Choosing the Number of Clusters in Monothetic Clustering

Tan V. Tran* Mark Greenwood*

Abstract

Monothetic clustering is a divisive clustering method based on recursive bipartitions of the data set determined by choosing splitting rules from any of the variables to conditionally optimally partition the multivariate responses. Like in other clustering methods, the choice of the number of clusters is important in this method. Connections between monothetic clustering and decision trees motivate the consideration of pruning methods as aids in selecting the number of clusters. We apply different cross-validation techniques to find the number of clusters that optimize prediction error and compare that approach to permutation-based hypothesis tests at each bi-splitting step, retaining splits with "small" p-values. A simulation study is performed to evaluate the performance of the new methods and compare to some other existing techniques.

Key Words: monothetic cluster analysis, non-parametric, cross-validation, number of clusters

1. Introduction

Clustering algorithms attempt to group subjects based on (multivariate) observations into clusters so that the dissimilarity within clusters is smallest while the between cluster dissimilarities are largest. Let y_{iq} be the i^{th} observation on variable q with $q = 1, \ldots, Q$, where Q is the number of response variables, $i = 1, \ldots, n$ where n is the sample size. We seek to divide the n objects into partitions $P_J = C_1, \ldots, C_J$, with J the number of clusters and C_j the j-th set of objects. One of the main tools for interpreting clusters is to describe the members of each cluster and so the choice of J impacts the group memberships and thus the interpretation of the results. If J is too small, it puts "unlike" subjects together. On the other hand, if J is too large, it would split observations that should be together. Picking the "correct" J is critical for any cluster analysis.

As a simple example to illustrate cluster analysis, we consider a data set introduced by Ruspini (1970) with 70 observations on two variables, x and y. The scatterplot of the data set is shown in Figure 1. Also, possible cluster solutions with different numbers of clusters can be seen in Figure 2. Visually, the two and four cluster solutions seem to be reasonable, but if five clusters are chosen, another solution can be given as in Figure 2c. Visually, for the two cluster solution, observations that have small x or small y will belong to one cluster, observations that have large x and y belong to another cluster. In the four cluster solution, the separation between clusters are still visually recognizable, but as we go into the five cluster case, the interpretation and even definition of clusters are less clear. Popular cluster analysis methods such as k-means and Ward's agglomerative hierarchical method can partition the data set into clusters, but they lack the capability to help researchers in interpreting the characteristics of each cluster and provide no clear way of predicting new observations. Monothetic cluster analysis (Chavent, 1998) provides at least partial solutions to both issues.

Here, we first describe the use of monothetic cluster analysis in Section 2. Next, we give an overview of some of well-known methods for choosing the number of clusters in a cluster analysis (Section 3) and introduce two novel methods that use the idea of cross-validation and permutation tests to pick the "correct" number of clusters (Section 4). Fi-

*Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717

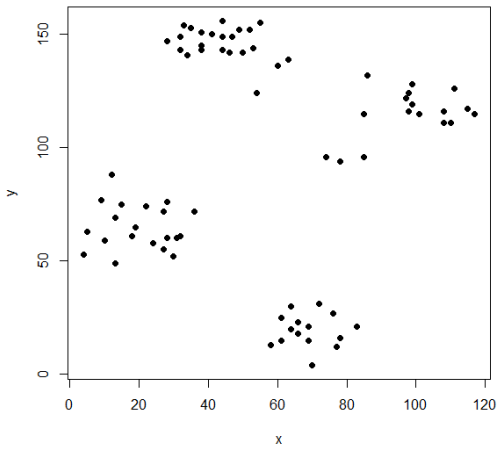
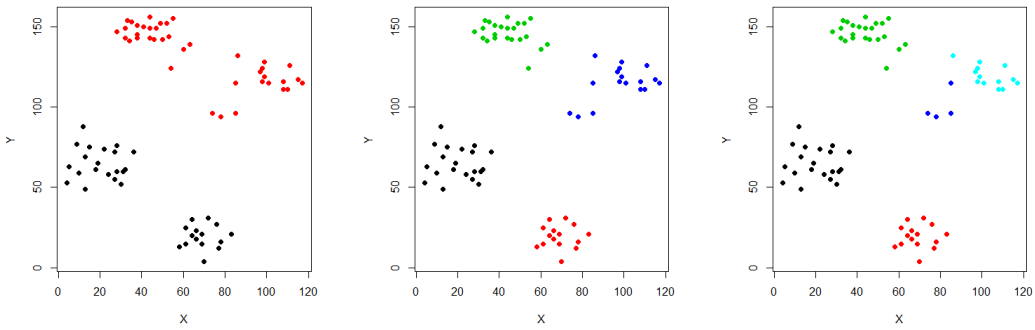


Figure 1: Ruspini data set



(a) Two clusters (b) Four clusters (c) Five clusters

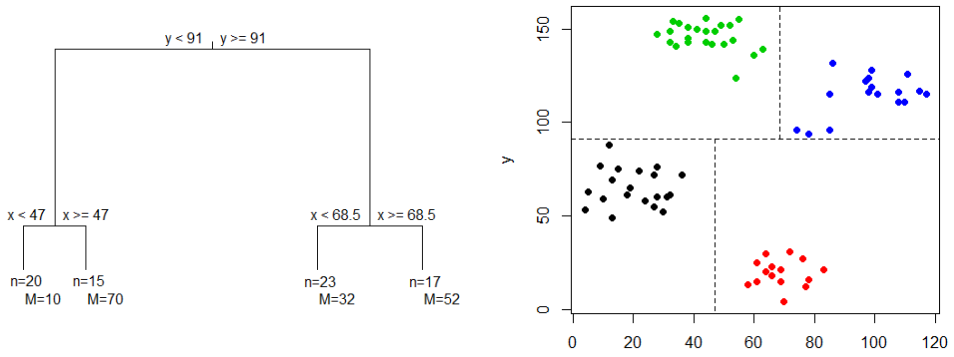
Figure 2: Different possible cluster solutions for Ruspini's data.

nally, in Section 5, a comparison among those methods is performed by simulation for different data set scenarios. Some conclusions are mentioned in Section 6.

2. Monothetic Cluster Analysis

Monothetic cluster analysis (Chavent, 1998) is an algorithm that provides a hierarchical, recursive partitioning of multivariate responses based on binary decision rules that are built from individual response variables. Inspired by regression trees (Breiman et al., 1984), the monothetic clustering algorithm searches for splits from each response variable that provide the best split of the multivariate responses in terms of a global criterion called inertia. It then recursively applies the same algorithm to each sub-partition, recording splitting rules to define the tree. The result of the algorithm is a set of hierarchical binary rules for determining cluster membership. Therefore the resulting hierarchy can be read and displayed as a decision tree.

Specifically, the within cluster inertia is the total variability around the cluster centroid.



(a) Decision tree

(b) Plot with splitting rules

Figure 3: Binary partitioning tree with maps of optimal splits. Three splits, four clusters.

In the case of Euclidean distance being used, the inertia for cluster j would be

$$I(C_j) = \sum_{i \in \text{Cluster}_j} \sum_{q=1}^{Q} (y_{iq} - \overline{y}_{\cdot q})^2. \tag{1}$$

The objective of the algorithm when splitting cluster C is to maximize the difference in inertia between C and the new sub-partition C_1 and C_2,

$$\max \left\{ I(C) - I(C_1) - I(C_2) \right\}, \tag{2}$$

which is then recursively applied to each sub-partition. James et al. (2013) and others have shown that the Euclidean distances for all observations within a cluster and variation around the means within a cluster are equivalent, so the within cluster inertia can be equivalently calculated from the dissimilarity matrix as

$$I(C_j) = \frac{1}{n} \sum_{i \in \text{Cluster}_j} \sum_{j=i+1} d_{ij}^2. \tag{3}$$

This result is used to justify the application of many distance-based methods to non-Euclidean dissimilarities (Anderson, 2001). While not used here, the monothetic clustering algorithm can also be directly applied to non-Euclidean distances and other dissimilarities.

Figure 3 shows an example of the splitting rules and the created clusters for the Ruspini data set using monothetic clustering. The monothetic clustering algorithm suggests the first split at the y-value of 91. In the newly created cluster that includes the data points that had $y > 91$, the algorithm suggest splitting at the x-value of 68.5, while for observations with lower y-values, it suggests partitioning at the x-value of 47. The set of splitting rules can be summarized in hierarchical tree form as shown in Figure 3a with each rule being generated from one variable at a time. The rules can be used to generate an interpretation of the four selected clusters as: Cluster 1 includes observations that have $y < 91$ and $x < 47$, while observations having $y < 91$ and $x > 47$ belong to cluster 2. Similarly, having $y > 91$ and $x < 68.5$ are the characteristics of the observations in cluster 3, and the rest of the data set that has large y (> 91) and x (> 47) belongs to cluster 4.

Compared to classical methods like Ward's hierarchical clustering or k-means, the benefit of monothetic clustering is the ability to interpret the clusters and predict for new observations by exploiting the decision tree. However, the partitioning based on one variable

at a time is not as flexible as some other algorithms, limiting its performance for complicated data structures and cluster shapes. Chavent et al. (2007) suggested that monothetic clustering should be used in studies where the interpretation of only a few clusters is the focus.

3. Some Popular Metrics for Choosing Number of Clusters

Many metrics for choosing the number of clusters have been mentioned in the clustering literature, such as a paper on the comparison of popular metrics by Milligan and Cooper (1985), and also implemented in popular statistical software such as the package `NbClust` (Charrad et al., 2014) in R (R Core Team, 2015). However, there is no universally "good" metric for all clustering problems or algorithms even though some are more popular than others. For the purpose of this paper, we chose two popular metrics that have good performance and are suitable for monothetic clustering to compare to our new approaches.

3.1 Average silhouette width

One common measure is the average silhouette width (Rousseeuw, 1987). The silhouette width is a measure of how "comfortable" an observation is in the cluster it resides in. Let $a(i)$ be the average dissimilarity between observation i and other observations in the same cluster, $d(i)_k$ be the average dissimilarity between i and other observations in cluster k, and $b(i) = \min_k(d(i)_k)$ be minimum "distance" from i to other clusters, then the silhouette width is defined to be

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}. \tag{4}$$

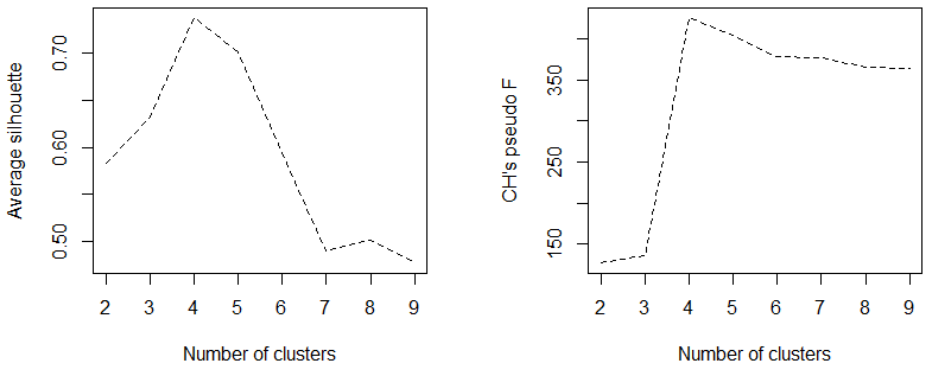
$s(i)$ can obtain the values from -1 to 1 corresponding to the state of observation i in its cluster. The recommended interpretation is that if the silhouette width is between 0 and 1 it is "happiest" in its existing cluster; if it is 0, the observation is ambivalent about cluster membership vs. next closest cluster; and if it is between -1 and 0 the observation "wants to leave" the current cluster.

A global measure for a cluster solution is found by averaging all n silhouette widths, defining the average silhouette width as

$$\bar{s} = \frac{\sum_{i=1}^{n} s(i)}{n}. \tag{5}$$

The cluster structure with J clusters that has the maximum average silhouette width will be considered as the "optimal" structure. Kaufman and Rousseeuw (1990) suggested that unusual observations be removed from the average silhouette width calculation as they had too much impact on it.

Average silhouette width is explicitly recommended by Kaufman and Rousseeuw (1990) for selecting the number of clusters in their Partitioning Around Medoids (PAM) algorithm. It can be applied to any cluster solution if the dissimilarity matrix and cluster memberships are available. Although average silhouette width is a clear criterion for choosing the number of clusters in a clustering problem, it has a major limitation in that it cannot select a single cluster solution because it is not defined on $J = 1$. In practice, large average silhouette width values for $J = 2$ are often observed when no real clusters exist in the data set, making its use for selecting $J = 1$ or $J = 2$ problematic.



(a) Average silhouette: selects 4 clusters (b) CH's pseudo-F: selects 4 clusters

Figure 4: The choice of clusters for the Ruspini data made by Average silhouette width and CH's pseudo-F methods. Both of them suggest the use of the $J = 4$ cluster solution.

3.2 Caliński and Harabasz (CH)'s pseudo-F

Caliński and Harabasz (1974) proposed the use of the idea of an F-statistic as a criterion to choose the number of clusters, J, in order to maximize the variation between clusters relative to the variation within clusters. Hence, their pseudo-F can be calculated as

$$\text{pseudo}-F = \frac{B(J)/(J-1)}{W(J)/(n-J)} \tag{6}$$

where $B(J)$ is the between cluster sums of squares (possibly from dissimilarity matrix using Equation 3) and $W(J)$ is the within cluster sums of squares which can also be found as a result of Equation 3 from a dissimilarity matrix. Because the pseudo-F is the ratio of the variance of the groups to the variance in the residuals, the setting of J clusters is considered good when the observations are similar within groups (small $W(J)$) but different between groups (large $B(J)$). However, like the average silhouette metric, the pseudo-F needs at least two clusters to be calculated so it cannot select a single cluster solution and often shows large values for $J = 2$ when only one cluster is present.

In Figure 4, the average silhouette width and CH's pseudo-F methods are applied to the Ruspini data to find the "optimal" number of clusters. In both methods, the criteria agree with each other and reach their maxima at $J = 4$, suggesting choosing the four cluster solution for this data set.

4. Proposed Methods for Choosing the Number of Clusters

Average silhouette width and pseudo-F are simple criteria for deciding on the optimal number of clusters but they both show the limitation of being unable to ever select a single cluster structure. Moreover, with the tree-based splitting rules of monothetic clustering resembling those in regression trees, popular methods inspired from regression and classification trees are possible to consider. Two methods which are explored in this paper are an adaptation of a cross-validation technique and using the pseudo-F to perform a permutation test.

4.1 M-fold Cross-Validation

Cross-validation (CV) is a popular method to "tune" many methods (see Hastie et al., 2009). In regression and classification trees, the size of tree is decided by a pruning algorithm in which a too complicated tree is firstly built, then cross-validation is used to prune the tree to a smaller size that balances fit and complexity of the tree. The M-fold cross-validation starts by randomly dividing the data set into M equal-sized subsets. A subset ($1/M$ of observations) then is withheld as the validating set, and the rest of the observations are used as the training set. In monothetic cluster analysis, each training set can be used to build the monothetic clustering splitting rule tree. Since the withheld observations are not used in the splitting process, the mean squared error of the observed y_{iq} and predicted responses $\hat{y}_{(-i)q}$ (which are the cluster means on the variable q from training data when Euclidean distance is used) for the withheld set provides an approximately unbiased estimate for the test error for the m-th subset as

$$MSE_m = \frac{1}{n_m} \sum_{q=1}^{Q} \sum_{i \in m} (y_{iq} - \hat{y}_{(-i)q})^2. \tag{7}$$

This process is repeated for the M subsets of the data set and the average of these test errors is the cross-validation-based estimate of the mean squared error of predicting a new observation. For a J cluster solution, the overall cross-validation based estimate is

$$CV_J = \frac{1}{M} \sum_{m=1}^{M} MSE_m. \tag{8}$$

Comparing CV_J for different values of J can be used in different ways to pick a solution. The smallest CV_J among them can be used as a choosing criterion (***minCV*** rule). Another approach is using the ***1SE*** rule (Breiman et al., 1984) in which the solution is the simplest one within 1 standard error of the minimum, where the standard error (SE) is

$$SE = \sqrt{\frac{1}{M-1} \sum_{i=1}^{M} (MSE_i - \overline{MSE})^2}. \tag{9}$$

Both approaches are used in regression or classification tree pruning, with the ***1SE*** rule typically giving more conservative tree size selection.

Because monothetic clustering is one of the few clustering algorithms that provides a clear cluster prediction rule (at least with Euclidean distance), we can use the binary rules to assign a new observation to a cluster and use the multivariate cluster mean to predict the response to find the cross-validation error estimate. Extensions of these ideas to other metrics or dissimilarities have not been developed as far as we know.

An example of the criteria for the Ruspini data is in Figure 5. The prediction mean squared error (CV_J) decreases when the number of clusters J increases. Ideally, we should see CV_J decrease and then increase, showing a clear choice of J that minimizes prediction error. In this case, the number of clusters explored was not large enough to observe an increase in CV_J, if it occurs. That is the reason why the ***minCV*** rule suggested the maximum number of clusters considered, which was 9. When adding the amount of 1 standard error to this minimum CV_J, it creates a region that covers solutions of 4 to 9 clusters. Because 4 is the smallest cluster solution in the region, it is the solution suggested by the ***1SE*** rule. We can see that ***1SE*** rule is much more conservative than the ***minCV*** rule, and it often picks the solution at or near the "elbow" in the CV_J plot.

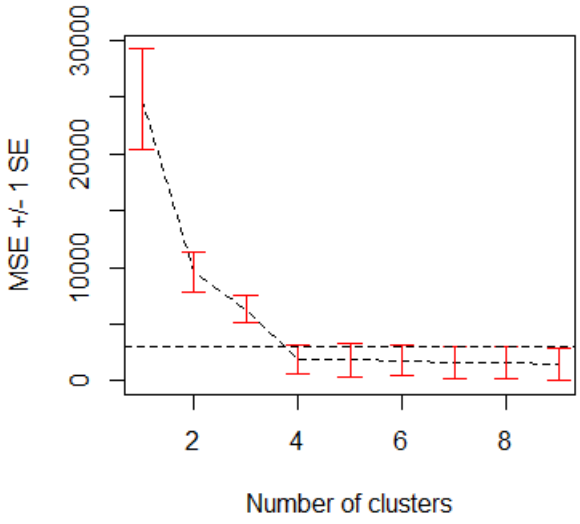


Figure 5: The choice of clusters for Ruspini data made by 10-fold CV where *minCV* selects 9 clusters and *1SE* selects 4.

Because the mean squared error can be calculated for $J = 1$, the cross-validation method can compare between the solution of one cluster vs. more than one cluster. However, the method cannot be applied for any non-Euclidean distance because the form of $\hat{y}_{(-i)q}$ is unclear with other metrics. Additionally, repeatedly building a new clustering structure for each validating subset and each J can be computationally expensive for large n or large Q.

4.2 Hypothesis tests at each bipartition

Each bipartition creates two distinct groups of observations. The need to split can be assessed using hypothesis testing with the null hypothesis being no difference in the two groups at the node, that the split should not be performed. The idea of a formal hypothesis test for trees using the p-value to stop tree growth has been used in the context of conditional inference trees (Hothorn et al., 2006). The tree is grown until a split has a p-value higher than a pre-determined threshold (say α of 0.01 or 0.05) and then that split or any other sub-divisions are not considered for that node. This threshold is continually adjusted when the split goes further down the tree to account for inflated Type I error rates as sequences of tests are combined.

In monothetic clustering, some adjustments need to be made to the process to account for the differences in the situation from those encountered by Hothorn et al. (2006). To allow applications with any dissimilarity measure, a nonparametric method based on a permutation test is used. Anderson (2001) developed a multivariate nonparametric testing approach called perMANOVA that involves calculating the pseudo-F-ratio directly from any symmetric distance or dissimilarity matrix using Equation 6 where the sum of squares are calculated directly from the dissimilarities (Equation 3). The p-value can then be calculated by tracking the pseudo-F across permutations and comparing the results to our observed result.

This p-value also needs to be adjusted to account for multiple hypothesis tests required

to reach a node further down in the tree. For example, in Figure 3, the hypothesis test at the second node to create clusters 1 and 2 (x is optimally cut at 47) is conditional on the hypothesis test at their father node (where y was cut at 91 having rejected the null hypothesis). The probability of a Type I error on at least one of these two tests is inflated unless we control for the accumulating number of tests. This probability is getting higher and higher as the depth in the tree grows. A simple solution to control the family-wise error rate for a set of tests is using the Bonferroni-adjustment which involves multiplying the p-value by the number of tests required to get to that level of tree. Specifically, we use

$$p - value_{adj} = \text{depth} \times \frac{\text{No. of } (F_{perm} \geq F_{observed})}{\text{No. of perm.}}, \tag{10}$$

where the depth is calculated as the height of the tree with the root node at depth = 1.

One other complication that arises in these tests is that in monothetic clustering, the split is based on the variable that has the maximum decrease in the difference inertia between the big cluster and two new clusters (Equation 2). This algorithm ensures that the splitting is the "best" possible choice at that node. This affects the permutation-based hypothesis test by creating smaller p-values than desired because the chosen split is already the most extreme result possible on the variable that defined the bipartition and creates another source of inflated Type I error rates in this situation. We suggest a potential solution by using only variation from the $Q - 1$ variables not used to define the candidate split in the calculation of the pseudo-F statistic. Then the hypothesis test will assess whether the binary split is useful on other variables by assessing the differences between the two groups on the other $Q - 1$ variables. This modification means the test cannot be performed in a data set that has only one variable ($Q = 1$).

An example of the results for the Ruspini data is in Figure 6. Each circle is a bipartition of the data set. In the first four partitions, the p-values from the permutation-based hypothesis tests are small (< 0.05), suggesting that all of the null hypotheses of no difference in the true multivariate group means should be rejected. However, at the fifth split (data are further split at a x-value of 45) it results in a very large adjusted p-value ($p = 1$), indicating that the split should not be made. Visually, in the cluster which has observations having $y > 91$ and $x < 68.5$, a partition at $x = 45$ looks like a slice in the middle of the points. If we consider the difference between two new partitions in terms of the y-values only, the observations are basically the same. So in this case, the hypothesis testing method suggests the five cluster solution.

The advantage of using this criterion in picking the cluster solution is that it can choose the one cluster solution. The test can be performed at the root of the decision tree to decide if the data set should be partitioned at all. It can also be applied in non-Euclidean settings. Another advantage is the test can be embedded into the tree building procedure, so that the test is performed every time a split is about to be made. With this combination, the cluster analysis algorithm stops when the predefined significance level is crossed, hence making the algorithm more computationally efficient. Nevertheless, the removal of the splitting variable would create a trade-off when one variable is dominant in deciding how the data set should be split. The test will return a large p-value for this test because the two new clusters may not be very different after removing the dominant variable. It may depend on the application to assess whether this is an advantage or limitation of this method.

5. Simulation Study

In order to see how the different criteria work for different types of data, we set up a simulation study including three scenarios with different numbers of clusters and varying

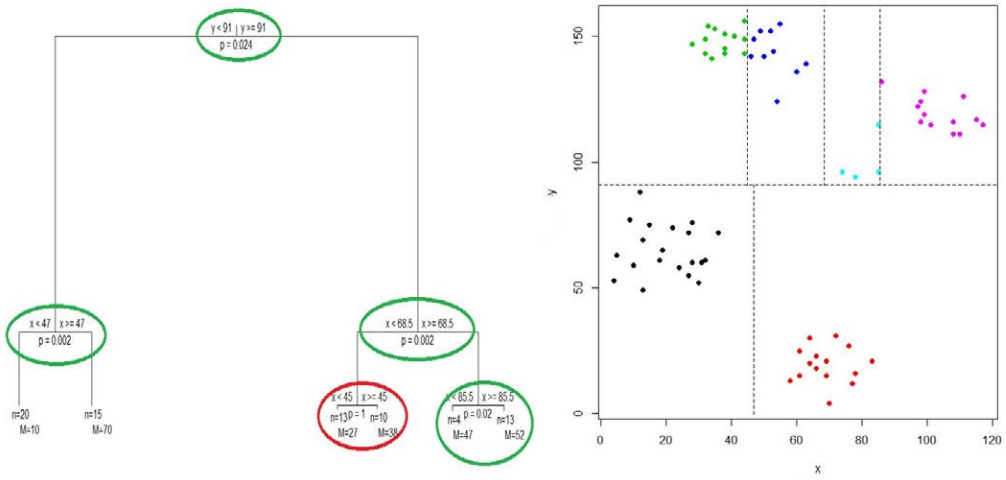


Figure 6: The choice of clusters for Ruspini data by hypothesis testing where 5 clusters are selected.

dimensions.

5.1 Simulation Set Up

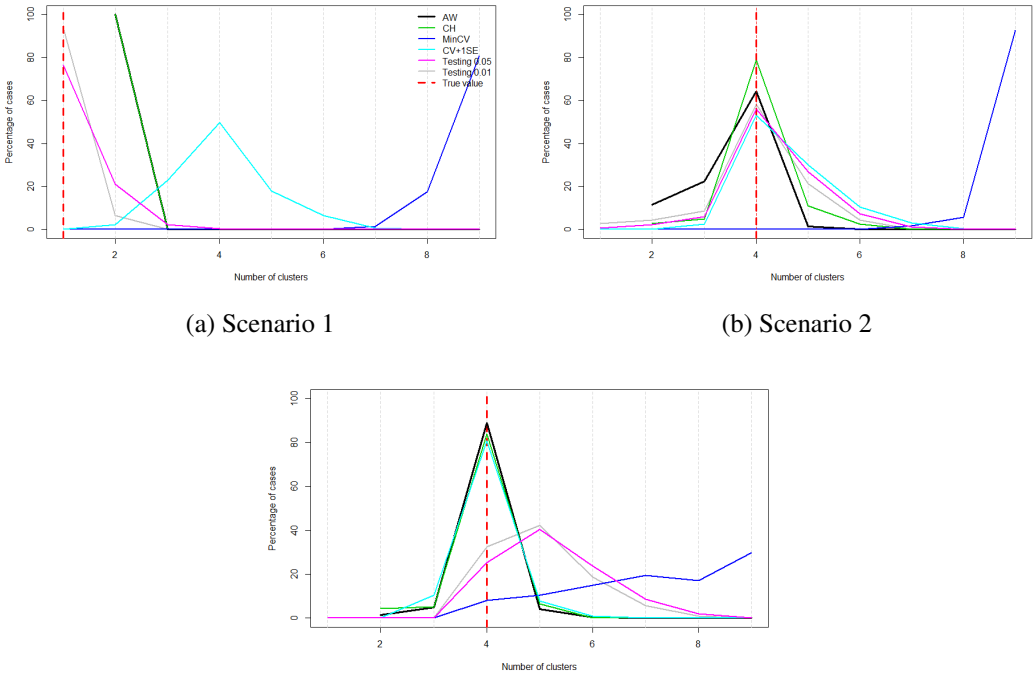
The data for the simulation study were created following three different settings, inspired by Tibshirani et al. (2001) and Walesiak and Dudek (2014).

1. The null model in 10 dimensions is generated with observations that are uniformly distributed within the range (-1, 1) in 10 dimensions. This is a special case with one big cluster.

2. The random 4-cluster model in 3 dimensions is generated with four clusters that have standard normal distributions. The mean values are sampled from a N(0; 5**I**) distribution and the cluster sizes are randomly chosen from either 25 or 50 observations. Every data set in which the distance between any pairs of clusters was less than 2 was discarded from the simulations. This is a difficult case, because the clusters are not well-separated (the smallest Euclidean distance between points in any two clusters is only 2 units).

3. The random 4-cluster model in 10 dimensions is generated with four clusters that have standard normal distributions. The mean values are sampled from a N(0; 1.9**I**) distribution and the cluster sizes are randomly chosen from either 25 or 50 observations. Similar to the previous case, every data set in which the distance between any pairs of clusters are less than 2 was discarded from the testing data. This is an even more difficult case than the first two. Although the minimum distance is the same as the previous case, increasing the dimensions makes the points in different clusters closer, because the distance is the sum over all dimensions.

In each setting, 500 data sets were created from each scenario. Each method was run on the same simulated data sets, with permutation-based hypothesis testing run with two different significance levels, $\alpha = 0.01$ and 0.05. The M-fold CV method was run with $M = 10$, and the minimum and one standard error criteria were applied to choose the optimal



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 7: Simulation results

solution. The maximum size of the monothetic cluster tree was set to be 9 clusters. This result should be far enough from the true number of clusters in all cases to be considered over-fitting.

Along with these methods, average silhouette width and CH's pseudo-F was run using the existing methods in the `cluster` package (Maechler et al., 2015). We wrote R code to implement the two new proposed methods which is available by request. The permutation tests are done using the `vegan` package (Oksanen et al., 2015).

Computational performance was tracked duting the simulation study and hypothesis testing is much faster than the CV process. The explanation is that the algorithm stops immediately when the terminal node adjusted p-values exceed the pre-defined significance level. The cross-validation method is slower because the monothetic cluster analysis needs to be built 10 times for each size of cluster solution (10-fold CV), and all size solutions need to be examined to pick the optimal one on the criteria. However, the differences in time are not prohibitive to the use of CV unless large Q or n are encountered.

5.2 Results of The Simulation Study

The results of simulation study with the number of cases that each method chose in each simulation scenario are summarized in Table **??** and Figure 7. The two traditional methods, average silhouette width and CH's pseudo-F, did quite well in selecting the correct number of clusters when there was actually more than one true cluster. Their correct detection rates were high and similar. Both methods were not capable of detecting the solution of one cluster and chose two clusters when one was correct, making any detection of two clusters with these methods suspicious.

In the first scenario with only one true cluster in 10 dimensions, only the hypothesis testing method works well. Particularly, when $\alpha = 0.01$, this method was quite successful

Table 1: Estimated number of clusters by different methods.
Unit: percent of time (out of 500). label

Setting	Estimates of number of clusters								
	1	2	3	4	5	6	7	8	9
Null model in 10 dimensions									
Average silhouette width	NA[a]	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CH's pseudo-F	NA[a]	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MinCV	**0.0**	0.0	0.0	0.0	0.0	0.2	1.4	17.6	80.8
CV + 1SE	**0.0**	2.2	22.8	49.8	17.8	6.6	0.6	0.2	0.0
Hypothesis testing ($\alpha = 0.01$)	**93.6**	6.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hypothesis testing ($\alpha = 0.05$)	**76.4**	21.0	2.2	0.4	0.0	0.0	0.0	0.0	0.0
Random 4-cluster model in 3 dimensions									
Average silhouette width	NA[a]	11.6	22.4	**64.4**	1.4	0.2	0.0	0.0	0.0
CH's pseudo-F	NA[a]	2.8	4.8	**79.0**	11.0	2.4	0.0	0.0	0.0
MinCV	0.0	0.0	0.0	**0.0**	0.0	0.0	1.6	5.8	92.6
CV + 1SE	0.0	0.0	2.6	**53.2**	30.2	10.6	3.0	0.4	0.0
Hypothesis testing ($\alpha = 0.01$)	2.8	4.4	8.6	**58.2**	21.4	4.4	0.2	0.0	0.0
Hypothesis testing ($\alpha = 0.05$)	0.6	2.2	6.0	**56.0**	26.8	7.2	1.2	0.0	0.0
Random 4-cluster model in 10 dimensions									
Average silhouette width	NA[a]	1.4	5.0	**89.0**	4.2	0.4	0.0	0.0	0.0
CH's pseudo-F	NA[a]	4.4	5.2	**83.8**	6.6	0.0	0.0	0.0	0.0
MinCV	0.0	0.0	0.0	**8.0**	10.6	15.0	19.4	17.2	29.8
CV + 1SE	0.0	0.2	10.4	**80.6**	7.8	1.0	0.0	0.0	0.0
Hypothesis testing ($\alpha = 0.01$)	0.0	0.0	0.0	**32.6**	42.2	18.8	5.6	0.8	0.0
Hypothesis testing ($\alpha = 0.05$)	0.0	0.0	0.0	**25.4**	40.4	23.6	8.6	2.0	0.0

[a] The smallest number of clusters chosen cannot be 1 for these methods.

at correctly preventing a search past the root node with the $J = 1$ solution rejected 6% of the time for this level. The inflated Type I error rates might come from assessing splits that are selected to be optimal globally from given choices even when no real difference exists. If monothetic clustering did not optimize a global criterion, this would not occur. This error rate was much higher when all variables were included in the test statistic (results not presented here) and these early results led to the modified test statistic used here. For a larger number of dimensions, Q, the hypothesis testing method struggled but the more conservative α value did provide more correct results.

In all scenarios, the *minCV* value was often the largest number of clusters considered. 9 clusters were picked more than 80% of the time in the null model and the 4-cluster, $Q = 3$ dimensions scenarios, and more than 30% in the $Q = 10$, 4-cluster scenario. This result suggests that the criterion does not provide much penalty for over-fitting and that it is not suitable as a criterion for choosing the number of clusters.

The *1SE* rule with CV is much better than *minCV* and close to other methods in performance except in *Scenario 1*. Some mistakes in *Scenarios 2* and *3* might be due to the variability in dividing the data into 10-fold validating and training data sets. In a particular application, the CV selection process could be repeated and a consensus of results selected which would decrease the variability due to random variation in validation-split membership.

6. Remarks

Some methods, such as CH's pseudo-F and average silhouette width, because of a limitation in their definition, cannot select the one cluster solution so they can't help a researcher to decide between one versus more than one cluster. In this regard, both new methods, M-fold cross validation and permutation-based hypothesis testing, are attractive as they provide direct information about the utility of a single cluster versus other sizes, but they also have their weaknesses. The hypothesis testing method works well in data sets with a relatively small number of dimensions but suffer as the dimension grows. Both hypothesis testing and cross-validation have some amount of randomness in their algorithms, but with a large number of permutations, the resulting p-value from hypothesis testing tends to be consistent, while the variability in the results of cross-validation is more serious and may be impacting its performance.

The speed of the monothetic clustering algorithm to build the clusters for a pre-defined size from a data set depends largely on the number of dimensions and, to a lesser degree, the sample size. Because of those reasons, cross-validation is the slowest among considered methods because it involves M re-fittings of the monothetic cluster solution for every potential cluster size. Conversely, permutation-based hypothesis testing is quite fast due to the speed of the permutation code and because only a part of a single monothetic cluster solution may be required for a run.

Whenever there are new observations, the prediction of their cluster along with its characteristics can be easily traced down from the created rules in the tree. If needed, the CV criterion can be used to estimate the prediction error. If the cross-validation based methods are to be used, the *1SE* criterion should be preferred. In order to overcome the randomness in the method to produce a more consistent result, multiple runs of cross-validation splitting process should be used to identify the most commonly selected optimal cluster size.

With the above remarks, we suggest an algorithm for finding the number of clusters in monothetic clustering using a combination of methods. First, the permutation test could be used to assess evidence for more than one cluster using the modified pseudo-F statistic and a small significance level, α. If the initial p-value is small enough to reject the null

hypothesis that the two groups are not different, then CH's pseudo-F maximization can be used to find the correct size of the cluster solution from two on.

Acknowledgments

This work was supported by a sabbatical for Greenwood in Fall 2014 and travel support from the College of Letters and Sciences and Department of Mathematical Sciences at Montana State University. Tran's Ph.D. program at Montana State University is sponsored by the Vietnam Education Foundation.

References

Anderson, M. J. A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26(1):32–46, February 2001. ISSN 14429985. doi: 10.1111/j.1442-9993.2001.01070.pp.x.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. *Classification and Regression Trees*. Chapman and Hall/CRC, 1 edition, 1984. ISBN 0412048418.

Caliński, T. and Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, June 1974.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. NbClust : An R Package for Determining the. *Journal of Statistical Software*, 61(6), 2014.

Chavent, M. A monothetic clustering method. *Pattern Recognition Letters*, 19(11):989–996, September 1998. ISSN 01678655. doi: 10.1016/S0167-8655(98)00087-7.

Chavent, M., Lechevallier, Y., and Briant, O. DIVCLUS-T: A monothetic divisive hierarchical clustering method. *Computational Statistics & Data Analysis*, 52(2):687–701, October 2007. ISSN 01679473. doi: 10.1016/j.csda.2007.03.013.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

Hothorn, T., Hornik, K., and Zeileis, A. Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, September 2006. ISSN 1061-8600. doi: 10.1198/106186006X133933.

James, G., Witten, D., Hasite, T., and Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*. Springer, 1st edition, 2013. ISBN 1461471370.

Kaufman, L. and Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 1 edition, 1990. ISBN 978-0471735786.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. *cluster: Cluster Analysis Basics and Extensions*, 2015. R package version 2.0.3.

Milligan, G. W. and Cooper, M. C. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, June 1985. ISSN 0033-3123. doi: 10.1007/BF02294245.

Oksanen, J., Blanchet, F. G., Kindt, R., Legendre, P., Minchin, P. R., O'Hara, R. B., Simpson, G. L., Solymos, P., Stevens, M. H. H., and Wagner, H. *vegan: Community Ecology Package*, 2015. R package version 2.3-1.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987. ISSN 03770427. doi: 10.1016/0377-0427(87)90125-7.

Ruspini, E. H. Numerical methods for fuzzy clustering. *Information Sciences*, 2(3):319–350, July 1970. ISSN 00200255. doi: 10.1016/S0020-0255(70)80056-1.

Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, May 2001. ISSN 1369-7412. doi: 10.1111/1467-9868.00293.

Walesiak, M. and Dudek, A. *clusterSim: Searching for optimal clustering procedure for a data set*, 2014. R package version 0.43-4.