A Resampling Approach to Clustering with Confidence

Yuan Chiam

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science

University of Washington

2012

Reading Committee:

Professor Werner Stuetzle, Chair

Professor Marina Meilă

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

A Resampling Approach to Clustering with Confidence

Yuan Chiam

Chair of the Supervisory Committee:
Professor Werner Stuetzle
Statistics

We propose a method for estimating the number of groups in a data set. Our method is an extension of Generalized Single Linkage clustering (GSL) (Stuetzle and Nugent 2010), a nonparametric clustering method based on the premise that groups in the data correspond to modes of the underlying data density. GSL starts with a nonparametric density estimate. It recursively splits the data into high density regions separated by valleys. The leaves of the resulting cluster tree correspond to modes of the density estimate. The problem is that nonparametric density estimates tend to have spurious modes due to sampling variability, giving rise to spurious splits in the cluster tree. We propose a resampling method aimed at assessing the significance of splits and a way of constructing a cluster tree making only significant splits. The only parameter is the significance level. Our method can identify highly non-linear groups. Simulation experiments suggest that the method is very conservative, which may explain its low power.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to Professor Werner Stuetzle for his invaluable guidance, sound advice and patience throughout my thesis-writing period. This thesis would have not have been possible without his support and encouragement. His enthusiasm and knowlegde have made statistics fun and enjoyable for me.

Chapter 1

INTRODUCTION

The goal of cluster analysis is to identify distinct groups in a data set. One of the major challenges in clustering is choosing the number of clusters. This has been an area of active research; we review some of the ideas in Chapter 2. Our approach is based on generalized single linkage clustering (GSL) (Stuetzle and Nugent 2010). We assume in the following that the reader is familiar with GSL. GSL assumes a correspondence between groups and modes of the underlying density $p(x)$, and it estimates the modes of the $p(x)$ by the modes of a nonparametric density estimate $\hat{p}(x)$. However, nonparametric density estimates typically have spurious modes due to sampling variability, and the challenge is to distinguish spurious modes from real ones. We first choose a confidence threshold and generate random partitions of data into half-samples X_{train} and X_{test}. For each partition, we construct a cluster tree from the training data such that the significance level of each split which is assessed using the test set is lower than the signifiance threshold. We then "average" the resulting cluster trees over partitions to arrive at a final tree. We describe our method in detail in Chapter 3. In Chapter 4, we present results of a Monte Carlo study of its performance. Chapter 5 concludes with a summary, discussion and ideas for future work.

Chapter 2

LITERATURE REVIEW

A number of methods for choosing the number of clusters have been proposed over the years. We will review three distinct approaches: a) Burman and Polonik (2009) examine the density along the line segments connecting potential modal candidates to determine if there is a dip which reflects distinct modal regions. Tibshirani, Walther and Hastie (2009) focus on k-means clustering and attempt to detect an "elbow" in the plot of within cluster dispersion versus number of clusters. Tibshirani and Walther (2005) use a measure of stability for determining the number of clusters.

2.1 Multivariate mode hunting: Data analytic tools with measures of significance - Burman and Polonik, 2009

Burman and Polonik (2009) proposed a method for locating isolated modes in a multivariate data set without pre-specifying their total number. Their method consists of the following three steps. First, they select initial modal candidates using an iterative nearest neighbor method which repeatedly employs two substeps: a) searching for a modal candidate, and b) eliminating its neighbors. Second, they test whether the level sets around each modal candidate are approximately elliptical. Modal candidates that fail the test are subsequently eliminated. However, the purpose of this step is not entirely clear. The third and most crucial step is to test whether the remaining candidates after steps 1 and 2 really represent different modal regions. For a given pair of modal candidates m_i and m_j, they consider the line segment connecting m_i and m_j and assess whether the density along the line has a dip. More formally, they define $SB(\alpha) := min\{logf(m_i), logf(m_j)\} - logf(m_\alpha)$ where $m_\alpha = (1-\alpha)m_i + \alpha m_j$ for $\alpha \in [0,1]$ denotes a point on the line connecting m_i and m_j. A test of $SB(\alpha) \leq 0$ which shows a significant overshoot over zero or a significant bump would indicate that modal regions m_i and m_j are in different modal regions of the underlying density. To decide which of the modal candidates are "real", they repeatedly apply the pairwise method above. Let $m_1, m_2 ... m_k$ be the candidates numbered in decreasing order of density. They first test to see whether m_i and m_j for i=1 and j=i+1, ..., k belong to the same modal region. If the test does not reject, W_j is eliminated as a modal candidate. This

results in a smaller subset of candidates $m_1...m_{k_1}$ with $k_1 \leq k$. They then set $i = i + 1$ and repeat the substeps iteratively until only distinct modal regions are left.

2.2 Estimating the number of clusters in a data set via the gap statistic - Tibshirani, Walther and Hastie, 2009

Tibshirani, Walther and Hastie (2009) proposed the 'gap statistic' to choose the appropriate number k of clusters in k-means clustering. The main idea is to standardize the graph of $log(W_k)$ where W_k is the within-cluster sum of squares (dispersion) for k clusters by comparing it with its expectation under an appropriate reference distribution of the data. They define the gap statistic as $Gap_n(k) = \mathbb{E}_n^*\{log(W_k)\} - log(W_k)$ where $\mathbb{E}_n^*$ denotes expectation for samples of size n from the reference distribution. The estimate of the optimal number of clusters $\hat{k}$ is the value maximizing $Gap_n(k)$. The authors consider two choices for the reference distribution: a) a uniform distribution on the smallest axis parallel hyperrectangle containing the data and b) a uniform distribution over a bounding box aligned with the principal components of the data. Method (a) has the advantage of simplicity while method (b) takes into account the shape of the data distribution. It is important to note that k-means clustering performs well only under the assumption of spherical groups.

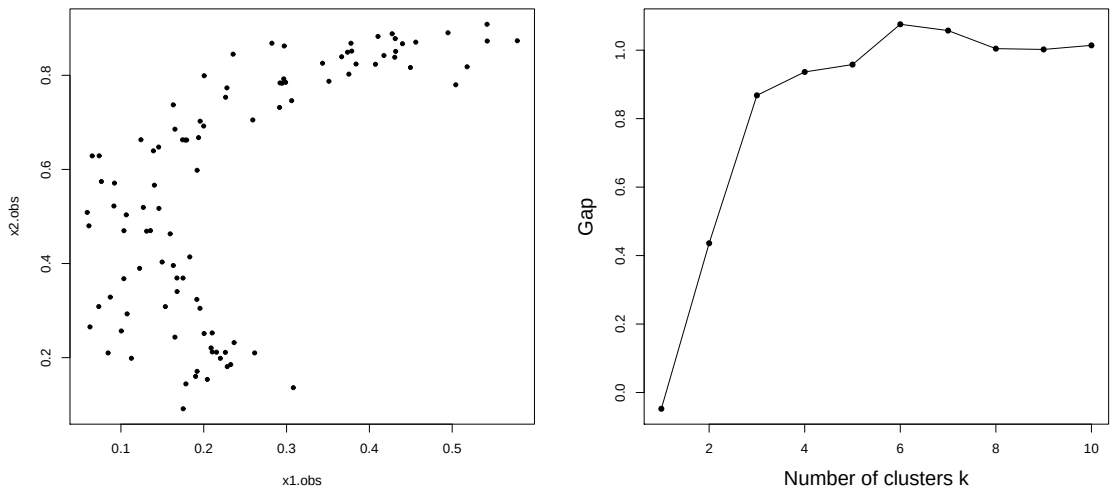


Figure 2.1: *(Left) Banana-shaped data, (Right) Gap curve*

If this assumption is violated, the gap statistic tends to over-estimate the number of groups. Consider the same banana-shaped data set (Fig. 2.1 *Left*), the maximum value of the gap statistic from the resulting gap curve (Fig 2.1 *Right*) misleadingly estimate 6 clusters.

2.3 Cluster Validation by Prediction Strength - Tibshirani and Walther, 2005

Tibshirani and Walther (2005) approached estimating of the number of clusters as a model selection problem, focusing on prediction error rather than within-cluster dispersion. Their main idea is to examine the stability of the clusters based on a prediction strength measure. The data X is first divided into X_{train} and X_{test} with g and $h = n - g$ observations respectively. Both X_{train} and X_{test} are then clustered into k clusters using any clustering procedure, obtaining cluster labels $y_1...y_g$ and $z_1...z_h$. The observations in X_{test} are assigned to the clusters of X_{train}, obtaining predicted cluster labels $\hat{z}_1...\hat{z}_h$. This is a supervised learning problem. With two partitions of X_{test}, one obtained by directly clustering X_{test}, the other by assigning the observations in X_{test} to the clusters obtained by clustering X_{train}, they test the stability of the clusters by checking the similarity of the two partitions. To measure agreement between the two partitions, specifically to measure how well the training set clusters predict co-memberships in the test set, Tibshirani and Walther introduced the idea of "prediction strength". For each pair of test observations

that are assigned to the same test cluster, they determine whether they are also assigned to the same cluster based on the training set. For each test cluster, they compute the proportion of observation pairs in that cluster that are also assigned to the same cluster by the training set. "Prediction strength", $ps(k)$ is hence defined as the minimum of this quantity over k test clusters. If $k = k_0$, the true number of clusters, then the k training set clusters will be similar to the k test set clusters, and will predict them well, resulting in a high $ps(k)$. The optimal number of clusters $\hat{k}$ is chosen to be the largest k such that $ps(k)$ is above some threshold. This procedure is repeated for random partitions and the results are "averaged" to obtain the number of clusters. The premise is that if the number of clusters is larger than the number of distinct groups in the data, then the partitions will be unstable. However, whether or not this premise holds depends on the clustering method. In their paper, Tibshirani and Walther focus on k-means clustering and they note that their method fails if groups in the data are non-spherical. This finding is not surprising and does not in itself indicate a problem with the prediction strength criterion. After all, k-means clustering assumes that the groups in the data are roughly sphered with the same variance, and if this assumption is violated, we should not be surprised if a stability-based criterion fails to identify the correct number of groups.

Chapter 3

METHODS

We first describe how we construct a cluster tree making only "significant" splits based on a random partition of the data set X into two half samples X_{train} and X_{test}. We then propose a method of "averaging" these trees over a number of random partitions. We assume that the reader is familiar with generalized single linkage clustering (Stuetzle and Nugent 2010).

3.1 Building a "confident tree" for a given partition

Step 1. Split the data set X into half-samples X_{train} and X_{test} (Fig. 3.1).

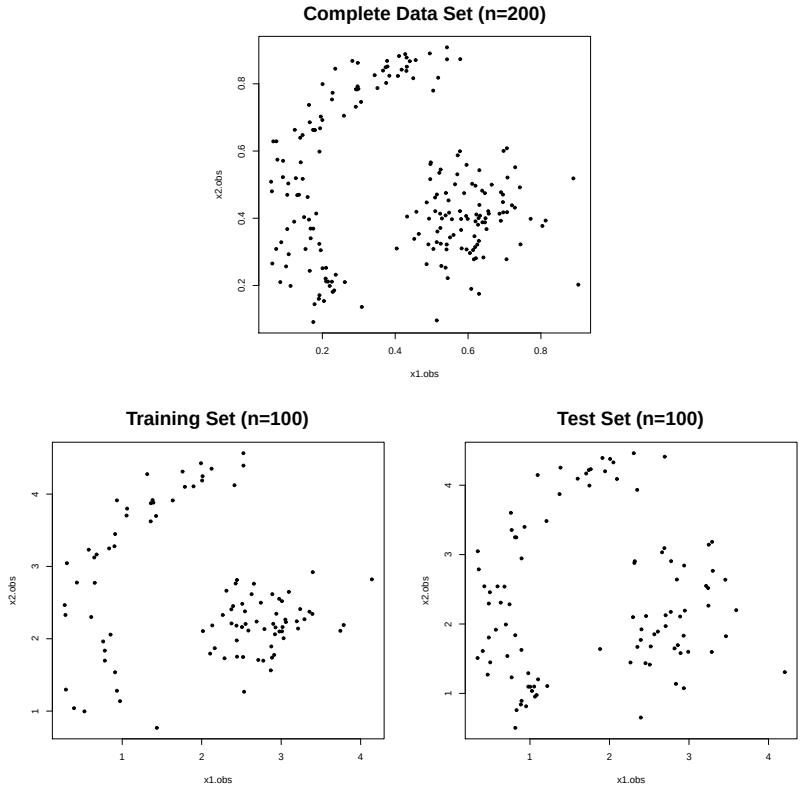


Figure 3.1: *(Top) Full Data, (Bottom Left) Training Data, (Bottom Right) Test Data*

Step 2. Apply generalized single linkage (GSL) to X_{train} and obtain a cluster tree T_{train} (Fig. 3.2). Each internal node of T_{train} corresponds to an edge of the maximum spanning tree (Fig. 3.2) which we call the "split edge".

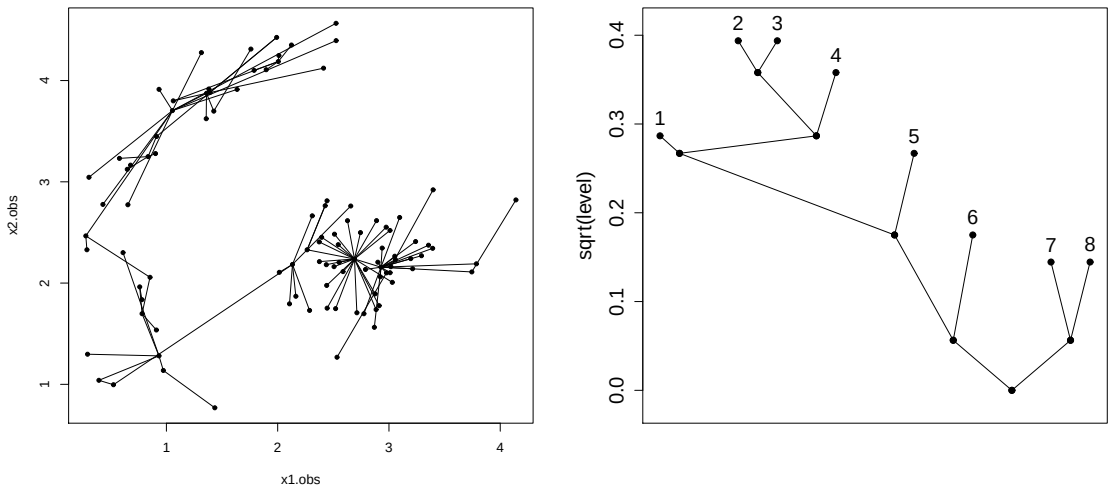


Figure 3.2: *(Left) Maximum Spanning Tree and (Right) Cluster Tree T_{train} based on Training Data*

Step 3. For each internal node in T_{train}, find split edge e, left cluster core c_l, right cluster core c_r and the corresponding x_{min}, x_l and x_r defined as

$$x_{min} = \arg\min_{x \in e} \hat{p}_{train}(x)$$
$$x_l = \arg\max_{x_i \in c_l} \hat{p}_{train}(x_i) \text{ (left mode)}$$
$$x_r = \arg\max_{x_i \in c_r} \hat{p}_{train}(x_i) \text{ (right mode)}$$

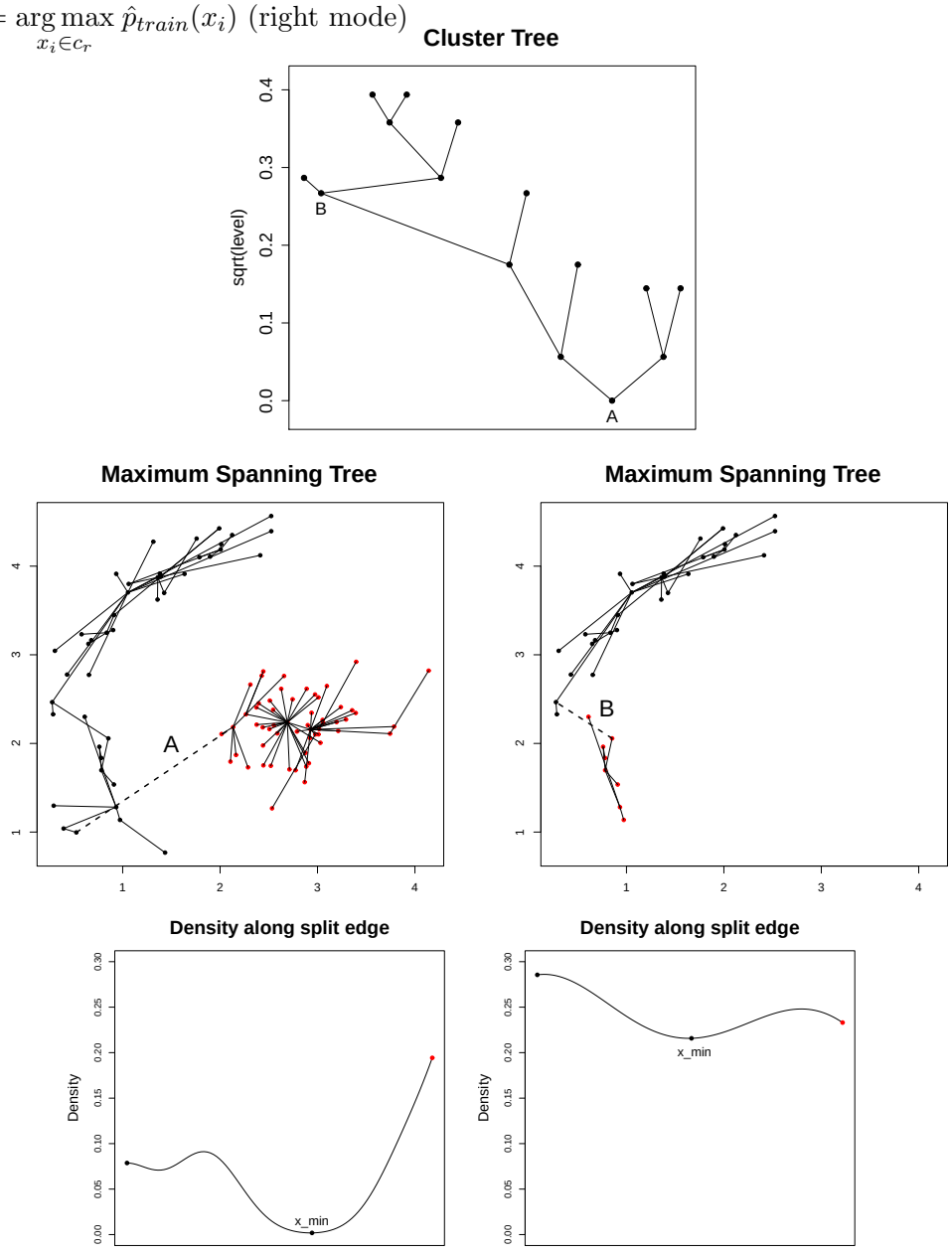


Figure 3.3: *(Top) Cluster tree T_{train} with Node A and B labelled, (Middle Left/Right) Dotted line shows the edge connecting the left and right cluster cores for split at Node A/B of cluster tree T_{train}, (Bottom Left/Right) Density $\hat{p}_{train}$ along the split edge*

Step 4. Draw n_{eval} half-samples from X_{test} (quarter-samples from data X) and obtain density estimates for these quarter samples $\hat{q}_1$, $\hat{q}_2$ ... $\hat{q}_{neval}$.

Step 5. Compute distribution (over n_{eval} quarter samples) of the statistic:

$$V_i = \log(\min(\hat{q}_i(x_l),\hat{q}_i(x_r)))\text{-}\log(\hat{q}_i(x_{min}))$$

for each of the internal nodes in T_{train}.

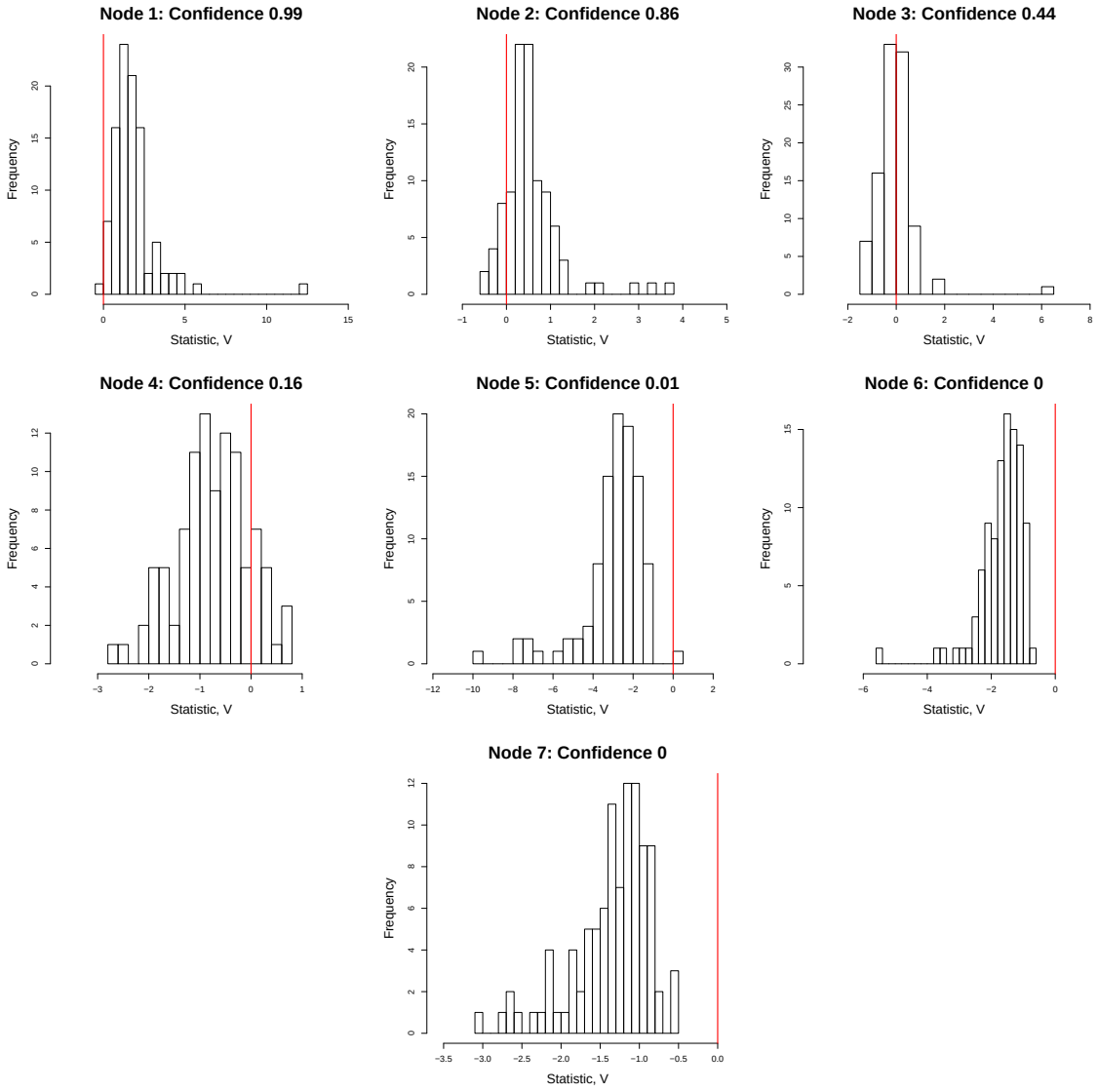


Figure 3.4: *Histograms of the statistic V for each internal node in sample Partition 1*

Step 6. Assign a "confidence level" defined as $c = P_{\hat{q}}(V{>}0)$ to each internal nodes of T_{train} (and the corresponding split edge of the maximum spanning tree M). We define the "significance" of a split as $1 - c$.

Step 7. Pick a confidence threshold and assign a binary label (significant or not) to each corresponding edge of the maximum spanning tree M (Fig. 3.5).

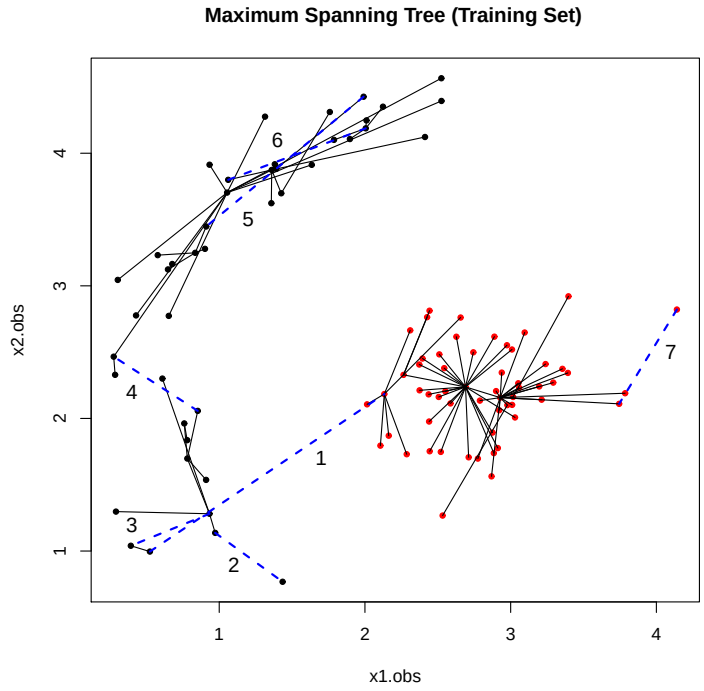


Figure 3.5: *Max Spanning Tree: Dotted lines show edges corresponding to internal nodes of cluster tree T_{train}. If a confidence threshold of 0.9 is chosen, based on the confidence measure, only edge 1 would correspond to a significant split.*

Step 8. Construct a new cluster tree T^*_{train} from the maximum spanning tree M allowing only splits of significant edges. First find the significant edge with the lowest edge weight. If there is no such edge, then the cluster tree consists only of the root node. Otherwise, break the edge, thereby splitting the maximum spanning tree into two segments; generate two daughters of the root node representing the segments; and recurse.

Step 9. Assign the observations in X_{test} to the nodes of T^*_{train} (Fig. 3.6) using a nearest neighbor classifier.

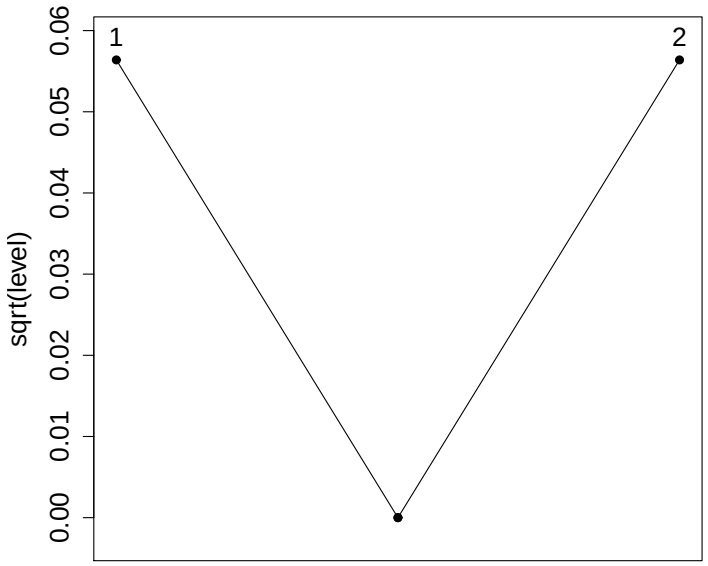


Figure 3.6: *New Cluster Tree* T^*_{train}

10. Construct a similarity matrix S for data X from the cluster tree T^*_{train}. The similarity matrix is a n by n matrix with S_{ij}=1 if X_i and X_j are in the same leaf of T^*_{train} and S_{ij}=0 otherwise.

3.2 "Averaging" over partitions

Step 11. Repeat Step 1 to 10 for n_{half} random partitions. For each partition, sort the confidence values for the splits in decreasing order. Let β_{ik} be the k^{th} largest confidence value for partition i. Table 3.1 shows the confidence values β_{ik} for 19 random partitions of our data set.

Splits	1	2	3	4	5	6	7	8	9	10	11	12	13	Significant Splits
Partition 1	0.99	0.86	0.44	0.16	0.01	0	0	-	-	-	-	-	-	1
Partition 2	1	0.49	0.14	0.1	0	0	-	-	-	-	-	-	-	1
Partition 3	1	0.33	0.14	0	0	-	-	-	-	-	-	-	-	1
Partition 4	0.98	0.6	0.14	0.11	0.1	0	-	-	-	-	-	-	-	1
Partition 5	1	0.24	0.23	0.19	0.04	0	0	-	-	-	-	-	-	1
Partition 6	1	0.34	0.2	0.06	0	0	0	-	-	-	-	-	-	1
Partition 7	1	0.15	0.12	0.09	0	-	-	-	-	-	-	-	-	1
Partition 8	1	0.79	0.47	0.37	0	0	0	0	-	-	-	-	-	1
Partition 9	1	0.48	0.44	0.02	0.02	0	-	-	-	-	-	-	-	1
Partition 10	1	0.49	0.44	0.28	0.26	0.09	0	-	-	-	-	-	-	1
Partition 11	1	0.46	0.27	0.2	0.15	0.1	0	-	-	-	-	-	-	1
Partition 12	1	0.53	0.41	0.38	0.26	0	-	-	-	-	-	-	-	1
Partition 13	0.99	0.43	0.33	0.1	0.08	0	0	-	-	-	-	-	-	1
Partition 14	0.98	0.46	0.27	0.17	0.06	0.02	0	0	-	-	-	-	-	1
Partition 15	1	0.65	0.07	0.01	0	0	-	-	-	-	-	-	-	1
Partition 16	1	0.6	0.38	0.17	0.05	0	-	-	-	-	-	-	-	1
Partition 17	0.94	0.59	0.17	0.08	0.03	0	0	0	0	-	-	-	-	1
Partition 18	0.99	0.83	0.22	0.08	0.07	0.02	0.01	0.01	0	0	0	0	0	1
Partition 19	1	0.34	0.2	0.18	0	0	-	-	-	-	-	-	-	1

Table 3.1: *Confidence of each node (sorted in decreasing order). The last column show the number of significant nodes using a confidence threshold of 0.9.*

Step 12. Given a confidence threshold, let k_i, $i = 1...n_{half}$ be the number of significant splits for partition i. Choose $\bar{k} = median(k_i)$ as the number of splits in the final output. For example, if a confidence threshold of 0.9 if chosen, all 19 partitions give one significant split, and therefore the final tree will have one split. However, if a confidence threshold of 0.4 is chosen, more than half of the 19 partitions give two significant splits, resulting in the median to be two significant splits.

Step 13. Calculate a similarity $\bar{S} = mean(S_i)$, where S_i is the similarity matrix obtained in Step 10, for partition i.

Step 14. Use GSL with similarity matrix $\bar{S}$ to create a cluster tree with $\bar{k}$ splits. This tree is the final output of the procedure.

Chapter 4

RESULTS

4.1 Simulation Scenarios

We generated $n_{rep} = 100$ independent realizations of data sets for each of 14 different scenarios:

Scenario 1. Uniform data in two dimensions: 200 data points uniformly distributed over the unit square in 2 dimensions (Fig. 4.1 *Left*).

Scenario 2. Standard gaussian data in two dimensions (Fig. 4.1 *Right*)

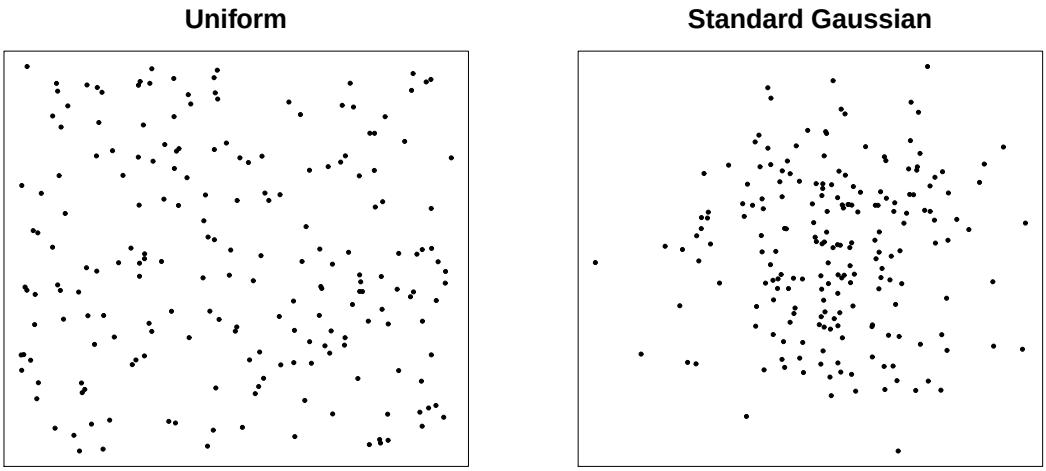


Figure 4.1: *(Left) Uniform Data, (Right) Standard Gaussian Data*

Scenarios 3-8. Bullseye data in two dimensions: 200 data points with parameter $fraction$ of data in the standard gaussian eye and the remaining data points in the ring made up of standard gaussians with centers uniformly chosen on the circle with parameter *radius* from the center of the eye. The parameter $fraction$ takes the values 0.25, 0.5 and 0.75 while the parameter *radius* takes the values 5 and 7. Abreviations are used to label these scenarios; for example, "b-f0.25-r5" identifies bullseye data with $fraction$ 0.25 of data in the eye with a *radius* of 5.

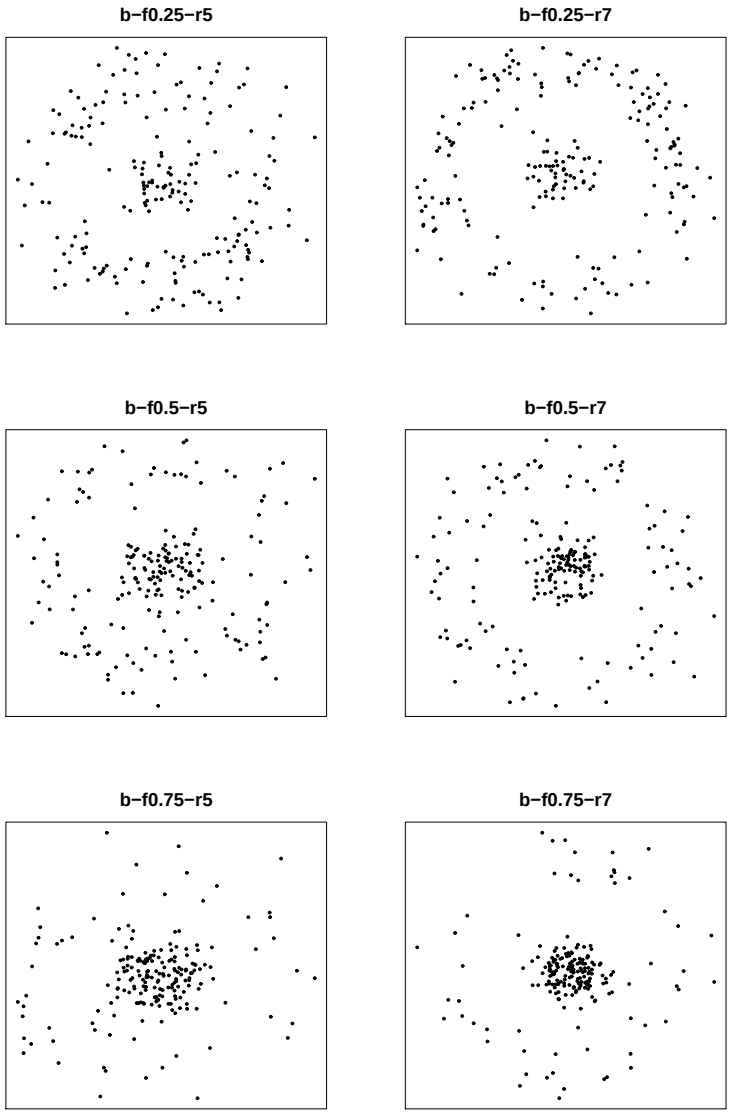


Figure 4.2: *Bullseye Data*

Scenarios 9-14. Bimodal standard gaussian data in two dimensions: 200 data points with parameter $fraction$ of data in the first standard gaussian and the remaining data points in the second standard gaussian, with center parameter *separation* away from the center of the first gaussian. The parameter $fraction$ takes the values 0.5, 0.7 and 0.9 while the parameter *separation* takes the values 4 and 7. Abreviations are used to label these scenarios; for example, "g-f0.7-s4" identifies a bimodal gaussian data with $fraction$ 0.7 of data in the first standard gaussian and a *separation* of 4 between the two standard gaussian centers.

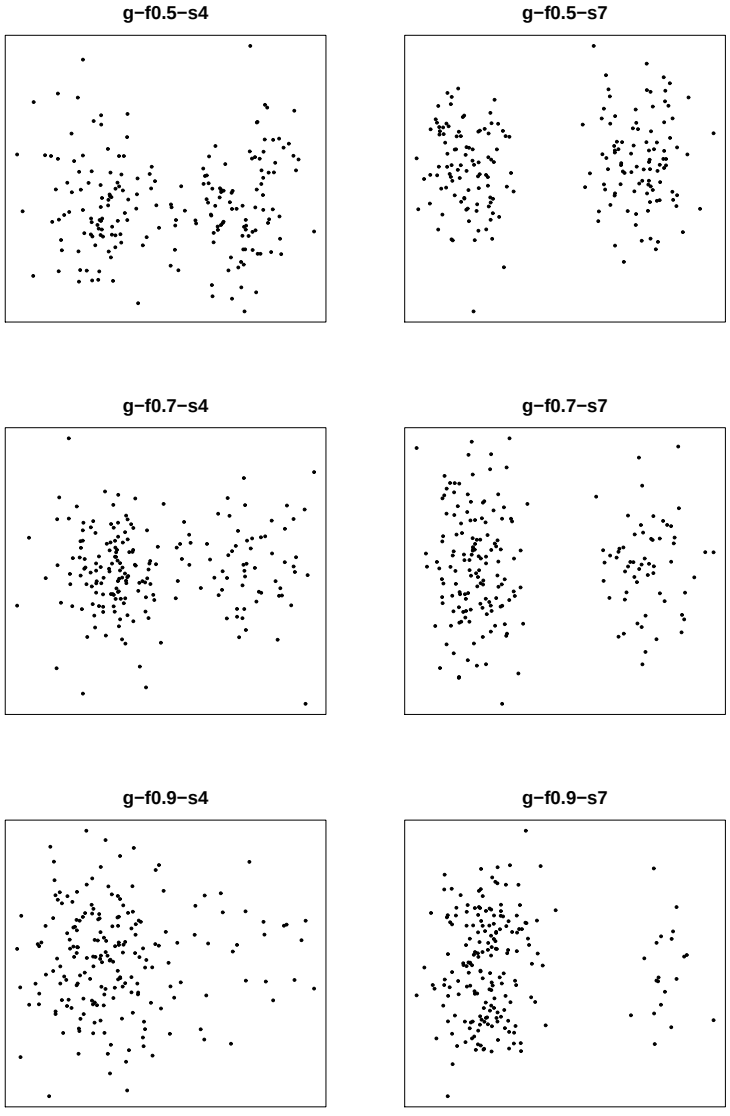
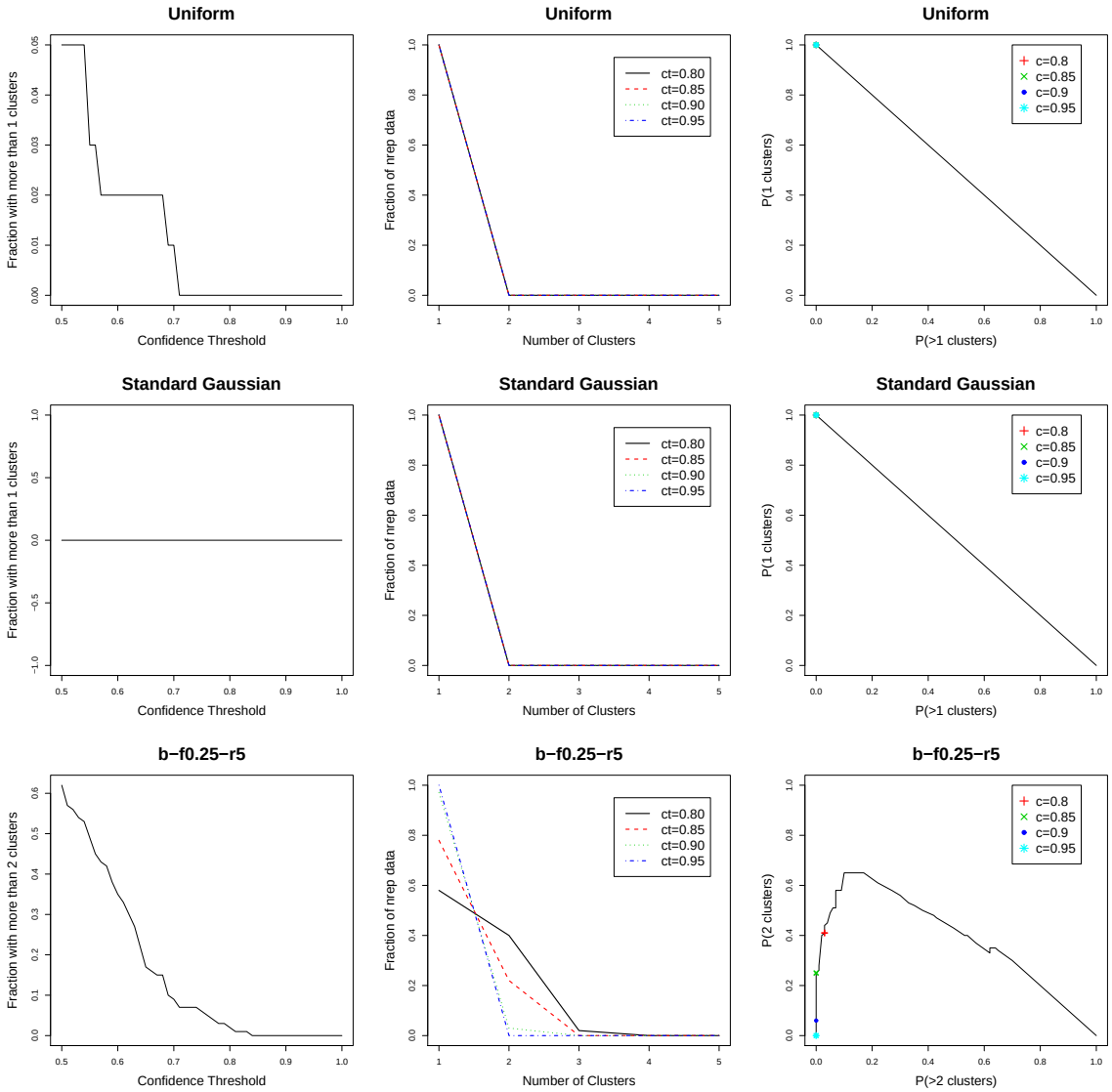
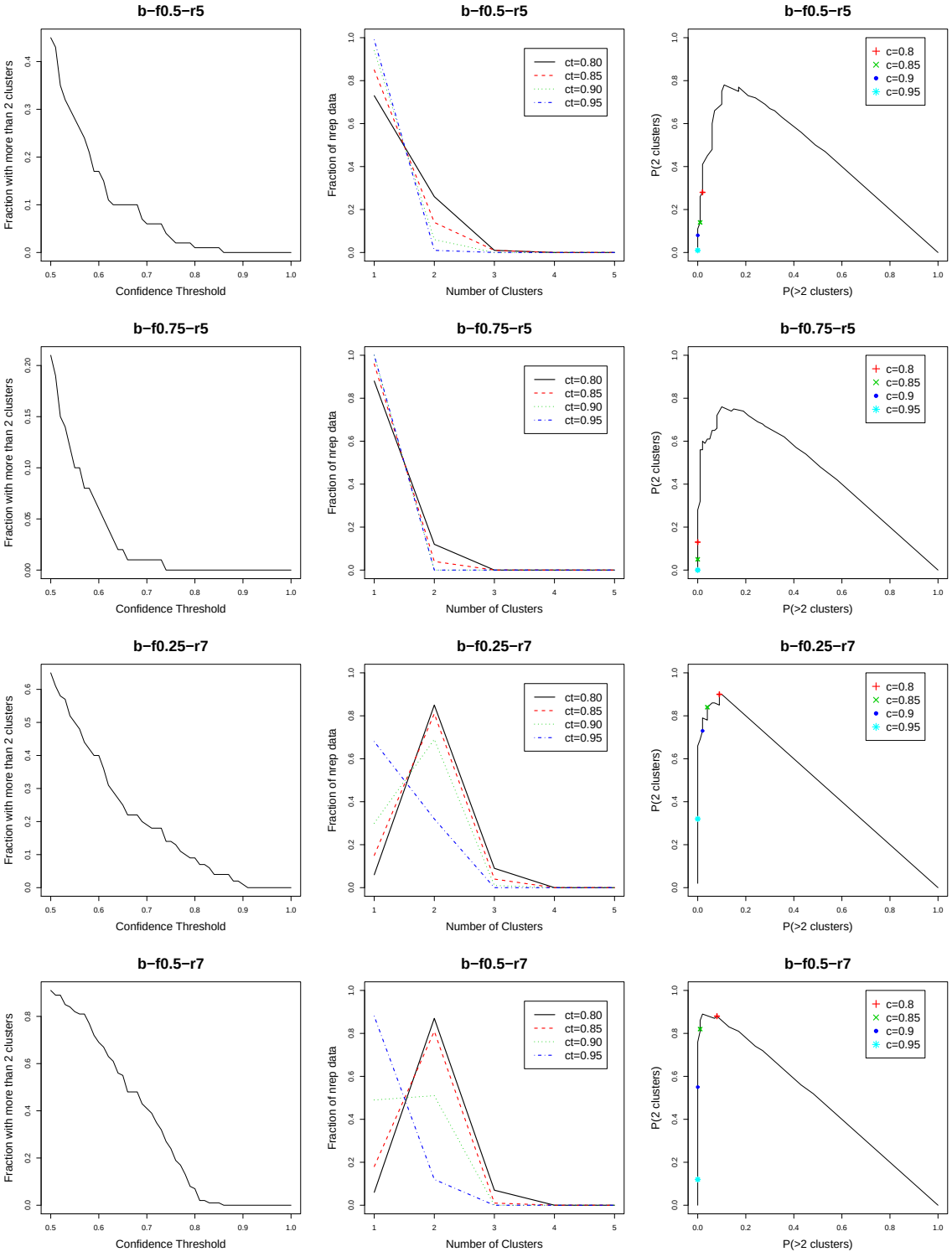


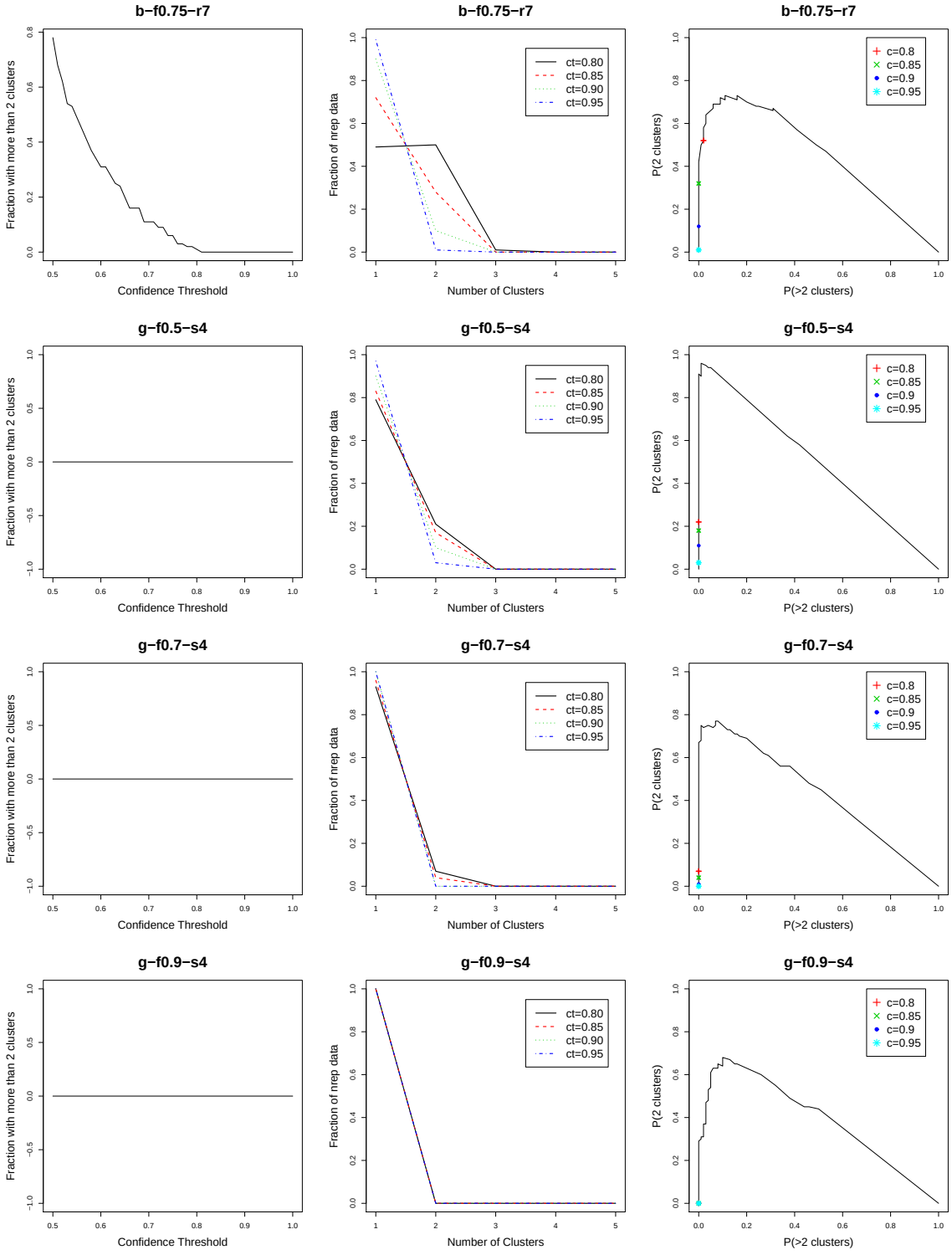
Figure 4.3: *Bimodal Standard Gaussian Data*

4.2 Simulation Results

For each of the 14 scenarios, we present 3 plots. First, the fraction of realizations with more than the correct number of clusters versus confidence threshold; second, the fraction of realizations versus the estimated number of clusters ; and third, the fraction of replicates with the correct number of clusters versus the fraction of replicates when the number of clusters over-estimates the number of groups, as a function of confidence level. The points corresponding to confidence levels 0.8, 0.85, 0.9 and 0.95 are marked on the curves.




b−f0.5−r5
Fraction with more than 2 clusters
Confidence Threshold
b−f0.5−r5
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
b−f0.5−r5
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
b−f0.75−r5
Fraction with more than 2 clusters
Confidence Threshold
b−f0.75−r5
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
b−f0.75−r5
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
b−f0.25−r7
Fraction with more than 2 clusters
Confidence Threshold
b−f0.25−r7
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
b−f0.25−r7
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
b−f0.5−r7
Fraction with more than 2 clusters
Confidence Threshold
b−f0.5−r7
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
b−f0.5−r7
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95


b−f0.75−r7
Fraction with more than 2 clusters
Confidence Threshold
b−f0.75−r7
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
b−f0.75−r7
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
g−f0.5−s4
Fraction with more than 2 clusters
Confidence Threshold
g−f0.5−s4
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
g−f0.5−s4
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
g−f0.7−s4
Fraction with more than 2 clusters
Confidence Threshold
g−f0.7−s4
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
g−f0.7−s4
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95
g−f0.9−s4
Fraction with more than 2 clusters
Confidence Threshold
g−f0.9−s4
Fraction of nrep data
Number of Clusters
ct=0.80
ct=0.85
ct=0.90
ct=0.95
g−f0.9−s4
P(2 clusters)
P(>2 clusters)
c=0.8
c=0.85
c=0.9
c=0.95

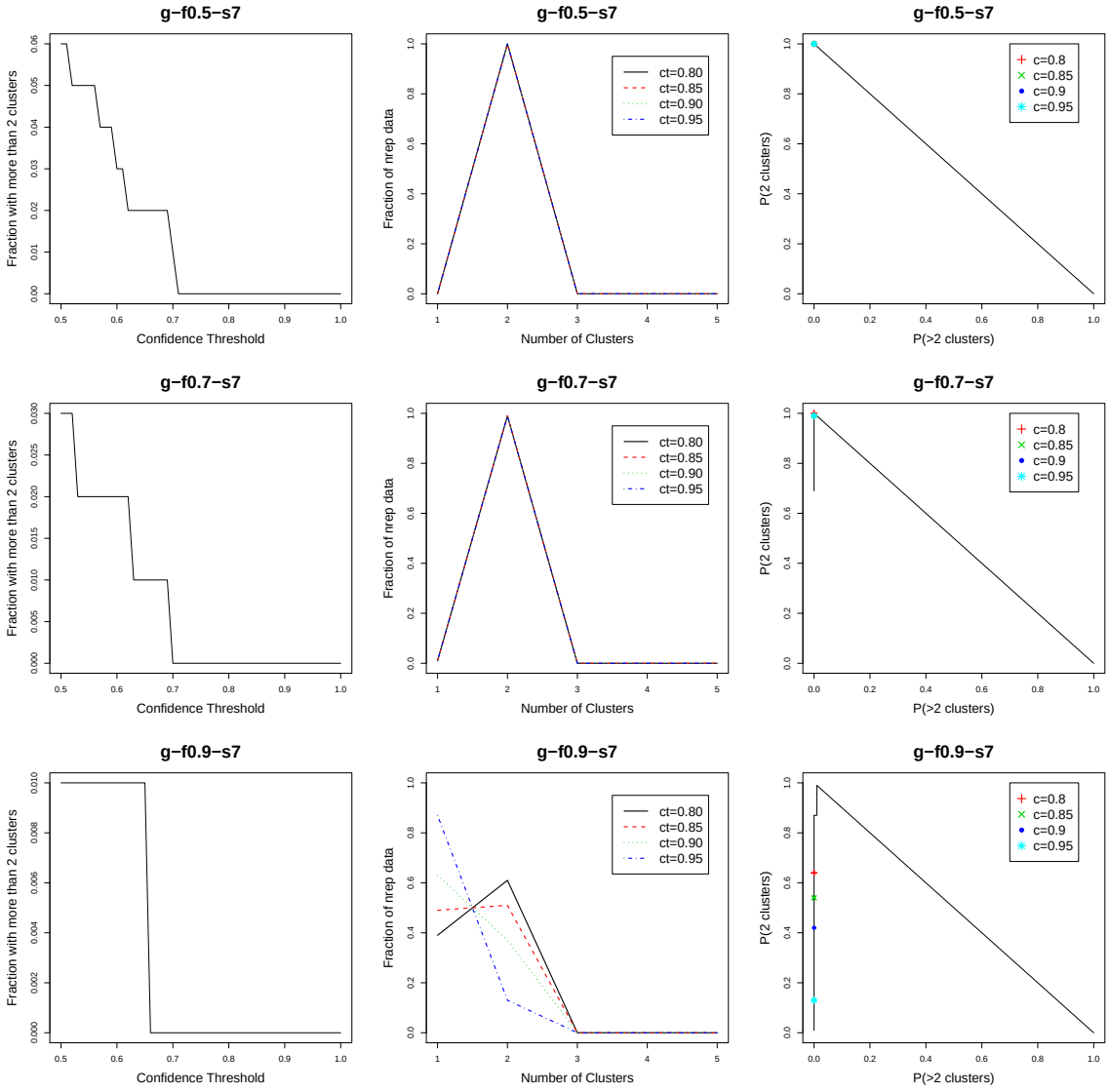


Figure 4.4: *(Left) Fraction of realizations with more than the correct number of clusters versus confidence threshold, (Center) Fraction of realizations versus the number of clusters estimated, (Right) Fraction of replicates with the correct number of clusters versus the fraction of replicates when the number of clusters over-estimates the number of groups*

4.3 Interpretation of Simulation Results

Our (admittedly very limited) simulation study suggests two hypotheses:

a) Our resampling method for estimating the number of groups is very conservative. The fraction of replications for which the method over-estimates the number of groups is consistently much lower than the nominal significance level.

b) This mis-calibration leads to the low power. Consider, for example the scenario "g-f0.5-s4". Even for significance level 0.2, the method over-estimates the number of groups for 0 out of 100 replications. It correctly identifies two groups only 20% of the time, while the corresponding calibration curve shows that, with correct calibration, it would produce the correct answer 80% of the time.

Chapter 5

SUMMARY/DISCUSSION

We have presented a clustering method that combines GSL with a resampling-based approach to determine the number of clusters. The only user input required is the desired significance threshold. Because it is based on GSL, our method can identify even highly non-linear (non-elliptical) groups. Our (admittedly limited) simulation study suggests that the method is very conservative: the fraction of replicates for which the method over-estimates the number of groups is much lower than the nominal significance. Correct calibration would result in a significant increase in power.

It is instructive to compare our method to the method of Burman and Polonik (2009). Both approaches are based on multivariate density estimation; both assume that distinct groups correspond to modes of the underlying density; and both consist of two main steps: a) finding modal candidates and b) checking if the modal candidates are separated by valleys. However, there is a fundamental difference in the way in which the two methods check for separation between modal candidates. The Burman and Polonik method looks at the estimated density along the line segment connecting the modal candidates whereas our method looks at the estimated density along the unique path in the maximum spanning tree connecting the modal candidates.

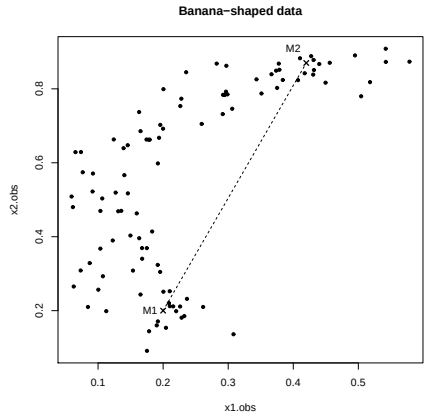


Figure 5.1: *Line segment connecting two spurious modal candidates*

Problems with the Burman and Polonik method can arise for non-convex groups. Consider a banana-shaped data set as in Figure 5.1. Spurious modal candidates occurring at the two ends of the banana create a line segment along which the estimated density has a significant dip, leading to the erroneous conclusion that there are 2 groups. Another difference to note is that Burman and Polonik uses asymptotics to assess the significance of a dip in density while we use a resampling approach.

There are several areas for future work. The most important one is to improve the calibration and thereby the power of the method. A more comprehensive simulation study would also be helpful.

BIBLIOGRAPHY

[1] Prabir Burman and Wolfgang Polonik. Multivariate mode hunting: Data analytic tools with measures of significance. *Journal of Multivariate Analysis*, 100:1198–1218, 2009.

[2] Rebecca Nugent and Werner Stuetzle. *Clustering with Confidence: A Low-Dimenstion Binning Approach*. Springer-Verlag Berlin Heidelberg, 2010.

[3] Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

[4] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via a gap statistic. *J. R. Statistical Society*, 63(2):411–423, 2001.

[5] Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.