot be computed. In the bottom three rows of Fig. 4.7, the original grayscale images (first column) are presented alongside the segmentation results and the number of segments (k) obtained by each method. UniSplit produces results closely resembling the original images in all cases, while the other methods often fall short. Notably, TailoredDip and FTC fail to segment the Moon image accurately, estimating fewer segments, while modclust and MEM generate noisy segmentations in the Man-Window image.

4.4.4 UDMM Naive Bayes for Classification

A machine learning algorithm that requires the statistical model of univariate data is the well-known Naive Bayes classifier [7]. This method assumes independence among all d features of each example, therefore the per class density of each feature $p(z_i|C_k)$



(a) Ground truth solution. (b) UDMM-NB solution. (c) GNB solution.

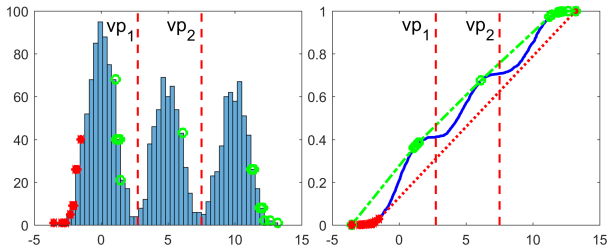
Figure 4.8: Data generated by three uniform rectangles assigned to two classes.

is estimated by considering the set of values of feature z_i for the examples belonging to class C_k. Once the densities $p(z_i|C_k)$ have been determined for all features z_i and classes C_k, the posterior probability that an example $z = (z_1, ..., z_d)$ belongs to class C_k is proportional to $P(C_k|z) \propto P(C_k) \prod_{i=1}^{d} p(z_i|C_k)$ where $P(C_k)$ are typically set equal to class frequencies and the example z is assigned to the class with maximum posterior probability.

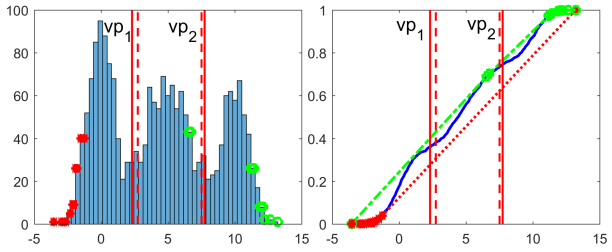
A widely approach is Gaussian Naive Bayes (GNB), which assumes that $p(z_i|C_k)$ follows a single Gaussian distribution. In this experiment we model each feature density $p(z_i|C_k)$ using a UDMM and we call the resulting method as UDMM-NB. We have considered one synthetic and several real datasets [5]. For each dataset, we used 10-fold cross validation to measure the accuracy. Table 4.6 provides the names and parameters (n: number of samples, d: number of features, K: number of classes) of each dataset in the first four columns, with the average and standard deviation of accuracy values for UDMM-NB and GNB in the fifth and sixth columns. UDMM-NB generally outperforms GNB, except for small datasets like Iris and Prestige, where sample sizes per class are low. A notable example is the synthetic 2-d dataset (Fig. 4.8), where UDMM-NB correctly discriminates the two classes, while GNB lacks the flexibility required for correctly modeling the data points (as also indicated in the first row of Table 4.6).

4.4.5 Examples with Noise and Outliers

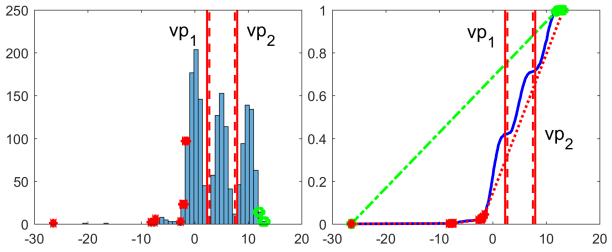
Noise and outliers can affect the ecdf shape, as well as the gcm/lcm points positions, but as shown in [74], the unimodality decisions by UU-test remain unaffected. Previous experiments with synthetic datasets containing outliers, such as distributions



(a) Original trimodal dataset.



(b) Trimodal dataset with uniform noise added to the valleys.



(c) Trimodal dataset with left-side Student's t-distributed noise (outliers).

Figure 4.9: Histogram and ecdf plots of a trimodal dataset before and after adding noise/outliers. The original valley points (dotted vertical lines) are almost identical to the final valley points (solid vertical lines).

D6, D8, and D12 (Table 4.2), demonstrated that UDMM outperformed other models in component detection and modeling accuracy, even with extreme values (e.g., in D6 the range is $[-6800, 330]$). We next provide an example that highlights UniSplit's robustness to noise and outliers. For a trimodal dataset (Fig. 4.9a), UniSplit identifies two valley points (vp_1, vp_2) (dotted vertical lines). When uniform noise is added between Gaussians (Fig. 4.9b), the ecdf changes, but the valley points (solid vertical lines) remain close to their original locations. Similarly, the addition of outliers (on

the left) generated from a Student's t distribution (Fig. 4.9c) shifts gcm/lcm points and modifies the ecdf significantly, however UniSplit detects correctly the number and positions of valley points (solid lines coincide with the original dotted lines), indicating robustness against noise and outliers.

4.4.6 Impact of the Statistical Significance Level

The UniSplit method automatically estimates the number of valleys in univariate multimodal data, leading to the automatic determination of the number of components in the UDMM, unlike other models requiring user-defined hyperparameters. UniSplit requires solely the significance level (α) of the uniformity test employed in UU-test, which was set to $\alpha = 0.01$ in all previous experiments.

To examine the influence of α, experiments were repeated with $\alpha = 0.05$ and $\alpha = 0.1$ using datasets from Table 4.1. Results showed minimal influence on UDMM's performance or component count. In 9 of 21 datasets, the number of components remained unchanged across all values of α. In 10 datasets, small increases (0.6%–12.3%) were observed as α increases. For example, in synthetic dataset D4, the average number of components increased from $k = 3$ (when $\alpha = 0.01$) to $k = 3.02$ (when $\alpha = 0.1$), and in D10, from $k = 4.05$ to $k = 4.55$. In the real datasets stamps and geyser, k increased more noticeably (from 1 to 3); however they could be considered as borderline cases of unimodality, as evident from histogram inspection.

4.5 Summary

In this chapter, we have proposed an approach for partitioning and statistical modeling of univariate datasets. The method relies on the notion of unimodality and partitions the dataset into unimodal subsets through a novel approach for determining valley points in the probability density. We have introduced properties of critical points (gcm/lcm points) of the data ecdf that provide indications on the existence of density valleys and further are exploited in the proposed UniSplit algorithm. UniSplit is non-parametric and automatically estimates the number of unimodal subsets. In contrast to other approaches, it requires only a statistical significance threshold as input and no other user specified hyperparameters. After splitting the datasets into unimodal subsets, our approach constructs a Unimodal Mixture Model (UDMM),

where each mixture component constitutes a statistical model of the corresponding unimodal subset in the form of a Uniform Mixture Model (UMM). The number of UDMM components is automatically obtained by the proposed UniSplit method, which constitutes a significant advantage over other models (e.g., GMM). In addition UDMM is very flexible and does not assume any specific parametric form for the unimodal mixture components. Experimental results on various modeling and clustering tasks indicate that UniSplit and UDMM are generally superior to competing methods without requiring any hyperparameter tuning.

Chapter 5

Unsupervised Decision Trees for Axis Unimodal Clustering

5.1 Introduction

5.2 Notations and Definitions

5.3 Axis Unimodal Clustering with a Decision Tree Model

5.4 Experimental Results

5.5 Summary

5.1 Introduction

As mentioned in previous chapters, the concept of unimodality has been employed in unimodality tests, statistical modeling and valley detection (mode estimation). In Chapter 1, Section 1.2.3, the application of unimodality tests in clustering methods is also discussed. Since most clustering methods handle multidimensional data, and most unimodality tests are applied to univariate data only, some techniques have been proposed that apply the unimodality tests on univariate datasets containing for example 1-d projections of the data or distances between data points. Unimodality assessment has been used either in a top-down fashion, by splitting clusters that are decided as multimodal [3, 29], or in a bottom-up fashion by merging clusters whose union is decided as unimodal [101]. In addition to intuitive justification, the use of unimodality provides a natural way to terminate the splitting or merging

procedure, thus allowing for the automated estimation of the number of clusters. Since those methods provide ellipsoidal or arbitrarily shaped clusters the results are not interpretable. We aim to tackle this issue by proposing a unimodality-based method for the construction of decision trees for clustering.

A detailed analysis of decision trees is provided in Chapter 1, Section 1.4. While decision trees are widely used in supervised learning tasks (e.g., classification), their construction becomes more challenging in unsupervised learning tasks (e.g., clustering), where data labels are absent, and only data points are available. Unlike typical clustering methods (e.g., k-means), which do not inherently provide explanations for the resulting clusters, decision trees offer an interpretable decision-making process.

Our approach is based on the notion of an *axis unimodal* cluster: a cluster where all features are unimodal, i.e., the set of values of each feature is unimodal as decided by a unimodality test [77]. The proposed method, called Decision Trees for Axis Unimodal Clustering (DTAUC), follows the typical top-down splitting paradigm for building axis-aligned decision trees (the data space is partitioned into hyperrectangular regions) and aims to partition the initial dataset into axis unimodal clusters. The decision rule at each node involves an appropriately selected feature and the corresponding threshold value. More specifically, given the dataset at each node, the multimodal features are first detected. For each multimodal feature, we follow a greedy strategy to detect the best threshold value (denoted as *split threshold*) that splits the set of feature values into subsets so that the unimodality of the partition is increased. We propose two criteria, criterion 1 and criterion 2, that rely on unimodality tests to assess the unimodality of the partition.

More specifically, criterion 1 is based on the p-values provided by Hartigans' dip-test [27] for unimodality. To improve performance, we combine this criterion with another one that measures the separation of data points before and after the split point, thus obtaining the final criterion used to assess the quality of splitting. Criterion 2 is based on the multimodality degree[1] of a multimodal feature. A high degree of multimodality indicates that this feature provides an appropriate split threshold. For assessing unimodality criterion 2 uses the UU-test [74] for unimodality.

Based on these criteria, the best-split threshold for a multimodal feature is determined. The procedure is repeated for every multimodal feature and the feature-threshold pair of highest quality (criterion 1) or highest multimodality degree (cri-

[1]For details on multimodality degree see Chapter 4, Section 4.2.1

terion 2) is used to define the decision rule of the node. When the data subset in a node does not contain any multimodal features, i.e., it is axis unimodal, no further splitting occurs, the node is characterized as leaf and a cluster label is assigned to this node. In this way at the end of the method, a partitioning of the original dataset into axis unimodal clusters has been achieved that is interpretable since it is represented by an axis-aligned decision tree.

The proposed DTAUC algorithm is direct (e.g., does not employ k-means as a preprocessing step), end-to-end and relies on the intuitively justified notion of unimodality. It is simple to implement and does not employ computationally expensive optimization methods. It contains no hyperparameters except for the statistical significance level of the unimodality test. The latter remark is important since most unsupervised decision tree methods include hyperparameters such as number of clusters, maximum tree depth, etc., which are difficult to tune in an unsupervised setting.

The rest of this chapter is organized as follows. In Section 5.2 we provide the necessary definitions and notations along with the unimodality tests (Hartigans' dip-test [27] and UU-test [74]) demonstrating illustrative figures. In Section 5.3 we describe the proposed method (DTAUC) for constructing unsupervised decision trees for clustering that mainly relies on the computation of appropriate split thresholds for partitioning multimodal features. The two proposed criteria for determining those split thresholds are also presented. Comparative experimental results are provided in Section 5.4, while Section 5.5 summarizes this chapter.

5.2 Notations and Definitions

In this section, we provide some definitions needed to present and clarify our method. At first, we briefly describe Hartigans' dip-test and UU-test, as they are employed in criterion 1 and 2, respectively. Illustrative figures are provided to clarify the concepts of the two criteria. Next, we present the definition for an *axis unimodal dataset* and finally, we provide notations related to the binary decision tree which is built by our method.

5.2.1 Dip-Test for Unimodality

Hartigans' *dip-test* [27] constitutes the most popular unimodality test and in this chapter, it is used for deciding unimodality of univariate datasets in criterion 1. In dip-test the null hypothesis H_a is that F is unimodal, while the alternative hypothesis H_1 suggests multimodality. H_0 is accepted at significance level a if p-value $> a$, otherwise it is rejected. A detailed description of dip-test is provided in Chapter 1, Section 1.2.2.

Fig. 5.1 illustrates examples of unimodal and multimodal datasets in terms of pdf plots (histograms) and ecdf plots. The p-values provided by the dip-test are also presented above each subfigure. In Fig. 5.1a a unimodal dataset generated by a Gaussian distribution is illustrated along with the corresponding p-value $= 0.98$. p-values close to 1 indicate unimodality and this is also evident in that case. In contrast, Fig. 5.1b presents a dataset generated by two close Gaussian distributions which does not clearly constitute a unimodal or multimodal dataset. This uncertainty is indicated by the computed p-value of 0.08, which is closer to 0. In this case, the significance level defined by the user plays a crucial role in determining unimodality. For example, if the significance level is set to $\alpha = 0.1$, then a p-value less than α leads to the dataset being determined multimodal. However, with significance levels of $\alpha = 0.01$ or $\alpha = 0.05$, the dataset would be considered unimodal. In Fig. 5.1c,d two strongly multimodal datasets are shown, with two and three peaks, respectively. The p-value is 0 in both datasets, and the test decides multimodality regardless of the significance level chosen by the user.

5.2.2 UU-Test for Unimodality

The UU-test [74] is a method for determining the unimodality of a dataset by constructing a piecewise linear (PL) approximation of its ecdf. In this chapter it is used in criterion 2 in order to assess unimodality of a feature. It uses critical points called gcm and lcm points to form an ordered set and attempts to build a unimodal PL function, where gcm points precede lcm points. For the approximation to be accurate, the data within each interval between successive gcm and lcm points must follow the uniform distribution, which is ensured by a uniformity test. The UU-test is described in detail in Chapter 2, Section 2.3.

In case UU-test fails constructing a good approximation of the ecdf, it indicates multimodality, which occurs when there exist non-uniform intervals between gcm and

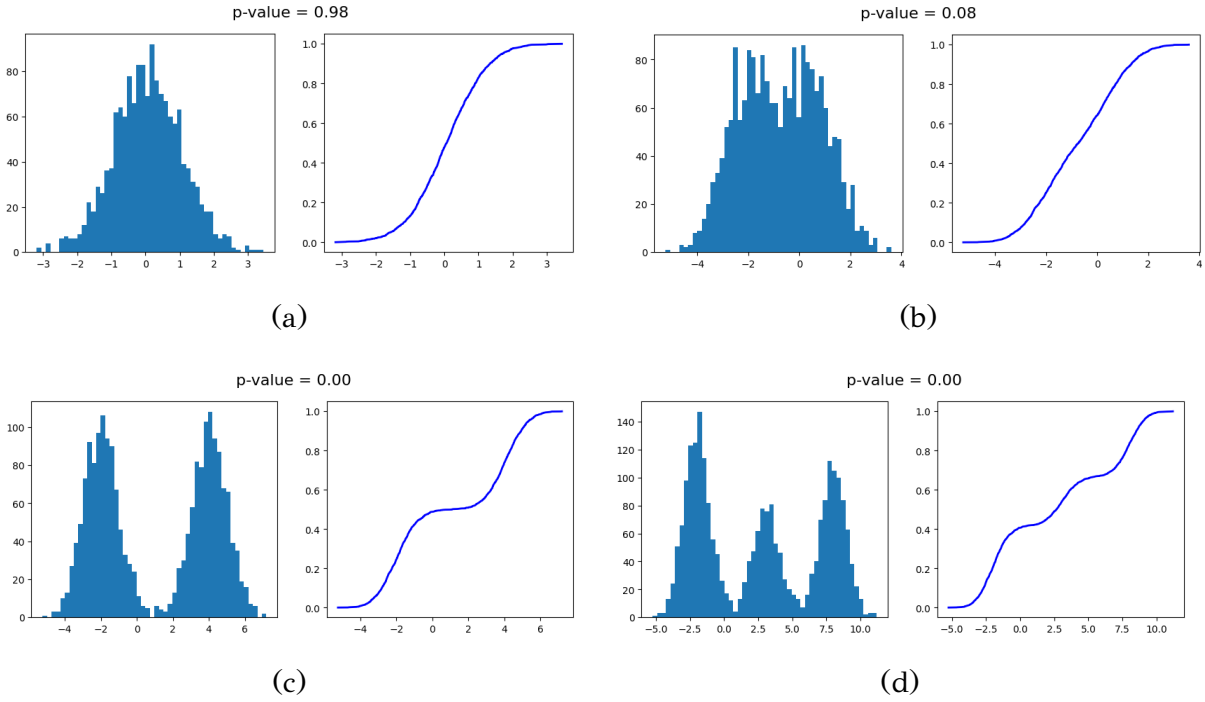


Figure 5.1: Histogram and ecdf plots of unimodal and multimodal univariate datasets. The p-values provided by the dip-test are also presented. (a) Unimodal dataset. (b) Borderline case of unimodal dataset (with two close peaks). (c) Multimodal dataset (with two peaks). (d) Multimodal dataset (with three peaks).

lcm points, indicating the presence of density valleys. In such cases, those non-uniform intervals are utilized to detect appropriate valley points. In Chapter 4, Section 4.2 an analysis is conducted on the existence and number of density valleys, by examining intervals $[a, b]$ between successive gcm or lcm points. Specifically, three main cases are discussed:

(a) If the data in [a,b] are uniformly distributed, the ecdf segment is linear, indicating no density valley, as both gcm or lcm points lie within the same mode's increasing or decreasing segment.

(b) A non-uniform but unimodal distribution in [a,b] results in a nonlinear ecdf segment, suggesting a single density valley between the successive modes represented by gcm or lcm points.

(c) When [a,b] is non-uniform and multimodal, the ecdf segment shows both convex and concave sections, pointing to multiple density valleys corresponding to the multiple modes within the interval.

Multimodality Degree

To quantify the degree of multimodality within a data interval, the distance among the peaks and the depth of the valley between the peaks on the data histogram is considered. Larger distances between peaks and deeper valleys indicate a higher multimodality degree, distinguishing intervals from uniformity. For an interval $[a, b]$, the multimodality degree is calculated as the maximum deviation d of the ecdf from the cdf of the uniform distribution, $F_U(x)$. Specifically, $d = \max_{x \in X(a,b)} (|F(x) - F_U(x)|)$, where d is marked by the maximum deviation (MD[2]) point, typically located near valley points.

In Fig. 4.4 (see Chapter 4, Section 4.2.1) larger distance between peaks results in a higher d value (top row), while for intervals with equal peak spacing, valley depth further influences d: deeper valleys increase d, indicating a stronger deviation from uniformity of the corresponding ecdf segment (bottom row). The location of the MD point aligns closely with valley points, helping to define an accurate valley point[3].

5.2.3 Axis Unimodal Dataset

Let $X \subseteq \mathbb{R}^d$ be a dataset consisting of data vectors in a d-dimensional space. Each point $x \in X$ can be represented as a vector $x = (x_1, x_2, \ldots, x_d)$, where $x_j \in \mathbb{R}$ denotes the j-th feature of x for $j = 1, 2, \ldots, d$. We also denote as X_j the j-th feature vector of the dataset X, which consists of the j-th feature values of all points in X. Given a dataset X, a feature j is characterized as unimodal or multimodal based on the unimodality or multimodality of X_j.

Definition 5.1. A d-dimensional dataset X is *axis unimodal* if every feature j is unimodal, i.e., each univariate subset X_j ($j = 1, \ldots, d$) (consisting of the j-th feature values) is unimodal [77].

Obviously, in order for a dataset X to be decided as axis unimodal, a unimodality test (e.g., dip-test, UU-test) should decide unimodality for each subset X_j. A dataset that is not axis unimodal will be called *axis multimodal*.

[2]More details on MD point are given in Chapter 4, Section 4.2.1.

[3]More details on computing valley points are given in Chapter 4, Section 4.3.1 and Fig. 4.2.

5.2.4 Node Splitting

Let u be a node during decision tree construction and X the corresponding set of data vectors to be split by applying a thresholding rule on a feature value. A split rule for u is defined as the pair $(j, sp) \in \{1, 2, \ldots, d\} \times \mathbb{R}$, where j is the feature on which the rule is applied and sp is the corresponding threshold level. A splitting rule of the form $\{x \in X : x_j \leq sp\}$ is then applied to the node. We denote the subset of X that satisfies this condition as X_L, while the set of points that do not satisfy this condition is denoted as X_R. Two child nodes u_L and u_R are then created corresponding to the subsets X_L and X_R, respectively. In this chapter, we aim to determine the feature-threshold pair that results in a decrease in the *multimodality of the partition* by computing two appropriately defined criteria. We denote the best-split pair as $(j^\star, sp^\star)$, where $j^\star$ and $sp^\star$ denote the *best feature* and the *best split threshold*, respectively. If for the dataset X, no features are detected for splitting, then node u is considered a leaf. This occurs when the dataset X is axis unimodal.

5.3 Axis Unimodal Clustering with a Decision Tree Model

The proposed method can be considered as a divisive (i.e., incremental) clustering approach that is based on binary cluster splitting and produces rectangular axis unimodal clusters. It starts with the whole dataset as a single cluster and, at each iteration, it selects an axis multimodal cluster and splits this cluster into two subclusters. The method terminates when all produced clusters are axis unimodal. Binary cluster splitting is implemented by applying a decision threshold on the values of a multimodal feature. In this way the cluster assignment procedure can be represented with a typical (axis-aligned) decision tree, ensuring the interpretability of the clustering decision. Obviously, the leaves of the decision tree correspond to axis unimodal clusters.

Since our objective is to produce axis unimodal clusters, we consider multimodal features for cluster splitting. Let X be the multimodal cluster to be split and X_j be the set of values of a multimodal feature j. Since X_j is multimodal, our objective is to determine a splitting threshold such that the splitting of X_j will result in two subsets X_{jL} and X_{jR} that are less multimodal than X_j (ideally they should be both unimodal). We define two criteria to evaluate the partition (X_{jL}, X_{jR}) in terms of

unimodality and separation (criterion 1) and multimodality degree (criterion 2).

In criterion 1, we follow the typical case for decision tree construction, i.e., we evaluate several partitions obtained by considering all multimodal features and several candidate threshold values for each feature. The best partition is determined according to the criterion 1 and the corresponding feature-threshold pair, which defines the decision rule for splitting cluster X. Criterion 2 identifies the best partition, i.e., the best multimodal feature and the best split threshold, using a more direct approach. By detecting non-uniform intervals between gcm and lcm points of the data ecdf and assessing their multimodality degree, it achieves identifying the best partition, without the need to explicitly test all possible threshold values.

The two criteria used to evaluate a partition are presented next.

5.3.1 Criterion 1

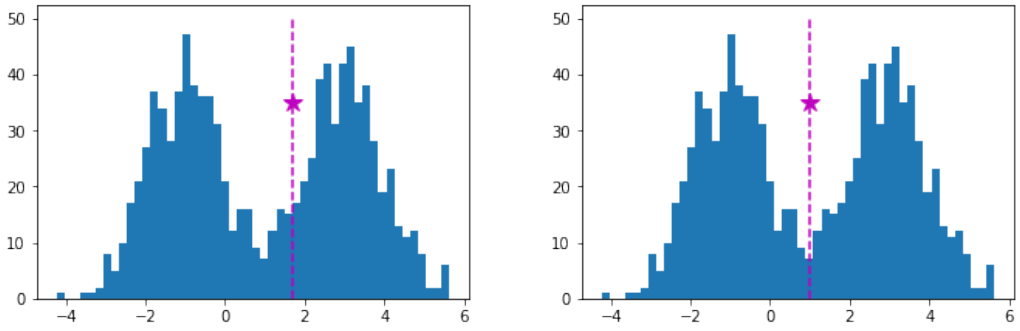
Let $S = \{s_1, s_2, \ldots, s_N\}$ denote the set of values corresponding to a multimodal feature. Initially, we sort the values s_i, $i = 1, 2, \ldots, N$ in ascending order. For each $i = 1, 2, \ldots, N-1$ we consider the average between s_i and its successor s_{i+1} as candidate split threshold sp. Given a threshold value sp, S is partitioned into two subsets: a left subset S_L (values on the left of sp) and a right subset S_R (values on the right of sp). Let also N_L and N_R be the sizes of S_L and S_R, respectively. The dip-test is then applied to both subsets to assess their unimodality, yielding two p-values: p_L for S_L and p_R for S_R. To evaluate the effectiveness of threshold sp, we define a *weighted p-value* of the partition: $p_{split} = \frac{N_L}{N} p_L + \frac{N_R}{N} p_R$.

An intuitive justification of the p_{split} formula is the following:

1. If both subsets S_L and S_R are multimodal, then the p_L-value and p_R-value are low resulting in a low p_{split} value.

2. If both subsets S_L and S_R are unimodal, then the p_L-value and p_R-value are high resulting in a high p_{split} value.

3. In case one subset (let S_L) is unimodal and the other (let S_R) is multimodal, we need to consider the size of each subset: if $N_L > N_R$ and since $p_L > p_R$ (unimodal S_L, multimodal S_R) then the resulting p_{split} value is high. In the opposite case, the set S_R becomes the dominant set and thus p_{split} demonstrates a lower value.



(a) (b)

Figure 5.2: Histogram of a bimodal dataset along with its split threshold (star) computed using criterion 1. (a) The split threshold was computed without utilizing the separation criterion. (b) The split threshold was computed taking into account the separation criterion.

Case 2 describes a scenario where the data splits into two unimodal sets resulting in high p_{split} values. In case 3 a dominant unimodal subset is compared against a relatively small multimodal subset, also yielding high p_{split} values. These findings indicate that high p_{split} values occur when the split highlights one or two unimodal subsets. By selecting the candidate threshold value providing the highest p_{split} value we obtain a partition of S into subsets of increased average unimodality.

Following the above procedure, we have experimentally noticed that although sensible splittings were generally obtained, the selected threshold sp was not always very accurate. For example, Fig. 5.2a illustrates the histogram of a bimodal dataset along with the computed sp marked with a star. Splitting can be considered successful since the dataset is split into two unimodal subsets; however, the sp presented in Fig. 5.2b is a more accurate split point than the one in Fig. 5.2a. Thus, there is some room for improvement in threshold determination. To tackle this issue we consider not only the unimodality partition, but also the *separation* of points before and after the threshold sp. More specifically, we consider a subset of w successive points right before sp and a second subset of w successive points right after sp. We define the *separation* (sep) of a threshold value sp as the average distance among all pairs of points belonging to different subsets. A large distance value indicates a high separation between the points before and after sp, thus sp lies in a density valley. Therefore, the split corresponding to sp is efficient, if the separation is high. We denote as $sep(sp)$ the separation of the points defined by a split point sp. We choose

Algorithm 5.1 $(sp^\star, q^\star) =$ best_split_point_c1(S, α)

p-value $\leftarrow$ dip-test(S, α)
if p-value $> a$ **return** $\emptyset$ // S: unimodal
$S \leftarrow sort(S)$
for all i with $w < i \leq N - w$ **do**
  $sp_i \leftarrow \frac{s_i + s_{i+1}}{2}$
  $S_{L_i} \leftarrow S(s_1, sp_i)$
  $S_{R_i} \leftarrow S(sp_i, s_N)$
  $p_{L_i} \leftarrow$ dip-test(S_{L_i}, α)
  $p_{R_i} \leftarrow$ dip-test(S_{R_i}, α)
  $p_{split_i} \leftarrow \frac{N_{L_i}}{N} p_{L_i} + \frac{N_{R_i}}{N} p_{R_i}$
  $sep_i \leftarrow sep(sp_i)$
  $q_i \leftarrow p_{split_i} \times sep_i$
end for
$i^\star \leftarrow \arg\max_i(q_i)$
$sp^\star \leftarrow sp_{i^\star}$
$q^\star \leftarrow q_{i^\star}$
return $(sp^\star, q^\star)$

a small value of w (e.g., $w = 0.01 \times N$), with the choice of value w not affecting the final result. Since sep computation requires at least w points before and after sp, we do not consider candidate threshold values defined by the first w and last w points of S.

Therefore, since our objective is to determine a threshold with both a high p_{split} value and high sep value, we define a new criterion $q = p_{split} \times sep$ to measure the quality of a split (criterion 1). A high q value provides a split into two highly separated (high sep value) and unimodal (high p_{split} value) subsets. Thus, we choose the candidate threshold sp resulting in a maximum q value as the *best split threshold* and denote it as $sp^\star$. Algorithm 5.1 presents the steps of computing the best-split point $sp^\star$ of a univariate dataset S using criterion 1. It first takes as input the univariate dataset S and the significance level α and returns the best-split point $sp^\star$ of S along with the corresponding $q^\star$ value. In case S is unimodal as decided by the dip-test, then Algorithm 5.1 returns the empty set.

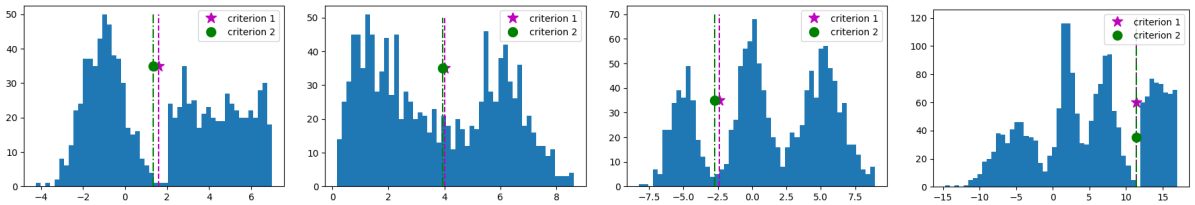


Figure 5.3: Histogram plots of synthetic datasets along with the best split thresholds found using criterion 1 (stars) and criterion 2 (circles).

5.3.2 Criterion 2

Non-uniform intervals defined by successive gcm or lcm points of the ecdf are called "candidate splitting intervals" indicating that they contain at least one valley (as described in Chapter 4, Section 4.3.1). Among them, the one with the highest degree of non-uniformity (highest multimodality degree) is selected as the "best splitting interval". Based on case (b) in Section 5.2.2, in case this interval is unimodal, a single valley exists. For the computation of the valley point, the maximum deviation (MD) point and the interval's boundary points are utilized. In case the boundary points are gcm points, the vp is computed as the middle point of MD and the upper boundary, while in case the boundaries are lcm points, the vp is computed as the middle of the lower boundary and the MD. Fig. 4.2 (see Chapter 4, Section 4.3.1) provides illustrative plots. In case the best splitting interval is multimodal, multiple valleys may exist, requiring recursive refinement. Then we focus on the most non-uniform subinterval and repeat the process until a unimodal interval is found. This ensures that the final valley point corresponds to a single valley within the best splitting interval. Fig. 4.3 (see Chapter 4, Section 4.3.1) provides illustrative plots.

Since the valley point is detected in the best splitting interval (which corresponds to the interval with the highest multimodality degree) the determined valley point in the interval demonstrates the best split threshold of the feature (given a multimodal feature). Algorithm 5.2 presents the steps of computing the best-split point $sp^\star$ of a univariate dataset S using criterion 2. It takes as input the univariate dataset S and the significance level α and returns the best-split point $sp^\star$ of S along with the corresponding $d^\star$ value. In case S is unimodal as decided by the UU-test, then Algorithm 5.2 returns the empty set. Fig. 5.3 illustrates the histogram plots and the best split points of several datasets which are generated by sampling from mixtures of Gaussian, uniform and triangular distributions. The best split points computed

Algorithm 5.2 $(sp^\star, d^\star)$ = best_split_point_c2(S, α)

if UU-test(S, α) decides unimodality for S **return** $\emptyset$ // S: unimodal
Compute GL set of S
$I \leftarrow$ set of candidate splitting intervals of GL
$T = [a^*, b^*] \leftarrow$ best splitting interval
if $X(a^*, b^*)$ is unimodal **then**
  $d^\star \leftarrow \max_{s \in S(a^*, b^*)} (|F(s) - F_U(s)|)$
  $s_{MD} \leftarrow$ compute MD point of T
  if a^*, b^* gcm points **then**
    $sp^\star \leftarrow \frac{s_{MD} + b^*}{2}$
  else
    $sp^\star \leftarrow \frac{a^* + s_{MD}}{2}$
  end if
  return $(sp^\star, d^\star)$
else
  $(sp^\star, d^\star) \leftarrow$ best_split_point_c2$(S(a^*, b^*))$
end if

using criterion 1 are marked with stars, whereas those computed using criterion 2 are marked with circles. Notably, both criteria yield nearly identical split points.

5.3.3 Decision Tree Construction

Next, we describe our method, called Decision Trees for Axis Unimodal Clustering (DTAUC), for obtaining interpretable axis unimodal partitions of a multidimensional dataset. Our method employs a divisive (top-down) procedure, thus we first assign the whole initial dataset to the root node. Assuming that at some iteration a node u contains a dataset X, our goal is to determine the splitting rule for node u. This involves determining the best pair consisting of a multimodal feature and the corresponding split threshold. The steps for determining the best pair according to criterion 1 and 2 are provided below.

Criterion 1

To identify the best split for u using criterion 1, we work as follows: first, we apply the dip-test to detect the multimodal features of X. If all features are unimodal,

node u is considered a leaf and no split occurs. If multimodal features exist, then for each multimodal feature j, Algorithm 5.1 is used to compute its best split threshold sp_j and the corresponding evaluation q_j of the resulting partition. Among the multimodal features, we select as best the one with maximum q_j value. Algorithm 5.3 describes the steps for determining the splitting rule of a dataset X using criterion 1. It takes the set X and a significance level α as input and returns the best pair $(j^\star, sp^\star)$ where $j^\star$ is the selected multimodal feature and $sp^\star$ the corresponding threshold.

Criterion 2

To identify the best split for u using criterion 2, we work as follows: first we apply the UU-test to detect the multimodal features of X. Similarly to criterion 1, if all features are unimodal, node u is considered as leaf and no split occurs. If multimodal features exist, then for each multimodal feature j, Algorithm 5.2 is used to compute its best split threshold sp_j and the corresponding multimodality degree d_j. Among the multimodal features we select as best the one with maximum d_j value. Algorithm 5.4 describes the steps for determining the splitting rule of a dataset X using criterion 2. It takes the set X and a significance level α as input and returns the best pair $(j^\star, sp^\star)$ where $j^\star$ is the selected multimodal feature and $sp^\star$ the corresponding threshold.

Algorithm 5.3 $(j^\star, sp^\star)$ = best_split_c1(X, α)

---

for all feature X_j **do**
  p_j-value $\leftarrow$ dip-test(X_j, a)
  if p_j-value $\leq a$ **then**
    $(sp_j, q_j) \leftarrow$ best_split_point_c1(X_j, α)
  end if
end for
if p_j-value$> a, \quad \forall j$ **return** $\emptyset$ // X: axis unimodal
$j^\star \leftarrow \arg\max_j(q_j)$
$sp^\star \leftarrow sp_{j^\star}$
return $(j^\star, sp^\star)$

Algorithm 5.4 $(j^\star, sp^\star)$ = best_split_c2(X, α)

for all feature X_j **do**
  unimodality $\leftarrow$ UU-test(X_j, a)
  if unimodality = False **then**
    $(sp_j, d_j) \leftarrow$ best_split_point_c2(X_j)
  end if
end for
if unimodality = True, $\forall X_j$ **return** $\emptyset$ // X: axis unimodal
$j^\star \leftarrow \arg\max_j(d_j)$
$sp^\star \leftarrow sp_{j^\star}$
return $(j^\star, sp^\star)$

In case the best split for u exists (i.e., u is not considered as a leaf), the data vectors of X are partitioned into two subsets, X_L and X_R, based on the feature $j^\star$ values: $X_L = \{x \in X : x_{j^\star} \leq sp^\star\}$ and $X_R = \{x \in X : x_{j^\star} > sp^\star\}$. Therefore two child nodes of u, denoted as u_L and u_R, are added to the tree, corresponding to sets X_L and X_R, respectively. Finally, the method is applied recursively on each resulting node, until all nodes are identified as leaves, i.e., the subsets in all nodes are axis unimodal. We assign each leaf a cluster label meaning that each leaf represents a single cluster. Therefore, an axis unimodal partition of the initial dataset X into hyperrectangles is obtained. Algorithm 5.5 describes the proposed DTAUC method. It takes a multidimensional dataset X and a significance level α as input and returns the constructed tree using either criterion 1 or criterion 2. It should be emphasized that the algorithm does not require as input the number of clusters which is automatically determined by the method.

5.3.4 An Illustrative Example

Table 5.1 presents the intermediate steps from the application of DTAUC on the two-dimensional dataset (called X) illustrated in the first plot of Fig. 5.4a. In this example, DTAUC uses criterion 1 to determine the best split pairs with the final partition being identical using both criteria. For each subset of X (listed in first column), we provide the feature along with its unimodal (U) / multimodal (M) property (second column) as determined by the dip-test. The best split thresholds $sp^\star$ and the corresponding q

Algorithm 5.5 DTAUC(X, α)

Create a root node u corresponding to X
$(j^\star, sp^\star) \leftarrow$ best_split(X, α) // using criterion 1 or 2
if $(j^\star, sp^\star) = \emptyset$ **then**
  return the leaf u
else
  $X_L = \{x \in X : x_{j^\star} \leq sp^\star\}$
  $X_R = \{x \in X : x_{j^\star} > sp^\star\}$
  $u_L \leftarrow$DTAUC(X_L, α)
  $u_R \leftarrow$DTAUC(X_R, α)
  return the decision tree rooted at u
end if

values for each feature are given in the third and fourth columns, respectively. The fifth column indicates whether to split or save the set mentioned in the first column. If a split decision is made, the best split feature is mentioned in parentheses. Either two subsets are created (in case of a split decision) or the set specified in the first column is axis unimodal, thus it is saved in set C which contains the axis unimodal subsets.

In Fig. 5.4 we provide illustrative plots corresponding to the step-by-step partition of the 2-D set X. Fig. 5.4a displays the 2-D plot of the initial dataset X, along with the histogram plots of feature vectors X_1 and X_2. A higher q value is computed for feature X_2 ($q_2 = 1.96 > q_1 = 0.47$) as shown in Table 5.1, thus we apply the split on feature X_2 with the threshold value $sp_2 = 4.14$. The partitioning of X into two subsets X_L and X_R is given in the right plot of Fig. 5.4a. The dotted line illustrates the split threshold sp_2. The plot of X_L is presented in Fig. 5.4b. The first feature is bimodal, while the second is unimodal, as indicated by the histogram plots and the q values for X_1 and X_2 in Table 5.1. Therefore, the split is applied considering the first feature using the threshold value $sp_3 = 6.17$ (dotted line in the right plot of Fig. 5.4b). This split results in two subsets, denoted as X_{LL} and X_{LR}. Fig. 5.4c illustrates the 2-D plots of X_{LL} and X_{LR} along with the corresponding histograms for each feature. It is clear that X_{LL} and X_{LR} are axis unimodal, thus we save them in C. The 2-D plot of X_R is presented in Fig. 5.4d, where it is clear that each feature is unimodal (the histogram plots and q values for X_R in Table 5.1 indicate unimodality), thus we save it in set C.

Table 5.1: Stepwise partitioning of the two-dimensional dataset of Figure 5.4a using criterion 1.

Sets	Features	$sp^\star$	q	Split ($j^\star$) or Save	Result
X	1 (M)	$sp_1 = 9.29$	$q_1 = 0.47$	Split X $(j^\star = 2)$	Sets X_L, X_R
	2 (M)	$sp_2 = 4.14$	$q_2 = 1.96$		
X_L	1 (M)	$sp_3 = 6.17$	$q_3 = 6.72$	Split X_L $(j^\star = 1)$	Sets X_{LL}, X_{LR}
	2 (U)	$\emptyset$	$\emptyset$		
X_{LL}	1 (U)	$\emptyset$	$\emptyset$	Save X_{LL}	$C = \{X_{LL}\}$
	2 (U)	$\emptyset$	$\emptyset$		
X_{LR}	1 (U)	$\emptyset$	$\emptyset$	Save X_{LR}	$C = \{X_{LL}, X_{LR}\}$
	2 (U)	$\emptyset$	$\emptyset$		
X_R	1 (U)	$\emptyset$	$\emptyset$	Save X_R	$C = \{X_{LL}, X_{LR}, X_R\}$
	2 (U)	$\emptyset$	$\emptyset$		

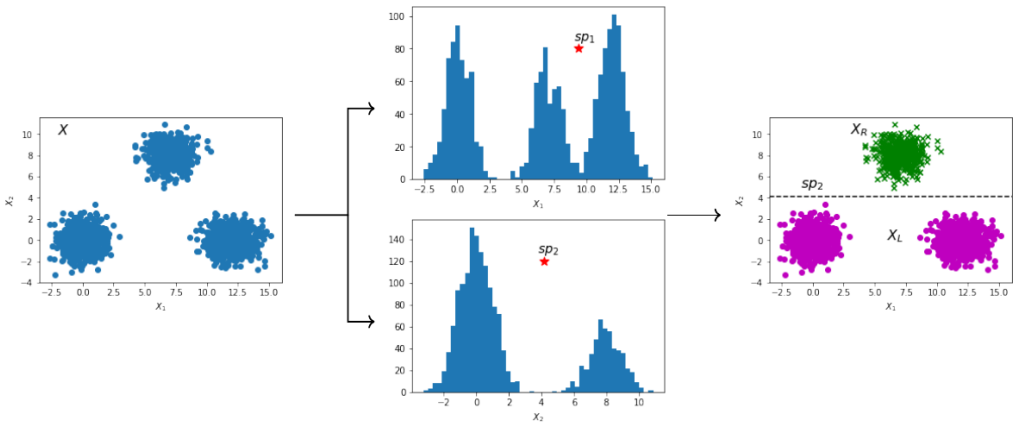
The final 2-D plot of X is given in Fig. 5.4e, where the two resulting split thresholds (horizontal split sp_2 and vertical split sp_3) and the final partition $\{X_{LL}, X_{LR}, X_R\}$ of X are illustrated. The corresponding binary decision tree for dataset X is presented in Fig. 5.5.

5.4 Experimental Results

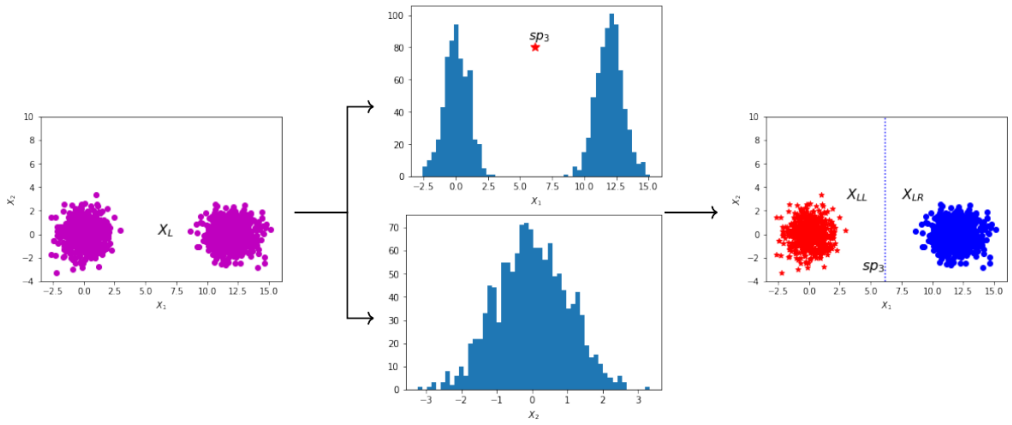
5.4.1 Evaluating DTAUC Performance

In this section we assess the performance of DTAUC on clustering synthetic and real data, focusing on the accurate estimation of the number of clusters and the quality of data partitioning. We compare the DTAUC method (using criteria 1 and 2) with the ICOT method [70] and the ExShallow method [62]. To the best of our knowledge, ICOT is the only method that provides a partition of the data into axis-aligned regions without using the ground-truth number of clusters during training. We also include an indirect method (ExShallow) in our experimental evaluation, in order to compare DTAUC and ICOT with a method that uses the ground-truth number of clusters.

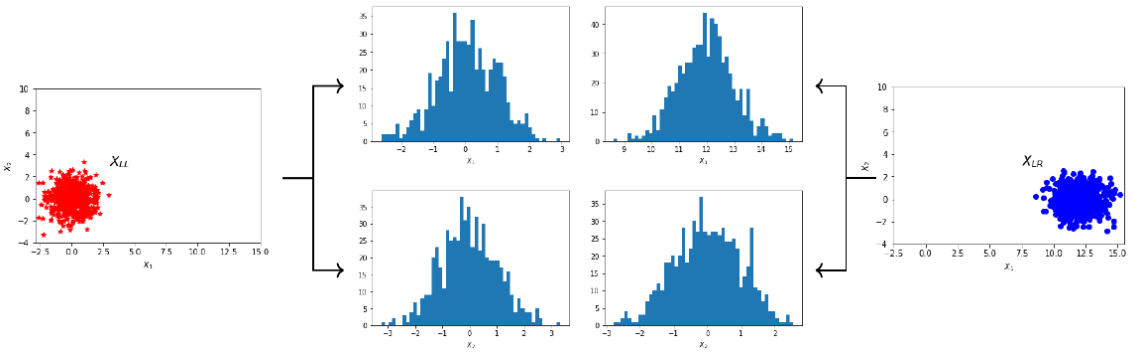
The three methods were applied to both synthetic (from the Fundamental Clustering Problems Suite (FCPS) [102]) and real datasets (from UCI [5]). Since ground



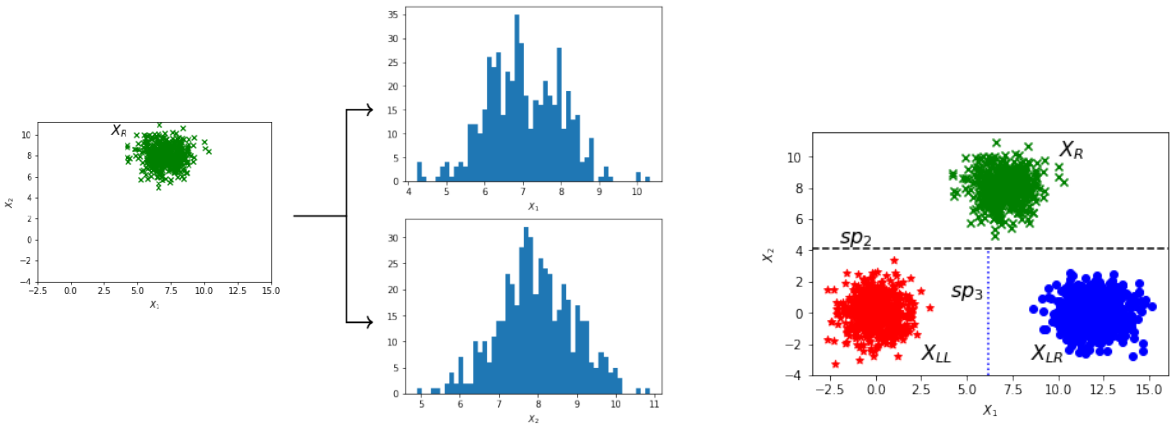
(a)



(b)



(c)



(d) (e)

Figure 5.4: Stepwise partitioning of a 2-D dataset (X) into axis unimodal rectangular regions using criterion 1. (a) 2-D plot of the original dataset X, with histogram plots of each feature, the obtained split points, and the resulting 2-D plot illustrating X split (by sp_2) into two clusters (X_L, X_R). (b) 2-D plot of X_L, with histogram plots of each feature, the obtained split point, and the resulting 2-D plot illustrating X_L split (by sp_3) into two clusters (X_{LL}, X_{LR}). (c) 2-D plots of X_{LL} and X_{LR}, along with the unimodal histogram plots of each feature. (d) 2-D plot of X_R, along with the unimodal histogram plots of each feature. (e) Final 2-D plot of X, illustrating the final split points (sp_2, sp_3) that partition X into three axis unimodal clusters (X_{LL}, X_{LR}, X_R).

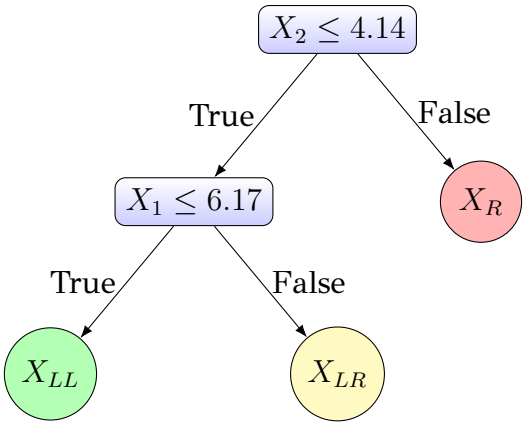


Figure 5.5: Binary decision tree constructed for the two-dimensional dataset of Figure 5.4a.

truth clustering information is available for each dataset, we evaluated the two methods in terms of splitting (clustering) performance using the widely used Normalized

Table 5.2: Parameters of synthetic and real datasets used in the experiments.

Dataset	n	d	$k^\star$
Synthetic			
Synthetic I	750	3	4
Hepta	212	3	7
Lsun	400	2	3
Tetra	400	3	4
TwoDiamonds	800	2	2
WingNut	1016	2	2
Real			
Boot Motor	94	3	3
Dermatology	366	33	6
Ecoli	327	7	5
Hist OldMaps	429	3	10
Image Seg.	210	19	7
Iris	150	4	3
Ruspini	75	2	4
Seeds	210	7	3

Mutual Information (NMI) score defined as follows:

$$NMI(Y, C) = \frac{2 \times I(Y, C)}{H(Y) + H(C)}, \tag{5.1}$$

where Y denotes the ground-truth labels, C denotes the cluster labels, $I(\cdot)$ is the mutual information measure and $H(\cdot)$ the entropy. This score ranges between 0 and 1, with a value close to 1 indicating that the ground truth partition has been found. All three methods build binary decision trees; therefore, we selected datasets suitable for partitioning into axis-aligned clusters for our experimental evaluation. Table 5.2 presents the parameters of each dataset (n: number of samples, d: number of features, $k^\star$: ground-truth number of clusters). We used min-max scaling for all datasets to ensure comparability with the ICOT method, which assumes features in the $[0, 1]$ range.

The DTAUC method uses a single parameter, the significance level α, which is necessary for the dip-test and UU-test to determine data unimodality during the splitting procedure. To determine an appropriate α for each dataset, we used the

Table 5.3: Partition results on synthetic data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported.

Dataset	$k^\star$/NMI	DTAUC_c1	DTAUC_c2	ICOT	ExShallow
Synthetic I	$k^\star = 4$	$k = 4$	$k = 4$	$k = 3$	$k^\star$ is given
	NMI	**0.99**	**0.99**	0.77	0.60
Hepta	$k^\star = 7$	$k = 7$	$k = 7$	$k = 4$	$k^\star$ is given
	NMI	0.95	0.98	0.74	**1.00**
Lsun	$k^\star = 3$	$k = 3$	$k = 3$	$k = 4$	$k^\star$ is given
	NMI	0.97	**0.99**	0.73	0.53
Tetra	$k^\star = 4$	$k = 4$	$k = 7$	$k = 4$	$k^\star$ is given
	NMI	0.94	0.92	**1.00**	**1.00**
Two Diamonds	$k^\star = 2$	$k = 2$	$k = 2$	$k = 2$	$k^\star$ is given
	NMI	**1.00**	**1.00**	**1.00**	**1.00**
WingNut	$k^\star = 2$	$k = 2$	$k = 2$	$k = 2$	$k^\star$ is given
	NMI	**1.00**	**1.00**	**1.00**	0.17

silhouette score [72] that is commonly used to assess the quality of a clustering solution. Specifically, for each dataset, we run the method for each value of $\alpha \in \{0.01, 0.05, 0.1\}$, compute the silhouette score for each obtained partition and keep the partition of maximum score as the final partition. In the ICOT method, we utilized a k-means warm start and retained the remaining parameters as specified in [70]. We encountered challenges running ICOT on datasets with a large number of features or clusters. This aligns with observations made by the authors in [70], who reported excessive runtimes for some datasets. For the ExShallow method, we provided the ground-truth number of clusters to run k-means and obtain the cluster labels. Then, a supervised binary decision tree is built by minimizing appropriate metrics as proposed in [62].

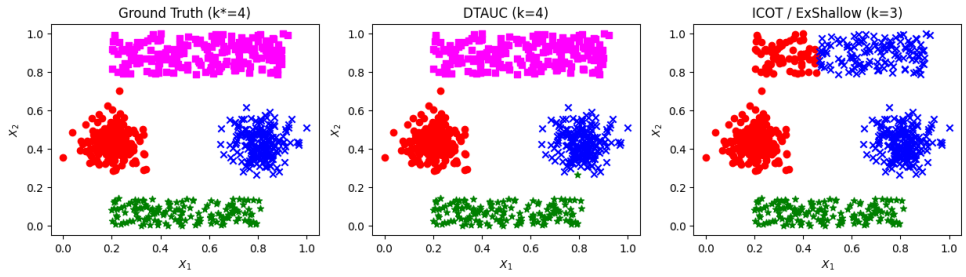
Tables 5.3 and 5.4 present for each dataset the NMI values and the number of clusters (k) as provided by the methods (it should be noted that in ExShallow the number of clusters is given). The ground-truth number of clusters ($k^\star$) is also provided in the second column of each table. DTAUC_c1 and DTAUC_c2 correspond to DTAUC method using criteria 1 and 2, respectively. The performance of DTAUC is

Table 5.4: Partition results on real data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported.

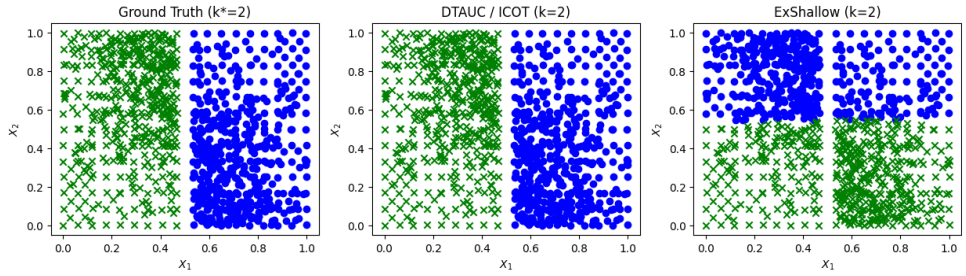
Dataset	$k^\star$/NMI	DTAUC_c1	DTAUC_c2	ICOT	ExShallow
Boot Motor	$k^\star = 3$	$k = 3$	$k = 3$	$k = 3$	$k^\star$ is given
	NMI	**1.00**	**1.00**	**1.00**	0.99
Dermatology	$k^\star = 6$	$k = 28$	$k = 67$	$k = 2$	$k^\star$ is given
	NMI	0.55	0.50	0.44	**0.83**
Ecoli	$k^\star = 5$	$k = 3$	$k = 3$	$k = 2$	$k^\star$ is given
	NMI	**0.61**	**0.61**	0.01	0.54
Hist OldMaps	$k^\star = 10$	$k = 10$	$k = 11$	$k = 2$	$k^\star$ is given
	NMI	0.74	**0.81**	0.03	0.75
Image Seg.	$k^\star = 7$	$k = 7$	$k = 8$	$k = 2$	$k^\star$ is given
	NMI	**0.69**	0.58	0.01	0.60
Iris	$k^\star = 3$	$k = 2$	$k = 4$	$k = 2$	$k^\star$ is given
	NMI	0.73	0.64	0.73	**0.81**
Ruspini	$k^\star = 4$	$k = 5$	-	$k = 4$	$k^\star$ is given
	NMI	0.89	-	**1.00**	**1.00**
Seeds	$k^\star = 3$	$k = 2$	-	$k = 2$	$k^\star$ is given
	NMI	0.63	-	0.53	**0.66**

superior compared to ICOT in most cases, achieving higher NMI values and closer estimations (k) of the ground-truth number of clusters ($k^\star$). However, DTAUC encounters challenges with some datasets, such as the Tetra dataset, where there is significant overlap among clusters. Another dataset where DTAUC_c1 demonstrates inferior performance is the Ruspini dataset. This dataset is relatively small ($n = 75$) and one of the four clusters is not compact. Consequently, DTAUC splits the non-compact cluster into two subclusters, detecting five clusters instead of four. For the Ruspini and Seeds datasets, DTAUC_c2 is not able to detect multimodal features or provide splits using the given α values ($0.01, 0.05, 0.1$). While it can generate splits with higher values of α, using $\alpha > 0.1$ is not a reliable choice in the context of statistical tests.

Another dataset to be discussed is Synthetic I, a three-dimensional dataset where



(a)



(b)

Figure 5.6: 2-D plots of (a) Synthetic I and (b) WingNut. The ground truth partition and the partitions obtained by DTAUC (using either criterion 1 or 2), ICOT and ExShallow are provided.

feature vectors X_1 and X_2 were generated using two Gaussian distributions and two uniform rectangles, and X_3 was generated using a uniform distribution. A 2-D plot of Synthetic I, with axes representing features X_1 and X_2, is provided in the left plot of Fig. 5.6a. It should be noted that since feature X_3 is uniformly distributed it does not contribute to the splitting process. This dataset is separated by axis-aligned splits; however, the ICOT and ExShallow methods fail in this task. As shown in Fig. 5.6a, ICOT fails to estimate the correct number of clusters ($k = 3$ instead of the actual $k^\star = 4$) (right plot), while the partition obtained by DTAUC (using either criterion 1 or 2) (middle plot) is successful. The 2-D plot of the ExShallow solution is almost identical to the ICOT plot (right plot in Fig. 5.6a).

DTAUC provides successful data partitions and accurately (or very closely) estimates the number of clusters for most synthetic and real datasets. In datasets (e.g., Ruspini) where sparse clusters exist, DTAUC demonstrates inferior performance compared to ICOT, since, based on the criterion of unimodality, it decides to split those clusters. However, ICOT is inferior in simple datasets, such as Synthetic I and Lsun, particularly when the clusters are close to each other and have a rectangular shape.

Table 5.5: Parameters of synthetic datasets used in the experiments comparing the two proposed criteria of the DTAUC method.

Dataset	n	d	$k^\star$
Synthetic II	800	2	5
Synthetic III	400	2	3
Synthetic IV	576	2	4
Synthetic V	790	2	5
Synthetic VI	2900	2	10
Synthetic VII	1900	2	3
Synthetic VII	2100	2	7

Table 5.6: Partition results of DTAUC method using criterion and criterion 2 on synthetic data reported: (i) The estimated number of clusters (k) and (ii) NMI values with respect to the ground truth labels. The ground truth number of clusters ($k^\star$) is also reported.

Dataset	$k^\star$/NMI	DTAUC_c1	DTAUC_c2
Synthetic II	$k^\star = 5$	$k = 6$	$k = 5$
	NMI	0.952	**0.995**
Synthetic III	$k^\star = 3$	$k = 3$	$k = 3$
	NMI	0.986	**1.00**
Synthetic IV	$k^\star = 4$	$k = 2$	$k = 4$
	NMI	0.880	**1.00**
Synthetic V	$k^\star = 5$	$k = 5$	$k = 5$
	NMI	0.986	**0.995**
Synthetic VI	$k^\star = 10$	$k = 11$	$k = 10$
	NMI	0.978	**0.992**
Synthetic VII	$k^\star = 3$	$k = 4$	$k = 3$
	NMI	0.890	**0.979**
Synthetic VIII	$k^\star = 7$	$k = 7$	$k = 7$
	NMI	0.959	**0.962**

In what concerns the indirect method (ExShallow), the information provided about the ground truth number of clusters seems to be helpful, in general. However, there

are simple datasets where it provides inferior results, such as Synthetic I, Lsun and WingNut. For example, in the case of the WingNut dataset (as illustrated in Fig. 5.6b), a single vertical line is required to split the data into two clusters (left plot); however, ExShallow fails to correctly determine this split (right plot). This mainly occurs due to an incorrect initial partition provided by the k-means algorithm, that is employed in the initial processing step. In this dataset, both DTAUC and ICOT provide a successful solution (middle plot).

5.4.2 Comparing Criterion 1 with Criterion 2

At this part of our experimental evaluation, we aim to compare the performance of the two proposed criteria (1 and 2) in DTAUC method. We created seven 2-d datasets and applied DTAUC method on each of them using the two criteria. Table 5.5 presents the parameters of each synthetic dataset.

In Table 5.6, we present the NMI values and the number of clusters (k) estimated by the DTAUC method using each criterion, along with the ground-truth number of clusters ($k^\star$), which is provided in the second column. Both criteria demonstrate high performance, with NMI results being similar. However, DTAUC_c2 achieves slightly higher NMI values across all datasets and provides cluster estimates that are closest to the ground truth numbers. In contrast, DTAUC_c1 produces less accurate estimates for some datasets, such as Synthetic II and IV.

The 2-D plots of the datasets listed in Table 5.6 are shown in Fig. 5.7, illustrating the original datasets (left plots), the solutions obtained by DTAUC_c1 (middle plots), and those by DTAUC_c2 (right plots). In most cases, the partitions are similar. However, for the Synthetic IV, VI, and VII datasets, the two criteria yield different solutions. Synthetic IV is a dataset containing four clusters (two very small and two very large) that can be partitioned with one vertical and one horizontal line. DTAUC_c1, however, fails to identify X_2 as a multimodal feature and, consequently, does not compute a horizontal split for X_2. In Synthetic VI and VII, DTAUC_c1 incorrectly splits a coherent rectangular cluster into two clusters, resulting in $k = 11$ and $k = 4$ clusters instead of the ground truth $k^\star = 10$ and $k^\star = 3$, respectively. In contrast, DTAUC_c2 successfully provides accurate partitions for these datasets.

Overall, although the two criteria demonstrate high performance and often produce nearly identical results, each one has its limitations. First, for a given feature,

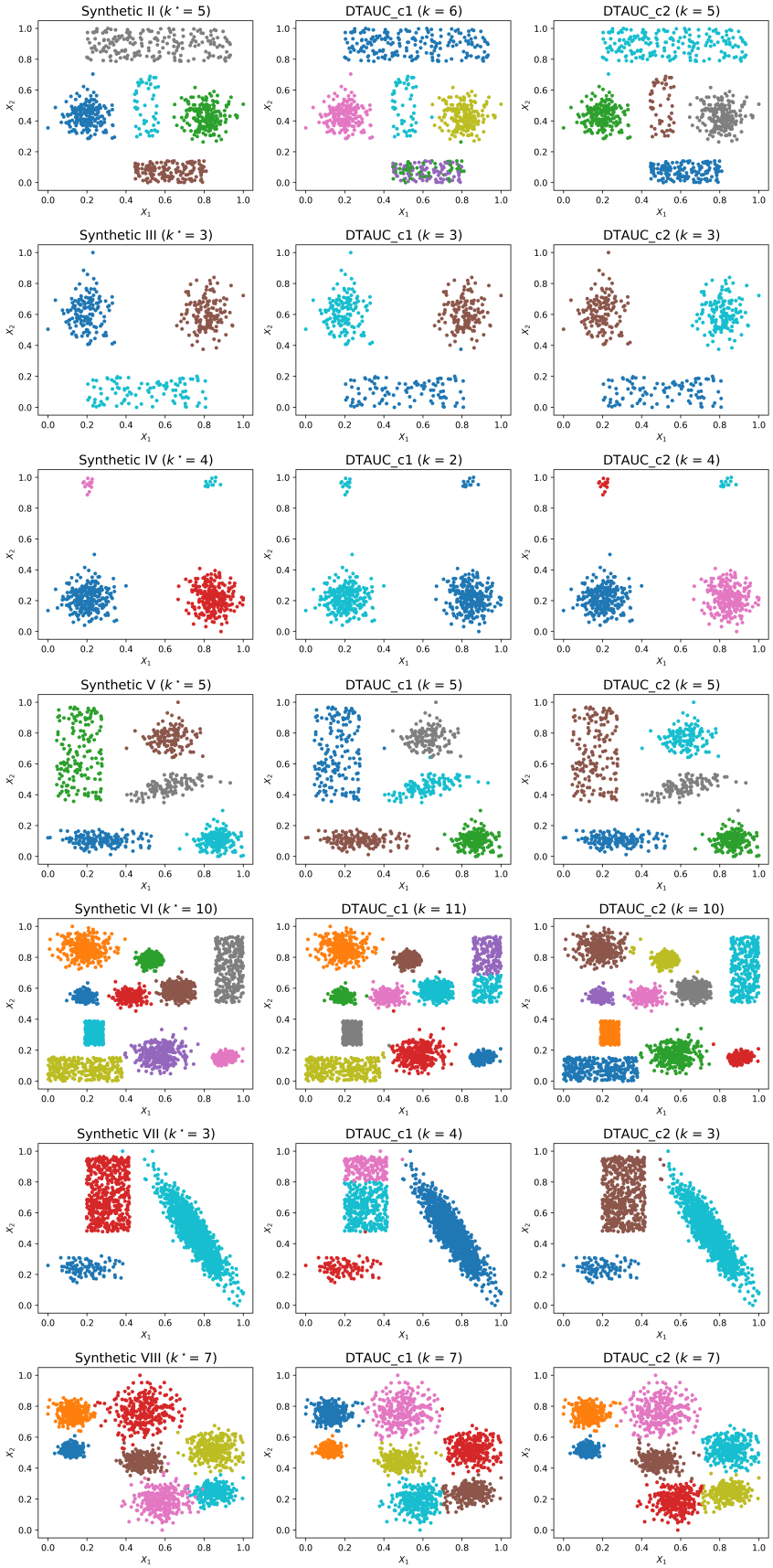


Figure 5.7: 2-D plots of synthetic datasets (Synthetic II – Synthetic VIII) illustrating the ground truth partition and the partitions obtained by DTAUC_c1 and DTAUC_c2.

DTAUC_c1 requires an explicit search to identify an appropriate split threshold among the feature values, whereas DTAUC_c2 is more straightforward, estimating the threshold directly using the splitting intervals provided by the UU-test. Both criteria depend on the significance level required for the dip-test (DTAUC_c1) and UU-test (DTAUC_c2). In our experiments, we ran DTAUC for $\alpha \in \{0.01, 0.05, 0.1\}$; however, DTAUC_c2 fails to provide a partition for two real datasets in Table 5.4 using these values, whereas DTAUC_c1 does not encounter this issue. Additionally, compared to DTAUC_c1, DTAUC_c2 produces more clusters than the ground truth in three real datasets. A limitation of DTAUC_c1 is its sensitivity to imbalanced clusters. In simple datasets from Table 5.6, where partitioning into hyperrectangles seems to be straightforward, DTAUC_c1 fails to deliver an entirely accurate partition, either due to the issue of imbalanced clusters or due to incorrect splitting of coherent clusters.

5.5 Summary

In this chapter, we have introduced the notion of axis unimodal cluster and proposed a method (DTAUC) for constructing binary trees for clustering based on axis unimodal partitions. This method follows the typical top-down paradigm for decision tree construction. It implements dataset splitting at each node by applying thresholding on the values of an appropriately selected multimodal feature. In order to select features and thresholds, two criteria have been proposed for the quality of the resulting partition that take into account unimodality and separation (criterion 1) and multimodality degree (criterion 2). The method automatically terminates when the subsets in all nodes are axis unimodal.

The DTAUC method relies on the idea of unimodality, which is closely related to clustering. It is simple to implement and provides axis-aligned partitions of the data, thus it offers interpretable clustering solutions. In addition, it does not involve any computationally expensive optimization technique, while it demonstrates the significant advantage that (apart from the typical statistical significance level) it does not include user-specified hyperparameters, for example, the number of clusters, the maximum depth of the tree or post-processing techniques, such as a pruning step.

In our experimental evaluation we assessed the performance of DTAUC on clustering both synthetic and real data, focusing on the accurate estimation of the number

of clusters and the quality of data partitioning. DTAUC, using either criterion 1 or criterion 2, was compared against methods that either require the number of clusters as input or determine it automatically. The experiments have shown that it provides successful data partitions and closely estimates the number of clusters across most synthetic and real datasets. Additionally, a comparison of the two criteria revealed their strong performance, with results often being nearly identical. However, each criterion has specific limitations, such as high dependence on the significance level or sensitivity to imbalanced clusters, making them complementary in handling challenging clustering problems.

Chapter 6

Conclusions and Future Work

The objective of this thesis was the development and implementation of machine learning methods based on the notion of unimodality. During the elaboration of the thesis we mainly focused on four different axes: i) creating a unimodality test for deciding data unimodality, ii) splitting multimodal data into unimodal subsets by detecting appropriate valley points, iii) building statistical models of univariate unimodal and multimodal data and iv) constructing (unsupervised) binary decision trees for clustering based on axis unimodal partitions.

Specifically, in Chapter 2, we proposed the Unimodal Uniform Test (UU-test), a novel approach for evaluating unimodality in one-dimensional datasets and constructing effective statistical models for unimodal data. The method operates by analyzing the empirical distribution function (ecdf) of a dataset, constructing a cumulative distribution function (cdf) that is piecewise linear, unimodal, and sufficiently models the data. The latter is ensured by applying uniformity tests on the data subsets corresponding to the linear segments. A key feature of UU-test is that it does not require any parameter estimation or bootstrapping, which makes it computationally efficient compared to methods like the dip-test. A notable advantage of UU-test is its dual purpose: it not only decides whether a dataset is unimodal but also generates a statistical model for the data in the form of a Uniform Mixture Model (UMM). This model provides a meaningful representation of the underlying distribution while maintaining simplicity and interpretability. Unlike typical methods that focus solely on determining unimodality, UU-test fills the gap by offering both a decision and a model,

which is particularly beneficial in applications that require further data processing or simulation.

Future research could focus on integration of the UU-test on various data analysis tasks exploiting the decisions on unimodality that it offers. UU-test could be used in clustering algorithms [3, 29] that currently rely on the dip-test for unimodality. As illustrated in Chapter 2, Section 2.6, in addition to the decision on unimodality (also provided by dip-test), UU-test directly suggests appropriate cut points in the case of multimodality. Such information is valuable for the clustering algorithm, since the cut points can be used for splitting the multimodal clusters. UU-test could also be used in applications that rely on statistical modeling to enhance the typical approach for unimodal data modeling by using the Uniform Mixture Model instead of using a single distribution (e.g. Gaussian, uniform, Student's t etc.). Another line of research concerns the generation of synthetic unimodal data that follow the same distribution as the original unimodal dataset. Finally, the UU-test method could prove useful in image thresholding problems that work with the image histogram [103, 104, 105].

In Chapter 3 we focused on improving UMM performance provided by UU-test, by substituting the uniform distribution with a more flexible distribution. Specifically, we proposed the Unimodal Π-sigmoid Mixture Model (UΠsMM), which replaces the uniform components of UMM with Π-sigmoid distributions. The Π-sigmoid distribution, defined as the difference of two translated logistic sigmoids, exhibits significant flexibility, enabling it to approximate a wide range of distributions, from Gaussian to uniform, by adjusting the slope of the sigmoids. This versatility allows UΠsMM to effectively capture the underlying characteristics of unimodal data. A critical aspect of our approach is the initialization of UΠsMM using the output of the UU-test algorithm, which ensures the unimodality of the initial mixture model. During training via the Expectation-Maximization (EM) algorithm, we addressed the challenge of maintaining the unimodality constraint. To this end, we introduced a mechanism to detect and correct any violations of unimodality during training by iteratively reducing the number of mixture components. This procedure not only ensures that the resulting model remains unimodal but also leads to a more parsimonious representation, improving its generalization ability.

While the current approach ensures unimodality during training, future work could focus on refining the optimization process to enhance convergence speed and computational efficiency. Incorporating modern optimization techniques or approx-

imate EM algorithms could further improve the training phase, especially for large datasets. Although the Π-sigmoid distribution provides notable flexibility, future research could investigate alternative distributions with similar or greater flexibility to model unimodal data. Comparing the performance of UΠsMM with these alternatives could provide insights into potential improvements or hybrid approaches. Additionally, since UΠsMM effectively models univariate unimodal data, similar to UMM, future research could focus on integrating the proposed approach into machine learning/data mining algorithms and methodologies that exploit univariate statistical modeling (e.g. Naive Bayes). Another major research direction is related to the statistical modeling of multimodal datasets using a mixture of UΠsMMs, where each UΠsMM component models a unimodal subset of the data.

In Chapter 4, we introduced the Unimodal Mixture Model (UDMM), a hierarchical statistical mixture model for effectively modeling univariate multimodal data. The UDMM builds on the Uniform Mixture Model (UMM) by leveraging its ability to model unimodal data and extends this capability to multimodal distributions through the use of a novel data partitioning technique, UniSplit. The proposed UniSplit algorithm determines valley points of univariate multimodal data achieving to split the original data into unimodal subsets. This approach relies on the idea of unimodality. We introduced properties of critical points (gcm/lcm points) of the data ecdf that provide indications on the existence of density valleys. These properties are exploited in the proposed UniSplit algorithm. The subsets provided by UniSplit are then modeled using UMMs, resulting in a flexible, non-parametric approach for density estimation that requires no training or manual hyperparameter tuning.

One of the key strengths of UDMM is its flexibility and independence from specific parametric assumptions about the underlying distributions of the data. This makes the method particularly appropriate for datasets generated by sources of different probability density (e.g., one Gaussian and one uniform). Additionally, the number of components in the UDMM is determined automatically, eliminating the need for user-defined parameters such as the number of components in Gaussian Mixture Models or the kernel bandwidth in mean shift algorithm. The lack of hyperparameters, apart from the statistical significance level of the uniformity test, underscores the method's ease of use and practical applicability. Through experiments on synthetic and real-world datasets, we demonstrated the efficacy of UDMM for statistical modeling and the robustness of the UniSplit algorithm for partitioning multimodal

data into unimodal subsets. The comparisons with alternative clustering and modeling approaches highlighted the advantages of our method in terms of flexibility, accuracy, and interpretability.

Since the proposed approach provides accurate statistical modeling of univariate data, it could be employed in any method or application requiring this type of modeling. Exploitation of the method for partitioning and statistical modeling of multidimensional datasets constitutes an important future research direction. For example, this could be achieved by determining appropriate univariate projections of the data where UniSplit could be employed for data splitting. Another direction for future work is to address the limitation of UDMM when applied to very small datasets, particularly in cases where limited data within specific intervals can lead to missing gcm/lcm points, resulting in unidentified valley points. To overcome this challenge, future research could explore alternative approaches for valley detection that do not solely rely on gcm/lcm points.

Finally, in Chapter 5 we introduced Decision Trees for Axis Unimodal Clustering (DTAUC), a novel clustering methodology that is based on the notion of axis unimodality to partition datasets into interpretable axis-aligned clusters. DTAUC builds decision trees in a top-down manner, where each split is guided by a carefully selected feature and an optimal threshold that increases the unimodality of the resulting partition. Two distinct criteria were proposed to evaluate the quality of splits: the first combines the p-values from Hartigans' dip-test with a separation metric, while the second uses the multimodality degree and the UU-test for unimodality. DTAUC is a simple method that offers interpretability. It avoids the complexity of preprocessing steps, computationally intensive optimization methods, or numerous hyperparameters that are typical in many unsupervised decision tree methods. Instead, it only requires setting the statistical significance level for the unimodality test, making it well-suited for unsupervised settings where hyperparameter tuning is challenging. Additionally, the axis-aligned decision tree structure provides a clear representation of the clustering process, enhancing the interpretability of the results.

Future work could focus on using a set of features/splitting rules (instead of a single feature/splitting rule) at each node, as oblique trees do. While this would make the resulting trees less interpretable, it would offer more accurate clustering solutions. It is also interesting to implement post-processing steps to improve the performance of DTAUC. In DTAUC each tree leaf represents a single cluster. Several methods

merge adjacent leaves into larger clusters, thereby capturing more complex structures in the data. After obtaining the final tree, we could consider the possibility of merging leaves if the unimodality assumption is retained.

Bibliography

[1] I. Stoepker and E. van den Heuvel, "Testing for multimodality," Ph.D. dissertation, BS thesis, Eindhoven University of Technology, Eindhoven, 2016.

[2] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouët, "Are your data gathered?" in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2210–2218.

[3] S. Maurus and C. Plant, "Skinny-dip: clustering in a sea of noise," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1055–1064.

[4] J. Delon, A. Desolneux, J.-L. Lisani, and A. B. Petro, "A nonparametric approach for histogram segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 253–261, 2006.

[5] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

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

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] I. Kononenko and M. Kukar, *Machine Learning and Data Mining*. Horwood Publishing, 2007.

[9] S. D. Silvey, *Statistical Inference*. Routledge, 2017.

[10] W. J. Conover, *Practical Nonparametric Statistics*. John Wiley & Sons, 1999, vol. 350.

[11] E. L. Lehmann, J. P. Romano, and G. Casella, *Testing Statistical Hypotheses*. Springer, 1986, vol. 3.

[12] Y. Dodge, "Kolmogorov–smirnov test," *The Concise Encyclopedia of Statistics*, pp. 283–287, 2008.

[13] G. McLachlan, "Finite mixture models," *A Wiley-Interscience Publication*, 2000.

[14] C. Loader, *Local Regression and Likelihood*. Springer Science & Business Media, 2006.

[15] S. Dharmadhikari and K. Joag-Dev, *Unimodality, Convexity, and Applications*. Elsevier, 1988.

[16] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes," *Ann. Math. Statist.*, vol. 23, no. 2, pp. 193–212, 06 1952.

[17] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 12 1965.

[18] J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivariate Behavioral Research*, vol. 5, no. 3, pp. 329–350, 1970.

[19] L. Engelman and J. A. Hartigan, "Percentage points of a test for clusters," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1647–1648, 1969.

[20] M. C. Minnotte, *A Test of Mode Existence with Applications to Multimodality*. Rice University, 1993.

[21] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 43, no. 1, pp. 97–99, 1981.

[22] P. Hall and M. York, "On the calibration of silverman's test for multimodality," *Statistica Sinica*, pp. 515–536, 2001.

[23] D. W. Muller and G. Sawitzki, "Excess mass estimates and tests for multimodality," *Journal of the American Statistical Association*, vol. 86, no. 415, pp. 738–746, 1991.

[24] G. Sawitzki, *The Excess Mass Approach and the Analysis of Multi-modality*. Springer, 1996.

[25] J. A. Hartigan and S. Mohanty, "The runt test for multimodality," *Journal of Classification*, vol. 9, no. 1, pp. 63–70, 1992.

[26] G. P. M. Rozál and J. Hartigan, "The map test for multimodality," *Journal of Classification*, vol. 11, no. 1, pp. 5–36, 1994.

[27] J. A. Hartigan, P. M. Hartigan *et al.*, "The dip test of unimodality," *The Annals of Statistics*, vol. 13, no. 1, pp. 70–84, 1985.

[28] A. Adolfsson, M. Ackerman, and N. C. Brownstein, "To cluster, or not to cluster: An analysis of clusterability methods," *Pattern Recognition*, vol. 88, pp. 13–26, 2019.

[29] A. Kalogeratos and A. Likas, "Dip-means: an incremental clustering method for estimating the number of clusters," in *Advances in Neural Information Processing Systems*, 2012, pp. 2393–2401.

[30] B. Schelling and C. Plant, "Diptransformation: Enhancing the structure of a dataset and thereby improving clustering," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 407–416.

[31] A. Krause and V. Liebscher, "Multimodal projection pursuit using the dip statistic," Ernst-Moritz-Arndt-Univ., Inst. für Mathematik und Informatik, Tech. Rep., 2005.

[32] L. H. Fraser, J. Pither, A. Jentsch, M. Sternberg, M. Zobel, D. Askarizadeh, S. Bartha, C. Beierkuhnlein, J. A. Bennett, A. Bittel *et al.*, "Worldwide evidence of a unimodal relationship between productivity and plant species richness," *Science*, vol. 349, no. 6245, pp. 302–305, 2015.

[33] C. Barichievy, D. G. Angeler, T. Eason, A. S. Garmestani, K. L. Nash, C. A. Stow, S. Sundstrom, and C. R. Allen, "A method to detect discontinuities in census data," *Ecology and Evolution*, vol. 8, no. 19, pp. 9614–9623, 2018.

[34] K. Johnsson, M. Linderoth, and M. Fontes, "What is a "unimodal" cell population? using statistical tests as criteria for unimodality in automated gating and quality control," *Cytometry Part A*, vol. 91, no. 9, pp. 908–916, 2017.

[35] X. Yao, J. Cafaro, A. J. McLaughlin, F. R. Postma, D. L. Paul, G. Awatramani, and G. D. Field, "Gap junctions contribute to differential light adaptation across direction-selective retinal ganglion cells," *Neuron*, vol. 100, no. 1, pp. 216–228, 2018.

[36] N. Schmitt and F. Westerhoff, "On the bimodality of the distribution of the s&p 500's distortion: Empirical evidence and theoretical explanations," *Journal of Economic Dynamics and Control*, vol. 80, pp. 34–53, 2017.

[37] D. Cliff, "Co-evolutionary dynamics in a simulation of interacting financial-market adaptive automated trading systems," in *34th European Modelling and Simulation Symposium*. CAL-TEK SRL, 2022, pp. 1–13.

[38] D. Cliff, "Parameterised response zero intelligence traders," *Journal of Economic Interaction and Coordination*, pp. 1–54, 2023.

[39] L. Scrucca, "A transformation-based approach to gaussian mixture density estimation for bounded data," *Biometrical Journal*, vol. 61, no. 4, pp. 873–888, 2019.

[40] J. Li and H. Zha, "Two-way poisson mixture models for simultaneous document classification and word clustering," *Computational Statistics & Data Analysis*, vol. 50, no. 1, pp. 163–180, 2006.

[41] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway, "Clustering on the unit hypersphere using von mises-fisher distributions." *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[42] J. E. Chacón, "Mixture model modal clustering," *Advances in Data Analysis and Classification*, vol. 13, pp. 379–404, 2019.

[43] R. A. Sampaio, J. D. Garcia, M. Poggi, and T. Vidal, "Regularization and optimization in model-based clustering," *Pattern Recognition*, vol. 150, p. 110310, 2024.

[44] J. Hartigan, *Clustering Algorithms*. New York: John Wiley & Sons, 1975.

[45] J. E. Chacón, "The modal age of statistics," *International Statistical Review*, vol. 88, no. 1, pp. 122–141, 2020.

[46] G. Menardi, "A review on modal clustering," *International Statistical Review*, vol. 84, no. 3, pp. 413–433, 2016.

[47] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[48] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[49] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoidshifts," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[50] J. N. Myhre, K. Ø. Mikalsen, S. Løkse, and R. Jenssen, "Robust clustering using a knn mode seeking ensemble," *Pattern Recognition*, vol. 76, pp. 491–505, 2018.

[51] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[52] Z. Rasool, S. Aryal, M. R. Bouadjenek, and R. Dazeley, "Overcoming weaknesses of density peak clustering using a data-dependent similarity measure," *Pattern Recognition*, vol. 137, p. 109287, 2023.

[53] J. Li, S. Ray, and B. G. Lindsay, "A nonparametric statistical approach to clustering via mode identification." *Journal of Machine Learning Research*, vol. 8, no. 8, 2007.

[54] L. Scrucca, "A fast and efficient modal em algorithm for gaussian mixtures," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 14, no. 4, pp. 305–314, 2021.

[55] L. G. Bauer, C. Leiber, C. Böhm, and C. Plant, "Extension of the dip-test repertoire-efficient and differentiable p-value calculation for clustering," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 109–117.

[56] C. Molnar, *Interpretable Machine Learning*. Lulu. com, 2020.

[57] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees*. CRC Press, 1984.

[58] J. R. Quinlan, "Discovering rules by induction from large collections of examples," *Expert Systems in the Micro Electronics Age*, 1979.

[59] J. R. Quinlan, "C4. 5: Programs for machine learning," 1993.

[60] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 2, pp. 119–127, 1980.

[61] J. R. Quinlan *et al.*, "Learning with continuous classes," in *5th Australian joint conference on artificial intelligence*, vol. 92. World Scientific, 1992, pp. 343–348.

[62] E. Laber, L. Murtinho, and F. Oliveira, "Shallow decision trees for explainable k-means clustering," *Pattern Recognition*, vol. 137, p. 109239, 2023.

[63] P. Tavallali, P. Tavallali, and M. Singhal, "K-means tree: an optimal clustering tree for unsupervised learning," *The Journal of Supercomputing*, vol. 77, no. 5, pp. 5239–5266, 2021.

[64] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," *arXiv preprint cs/0011032*, 2000.

[65] J. Basak and R. Krishnapuram, "Interpretable hierarchical clustering by constructing an unsupervised decision tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 121–132, 2005.

[66] R. Fraiman, B. Ghattas, and M. Svarc, "Interpretable clustering using unsupervised binary trees," *Advances in Data Analysis and Classification*, vol. 7, pp. 125–145, 2013.

[67] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, 2000, pp. 20–29.

[68] M. Gabidolla and M. Á. Carreira-Perpiñán, "Optimal interpretable clustering using oblique decision trees," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 400–410.

[69] D. Heath, S. Kasif, and S. Salzberg, "Induction of oblique decision trees," in *IJCAI*, vol. 1993. Citeseer, 1993, pp. 1002–1007.

[70] D. Bertsimas, A. Orfanoudaki, and H. Wiberg, "Interpretable clustering: an optimization approach," *Machine Learning*, vol. 110, no. 1, pp. 89–138, 2021.

[71] D. Bertsimas and J. Dunn, "Optimal classification trees," *Machine Learning*, vol. 106, pp. 1039–1082, 2017.

[72] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[73] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[74] P. Chasani and A. Likas, "The uu-test for statistical modeling of unimodal data," *Pattern Recognition*, vol. 122, p. 108272, 2022.

[75] P. Chasani and A. Likas, "Statistical modeling of univariate unimodal data using Π-sigmoid mixture models," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, J. Macintyre, M. Avlonitis, and A. Papaleonidas, Eds. Cham: Springer Nature Switzerland, 2024, pp. 349–361.

[76] P. Chasani and A. Likas, "Statistical modeling of univariate multimodal data," 2024. [Online]. Available: https://arxiv.org/abs/2412.15894

[77] P. Chasani and A. Likas, "Unsupervised decision trees for axis unimodal clustering," *Information*, vol. 15, no. 11, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/11/704

[78] T. Robertson, F. Wright, and R. Dykstra, *Order Restricted Statistical Inference*. New York: John Wiley and Sons, 1988.

[79] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, New York, 2000.

[80] P. F. Craigmile and D. Tirrerington, "Parameter estimation for finite mixtures of uniform distributions," *Communications in Statistics-Theory and Methods*, vol. 26, no. 8, pp. 1981–1995, 1997.

[81] N. Bouguila and W. Fan, *Mixture Models and Applications*. Springer, 2020.

[82] S. Chen and M. Wang, "Seeking multi-thresholds directly from support vectors for image segmentation," *Neurocomputing*, vol. 67, pp. 335–344, 2005.

[83] J. Fox and S. Weisberg, *An R Companion to Applied Regression*, 3rd ed. Thousand Oaks CA: Sage, 2019. [Online]. Available: https://socialsciences.mcmaster.ca/jfox/Books/Companion/

[84] T. Chamalis and A. Likas, "The projected dip-means clustering algorithm," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–7.

[85] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Springer Science & Business Media, 2011.

[86] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.

[87] I. T. Jolliffe, *Principal Component Analysis*,, 2nd ed. New York, Springer, 2002.

[88] M. Roux, "A comparative study of divisive and agglomerative hierarchical clustering algorithms," *Journal of Classification*, vol. 35, no. 2, pp. 345–366, 2018.

[89] D. Boley, "Principal direction divisive partitioning," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 325–344, 1998.

[90] G. Hamerly and C. Elkan, "Learning the k in k-means," ser. NIPS'03. Cambridge, MA, USA: MIT Press, 2003, pp. 281–288.

[91] A. Alivanoglou and A. Likas, "Probabilistic models based on the π-sigmoid distribution," in *Artificial Neural Networks in Pattern Recognition: Third IAPR Workshop, ANNPR 2008 Paris, France, July 2-4, 2008 Proceedings 3*. Springer, 2008, pp. 36–43.

[92] G. J. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley Series in Probability and Statistics, 2000.

[93] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, 2007.

[94] H. D Jr, "Hedonic prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, pp. 81–102, 1978.

[95] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery, "mclust 5: clustering, classification and density estimation using gaussian finite mixture models," *The R journal*, vol. 8, no. 1, p. 289, 2016.

[96] L. Scrucca, C. Fraley, T. B. Murphy, and A. E. Raftery, *Model-Based Clustering, Classification, and Density Estimation Using mclust in R*. Chapman and Hall/CRC, 2023.

[97] J. Ameijeiras-Alonso, R. M. Crujeiras, and A. Rodriguez-Casal, "multimode: An r package for mode assessment," *Journal of Statistical Software*, vol. 97, no. 9, p. 1–32, 2021. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v097i09

[98] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2015.

[99] B. Silverman, "Density estimation for statistics and data analysis," *Monographs on Statistics and Applied Probability*, 1986.

[100] C. Leiber, L. Miklautz, C. Plant, and C. Böhm, "Benchmarking deep clustering algorithms with clustpy," in *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2023, pp. 625–632.

[101] G. Vardakas, A. Kalogeratos, and A. Likas, "Uniforce: The unimodality forest method for clustering and estimation of the number of clusters," *arXiv preprint arXiv:2312.11323*, 2023.

[102] A. Ultsch, "Fundamental clustering problems suite (fcps)," Technical report, University of Marburg, Tech. Rep., 2005.

[103] P. L. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.

[104] N. Coudray, J.-L. Buessler, and J.-P. Urban, "Robust threshold estimation for images with unimodal histograms," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 1010–1019, 2010.

[105] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.

Author's Publications

1. Paraskevi Chasani and Aristidis Likas, The UU-test for statistical modeling of unimodal data, *Pattern Recognition*, vol. 122, p. 108272, 2022
   doi: https://doi.org/10.1016/j.patcog.2021.108272

2. Paraskevi Chasani and Aristidis Likas, Statistical Modeling of Univariate Unimodal Data Using Π-Sigmoid Mixture Models, *IFIP Advances in Information and Communication Technology*, pp. 349–361, 2024
   doi: https://doi.org/10.1007/978-3-031-63219-8_26

3. Paraskevi Chasani and Aristidis Likas, Unsupervised Decision Trees for Axis Unimodal Clustering, *Information*, vol. 15, no. 11, pp. 704–704, 2024
   doi: https://doi.org/10.3390/info15110704

4. Paraskevi Chasani and Aristidis Likas, Statistical Modeling of Univariate Multimodal Data, submitted for publication
   doi: https://doi.org/10.48550/arXiv.2412.15894

Short Biography

Paraskevi Chasani received her B.Sc. degree in Mathematics (2015) (grade 8.67/10 "Excellent") from the Department of Mathematics, University of Ioannina, Greece. In 2019 she received her M.Sc. degree in Computer Science (grade 9.58/10 "Excellent") from the Department of Computer Science and Engineering, University of Ioannina, Greece. Since 2019 she has been a Ph.D. candidate at the same department. She received various scholarships during her undergraduate and graduate studies. Her research interests include statistical modeling, machine learning and data mining.