

Model Based and Hybrid Clustering of Large Datasets

Jeremy Tantrum

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

Doctor of Philosophy

University of Washington

2003

Program Authorized to Offer Degree: Statistics

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Jeremy Tantrum

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Co-Chairs of Supervisory Committee:

Alejandro Murua

Werner Stuetzle

Reading Committee:

Doug Martin

Alejandro Murua

Werner Stuetzle

Date:

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Model Based and Hybrid Clustering of Large Datasets

Jeremy Tantrum

Co-Chairs of Supervisory Committee:

Assistant Professor Alejandro Murua

Statistics

Professor Werner Stuetzle

Statistics

The goal of clustering is to identify distinct groups in a dataset. The basic idea of model based clustering is to approximate the data density by a mixture model, typically a mixture of Gaussians. The number of distinct groups in the data is then taken to be the number of mixture components, and the observations are partitioned into clusters (estimates of the groups) using Bayes' rule.

Mixture models can be estimated using the EM algorithm; however in order to be successful the EM algorithm requires good initial cluster assignments for the observations. Such assignments can be obtained by hierarchical model based clustering. The problem is that model based hierarchical clustering is only practical for at most a few thousand observations. We review an idea called Fractionation, originally conceived by Cutting, Karger, Pedersen and Tukey for non-parametric hierarchical clustering of large datasets, and describe an adaptation to model based clustering. A further extension, called Refractionation, leads to a procedure that can be successful even in the difficult situation where there are large numbers of small groups.

If the groups in the data are well separated and look Gaussian, then model based clustering will produce clusters that indeed tend to be “distinct” in the most common sense of the word – contiguous, densely populated areas of feature space, separated by contiguous, relatively empty regions. If the groups are not Gaussian or if the covariance structure of the mixture model is incorrect, this correspondence may break down; an isolated group with a non-elliptical distribution, for example, may be modeled by not one, but several mixture components, and the corresponding clusters will no longer be well separated. We present methods for assessing the degree of separation between the components of a mixture model and between the corresponding clusters. We also propose a Hybrid Clustering algorithm that combines some of the advantages of non-parametric and model based clustering. The algorithm starts with the hierarchical clustering tree corresponding to the mixture model chosen by the Bayesian Information Criterion. It then progressively merges clusters that do not appear to correspond to different modes of the data density.

TABLE OF CONTENTS

List of Figures	v
List of Tables	x
Chapter 1: Introduction	1
Chapter 2: Converting Documents Into Vectors	4
2.1 Term Document Matrix	4
2.2 Frequency Transformation and Term Weighting	4
2.3 Dimensionality Reduction	6
2.3.1 Principal Component Analysis	7
2.3.2 Latent Semantic Indexing	8
Chapter 3: Model Based Clustering	10
3.1 Parsimonious Mixture Models	10
3.2 Hierarchical Model Based Clustering	11
3.2.1 Clustering Singletons	12
3.2.2 Computing the Likelihood Distance for Clusters with Diagonal Co- variance Matrices	13
3.3 Choosing Between Models	14

3.4	Model Selection with Cost Complexity Pruning	15
3.4.1	Definition of Cost Complexity Pruning	15
3.4.2	Bottom Up Recombination	16
3.4.3	Equality of Optimally Terminated Subtree Sequence and Hierarchical Clustering Subtree Sequence	16
Chapter 4:	Application to the TDT data	19
4.1	The Topic Detection and Tracking Corpus	19
4.2	The Effects of Frequency Transformation, Term Weighting and Dimension- ality on the Performance of a Document Classifier	24
4.2.1	Design of The Experiment	24
4.2.2	Results	25
4.3	Comparing Two Partitions	27
4.3.1	Fowlkes-Mallow Index	27
4.3.2	The Adjusted Rand Index	27
4.3.3	The F1 Index	28
4.4	The Effects of Term Weighting and Dimensionality on the Performance of Model Based Document Clustering	29
4.5	Using the BIC for Estimating the Number of Topics	30
4.5.1	Gaussian Data with Diagonal Covariances	30
4.5.2	Gaussian Data with Non-Diagonal Covariances	31
4.5.3	Resampling from a Kernel Density Estimate	32
4.5.4	Summary	32

4.6	Clustering the “1100 TDT” Collection	32
Chapter 5:	Fractionation	35
5.1	Model Based Fractionation	37
5.2	Illustration	37
Chapter 6:	Model Based Refractionation	40
6.1	Illustration	42
6.2	Properties of Fractionation and Refractionation	43
Chapter 7:	Fractionation and Refractionation Examples	49
7.1	Fractionation Example 1	49
7.2	Fractionation Example 2	50
7.3	Refractionation Example 1	50
7.4	Refractionation Example 2	53
7.5	Clustering the entire TDT Data Set	54
Chapter 8:	Assessment and Visualization	59
8.1	Assessing separation between mixture components	60
8.1.1	Assessing separation using posterior probabilities	61
8.1.2	Assessing separation using margins	62
8.1.3	Assessing separation using misclassification probabilities	62
8.2	Assessing separation between clusters	64

Chapter 9:	Hybrid Clustering	65
9.1	The Hybrid Clustering Algorithm	65
9.2	Illustration of hybrid clustering	66
9.3	Testing for unimodality	67
9.4	Remarks	70
Chapter 10:	Assessment and Hybrid Clustering Examples	71
10.1	Diagnostics and Hybrid Clustering of the Olive Oil Data	71
10.2	Diagnostics and Hybrid Clustering of Simulated “Olive Oil Data”	73
10.3	Diagnostics and Hybrid Clustering for the “1100 TDT” Collection	75
10.4	Diagnostics and Hybrid Clustering of the Full TDT Data Set	77
10.4.1	Pruning	78
10.4.2	Validation of Clusters	79
10.5	Clustering Uniform data in 100 dimensions	79
Chapter 11:	Discussion and Future Work	90
11.1	Discussion	90
11.2	Future Work	91
11.2.1	Massive Data Sets with Many Clusters	91
11.2.2	Mixtures of Factor Analyzers	92
Bibliography		93

LIST OF FIGURES

4.1	Projection of the labeled TDT data onto principal components 1 and 4. . . .	20
4.2	XGobi projection of the labeled TDT, showing topic 15.	21
4.3	XGobi projection of the labeled TDT, showing topic 8.	22
4.4	XGobi projection of the labeled TDT, showing topic 11.	22
4.5	Performance of the transformation/weighting combinations as a function of dimensionality	25
4.6	Performance of the transformation/weighting combinations as a function of dimensionality (log-log scale).	26
4.7	Performance of weighting schemes on hierarchical model based clustering as a function of dimensionality. Confidence intervals for the curves are shown as solid bands.	30
4.8	Two way contingency table of topics vs clusters for the “1100 TDT” collec- tion. Topic labels are shown on the vertical axis and mixture components are shown on the horizontal axis.	33
5.1	Fractionation Algorithm	36
5.2	The data with each fraction shown in a different color.	38
5.3	The data from each fraction with the meta-observations superimposed.	38
5.4	All 40 meta-observations and the final 4 clusters chosen by the BIC.	39

6.1	Refractionation Algorithm	41
6.2	Observations and with fraction membership shown in color.	43
6.3	Meta-observations obtained by clustering the initial four fractions.	44
6.4	Clusters chosen by the BIC after the first pass of Fractionation.	44
6.5	Fractions formed by the first pass of Fractionation.	45
6.6	Meta-observations obtained by clustering the four fractions in the second pass of Fractionation.	45
6.7	Clusters chosen by the BIC after the second pass of Fractionation.	46
6.8	Fractions formed by the second pass of Fractionation.	46
6.9	Meta-observations obtained by clustering the four fractions in the third pass of Fractionation.	47
6.10	Clusters chosen by the BIC after the third pass of Fractionation.	47
7.1	Fowlkes-Mallows index vs number of clusters chosen by the BIC for the data set of Fractionation Example 1.	50
7.2	Fowlkes-Mallows index vs number of clusters chosen by the BIC for the data set of Fractionation Example 2.	51
7.3	Fowlkes-Mallows index vs number of clusters chosen by the BIC for Refrac- tionation Example 2.	54
8.1	Data set with fitted Gaussian mixture. The modes of the mixture are indi- cated by the three white dots. (This example is referred to as the running example in the remainder of the thesis.)	60

8.2	Running example: Rootograms of the posterior probabilities $P(Y = g X)$ for X distributed according to the mixture model.	62
8.3	Running example: Cumulative distribution function of the margin.	63
8.4	Running example: Rootograms of the posterior probabilities $P(Y = g \mathbf{x}_i)$ for the data.	64
9.1	Running example: Tree generated by hierarchical model based clustering and diagnostic plot for the circled node.	67
9.2	Running example: Tree generated by hierarchical model based clustering after first step of pruning, and diagnostic plot for the circled node.	68
9.3	Running example: Tree generated by hierarchical model based clustering after second step of pruning, and diagnostic plot for the circled node.	69
9.4	Illustration of the DIP statistic.	69
10.1	Olive oil data: Original tree (all nodes) and pruned tree (dark nodes).	72
10.2	Two way contingency table of areas vs pruned clusters for the olive oil data before and after pruning. The 9 areas are shown on the vertical axis and mixture components are shown on the horizontal axis.	73
10.3	Olive oil data: Histograms of posterior probabilities $P(Y = g \mathbf{x}_i)$ for the data, before (a) and after (b) pruning.	74
10.4	Olive oil data: Misclassification probabilities MC_g for the 28 components of the mixture model, before (left) and after (right) pruning.	75
10.5	Pruned node of olive oil tree	75
10.6	Non pruned node of olive oil tree	76

10.7 Olive oil data: cumulative distribution function of the margins before pruning (black line) and after pruning (grey line).	76
10.8 Simulated olive oil data: Original tree (all nodes) and pruned tree (black nodes).	77
10.9 Simulated olive oil data: Two way contingency table of areas versus clusters before pruning.	78
10.10 Simulated olive oil data: Two way contingency table of areas versus clusters after pruning.	79
10.11 Clustering tree for the “1100 TDT” collection. The colored leaves correspond to topics which are split across several mixture components.	83
10.12 Histograms of posterior probabilities $P(Y = g \mathbf{x}_i)$ for the “1100 TDT” col- lection, before pruning	84
10.13 Cumulative distribution function of the margin of the model for the “1100 TDT” collection on the log scale.	84
10.14 Two way contingency table of topics vs pruned clusters for the “1100 TDT” collection. Topic labels are shown on the vertical axis and mixture compo- nents are shown on the horizontal axis.	85
10.15 Clustering tree for the full TDT data with pruned leaves shown in grey. . . .	86
10.16 Histograms of posterior probabilities $P(Y = g \mathbf{x}_i)$ for the full TDT data, before pruning.	86
10.17 Cumulative distribution function of the margin of the model for the full TDT data on the log scale.	87
10.18 Clustering Tree for the Uniform data with pruned leaves shown in grey. . . .	87

10.19	Histograms of posterior probabilities $P(Y = g \mathbf{x}_i)$ for the Uniform data, before pruning	88
10.20	Misclassification probabilities for each of the mixture components of the model fitted to the Uniform data.	88
10.21	Cumulative distribution function of the margin of the models for the Uniform data.	89
10.22	GKL-Silverman and GKL-DIP plots for the full TDT data split on the root node.	89

LIST OF TABLES

4.1	The principal component direction which correspond to topics. High values of the 25th direction corresponds to topic 10 and low values corresponds to both topics 1 and 13. Only part of topic 11 is separated along component 19.	20
4.2	The TDT Corpus Topics	23
4.3	Comparison between clusters (rows) and groups (columns). The cell count n_{ij} is the number of observations in cluster i and group j	28
4.4	Number of clusters chosen by the BIC when the model is correct. The true number of groups is 19.	31
4.5	Number of clusters chosen by the BIC for data simulated from a mixture of factor analyzers. The true number of groups is 19.	31
4.6	Number of clusters chosen by the BIC for non-diagonal Gaussian data – 19 is the true number of groups.	32
6.1	The distribution of the number of fractions across which groups are scattered at the start of each Fractionation pass.	43
7.1	Refractionation Example 1 – agreement between clusters and groups after each Fractionation pass.	52
7.2	Refractionation Example 1 – distribution of the number of fractions in which groups are represented, at the start of each Fractionation pass.	52

7.3	Refractionation Example 1 – distribution of the number of groups represented in each fraction at the start of each Fractionation pass. Here n_f is the number of fractions and the last column is the minimum of the average number of groups per fraction.	52
7.4	Improvement of clustering and change in clusters for the 15,863 TDT documents when using time for the original fractioning. The first column shows the Fowlkes-Mallows index associated to consecutive clusterings generated by consecutive refractionation passes. The last column refers to the FM1 index associated to the labeled data.	57
7.5	Improvement of clustering and change in clusters for the 15,863 TDT documents when using a random assignment for the initial fractions. The first column shows the Fowlkes-Mallows index comparing clusterings generated by consecutive refractionation passes. The last column shows the FM1 index for the labeled data.	58
8.1	Misclassification matrix for the running example.	63
10.1	Document Headlines for Cluster 33	80
10.2	Document Headlines for Cluster 72	81

ACKNOWLEDGMENTS

I would like to thank and acknowledge my advisors Alejandro Murua and Werner Stuetzle for their advice and support during the past six years. Werner's most notable saying is, "*grad school is the best time of your life*". Alejandro was of immeasurable support while I was starting out in research, and Werner has spent many many hours fine tuning both papers and this thesis. I would also like to thank the other members of my committee, Susan Dumais of Microsoft Research, Doug Martin, and Adrian Raftery for their insightful comments and assistance. Chris Fraley also has been most helpful in providing insight into model based clustering. I am grateful to the National Security Agency which has funded me for most of my degree under contracts 62-1942 and 62-2948.

I would like to thank my fellow graduate students for their support and camaraderie during my time in the Statistics department. I would also like to thank the members of the Graduate Chapter of InterVarsity Christian Fellowship for their support, encouragement and prayers.

I would especially like to thank the faculty of the University of Auckland who not only taught me statistics for my undergraduate degree, but encouraged me to pursue a graduate degree at the University of Washington. I would also like to thank my parents and grandparents for their enabling me to maintain contact with New Zealand while being in a foreign country.

My wife Barbara has been a great support and encouragement during these last two years of my degree (and first two of our marriage). Her support has made this foreign country a little less foreign.

Lastly, I would like to thank the Lord God Almighty who has sustained me through both the trials and the triumphs that constitute a graduate program.

Chapter 1

INTRODUCTION

This thesis is about clustering. The goal of clustering is to identify distinct groups in a dataset – contiguous, densely populated areas of feature space separated by contiguous, relatively empty regions (Carmichael, George, and Julius 1968).

To cast clustering as a statistical problem we regard the data $\mathbf{x}_1, \dots, \mathbf{x}_n$ as a sample from some unknown probability density $p(\mathbf{x})$. There are two statistical approaches to clustering. *Non-parametric clustering* (Wishart 1969; Hartigan 1981, 1985; Wong and Lane 1983; Rozal and Hartigan 1994; Ester, Kriegel, Sander, and Xu 1996; Ankerst, Breuning, Kriegel, and Sander 1999; Stuetzle 2003) is based on the premise that groups correspond to modes of the density $p(\mathbf{x})$. The goal then is to estimate the modes and assign each observation to the “domain of attraction” of a mode. In contrast, *model based clustering* (McLachlan and Basford 1988; Banfield and Raftery 1993; McLachlan and Peel 2000, and references therein; Fraley and Raftery 2002) assumes that each group g is represented by a density $p_g(\mathbf{x})$ that is a member of some parametric family, such as the multivariate normal family. The density $p(\mathbf{x})$ then is a mixture of the group densities. The parameters of the mixture components as well as their number can be estimated from the data. The ability to estimate the number of groups is an important strength of the model based approach. There is, as yet, no comparable method for non-parametric clustering in more than one dimension. The advantage of non-parametric clustering is that it makes no assumptions about the shapes of the groups. In this thesis we focus on model based clustering; a brief introduction is presented in Chapter 3.

While the techniques presented in this thesis are useful in many application areas, our

main motivation is document clustering, a particularly challenging problem due to high dimensionality. Document clustering has many uses, such as improving the presentation of search results and automatic filing of e-mails. The benefits of clustering search results are eloquently described on the web site of *Vivisimo.com* (2003):

Our flagship product, Vivisimo Clustering Engine, automatically organizes search or database query results into meaningful hierarchical folders completely on-the-fly, out-of-the-box. It cleanly interfaces with any search engine or document database, transforming long lists of search results into categorized information without any clumsy pre-processing of the source documents.

The first step in document clustering is to map the documents into a vector space, as described in Chapter 2. In Chapter 4 we compare the performance of mapping methods in the context of both document classification and clustering. We also illustrate some properties of hierarchical model based clustering.

Chapters 5 – 10 contain the main contributions of this thesis.

The first contribution, described in Chapters 5 and 6, is *Model Based Fractionation and Refractionation*, a linear time method that makes hierarchical model based clustering practical for large data sets. It is an adaptation of an algorithm for nonparametric clustering originally proposed by Cutting, Karger, Pedersen, and Tukey (1992). The basic idea is to split the data into small manageable sized “fractions” and then combine the results of clustering those fractions. The model based Fractionation algorithm differs only slightly from hierarchical model based clustering. Examples are presented in Chapter 7.

Approaches to model based hierarchical clustering of large data sets have previously been suggested by Posse (2001) and Wehrens, Buydens, Fraley, and Raftery (2003). Posse’s basic idea is to start the hierarchical clustering not from individual observations, but from an initial partition formed by analyzing the minimal spanning tree of the data. A Euclidean minimal spanning tree can be computed in expected time $O(n \log n)$ (Bentley and Friedman 1978); in practice however, this bound is realistic only if the dimensionality is small relative to the number of observations n . In high dimensions, the Bentley and Friedman algorithm is

not much faster than the naive algorithm of Prim (1957) that evaluates all pairwise distances (Nene and Nayar 1997).

Wehrens, Buydens, Fraley, and Raftery (2003) apply model based hierarchical clustering to a sample and generate several models. These models are then used as starting guesses for the EM algorithm applied to the entire data set. They show that their method is successful in a number of examples; however there is always the danger that small groups are not represented in the sample and therefore will be missed altogether.

The second contribution of the thesis, described in Chapter 8, is a collection of tools for *Model Assessment*. The main purpose of these tools is to quantify the overlap between mixture components and verify that the clusters indeed correspond to distinct groups in the data. We assess overlap by considering the distribution of the margin of the model; the posterior probabilities of class assignment; and the misclassification probabilities.

The third contribution, described in Chapter 9, is a new *Hybrid Clustering* method which combines some of the strengths of model based and non-parametric clustering. The hybrid clustering algorithm prunes the cluster tree generated by hierarchical model-based clustering. Starting with the tree corresponding to the mixture model chosen by the Bayesian Information Criterion, it progressively merges clusters that do not appear to correspond to different modes of the data density. The decision on whether or not to merge two clusters is based on the DIP test of unimodality (J.A. Hartigan and P.M. Hartigan 1985) applied to the projection of the data onto the Fisher linear discriminant direction. An example of hybrid clustering is shown in Chapter 10.

Chapter 11 concludes the thesis with a discussion and ideas for future work.

Chapter 2

CONVERTING DOCUMENTS INTO VECTORS

The input to model based clustering is a collection of points in Euclidean space. Documents, on the other hand, are collections of words of varying lengths. Therefore, in order to apply model based clustering we need to map *bags of words* into vectors. (The term “bag of words” indicates that we ignore the order of words in a document.)

2.1 Term Document Matrix

The documents in a collection are first decomposed into word or sub-word units usually referred to as *terms*. Since *stop words* such as “the”, “and”, etc. provide no information about the topic of a document, they are excluded. The remaining terms in the collection are arbitrarily assigned sequence numbers between 1 and the number of terms p . Each document in the collection is then represented by a p -dimensional vector of term frequencies, and the collection of n documents is represented by a $n \times p$ term-frequency matrix $F = \{f_{ij}\}$. In many applications the size of the vocabulary is much larger than the typical document size, giving rise to extremely sparse term frequency matrices.

2.2 Frequency Transformation and Term Weighting

The raw term-frequency matrix F is usually subjected to various transformations. The first step often is to replace the term frequencies by their square-root or logarithm. This reduces the influence of high counts, which is motivated by the belief that the difference between a term occurring 10 times versus 11 times in a document is not as significant as the difference between a term occurring once versus not occurring at all. A more extreme step in the

same direction is to convert F to a binary matrix indicating whether a term does or does not occur in a document.

A document is usually characterized by a few key terms; these terms indicate what topic the document is covering, and do not necessarily appear multiple times within the document. Hence total term frequency is not always indicative of a term's information content; for example a term that occurs in all documents contains no information about the topics. To account for this disparity in the information content of terms, several term weighting schemes have been proposed. These schemes are global in the sense that the weight of a term depends on the distribution of the term over the entire document collection. Some proposed choices for the weight, w_j , assigned to the j -th term are:

Identity: $w_j = 1$

Normal: $w_j = 1/\sqrt{\sum_i f_{ij}^2}$

Global frequency Inverse document frequency (GfIdf):

$$w_j = \sum_i f_{ij} / \sum_i I_{(f_{ij}>0)}, \text{ where } I_{(\cdot)} \text{ denotes the indicator function.}$$

Inverse document frequency (Idf): $w_j = \log \left(n / \sum_i I_{(f_{ij}>0)} \right)$

Entropy: $w_j = 1 + \sum_i p_{ij} \log p_{ij} / \log n$, with $p_{ij} = f_{ij} / \sum_i f_{ij}$.

The normal weighting scheme normalizes the term counts over the document collection. Hence a term which occurs infrequently will make the same contribution to the distance between documents as a very common term. The global frequency inverse document frequency (GfIdf) weighting scheme weights each term by the average frequency of the term in documents containing the term. Among two terms with equal total frequency $\sum_i f_{ij}$, GfIdf favors the one that occurs in a smaller number of documents. The inverse document frequency (Idf) weighting scheme gives lower weights to terms occurring in a large number of documents. The entropy weighting scheme is based on information-theoretic ideas, and the weight is actually one minus the entropy. The entropy of a frequency distribution is maximized if all the frequencies are the same. This case is thought of as least informative: if a given term is equally likely to be present in all documents, then this term is

not informative about the topic of any particular document. A frequency distribution that is concentrated at a single document is at the other extreme (entropy = 0) – the term completely distinguishes a particular document from the others.

Previous studies by Dumais (1991) suggest that entropy weighting outperforms other weighting schemes in the context of information retrieval; however her study focuses only on untransformed frequencies and log-transformed, entropy weighted frequencies. In an experiment, described in Chapter 4, we evaluate all 15 combinations of the three term frequency transformations (untransformed, square root and logarithm) with the five weighting schemes listed above. We first transform the term frequencies, multiply the transformed frequencies by their global weights, and then normalize each document vector to have unit length. The last step eliminates the influence of document length on distance. More precisely, with the frequency of term j in document i being f_{ij} , with term weights w_j and with a transformation $g(\cdot)$ (square-root, log or identity), the transformed and weighted term frequencies are given by

$$x_{ij} = \frac{w_j \times g(f_{ij})}{\sqrt{\sum_k (w_k \times g(f_{ik}))^2}},$$

In the following, X will denote the transformed and weighted term frequency matrix.

2.3 Dimensionality Reduction

The number of terms (p) occurring in a document collection can easily be in the thousands and may be larger than the number of documents (n). Representing each document by such a high dimensional vector has several disadvantages.

From a statistical viewpoint, high dimensionality makes it hard to build parsimonious models. For example, fitting a Gaussian distribution to a collection of p dimensional vectors requires estimation of $p + p(p + 1)/2$ parameters.

Representing documents by high dimensional term frequency vectors can also be detrimental to performance. This was first noted in the context of document retrieval and led to the discovery of *Latent Semantic Indexing* (LSI). LSI was conceived with the goal of obtaining a

measure of similarity between documents that is more reflective of “semantic content” than lexical matching (Berry, Dumais, and O’Brien 1995). Lexical matching between words has been observed to be quite ineffective in information retrieval since only documents having at least one word in common with the query are retrieved. In fact, compared to LSI, lexical matching yields low *recall* (many relevant documents are missed) and low *precision* (many unrelated documents are retrieved). According to Berry et al. (1995), the premise of LSI is that “*there is an underlying or latent structure in the pattern of word usage that is partially obscured by the variability of word choice.*” LSI is designed to uncover this structure.

LSI is essentially the same as principal component analysis, the standard statistical tool for dimensionality reduction. Principal component analysis maps a collection of high-dimensional vectors into some lower dimensional space while (hopefully) preserving the essential structure.

2.3.1 Principal Component Analysis

Given document vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in R^p$ and a target dimensionality q , principal component analysis finds the q -dimensional affine subspace S of R^p that is closest to the document vectors, in that it minimizes $\sum \mathbf{d}^2(\mathbf{x}_i, S)$, where $\mathbf{d}(\cdot, \cdot)$ denotes Euclidean distance. It is a standard result (Mardia, Kent, and Bibby 1979, Chapter 8) that S passes through the mean of the document vectors and is spanned by the q eigenvectors of the term covariance matrix Σ with the largest eigenvalues. The term covariance matrix is defined as

$$\begin{aligned}\Sigma &= 1/n \tilde{X}^t \tilde{X}, & \text{where} \\ \tilde{X} &= (I - 1/n \mathbf{1}\mathbf{1}^t) X.\end{aligned}$$

\tilde{X} is obtained from X by mean centering the columns. Here $\mathbf{1} = (1, \dots, 1)$. Let

$$\Sigma = A \Lambda A^t.$$

be the eigen-decomposition of Σ . The columns of A are the normalized eigenvectors of Σ , and $\Lambda = \mathbf{diag}(\lambda_1, \dots, \lambda_p)$ is the diagonal matrix of eigenvalues, in decreasing order. The projection \mathbf{y} of a document vector \mathbf{x} on the space spanned by the first q eigenvectors of Σ

is given by $\mathbf{y} = A_q^t \mathbf{x}$, where A_q denotes the $p \times q$ matrix consisting of the q leading columns of A .

Dimensionality reduction by principal component analysis has another interesting property: it preserves distances between documents to the largest extent possible (Mardia et al. 1979, Chapter 14.4). For a set of feature vectors $\mathbf{z}_1, \dots, \mathbf{z}_n \in R^q$, define

$$E(\mathbf{z}_1, \dots, \mathbf{z}_n) = \sum_{i,j} (\mathbf{d}^2(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{d}^2(\mathbf{z}_i, \mathbf{z}_j))^2$$

The figure of merit $E(\mathbf{z}_1, \dots, \mathbf{z}_n)$ measures how well the inter-point distances of the q -dimensional feature vectors $\mathbf{z}_1, \dots, \mathbf{z}_n$ match those of the p -dimensional document vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$. It is optimized by choosing $\mathbf{z}_i = A_q^t \mathbf{x}_i$. As the “structure” of the document cloud is captured by its inter-point distance matrix it is justified to say that, in a sense, dimensionality reduction by principal component analysis preserves structure to the largest extent possible.

Rather than computing the eigen-decomposition of $\Sigma = \tilde{X}^t \tilde{X}$, we can find principal components by computing the singular value decomposition $\tilde{X} = \tilde{U} \tilde{\Phi} \tilde{V}^t$ of \tilde{X} . Here \tilde{U} is $n \times n$ orthogonal, \tilde{V} is $p \times p$ orthogonal, and $\tilde{\Phi}$ is the diagonal matrix of singular values, in decreasing order. We have

$$\tilde{X}^t \tilde{X} = \tilde{V} \tilde{\Phi}^2 \tilde{V}^t$$

which shows that $\tilde{V} = A$ and $\tilde{\Phi}^2 = \Lambda$ (up to sign changes). This alternative algorithm highlights the similarity between principal components and LSI.

2.3.2 Latent Semantic Indexing

Latent semantic indexing is a dimensionality reduction tool very similar to principal component analysis. In contrast to principal component analysis, it finds the singular value decomposition $X = U\Phi V^t$ of X – the columns of X are not mean-centered. Dimensionality reduction is achieved as in principal component analysis: $\mathbf{y}_i = V_q^t \mathbf{x}_i$, where V_q is the matrix consisting of the leading q columns of V . *Latent semantic indexing reduces dimensionality*

by projecting the document vectors onto the closest q -dimensional linear subspace, whereas principal component analysis projects them onto the closest affine subspace. An operational advantage of LSI is that there are SVD algorithms (Berry et al. 1995; Berry, Drmac, and Jessup 1999) that take advantage of the sparsity of the matrix X .

A more recent approach to reducing the dimensionality of documents is Probabilistic LSI (Hofman 1999; Griffiths and Steyvers 2002). Probabilistic LSI regards the term counts of a document d as a sample from a multinomial distribution $Mult(n_d, p_d)$ where n_d is the number of words in the document and p_d is a document specific probability vector. A q dimensional subspace of the term space is defined by $q + 1$ probability vectors w_1, \dots, w_{q+1} . The projection of a document onto this subspace is the convex combination of w_1, \dots, w_{q+1} that maximizes the likelihood of the term counts for the document. The optimal subspace is the given by the $q + 1$ probability vectors maximizing the likelihood of the projections for the entire document collection.

Chapter 3

MODEL BASED CLUSTERING

The premise of model based clustering is that each group g in the data is represented by a density $p_g(\mathbf{x})$ that is a member of some parametric family. The overall density $p(x)$ then is a mixture of group densities:

$$p(\mathbf{x}) = \sum_{g=1}^G \pi_g p_g(\mathbf{x}), \quad (3.1)$$

where π_g is the probability that a randomly chosen observation belongs to group g . A common assumption is that the group densities p_g are multivariate Gaussian with mean μ_g and covariance matrix Σ_g , and we use $\phi(\cdot; \mu_g, \Sigma_g)$ to denote this density. The log likelihood of a sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ is

$$l = \sum_{i=1}^n \log\left(\sum_{g=1}^G \pi_g \phi(\mathbf{x}_i; \mu_g, \Sigma_g)\right). \quad (3.2)$$

For a given number of mixture components this log-likelihood can be optimized over π_g, μ_g and Σ_g using the EM-algorithm (McLachlan and Peel 2000, Chapter 2.8).

Throughout this thesis, we shall use “groups” to denote the actual groups in the data and “clusters” to denote the estimates of these groups.

3.1 Parsimonious Mixture Models

There are a range of different restrictions which can be placed on the model (3.1) to make it more parsimonious (Banfield and Raftery 1993; Fraley and Raftery 1998; Fraley and Raftery 2002). These more parsimonious models restrict the shape, orientation and volume of the covariances. The simplest model assumes that all the covariances are identical and spherical

– that is $\Sigma_g = \sigma^2 I$. The next simplest model is one where each component has a spherical covariance matrix with a different variance – that is $\Sigma_g = \sigma_g^2 I$. Then there are various models with diagonal covariance matrices. On the upper end of the complexity range is the unrestricted model, where each component of the mixture has its own covariance matrix with no restrictions on the shape, orientation or volume.

We focus on a model where each mixture component has its own diagonal covariance matrix. An advantage of this restriction is that a non-degenerate model can be estimated from as few as two observations. To get a non-degenerate estimate for the full model, on the other hand, requires $p + 1$ observations.

A potential problem with using diagonal covariance matrices is that we are implicitly assuming that there is no correlation between the variables within each group, which of course might be false.

3.2 Hierarchical Model Based Clustering

In practice we never know the number of distinct groups in our data, and therefore the number of mixture components (G) is unknown as well. We can use the method of Fraley and Raftery (1998) and Das Gupta and Raftery (1998) to solve this problem by fitting mixture models for a range of values of G and then choosing the best one according to some criterion, trading off goodness of fit against model complexity (see section 3.3).

An efficient way of initializing estimation algorithms for a range of values of G is model based hierarchical clustering (Banfield and Raftery 1993). Model based hierarchical clustering follows the same general outline as all hierarchical agglomerative clustering procedures:

1. Initialize every point to be its own cluster.
2. Repeatedly merge the two closest clusters until only one cluster is left.

We now describe how we measure the closeness of clusters. Let C_i , $i = 1, 2$ be two clusters with counts n_i , mean vectors μ_i and sample covariance matrices Σ_i . Let C_{12} be the cluster obtained by merging C_1 and C_2 with count $n_{12} = n_1 + n_2$, mean vector μ_{12} , covariance

matrix Σ_{12} . We measure distance between C_1 and C_2 by the decrease in log-likelihood resulting from the merge:

$$\Delta l(1, 2) = l_1 + l_2 - l_{12} \quad (3.3)$$

where $l_1 = \sum_{i \in C_1} \log \phi(\mathbf{x}_i; \mu_1, \Sigma_1)$ and l_2, l_{12} are defined analogously.

Note that $\Delta l(1, 2)$ is the test statistic of the log likelihood ratio test for the hypothesis that clusters 1 and 2 are sampled from the same Gaussian distribution with diagonal covariance matrix.

Hierarchical model based clustering for a model with uniform spherical covariance matrices is identical to Ward's method (Ward 1963).

3.2.1 Clustering Singletons

At the start of hierarchical agglomerative clustering every observation is its own cluster. This presents a problem, as there is no obvious way to assign a covariance matrix to a singleton. Fraley (1998) proposes heuristics for dealing with this problem for the variable spherical and unconstrained models. We focus on models with diagonal covariances which require only two observations per cluster to get non-degenerate estimates of the covariances. We therefore start the clustering by pairing the observations.

To find a good pairing, we first compute the Euclidean distance between all pairs of observations. We then repeatedly merge the closest pair of observations into a cluster and remove them from consideration. If at any stage the distance between the closest pair of observations is significantly larger than the distance between one of the observations and its nearest neighbor, then we instead assign these observations to the clusters containing their nearest neighbors. This tends to avoid merging observations from different groups. We have empirically observed that a good default is to keep a pair intact if the distance between the observations is less than 1.3 times the distance to the nearest neighbor for either observation.

If there are indeed groups in the data, then single observations will be connected to a pair from the same group because the distance between a point and its nearest neighbor (which will be in the same group) will be significantly smaller than the distance between that point and any other point from a different group.

3.2.2 Computing the Likelihood Distance for Clusters with Diagonal Covariance Matrices

For the diagonal model, there are efficient formulas for evaluating the log-likelihood distance (3.3). Define

$$\begin{aligned}\widehat{\Sigma}_i &= n_i W_i, \\ \lambda_i &= \frac{1}{n_i} |\mathbf{diag}(W_i)|^{1/p}, \\ D_i &= \frac{\mathbf{diag}(W_i)}{|\mathbf{diag}(W_i)|^{1/p}}.\end{aligned}$$

Note that λ_i measures the volume of the i th cluster and D_i describes its shape.

The corresponding parameters for the merged cluster C_{12} are

$$\hat{\mu}_{12} = \frac{n_1}{n_{12}} \hat{\mu}_1 + \frac{n_2}{n_{12}} \hat{\mu}_2, \quad (3.4)$$

$$W_{12} = W_1 + W_2 + n_1(\hat{\mu}_{12} - \hat{\mu}_1)(\hat{\mu}_{12} - \hat{\mu}_1)^t + n_2(\hat{\mu}_{12} - \hat{\mu}_2)(\hat{\mu}_{12} - \hat{\mu}_2)^t, \quad (3.5)$$

$$\lambda_{12} = \frac{1}{n_{12}} |\mathbf{diag}(W_{12})|^{1/p}. \quad (3.6)$$

Therefore the likelihood distance from Equation (3.3) becomes

$$\begin{aligned}\Delta l(1, 2) &= p(n_{12}) \log \lambda_{12} + \frac{1}{\lambda_{12}} \text{trace}(W_{12} D_{12}^{-1}) \\ &\quad - \left(p n_1 \log \lambda_1 + p n_2 \log \lambda_2 + \frac{1}{\lambda_1} \text{trace}(W_1 D_1^{-1}) + \frac{1}{\lambda_2} \text{trace}(W_2 D_2^{-1}) \right) \\ &= p(n_{12}) \log \lambda_{12} - p n_1 \log \lambda_1 - p n_2 \log \lambda_2.\end{aligned} \quad (3.7)$$

3.3 Choosing Between Models

One of the big advantages of Model Based Clustering is that it provides a theoretical basis for estimating the number of mixture components, which is taken to be equal to the number of groups.

Typical methods estimate the number of mixture components by maximizing a criterion trading off goodness of fit and model complexity. Examples include Akaike's Information Criterion (Akaike 1973; 1974)

$$Akaike = 2 \times \log l(\cdot) - 2r, \quad (3.8)$$

which tends to overestimate the number of components (Koehler and Murphee 1988), and the Bayesian Information Criterion (BIC) (Schwarz 1978)

$$BIC = 2 \times \log l(\cdot) - r \log(n), \quad (3.9)$$

where r is the number of parameters, n is the number of observations and $l(\cdot)$ is the likelihood of the model.

The BIC was developed by Schwarz (1978) as an approximation to twice the logarithm of the Bayes Factor, originally described by Jeffreys (1935) but better explained by Kass and Raftery (1995). Also note that Fraley and Raftery (2002) give an explanation of the use of BIC in the model based clustering context. Let \mathcal{D} denote the data, and let $\mathcal{M}_1, \mathcal{M}_2$ be two different mixtures models. The Bayes factor for model \mathcal{M}_2 against model \mathcal{M}_1 is the ratio

$$P(\mathcal{D}|\mathcal{M}_2)/P(\mathcal{D}|\mathcal{M}_1).$$

This corresponds to the posterior odds for \mathcal{M}_2 against \mathcal{M}_1 , assuming that *a priori* the two models is equally likely.

Use of BIC for choosing the number of mixture components is only justified if the mixture models are fitted by maximum likelihood. Strictly speaking this requires running the EM algorithm to convergence, which can be computationally expensive for large data sets. Also note that running the EM algorithm to convergence would destroy the hierarchical structure

of the partitions that we obtain. As a compromise, we perform an E-step, followed by an M-step and use the resulting mixture likelihood in the BIC formula.

3.4 Model Selection with Cost Complexity Pruning

Hierarchical model based clustering produces a nested sequence of trees, $T_1 \subset \dots \subset T_i \subset \dots \subset T_n$, where T_i represents a mixture model with i mixture components. In the previous section we described how to use the BIC for finding the best tree in this sequence.

Note that any subtree of T_n (not only the subtrees in the sequence constructed by hierarchical agglomeration) represents a mixture model. An alternative method for generating a nested sequence of subtrees of the full tree T_n is *cost complexity pruning*, first proposed in the context of Classification And Regression Trees (CART) (Breiman, Friedman, Olshen, and Stone 1984). We now compare these two approaches.

3.4.1 Definition of Cost Complexity Pruning

The basic idea of cost complexity pruning is to define a notion of tree cost, incorporating both complexity and goodness of fit of a tree. The cost of a clustering tree is recursively defined as follows:

- Cost of terminal node = negative two times the log-likelihood for the observations in that node.
- Cost of interior node = cost of left child + cost of right child + α , where α is the complexity penalty.
- Cost of tree = cost of root node.

If we define \tilde{T} as the set of leaves of tree T , $|\tilde{T}|$ as the number of leaves in the tree and $R(s)$ as negative two times the log-likelihood of the observations in node s , then the cost of T is

$$R_\alpha(T) = \sum_{s \in \tilde{T}} R(s) + \alpha(|\tilde{T}| - 1). \quad (3.10)$$

If we let r_0 be the number of parameters required by a terminal node ($2p$ for the the diagonal model) and then set $\alpha = r_0 \log n$, then the cost of the tree is the value of the BIC (up to an additive constant).

3.4.2 Bottom Up Recombination

We can find the minimum cost subtree for any value of the complexity penalty α by *bottom up recombination*. By increasing α , we get a nested sequence of trees; these are called the *optimally terminated subtrees*. The optimally terminated subtree for $\alpha = 0$ is the full tree T_n ; for sufficiently large values of α , the optimally terminated subtree consists of the root node T_1 .

For any particular node t of a given tree, we define the critical value $\tilde{\alpha}_t$ as the minimum value of α for which the node would be made terminal. Recall that the cost of the subtree T_t rooted at node t is $R_\alpha(T_t) = -2 \log L_{T_t} + \alpha(|\tilde{T}_t| - 1)$ and the cost of t if it were to be made terminal is $R(t) = -2 \log L_t$. Here L_t is the likelihood of the data in node t if it is terminal and L_{T_t} is the likelihood of the data for the mixture model defined by the leaves of the subtree T_t . The critical value $\tilde{\alpha}_t$ can be found by equating the cost of the subtree rooted at node t to the cost of the node if it were to be made terminal: $R_{\tilde{\alpha}_t}(T_t) = R(t)$. This gives

$$\tilde{\alpha}_t = \frac{2 \log L_{T_t} - 2 \log L_t}{|\tilde{T}_t| - 1}. \quad (3.11)$$

We can now find the complete sequence of optimally terminated subtrees by repeatedly (i) finding $\hat{t} = \arg \min_t \tilde{\alpha}_t$ for the current tree; (ii) pruning the subtree rooted at node \hat{t} .

3.4.3 Equality of Optimally Terminated Subtree Sequence and Hierarchical Clustering Subtree Sequence

We will now give a sufficient condition under which cost complexity pruning gives the same sequence of subtrees as hierarchical clustering.

As before let $T_1 \subset \dots \subset T_n$ be the sequence of subtrees generated by hierarchical clustering and $T'_1 \subset \dots \subset T'_n$ be the sequence of subtrees generated by cost complexity pruning. (Obviously $T_1 = T'_1$ and $T_n = T'_n$.)

Let t_i be the node whose children are pruned from T_{i+1} to form T_i , so t_{n-1} is the first interior node generated by the hierarchical clustering algorithm.

For any node t with children t_l and t_r , define $\Delta R(t) = R(t) - R(t_l) - R(t_r)$. This is the decrease in log likelihood resulting from the merge of the clusters corresponding to t_l and t_r . Note that if t_l and t_r both are leaves then the critical value of α for node t is $\tilde{\alpha}_t = \Delta R(t)$.

Theorem: If

$$\Delta R(t_1) > \Delta R(t_2) > \dots > \Delta R(t_{n-1}).$$

then $T_i = T'_i$ for all $i = 1, \dots, n$.

Proof: The proof is by induction. By definition $T_n = T'_n$. Suppose we have shown that $T_{k+1} = T'_{k+1}$. We have to prove that the critical value $\tilde{\alpha}_{t_k}$ for node t_k is the minimum value of $\tilde{\alpha}_t$ for all the interior nodes t of T_{k+1} .

Note that by rearranging the definition of $\Delta R(s)$ we get $R(s) = R(s_l) + R(s_r) + \Delta R(s)$. Repeatedly using this identity gives

$$R(t) = \sum_{s \in T_t \setminus \tilde{T}_t} \Delta R(s) + \sum_{s \in \tilde{T}_t} R(s), \quad (3.12)$$

where $T_t \setminus \tilde{T}_t$ is the set of interior nodes of tree T_t with cardinality $|T_t \setminus \tilde{T}_t| = |\tilde{T}_t| - 1$. Recall that $R(t) = -2 \log L_t$ and $-2 \log L_{T_t} = \sum_{s \in \tilde{T}_t} R(s)$.

By substituting Equation 3.12 into Equation 3.11 we obtain

$$\begin{aligned} \tilde{\alpha}_t &= \frac{\sum_{s \in T_t \setminus \tilde{T}_t} \Delta R(s)}{(|\tilde{T}_t| - 1)} \\ &> \min_{s \in T_t \setminus \tilde{T}_t} \Delta R(s). \end{aligned}$$

By assumption,

$$\min_{s \in T_t \setminus \tilde{T}_t} \Delta R(s) \geq \Delta R(t_k),$$

with equality iff t_k is a descendent of t . Since $\Delta R(t_k) = \tilde{\alpha}_{t_k}$ the theorem is proven.

Note: Since $\Delta R(t) = -2\Delta l(t_l, t_r)$, and the hierarchical clustering algorithm outputs the $\Delta l(t_l, t_r)$, we can plot these to determine whether the condition holds.

Chapter 4

APPLICATION TO THE TDT DATA

In this chapter we illustrate and evaluate some of the methods introduced in Chapters 2 and 3 using the *Topic Detection and Tracking* (TDT) corpus (Allan, Carbonell, Doddington, Yamron, and Yang 1998). After describing the TDT corpus, we present the results of two experiments evaluating the effects of transformations, term weightings and feature space dimensionality on document classification and clustering.

4.1 The Topic Detection and Tracking Corpus

The TDT corpus consists of 15,863 news stories (documents) taken from Reuters and CNN between July 1, 1994, and June 30, 1995. The TDT project investigators classified 1131 of these documents into 25 pre-selected topics shown in Table 4.2. The number of documents in a given topic in the “labeled TDT” collection ranges from 2 to 273, and most topics have between 10 and 60 documents. In some of our examples we only use topics that have at least 10 documents. The reduced data set consisting of 1100 documents will be called the “1100 TDT” collection.

We carried out a visual exploration of the labeled TDT collection using the data exploration tool *XGobi* (Swayne, Cook, and Buja 1998). We applied log-Idf weighting to the term document matrix and took the first 50 principal components, resulting in a data set with 1131 observations and 50 variables. Looking at pairwise scatter plots of principal components, we found that many of the principal components correspond directly to specific topics (see Figure 4.1). Topic 9 is separated from the rest of the data along principal component 1, topic 6 is separated along principal component directions 4 and 5, etc.

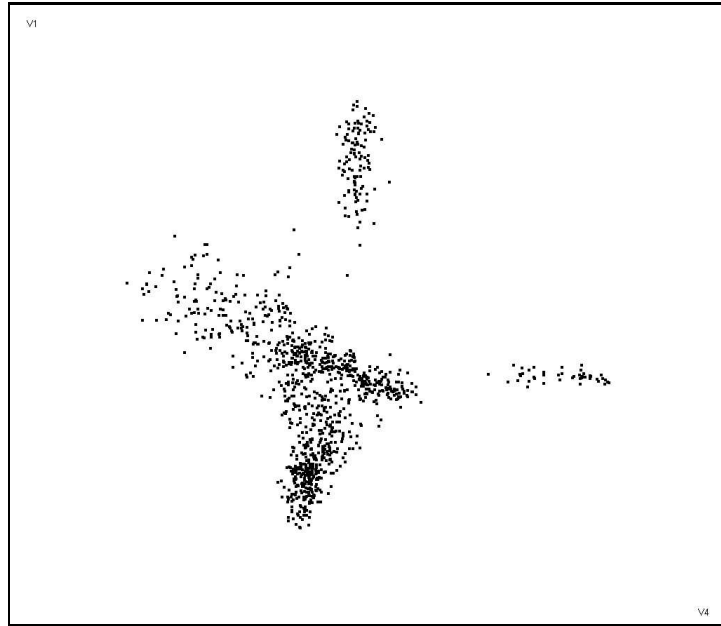


Figure 4.1: Projection of the labeled TDT data onto principal components 1 and 4.

Table 4.1: The principal component direction which correspond to topics. High values of the 25th direction corresponds to topic 10 and low values corresponds to both topics 1 and 13. Only part of topic 11 is separated along component 19.

Principal Component	1	4,5	8	7	11	14	17,18	19	21	25	25	33
Topic	9	6	16	5	17	12	20	[11]	23	10	1,13	2

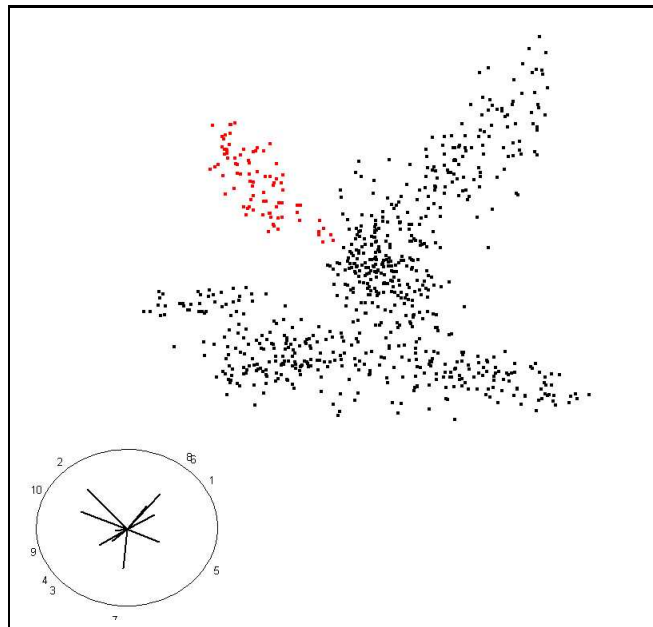


Figure 4.2: XGobi projection of the labeled TDT, showing topic 15.

After removing the topics in Table 4.1 from the data set (except topic 11), we browsed the data using the Grand Tour. The first cluster which became apparent was topic 15; the corresponding view is shown in Figure 4.2. (The circle in the corner shows which coordinates are projected in which direction.) After removing this cluster and continuing the Grand Tour, we found the cluster corresponding to topic 8 (Figure 4.3). Once this was removed, the cluster corresponding to topic 11 became visible (Figure 4.4). Continuing this process showed that most of the topics, including some of the smaller ones, can be found by browsing. However, this is time consuming and clusters which are close together are hard to find. The goal of clustering methods like model based clustering is to automate this process.

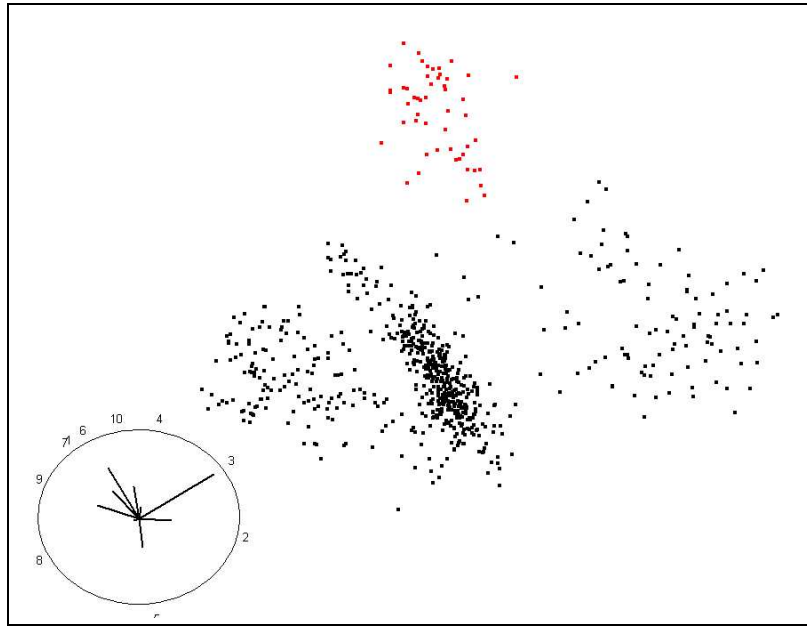


Figure 4.3: XGobi projection of the labeled TDT, showing topic 8.

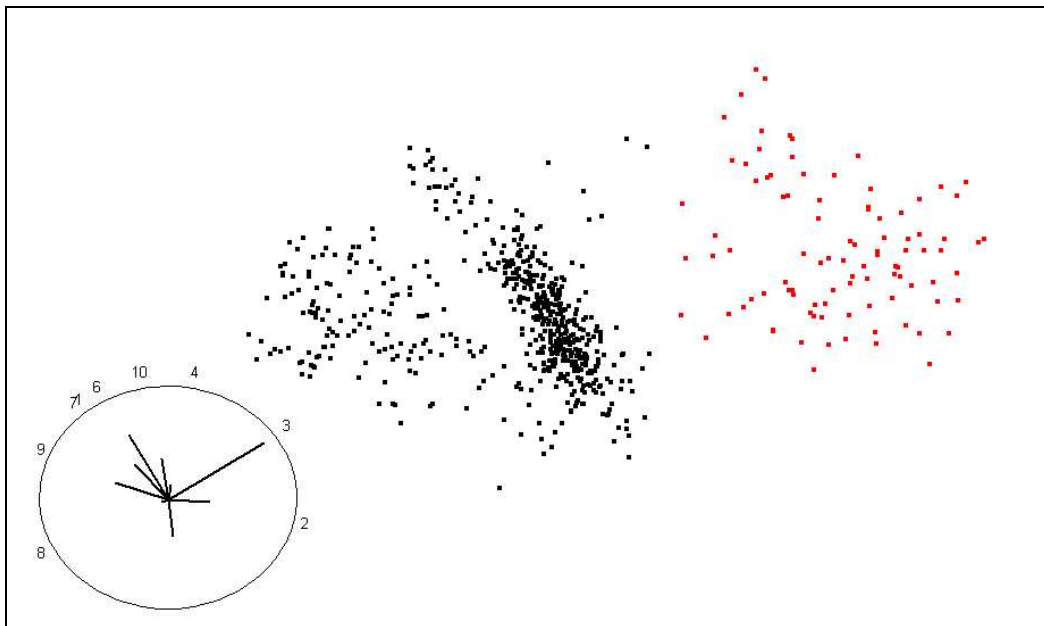


Figure 4.4: XGobi projection of the labeled TDT, showing topic 11.

Table 4.2: The TDT Corpus Topics

Number	Topic	Number of Stories
1	Aldrich Ames	8
2	Carlos the Jackal	10
3	Carter in Bosnia	34
4	Cessna on White House	14
5	Clinic Murders (Salvi)	41
6	Comet into Jupiter	45
7	Cuban riot in Panama	2
8	Death of Kim Jong Il (N. Korea)	58
9	DNA in OJ trial	114
10	Haiti ousts observers	12
11	Hall's copter (N. Korea)	97
12	Humble, TX, flooding	22
13	Justice-to-be Breyer	8
14	Karrigan/Harding	2
15	Kobe Japan quake	84
16	Lost in Iraq	44
17	NYC Subway bombing	24
18	OK-City bombing	273
19	Pentium chip flaw	4
20	Quayle lung clot	12
21	Serbians down F-16	65
22	Serbs violate Bihac	90
23	Shannon Faulker	7
24	USAir 427 crash	39
25	WTC Bombing trial	22

4.2 *The Effects of Frequency Transformation, Term Weighting and Dimensionality on the Performance of a Document Classifier*

The goal of this experiment was to assess the influence of three factors — frequency transformation $g(\cdot)$, term weighting w_j , and dimensionality q of the feature space — on the ability to predict the topic of a document from its feature vector.

We tried all 15 frequency transformation / term weighting combinations described in Section 2.2. We used principal component analysis for dimensionality reduction, with a range of dimensions between 5 and 500.

4.2.1 *Design of The Experiment*

We used a k-nearest-neighbor classifier for document classification. To classify a document with feature vector \mathbf{x} , a k-nearest-neighbor classifier finds its k nearest neighbors among the training observations and takes a majority vote. The number k of neighbors is a parameter of the procedure. We used cross-validation to estimate the optimal k from the training sample. Here is a detailed description of the experiment:

- Randomly partition the $n = 1131$ labeled documents into five groups $\mathcal{D}_1, \dots, \mathcal{D}_5$ of roughly equal size.
- Choose a combination of the experimental factors (frequency transformation, term weighting, and dimensionality).
- For $i = 1, \dots, 5$, use group \mathcal{D}_i as the test set and the union \mathcal{D}_{-i} of the remaining groups as the training set. Estimate the optimal k from \mathcal{D}_{-i} by cross-validation. Classify the documents in \mathcal{D}_i using the optimal k . Let E_i denote the number of errors. Measure the merit of the current combination of experimental factors by the error rate $E = \sum_{i=1}^5 E_i/n$.

4.2.2 Results

Figures 4.5 and 4.6 summarize the results of the experiment. In both figures the error rate E is plotted on the vertical axis and the dimension of the feature space is plotted on the horizontal axis. For clarity we only show three curves. The solid curve corresponds to log transformation and Idf weighting, but the curves for log-entropy, log-unweighted, square root-entropy and square root-unweighted are very similar. The dotted line corresponds to untransformed Idf weighting. The transformed GIdf weightings lie somewhere in between the solid and dotted lines. The normally weighted curves are all much higher than the others and are similar to the dashed line, which is for the log transformation. The grey band is a ± 2 standard error band for log-Idf.

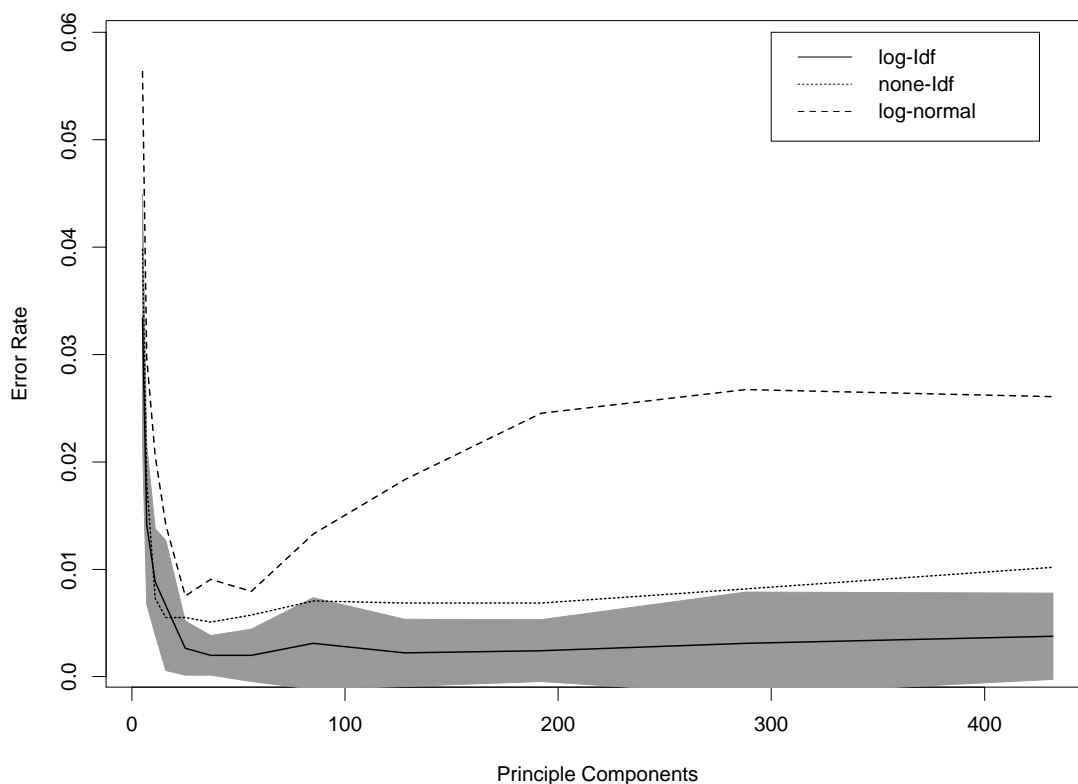


Figure 4.5: Performance of the transformation/weighting combinations as a function of dimensionality

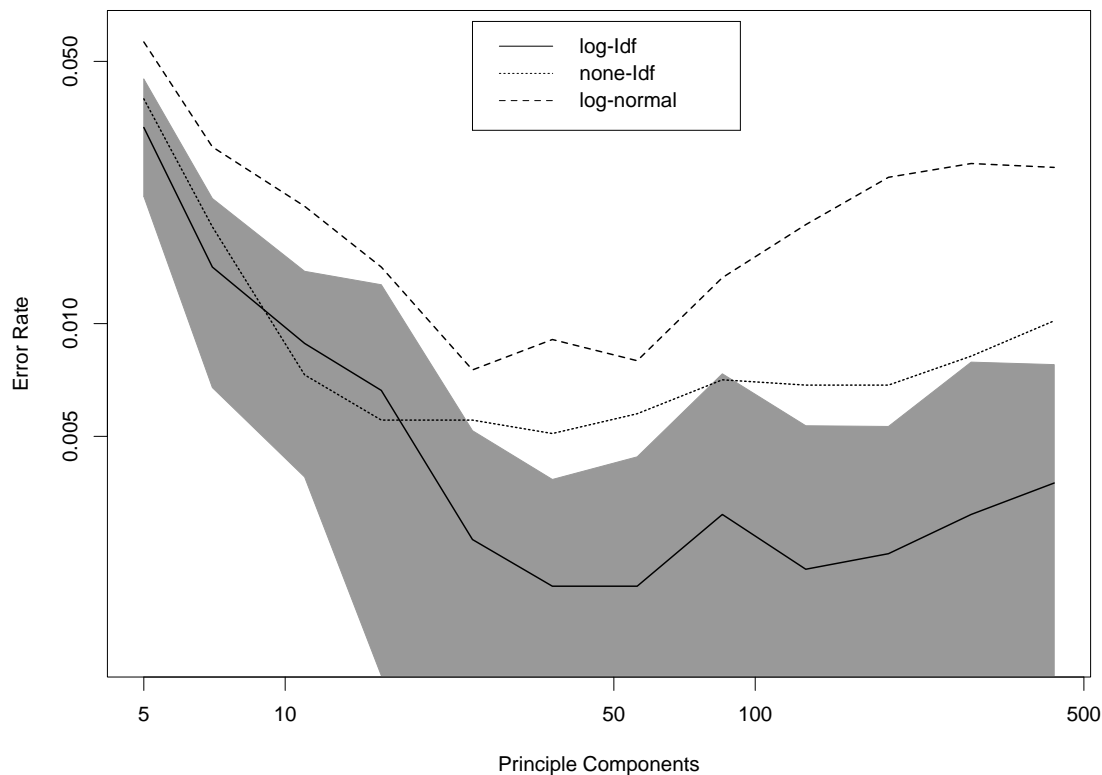


Figure 4.6: Performance of the transformation/weighting combinations as a function of dimensionality (log-log scale).

We also analyzed the results of our experiment by fitting a Poisson generalized linear model to the error counts. The analysis showed that the best combinations of transformation and term weighting are log-Idf, square root-Idf, log-entropy and square root-entropy. All other combinations resulted in a significantly worse error rate. The normal term weighting performs much worse than all of the other schemes, which has been observed by other researchers (Dumais 1991). Using fewer than 20 dimensions results in rapidly deteriorating performance.

For the optimal selections of frequency transformation, term weighting and feature space dimension the misclassification rate is less than 1%; most of the errors correspond to misclassification of documents representing very rare topics, e.g. *Cuban riot in Panama* (two

documents), *Karrigan/Harding* (two documents), and *Pentium chip flaw* (four documents).

4.3 Comparing Two Partitions

When we want to evaluate the performance of a clustering method we have to apply it to a labeled data set. We can then compare the partition of the data set induced by the labels to the partition produced by the clustering method. We now describe three indices measuring the similarity between partitions: the Fowlkes-Mallows index, the Adjusted Rand index and the F1 index.

4.3.1 Fowlkes-Mallow Index

The Fowlkes-Mallows index (Fowlkes and Mallows 1983) is the geometric mean of two probabilities: the probability that two randomly chosen observations are in the same cluster given that they are in the same group, and the probability that two randomly chosen observations are in the same group given that they are in the same cluster. Hence a Fowlkes-Mallows index near 1 means that the clusters are good estimates of the groups.

To compute the Fowlkes-Mallows index we construct a contingency table of the groups and the clusters, as shown in Table 4.3. Let $n_{i\cdot}$ be the sum over the i -th row of the table and let $n_{\cdot j}$ be the sum over the j -th column. Then the Fowlkes-Mallows index is given by

$$\sum_{i,j} \binom{n_{ij}}{2} / \sqrt{\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}} \quad (4.1)$$

4.3.2 The Adjusted Rand Index

The Rand index (Rand 1971) is the probability of two randomly chosen observations being either both in the same group and in the same cluster, or in different groups and different clusters. The Adjusted Rand index (Hubert and Arabie 1985) is the Rand index, normalized to have mean zero if there is no association between group labels and cluster labels, and to have maximum value one. The formula is given in Equation 4.2.

Table 4.3: Comparison between clusters (rows) and groups (columns). The cell count n_{ij} is the number of observations in cluster i and group j .

clusters	groups				Total
	1	2	...	J	
1	n_{11}	n_{12}	...	n_{1J}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2J}	$n_{2.}$
...
I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I.}$
Total	$n_{.1}$	$n_{.2}$...	$n_{.J}$	n

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2} \right] - \sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} / \binom{n}{2} / \binom{n}{2}} \quad (4.2)$$

4.3.3 The F1 Index

In the context of document clustering, an appealing index for the similarity of partitions can be created by combining the recall and precision measures which are frequently used in document retrieval. Suppose we were to use our clustering for document retrieval: when presented with a query document d we retrieve all the documents assigned to the same cluster $C(d)$ as d . Ideally we should have retrieved the documents in the same group $G(d)$ as document d . We define the recall $r(d)$ associated with document d as the fraction of documents in $G(d)$ which were retrieved. We define the precision $p(d)$ as the fraction of documents retrieved that are in $G(d)$. As in the Table 4.3, the number of documents in $C(d)$ is $n_{i.}$, the number of documents in $G(d)$ is $n_{.j}$ and the number of documents in both $G(d)$ and $C(d)$ is n_{ij} . Consequently

$$r(d) = n_{ij}/n_{.j} \quad \text{and} \quad p(d) = n_{ij}/n_{i.}$$

A reasonable index for similarity of partitions has to take into account the recall and precision associated with all documents in the collection. A popular index in the document

retrieval literature is the so-called F index (Allan et al. 1998; Van Rijsbergen 1979) (also denoted by $F1$ in some papers). We first combine recall and precision for a single document d by:

$$F(d) = 2 \frac{p(d)r(d)}{p(d) + r(d)} = \left\{ \frac{1}{2} \left(\frac{1}{r(d)} + \frac{1}{p(d)} \right) \right\}^{-1} \quad (4.3)$$

Averaging over all documents in a collection yields

$$F1 = \frac{1}{n} \sum_d F(d) = \frac{2}{n} \sum_{i,j} \frac{n_{ij}^2}{n_{i.} + n_{.j}}. \quad (4.4)$$

In our experiments the Fowlkes-Mallows and $F1$ indexes give very similar results.

4.4 The Effects of Term Weighting and Dimensionality on the Performance of Model Based Document Clustering

The goal of this experiment was to assess the influence of term weighting and feature space dimensionality on the ability of a clustering method to reproduce the partition of the “1100 TDT” collection into topics. We only used the log transformation of term frequencies. The clustering method studied was hierarchical model based clustering with diagonal covariances. We fixed the number of mixture components at 19, which is the number of topics in the “1100 TDT” collection.

Figure 4.7 shows the results of the experiment. The Fowlkes-Mallows index is plotted on the vertical axis, and the dimension of the feature space is plotted on the horizontal axis. The solid curves are the averages over 30 random half samples. The bands are ± 2 standard error intervals for the weightings. (We used half sampling instead of bootstrapping because bootstrap samples will contain observations multiple times which can give a misleading picture of the performance of clustering methods.)

The results are similar to those obtained from the classification experiment described in Section 4.2. The best weightings are the entropy and Idf . The normal and $GfIdf$ weightings perform poorly. The normal weighting does not reach the same peak performance as the other weightings. The performance deteriorates rapidly as the dimension drops below 20.

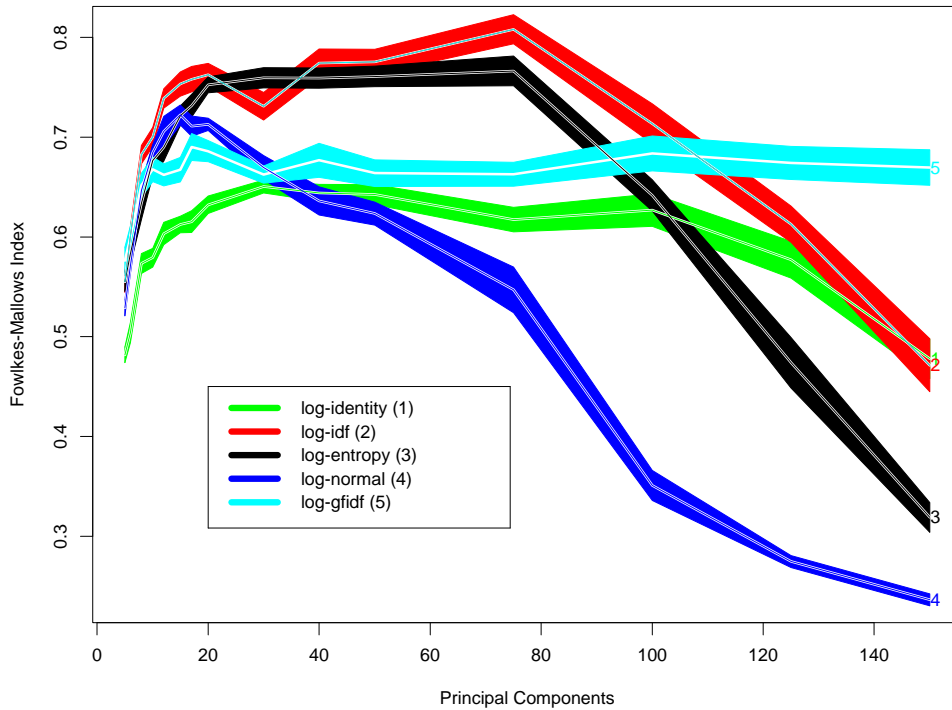


Figure 4.7: Performance of weighting schemes on hierarchical model based clustering as a function of dimensionality. Confidence intervals for the curves are shown as solid bands.

4.5 Using the BIC for Estimating the Number of Topics

We performed 3 experiments to investigate performance of the BIC. In these experiments we used the “1100 TDT” collection, log-Idf transformation and a 50 dimensional feature space.

4.5.1 Gaussian Data with Diagonal Covariances

In this experiment we estimated means and diagonal covariance matrices for the 19 topics of the “1100 TDT” data. We scaled the covariances to make the groups more or less compact. We then drew 15 samples for each scale factor from a Gaussian mixture model with these means and covariances. Table 4.4 shows the frequency distribution of the number of mixture components chosen by the BIC.

Table 4.4: Number of clusters chosen by the BIC when the model is correct. The true number of groups is 19.

Scale factor	14	15	16	17	18	19
2.2	3	7	5			
1.8			1	2	8	4
1.4						15
1.2						15
1.0						15
0.8						15

Table 4.5: Number of clusters chosen by the BIC for data simulated from a mixture of factor analyzers. The true number of groups is 19.

Scale factor	14	14	15	16	17	18			
1.4	8	3	2	2					
1.2	8	3		2	2				
1.0	1		1	5	7	1			
0.8				1	10	4	0.6	8	7

The BIC reliably selects the correct number of clusters when the variance of the groups is not significantly larger than the actual variance in the “1100 TDT” collection. As we blow up the groups by increasing the scale factor and thereby increasing the overlap between groups, the number of mixture components chosen by the BIC decreases.

4.5.2 Gaussian Data with Non-Diagonal Covariances

This experiment exactly parallels the previous one except that we used factor analysis (Hinton, Dayan, and Revow 1997) to estimate non-diagonal covariance matrices for the topics. Table 4.5 shows the frequency distribution of the number of mixture components chosen by the BIC.

Table 4.6: Number of clusters chosen by the BIC for non-diagonal Gaussian data – 19 is the true number of groups.

Scale factor	14	14	15	16	17	18
1.4	6	2	2	2	2	1
1.2	3	3	2	4	3	
1.0		2	2	4	6	1
0.8				2	11	2
0.6					11	4

4.5.3 Resampling from a Kernel Density Estimate

In this experiment we generated data by sampling from a kernel density estimate. We estimated the feature distribution for topic j by applying a Gaussian kernel estimate to the observations. We chose the covariance of the kernel to be the same as the covariance of the topic. We then rescaled the data to have the same variances in each coordinate as in the original data. Table 4.6 shows the distribution of the number of mixture components chosen by the BIC for scale factors.

4.5.4 Summary

The combination of using the BIC criterion with hierarchical model based clustering seems to perform well when the model is correct. Lack of fit of the model seems to lead to an under estimate of the number of groups.

4.6 Clustering the “1100 TDT” Collection

When applying hierarchical model based clustering with diagonal covariance matrices to the “1100 TDT” collection, the BIC chooses a model with 42 mixture components. The Fowlkes-Mallows index for the corresponding clustering is 0.49 and the Adjusted Rand

index is 0.38. (Using the code of (Fraley and Raftery 2002) and fitting an unconstrained hierarchical model followed by EM with a diagonal covariance, the BIC chooses 46 mixture components.)

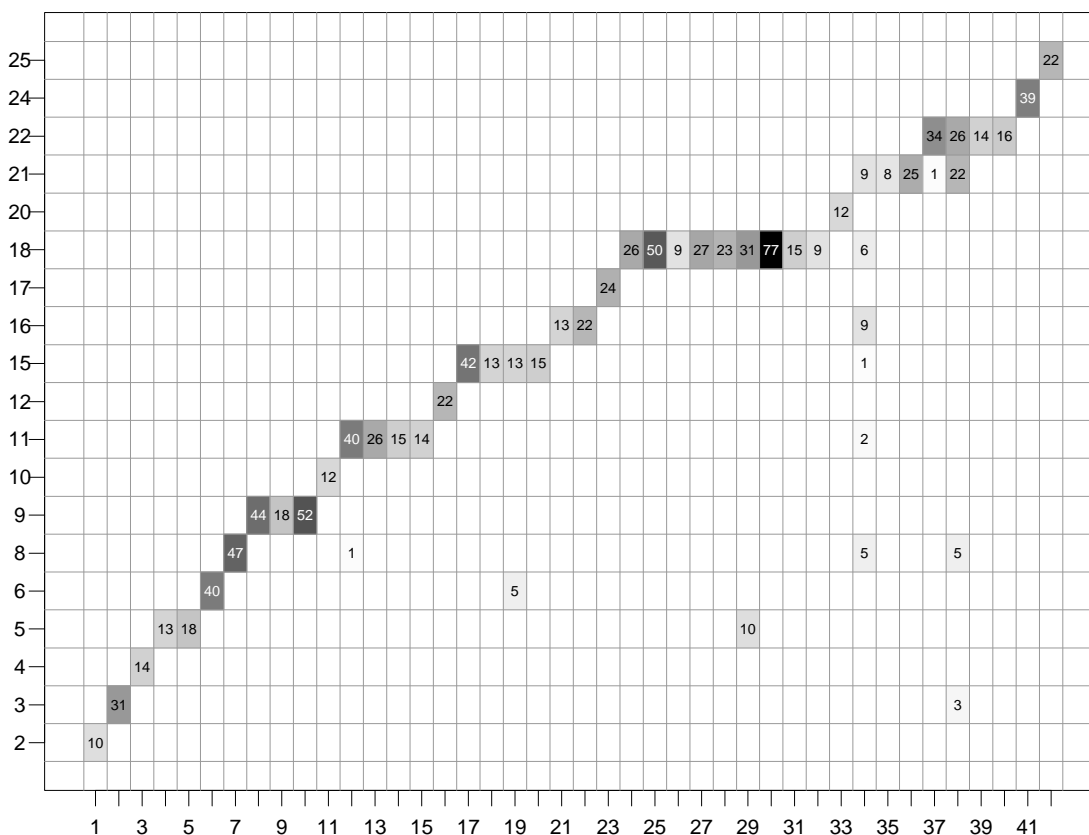


Figure 4.8: Two way contingency table of topics vs clusters for the “1100 TDT” collection. Topic labels are shown on the vertical axis and mixture components are shown on the horizontal axis.

Figure 4.8 shows a two way contingency table with cluster index on the horizontal axis and topic on the vertical axis. The clusters have good precision: most of them consist of documents from exactly one topic. The few exceptions do not seem unreasonable, for example topics 21 and 22 which are intermingled in cluster 40 both refer to Serbian aggression; cluster 21 contains observations from the topics “*Comet into Jupiter*” and “*Kobe Japan quake*” suggesting that it consists of stories about natural disasters; cluster 31 contains documents from two of the murder trial topics, “*OK-City Bombing*” and “*Clinic Murders (Salvi)*”.

While precision of the clustering is good, recall is low: many topics are split up between several clusters. At least for some of the topics such a split might not be completely spurious; they actually seem to consist of multiple subtopics (see Section 10.3).

Chapter 5

FRACTIONATION

The time and space complexity of hierarchical model based clustering is at least $O(n^2)$, which makes it impractical for data sets consisting of more than a few thousand observations. *Fractionation*, originally proposed by Cutting, Karger, Pedersen, and Tukey (1992) in the context of non-parametric hierarchical clustering, offers a way out of this dilemma.

Let M be roughly the problem size for which a $O(M^2)$ hierarchical clustering problem is computationally tractable. The original Fractionation algorithm for partitioning a data set into G clusters proceeds as follows:

- 1 Split the data into subsets or fractions that are approximately of size M .
- 2 Cluster each fraction into a fixed number αM of clusters, with $\alpha < 1$. Summarize each cluster by its mean. We refer to these cluster means as *meta-observations*.
- 3 If the total number of meta-observations is significantly greater than M , return to step (1), with the meta-observations taking the place of the original data.
- 4 Cluster the meta-observations into G clusters.
- 5 Assign each individual observation to the cluster with the closest mean.

The number of fractions in the i -th iteration is approximately $\alpha^{i-1}n/M$ and the work involved in clustering a fraction is $O(M^2)$ independent of n . Consequently the i th iteration in the fractionation process takes $O(\alpha^{i-1}nM)$ operations and the total number of operations is

$$\sum_{i=1}^I \alpha^{i-1}nM = nM \frac{1 - \alpha^I}{1 - \alpha},$$

which shows that the total run time is $O(n)$. Both α and M are user defined parameters.

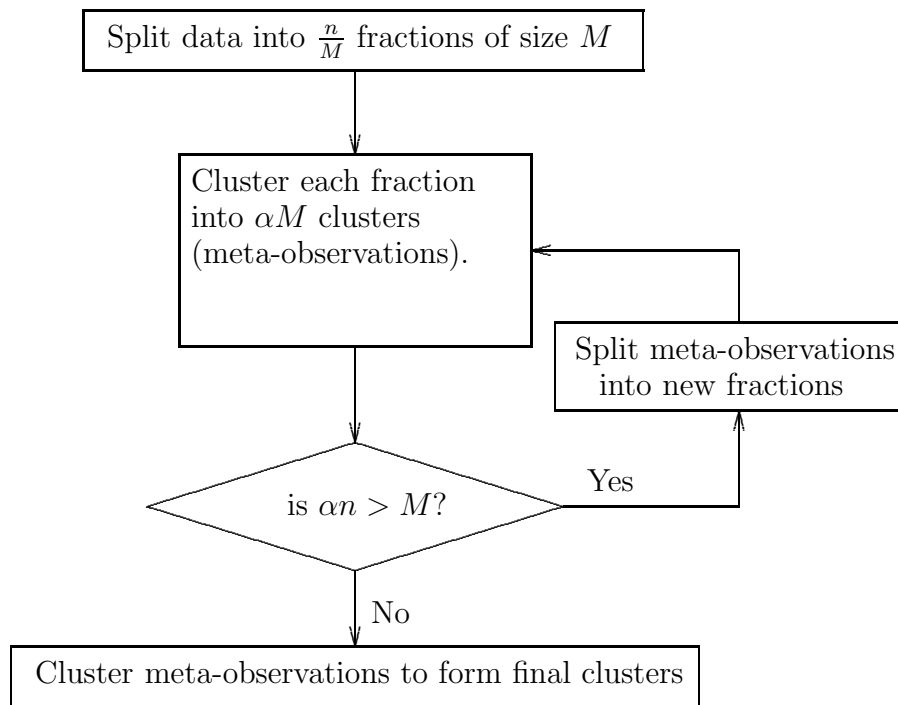


Figure 5.1: Fractionation Algorithm

5.1 *Model Based Fractionation*

If we use hierarchical model-based clustering as the base clustering method in Fractionation, then we get *Model Based Fractionation*. The main difference between the original Fractionation method of Cutting et al. (1992) and model-based Fractionation is that in model-based Fractionation a meta-observation is not characterized just by a mean, but by all the sufficient statistics (the mean, the covariance, and the number of observations in the cluster); we retain not only the location of each cluster, but also its size, shape and volume.

A nice feature of model-based Fractionation is that it only differs slightly from hierarchical model-based clustering. The only difference is that in step 2 of the Fractionation algorithm the search for the closest pair of clusters is restricted to clusters within the same fraction, whereas model-based clustering would search for the closest pair across the entire data set.

In our application the number G of groups is typically unknown. We use the BIC in step 4 of the procedure to choose the number of mixture components.

5.2 *Illustration*

Consider a data set in the plane consisting of 400 observations in 4 Gaussian groups (Figure 5.2). Suppose that the order of magnitude of the largest practical problem size for our hierarchical clustering procedure was $M = 100$ and that we chose $\alpha = 0.1$.

We randomly split the data into 4 fractions of equal size. Each fraction is clustered into $\alpha M = 0.1 \times 100 = 10$ meta-observations, shown in Figure 5.3. The circles in this and the following figures are isopleths of the component densities containing 95% of the mass.

As the total number of meta-observations (40) is less than M no further iterations of fractionation are necessary. Instead we can just cluster the 40 meta-observations. The BIC chooses 4 mixture components, which are shown in figure 5.4.

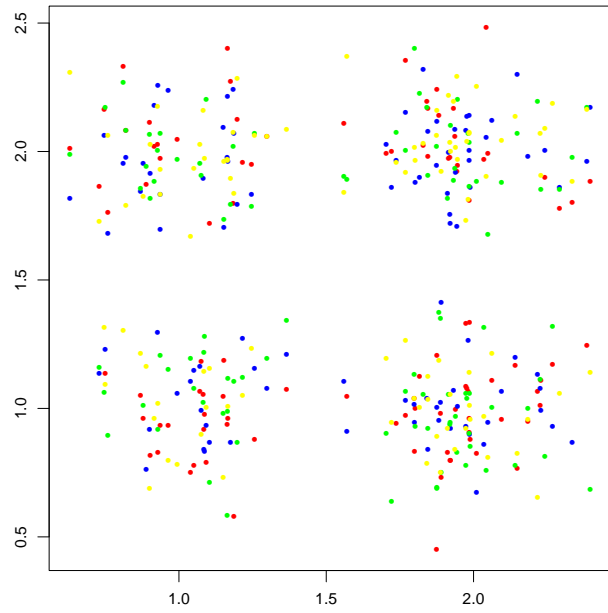


Figure 5.2: The data with each fraction shown in a different color.

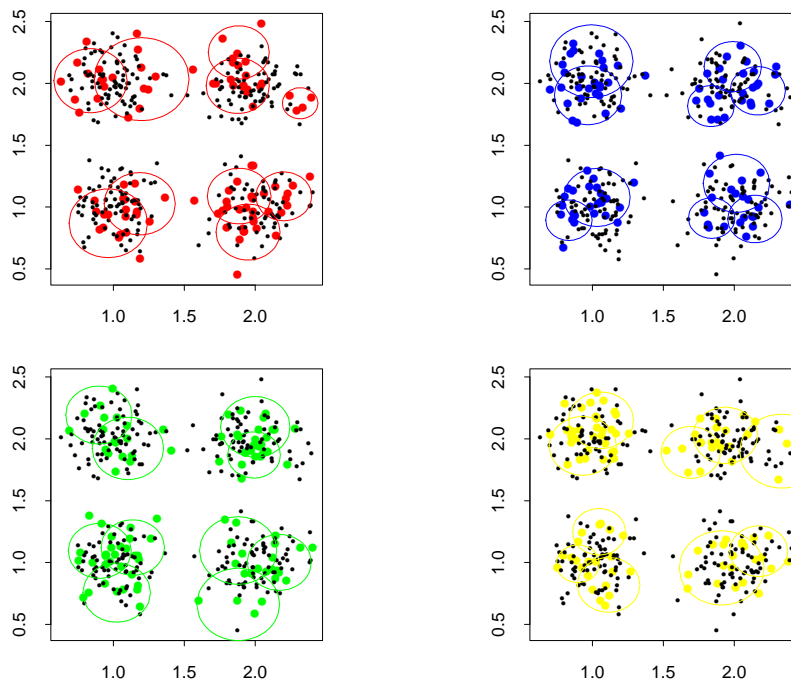


Figure 5.3: The data from each fraction with the meta-observations superimposed.

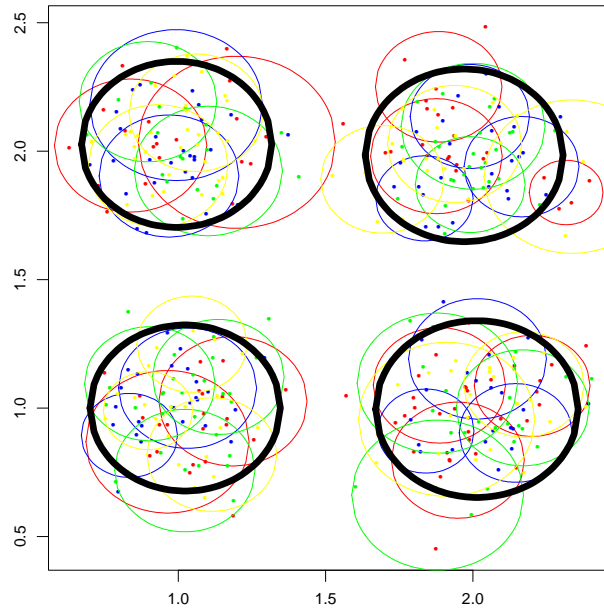


Figure 5.4: All 40 meta-observations and the final 4 clusters chosen by the BIC.

Chapter 6

MODEL BASED REFRACTIONATION

A major problem with Fractionation is that once observations from different groups have been assigned to the same meta-observation this error will never be corrected. Such erroneous assignments are less likely to occur if fractions are pure, that is; contain observations from few groups or, equivalently, if groups are split over few fractions. We could form purer fractions if we knew the group labels of the observations. This observation suggests applying Fractionation repeatedly and forming the fractions for Step 1 of the i -th pass based on the clustering produced in the $(i - 1)$ st pass. Conceptually, Step 4 of the Fractionation algorithm is replaced by two steps, both involving hierarchical model based clustering of the meta-observations generated by Step 3:

4a Cluster the meta-observations into G clusters, where G is determined by the BIC.

4b Define the fractions for the i -th pass: as soon as a cluster formed during the merging step in hierarchical model based clustering represents more than $M/2$ observations, make those observations into a fraction and remove the cluster from the merge process.

We stop the Refractionation passes when the changes in the number G of clusters and the cluster compositions is small enough. The fraction sizes will vary between $M/2$ and M .

There are two considerations for choosing the fraction size M . The total run time of a fractionation pass is proportional to nM and therefore smaller fraction size results in a shorter run time. On the other hand, the goal of refractionation is to purify the fractions, i.e. to reduce the number of groups which are split across multiple fractions. This is easier if the fraction size M is large. As a compromise we choose M as large as computationally practical, but choose the initial fractions sizes to be close to $M/2$.

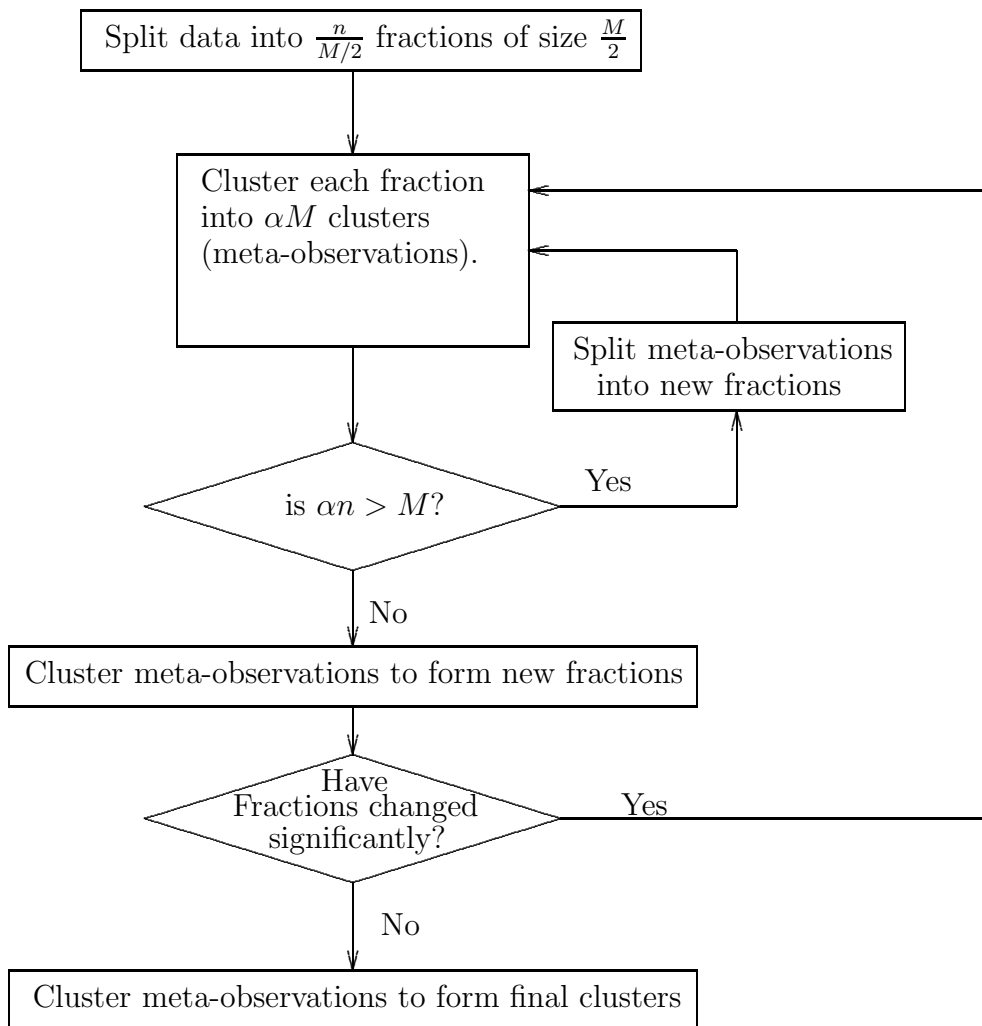


Figure 6.1: Refractionation Algorithm

6.1 *Illustration*

Consider an example in two dimensions with 25 equally spaced Gaussian groups containing 16 points each (Figure 6.2). We choose $\alpha = 0.1$ and kept the number of fractions constant at four to simplify the exposition.

We randomly split the data into four fractions of 100 observations each (Step 1 of the Fractionation algorithm), and then use model based hierarchical clustering to partition each fraction into 10 clusters (Step 2 of the algorithm). The fractions and their clusters (meta-observations) are shown in Figure 6.3.

As the number of meta-observations produced by clustering the fractions in this case is 40, no further iterations of fractionation are necessary and we can therefore proceed to steps 4a and 4b.

Partitioning the 40 meta-observations into 17 clusters as chosen by the BIC (Step 4a) produces the mixture model whose component densities are shown in Figure 6.4. Clearly, this clustering in no way reflects the structure of the data, meaning that Fractionation would fail.

Clustering the 40 meta-observations into new fractions (Step 4b) results in fraction sizes of 97, 108, 104, and 91. Figure 6.5 shows the new fractions.

We now start the second pass of Fractionation. Each of the new fractions is partitioned into 10 clusters (Step 2) shown in Figure 6.6.

Partitioning the 40 meta-observations into 25 clusters as chosen by the BIC (Step 4a) produces the mixture model shown in Figure 6.7. We have essentially recovered the structure of the data.

A third pass of Fractionation (Figures 6.8, 6.9 and 6.10) leads to almost the same mixture model (Figure 6.10) as the second pass (Figure 6.7), and the Refractionation process stops.

Table 6.1 gives numerical summaries of the purity of the fractions. At the beginning of the first Fractionation pass, each of the 25 groups is scattered over all four fractions, whereas

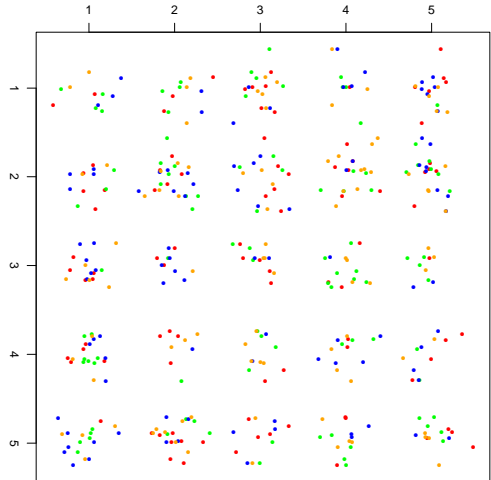


Figure 6.2: Observations and with fraction membership shown in color.

Table 6.1: The distribution of the number of fractions across which groups are scattered at the start of each Fractionation pass.

Pass	Min	Median	Max	> 1	> 2
1	4	4	4	25	25
2	1	1	2	10	0
3	1	1	2	1	0

at the beginning of the third pass only one of the groups is split across multiple fractions.

6.2 Properties of Fractionation and Refractionation

We can make statements about the performance of Fractionation and Refractionation in a highly idealized situation where there is no within group scatter – the observations within each group are exactly tied.

Fractionation is guaranteed to succeed if the number of groups G is no bigger than the number αM of clusters generated from each fraction. This is because in this case the algorithm will never merge (meta-)observations from different groups.

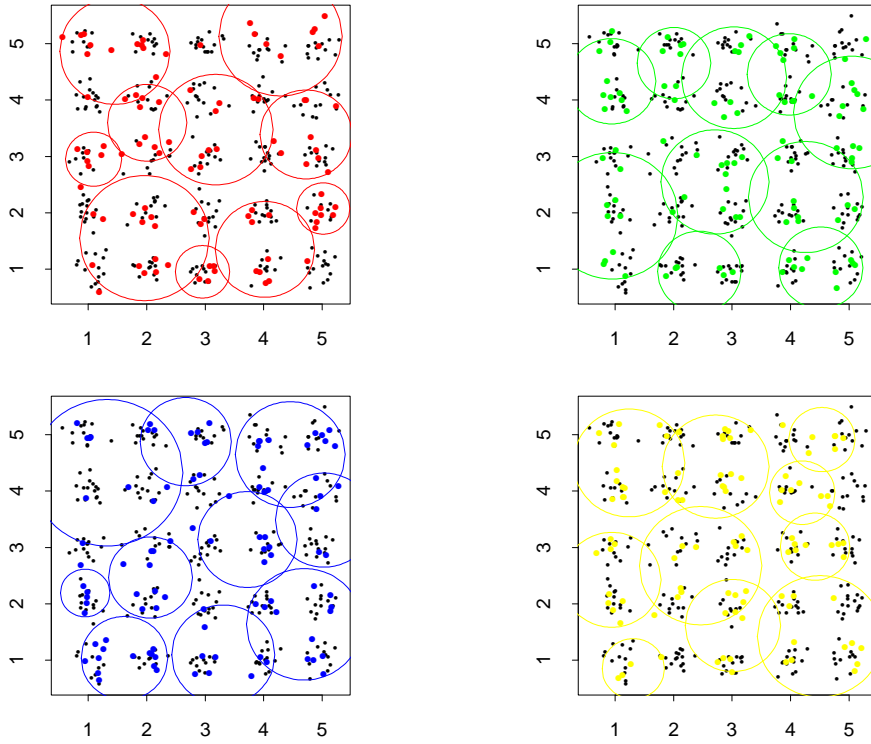


Figure 6.3: Meta-observations obtained by clustering the initial four fractions.

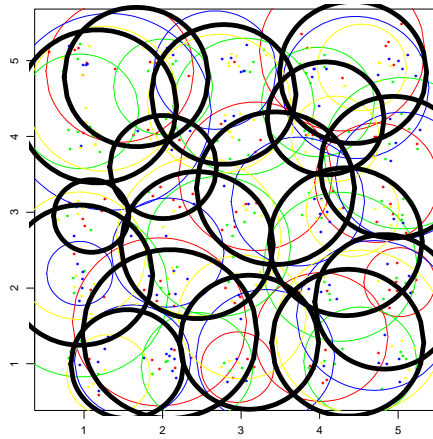


Figure 6.4: Clusters chosen by the BIC after the first pass of Fractionation.

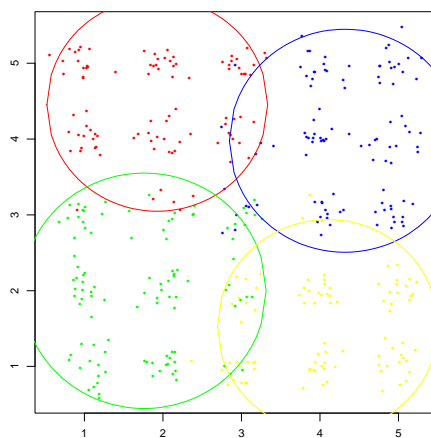


Figure 6.5: Fractions formed by the first pass of Fractionation.

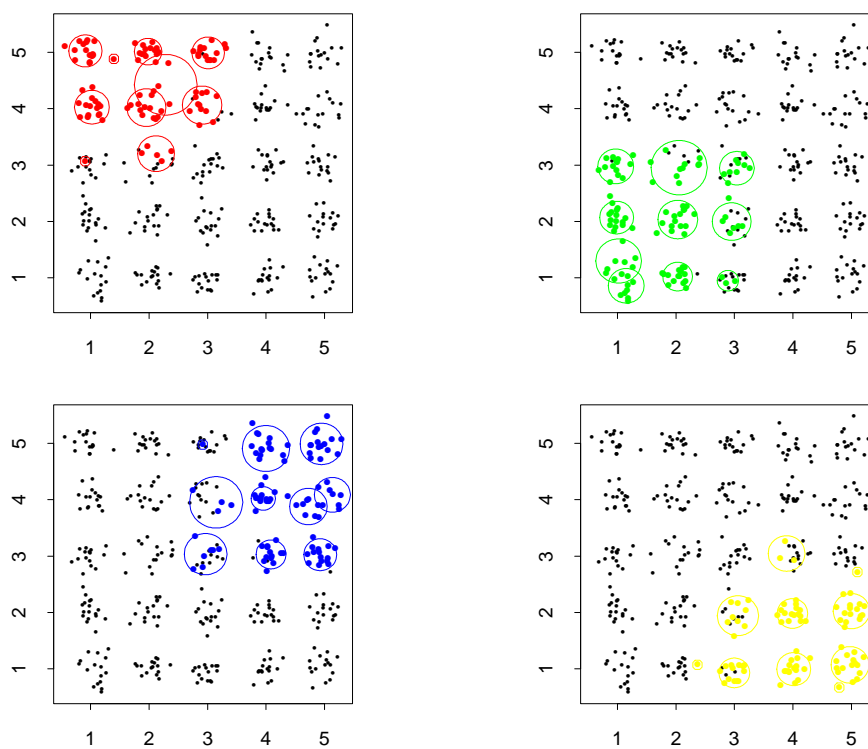


Figure 6.6: Meta-observations obtained by clustering the four fractions in the second pass of Fractionation.

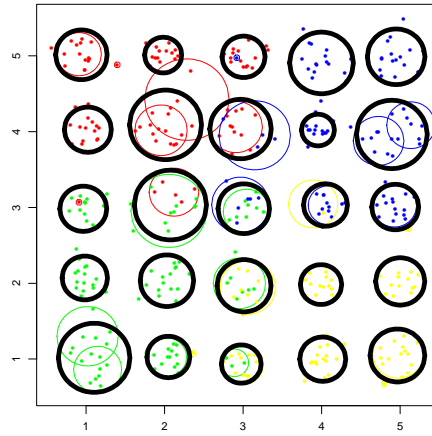


Figure 6.7: Clusters chosen by the BIC after the second pass of Fractionation.

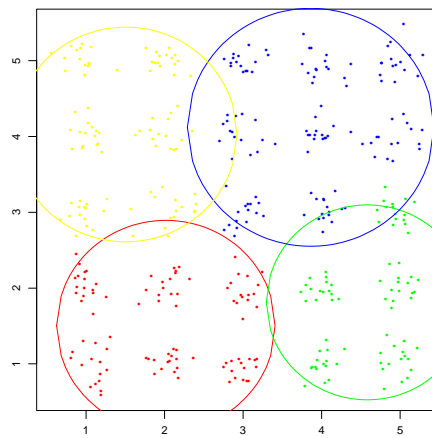


Figure 6.8: Fractions formed by the second pass of Fractionation.

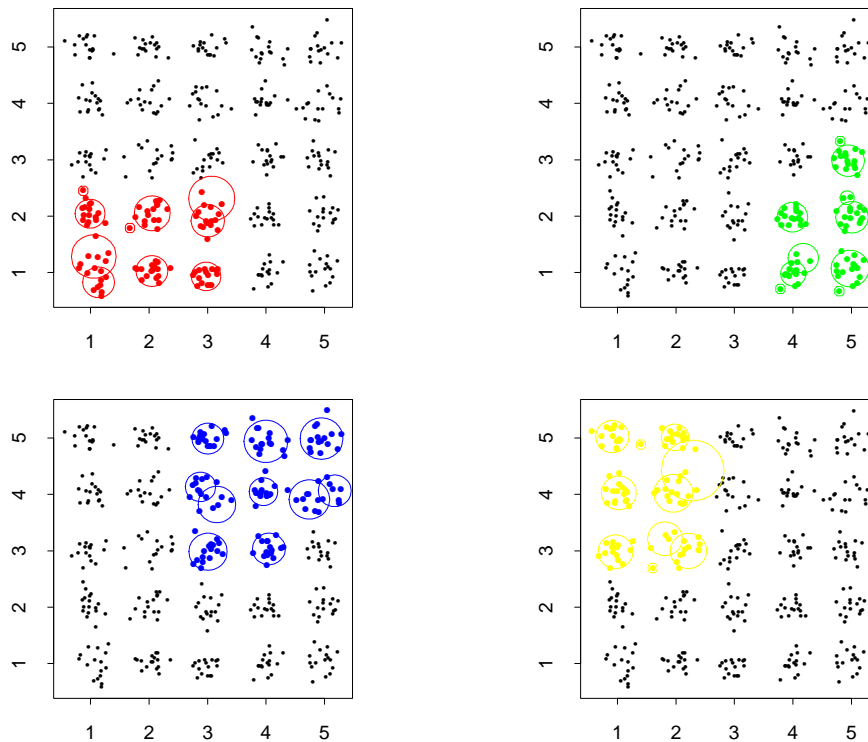


Figure 6.9: Meta-observations obtained by clustering the four fractions in the third pass of Fractionation.

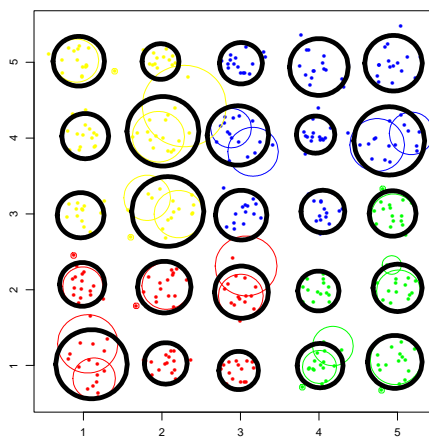


Figure 6.10: Clusters chosen by the BIC after the third pass of Fractionation.

Refractionation is guaranteed to fail if the number of groups is larger than αn , which is the number of clusters generated from each fraction times the number of fractions. If there are that many groups then it will not be possible for all fractions to have less than αM groups. Conversely, Refractionation may succeed if it reaches a state where each fraction has no more than αM groups.

As the example in Section 6.1 shows, there are instances where Fractionation fails, but Refractionation succeeds.

Chapter 7

FRACTIONATION AND REFRACTIONATION EXAMPLES

In this chapter we will show some examples for the use of Fractionation and Refractionation, all based on the TDT data set.

We will first use the “1100 collection” to simulate large data sets and then cluster the full TDT collection.

7.1 Fractionation Example 1

To create the data for this example, we estimated the mean vector and covariance matrix for each of the 19 groups in the TDT dataset. We then generated 20 times the number of observations in each group from a Gaussian distribution with the group mean vector and covariance matrix. This gave a dataset with $n = 22,000$ observations. We randomly partitioned the data into 22 fractions of $M = 1,000$ observations each, and clustered the fractions into $M/10 = 100$ clusters. As the number of groups (19) is small relative to the number of clusters generated in each fraction, one pass of Fractionation was sufficient; no Refractionation was needed. We determined the final number of clusters using the BIC criterion. We repeated this experiment — generating a sample of size 22,000 and clustering it — ten times. In these ten replications, the BIC criterion chose 19 clusters 5 times, 20 clusters 4 times and 21 clusters once. The average Fowlkes-Mallows index for the runs that chose 19 clusters was 0.9955 and for those that chose 20 it was 0.9932 (see Figure 7.1), indicating almost perfect agreement between groups and clusters. This is reassuring — after all, the data were generated from a Gaussian mixture, and we would hope that model based clustering would do well.

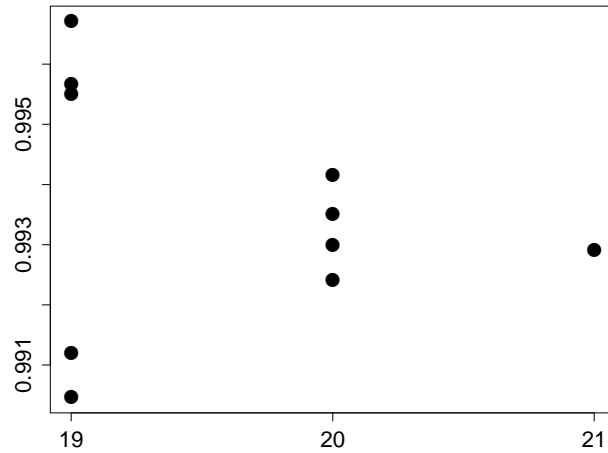


Figure 7.1: Fowlkes-Mallows index vs number of clusters chosen by the BIC for the data set of Fractionation Example 1.

7.2 Fractionation Example 2

The data in this example were obtained by estimating each group density by a kernel density estimate (Scott 1992) and then sampling from this estimate, again generating 20 times the number of observations in the group. We used a Gaussian kernel with the same covariance matrix as the corresponding group scaled by a factor of 1/10. As in Fractionation Example 1 this resulted in a dataset of 22,000 observations, but the data are no longer sampled from a mixture of 19 Gaussians. We clustered the dataset using one pass of Fractionation and the BIC criterion for choosing the final number of clusters. In ten replications of the experiment the BIC criterion chose 21 clusters once, 22 clusters 3 times and 23 clusters 6 times. The values of the Fowlkes-Mallows index were between 0.83 and 0.94 (see Figure 7.2.)

7.3 Refractionation Example 1

Fractionation Examples 1 and 2 are easy: the number of groups is small, and all the groups are large. They could certainly have been recovered by clustering a random sample of manageable size. Refractionation example 1 is more challenging.

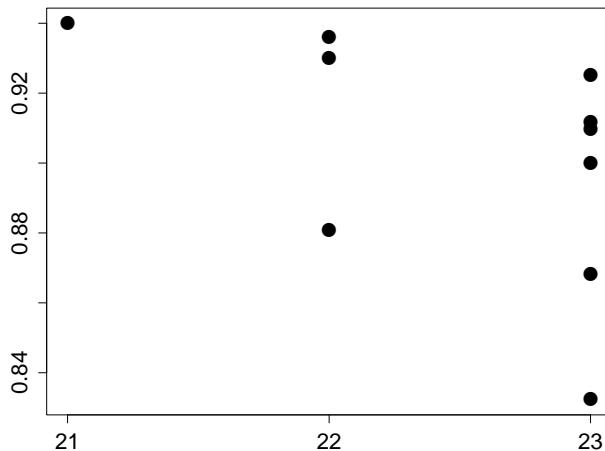


Figure 7.2: Fowlkes-Mallows index vs number of clusters chosen by the BIC for the data set of Fractionation Example 2.

We generated the data for this example by essentially replicating the labeled TDT dataset 19 times, replacing each group by a scaled and shifted version of the entire dataset: Let μ_i and Σ_i be the mean vector and covariance matrix of the i -th group. We obtained the i -th replicate by scaling and shifting the entire dataset to have mean vector μ_i and covariance matrix Σ_i . We end up with $19 \times 19 = 361$ groups and $19 \times 1100 = 20,900$ observations.

We randomly split these 20,900 observations into $M = 20$ fractions of 1,045 observations each and clustered fractions into 100 clusters. Because the number of groups (361) is larger than the number of clusters per fraction (100), and initial fractions typically contain observations from more than 100 groups, a single pass through Fractionation does not result in a good clustering of the data, and Refractionation is necessary.

Table 7.1 shows the Fowlkes-Mallows index of the clustering into 361 clusters after the first four passes through Fractionation. The index almost doubles, indicating that the agreement between groups and clusters improves dramatically. This improvement goes along with an equally drastic decrease in the number of non zero entries in the 361×361 contingency table (comparable to Table 4.3).

Tables 7.2 and 7.3 confirm that Refractionation indeed increases the purity of the fractions.

Table 7.1: Refractionation Example 1 – agreement between clusters and groups after each Fractionation pass.

Pass	Fowlkes	non zero
	Mallows	entries
1	0.325	1729
2	0.554	908
3	0.616	671
4	0.613	651

Table 7.2: Refractionation Example 1 – distribution of the number of fractions in which groups are represented, at the start of each Fractionation pass.

Pass	Min	Median	Max	> 1	> 2
1	6	18	20	361	361
2	1	4	10	350	287
3	1	1	3	68	7
4	1	1	2	41	0

Table 7.3: Refractionation Example 1 – distribution of the number of groups represented in each fraction at the start of each Fractionation pass. Here n_f is the number of fractions and the last column is the minimum of the average number of groups per fraction.

Pass	Min	Median	Max	n_f	$361/n_f$
1	270	289	296	20	18.0
2	18	88	150	18	20.1
3	18	19	60	17	21.2
4	19	19	58	16	22.6

Table 7.2 shows that, initially, groups are scattered over many fractions, while after the fourth pass through Fractionation 320 of the 361 groups are contained entirely in a single fraction, and the remaining 41 groups are each split across two fractions.

Table 7.3 gives the number of groups represented in each fraction at the beginning of each Fractionation pass. At the beginning of the first pass the least diverse fraction contains observations from 270 groups, and the most diverse fraction contains observations from 296 groups. The median number of groups per fraction is 289. In contrast, at the beginning of the fourth Fractionation pass the least diverse fraction contains observations from 19 groups, and the most diverse fraction contains observations from 58 groups. The median number of groups per fraction is 19. These numbers again demonstrate how successful Refractionation is at purifying the fractions. There is no change in clustering after the 4th run of fractionation.

The groups in this example are too small to fit a mixture model whose components have unconstrained covariance matrices: in 50 dimensions we need at least 51 observations to obtain a non-singular sample covariance matrix. In order to avoid this problem, we constrained the covariance matrices of the mixture components to be diagonal. The results above indicate that this works well if we make the algorithm produce the correct number of clusters, 361.

Fitting accurate and parsimonious mixture models to datasets with many small groups requires Mixtures of Factor Analyzers (Hinton et al. 1997; McLachlan and Peel 2000), combined with a criterion like BIC to estimate the number of components. As we have not implemented this approach, we shall instead consider a simulated example where the group covariance matrices are diagonal, and therefore a mixture model with diagonal covariance matrices will fit well.

7.4 Refractionation Example 2

We generated the data for this example in a way very similar to Refractionation Example 1. However, we replaced the observations in each of the 361 groups by simulated data

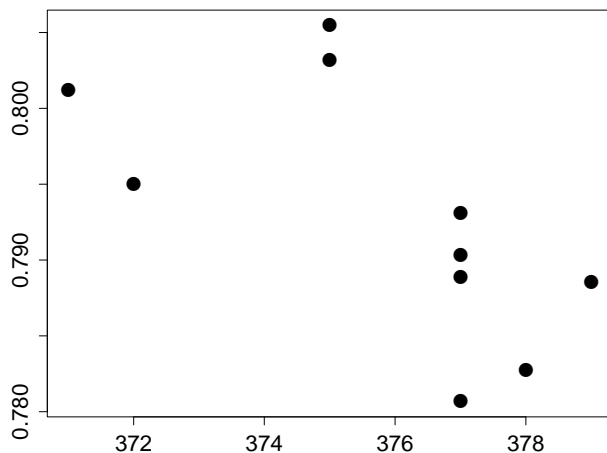


Figure 7.3: Fowlkes-Mallows index vs number of clusters chosen by the BIC for Refractionation Example 2.

from a multivariate Gaussian distribution with the same mean as the group and a diagonal covariance matrix obtained by setting the off-diagonal elements of the sample covariance matrix to zero.

We simulated ten datasets of size 20,900 from this mixture distribution with 361 axis-parallel components and clustered them with the same algorithm as in Refractionation Example 1, except that we chose the number of clusters using the BIC criterion. In ten replications of this experiment, the number of clusters chosen by the BIC criterion was between 371 and 379, with values of the Fowlkes-Mallows index between 0.781 and 0.806 (see Figure 7.3.) The Fowlkes-Mallows index for models with 361 clusters is between 0.797 and 0.820, which is only slightly better than for the models obtained using the BIC criterion.

7.5 Clustering the entire TDT Data Set

For this example we clustered the entire set of 15,863 TDT documents in 100-dimensional space. The documents are ordered according to the time and date of the story. We used this ordering to split the data into 15 fractions of approximately 1000 documents each and then applied the refractionation process.

Only 1,100 of the 15,863 documents are labeled, which raises the issue of measuring the quality of clustering for partially labeled data. Our first suggestion is to use the labeled documents only: each labeled document has both a topic label and a cluster label, and we can measure agreement between these labels using the Fowlkes-Mallows index. We call this quality measure the FM1 index.

The results are shown in Table 7.4. The second column gives the agreement between the clusterings obtained in successive passes of fractionation. After 10 passes the agreement no longer improved, which led us to stop the refractionation process.

The third column shows the FM1 index for successive passes of fractionation. There is no improvement in the index, which suggests that refractionation was either unsuccessful or unnecessary. We suspect the latter. Our explanation for the lack of improvement is that documents on the same topic are concentrated in time, and therefore the initial fractions (obtained by splitting the collection according to time) were already quite pure.

To verify this explanation we repeated the experiment, but randomly assigned documents to the initial fractions. The results are shown in Table 7.5. This time there is a significant increase in the FM1 index over successive passes of fractionation. The final values of the FM1 index for the two different choices of initial fractions are close, suggesting that the penalty for a poor choice of the initial fractions is small.

A criticism of the FM1 index is that it only reflects “recall” and ignores “precision”. Consider the ideal situation where the labeled documents for each topic are concentrated in exactly one cluster in such a way that labeled documents from different topics fall in different clusters. This situation would result in an FM1 index of 1, no matter how many additional unlabeled documents are contained in these clusters. This observation suggests an alternative measure of goodness of clustering, which we call the FM2 index. We collect all documents in all the clusters containing labeled observations. If a document in this collection was not assigned to a topic then we label it as “*other*”. Each of the documents now has two labels, the topic label (possibly *other*) and the cluster label, and we can measure agreement between them using the Fowlkes-Mallows index. Since the FM2 index uses the

same data as the FM1 index plus the additional “topic” *other*, the value of FM2 cannot exceed that of FM1.

The labeled subset of the TDT collection was generated by selecting a list of topics and then identifying all the documents referring to those topics. Consequently, the clusters containing labeled documents should have no unlabeled documents, and the FM2 index should be close to FM1 in value. This turns out not to be the case. The optimal number of clusters according to the BIC criterion is 209, and the FM2 index for the corresponding clustering is only 0.12. Even in clusters containing documents from one of the selected topics those documents form a minority.

We do not have a conclusive explanation for this contradiction. Applying the tools described in the subsequent chapters suggests that these clusters do correspond to distinct groups in the data.

Pass	Similarity between consecutive clusterings	Fowlkes-Mallows (FM1) Index
1		0.545
2	0.269	0.516
3	0.475	0.539
4	0.557	0.530
5	0.624	0.492
6	0.626	0.530
7	0.667	0.530
8	0.692	0.535
9	0.711	0.541
10	0.733	0.546
11	0.727	0.545

Table 7.4: Improvement of clustering and change in clusters for the 15,863 TDT documents when using time for the original fractioning. The first column shows the Fowlkes-Mallows index associated to consecutive clusterings generated by consecutive refractionation passes. The last column refers to the FM1 index associated to the labeled data.

Pass	Similarity between consecutive clusterings	Fowlkes-Mallows (FM1) Index
1		0.479
2	0.248	0.500
3	0.528	0.517
4	0.539	0.517
5	0.594	0.526
6	0.625	0.524
7	0.621	0.530
8	0.665	0.555
9	0.665	0.518
10	0.711	0.544
11	0.713	0.569
12	0.626	0.536

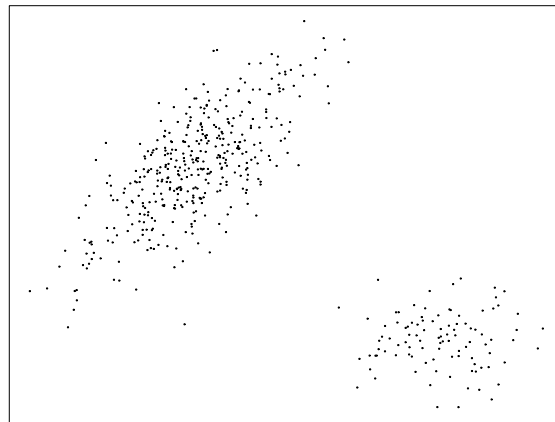
Table 7.5: Improvement of clustering and change in clusters for the 15,863 TDT documents when using a random assignment for the initial fractions. The first column shows the Fowlkes-Mallows index comparing clusterings generated by consecutive refractionation passes. The last column shows the FM1 index for the labeled data.

Chapter 8

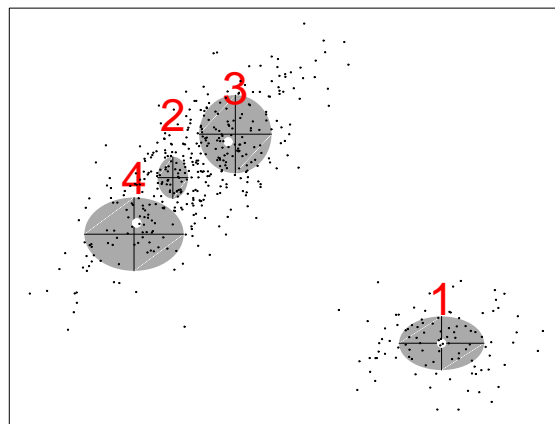
ASSESSMENT AND VISUALIZATION

The goal of clustering is to identify distinct groups in a dataset. Model based clustering relies on the premise that there is a one to one correspondence between groups in the data and mixture components in the model. This assumption could be violated for several reasons. First, groups might not be Gaussian. An isolated group with a non-elliptical distribution might be modeled by not one, but several mixture components, and the corresponding clusters would no longer be distinct. Second, the groups might be Gaussian but our covariance model might be too restrictive. This problem seems hard to avoid in high dimensions, where estimating unrestricted covariances requires large group sizes. The problem is illustrated in Figure 8.1a. There are two Gaussian groups, the left one with a non-diagonal covariance matrix. If we fit a mixture model with diagonal covariance matrices, the BIC criterion chooses a model with three components for this group; Figure 8.1b shows regions containing 60% of the mass of each component.

In this chapter we present methods for assessing the degree of separation between the components of a mixture model and between the corresponding clusters.



(a)



(b)

Figure 8.1: Data set with fitted Gaussian mixture. The modes of the mixture are indicated by the three white dots. (This example is referred to as the running example in the remainder of the thesis.)

8.1 Assessing separation between mixture components

Roughly speaking, we would expect mixture components modeling different groups in the data to be well separated. On the other hand, mixture components modeling parts of the same group would be expected to exhibit significant overlap.

We now put this concept in probability terms. We can generate observations from a mixture density $\sum_g \pi_g p_g(\mathbf{x})$ by first generating a component label Y with $P(Y = g) = \pi_g$, and then

generating X from p_Y . According to Bayes rule, the posterior probability $P(Y = g|X)$ is

$$P(Y = g|X) = \frac{\pi_g p_g(X)}{\sum_{j=1}^G \pi_j p_j(X)}.$$

Component g is well separated from all the other components if $P(Y = g|X)$ only takes extreme values, either close to zero or close to one - one for observations actually generated from component g , and zero for all others.

Exactly evaluating the distributions of $P(Y = g|X)$ for the G components is impossible when the dimension m is larger than 1, and hence we resort to Monte Carlo simulation.

In the following we present three methods for assessing the separation between mixture components, based on the posterior probabilities, the margins, and the misclassification probabilities.

8.1.1 Assessing separation using posterior probabilities

Figure 8.2 shows rootograms of the posterior probabilities $P(Y = g|X)$ for the four components of the mixture model in our running example. (A rootogram is a variant of a histogram where the heights of the bars encode the square roots of the bin counts, instead of the bin counts themselves. This makes low counts more visible.) These rootograms are based on a sample of 20,000 observations from the estimated mixture model. We have omitted the bin containing $P(Y = g|X) = 0$ in the rootograms, because it would have by far the largest bin count and would obscure the information in the remaining bins.

The rootogram for component one has a large peak at $P(Y = 1|X) = 1$ and is essentially zero elsewhere, indicating clear separation of component one from all the other components. On the other extreme, the rootogram for component two has no peak at $P(Y = 2|X) = 1$. This is due to the fact that component two is completely overlapped by components three and four, and hence there is always a substantial posterior probability that an observation generated from p_2 might have come from p_3 or p_4 . Furthermore, the significant mass away from $P(Y = g|X) = 1$ in the rootograms for components two, three, and four shows that these components are not well separated.

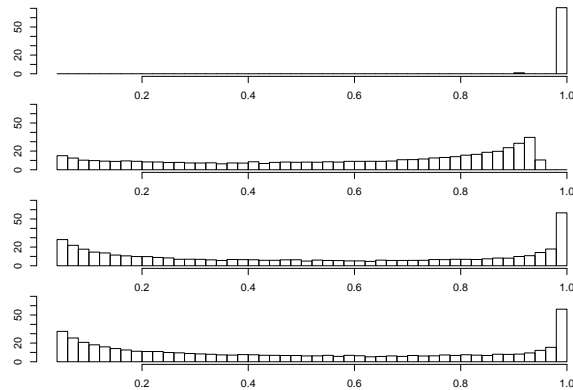


Figure 8.2: Running example: Rootograms of the posterior probabilities $P(Y = g|X)$ for X distributed according to the mixture model.

8.1.2 Assessing separation using margins

An alternative to looking at the posterior probabilities is to consider the margins. Let $\hat{Y}(X)$ be the estimated component label assigned to X by Bayes' rule:

$$\hat{Y}(X) = \arg \max_g P(Y = g|X).$$

The margin of X drawn from component Y of the model is given by

$$\text{margin}(X, Y) = P(\hat{Y}(X) = Y|Y) - \max_{g \neq Y} P(\hat{Y}(X) = g|Y).$$

Note that a negative margin means that X is assigned to the wrong component, and that a small margin means that X lies in a region where components overlap significantly.

Figure 8.3 shows the cumulative distribution function (cdf) of the margin for observations drawn from the four component mixture model of our running example. There is a substantial proportion of small margins (below < 0.5) indicating substantial overlap between the components.

8.1.3 Assessing separation using misclassification probabilities

When the number of clusters is moderate, we can look at the misclassification matrix to detect well separated as well as overlapping components of a mixture model. Table 8.1 shows

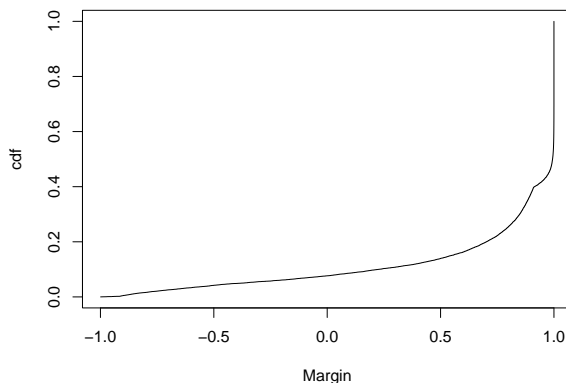


Figure 8.3: Running example: Cumulative distribution function of the margin.

	1	2	3	4	MC_g	π_g
1	1	0	0	0	0	0.196
2	0	0.923	0.086	0.107	0.193	0.202
3	0	0.035	0.900	0.022	0.057	0.404
4	0	0.042	0.014	0.871	0.056	0.198

Table 8.1: Misclassification matrix for the running example.

the misclassification matrix for the mixture model in our running example. Let $m_{gg'}$ be the probability that the Bayes rule assigns an observation from component g to component g' .

From the misclassification matrix we can extract information at three different levels of detail. At the coarsest level we can look at the overall misclassification probability given by $\sum_g \pi_g (1 - m_{gg})$. The lower this probability is, the better the separation. At the next higher level of detail, we can look at the component-wise misclassification probabilities MC_g . In our example (Table 8.1) the misclassification probability for component one is zero ($MC_1 = 0$), indicating that component one is well separated. The misclassification probabilities for the other components are substantially larger. On the most detailed level, the values of $m_{gg'}$ and $m_{g'g}$ indicate which other components overlap component g . The pattern of entries in Table 8.1 shows that components two, three and four are mutually overlapping. We could not see this from the less detailed views.

8.2 Assessing separation between clusters

A mixture model is only an estimate for the true underlying density of the data. Therefore the degree of separation between mixture components (or lack thereof) does not always accurately reflect the actual separation between the clusters.

We cannot compute the matrix of misclassification probabilities for the observed data $\mathbf{x}_1, \dots, \mathbf{x}_n$, nor the margins, because those require knowing the true labels. However we can compute the posterior probabilities $P(Y = g|\mathbf{x}_i)$, and therefore generate a plot analogous to Figure 8.2, shown in Figure 8.4. The rootogram for $P(Y = 2|\mathbf{x}_i)$ looks basically flat, from which we can conclude that cluster 2 almost certainly does not correspond to a distinct group in the data.

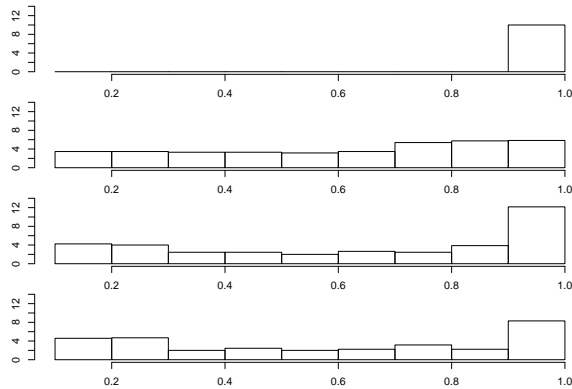


Figure 8.4: Running example: Rootograms of the posterior probabilities $P(Y = g|\mathbf{x}_i)$ for the data.

Chapter 9

HYBRID CLUSTERING

As pointed out in Chapter 8, there is a conceptual problem with model based clustering. If the groups in the data are either not Gaussian or Gaussian with a more complicated covariance structure than is assumed in the mixture model, then those groups will be represented by not one, but several mixture components.

In this chapter we propose a hybrid clustering algorithm that shares some of the advantages of model based and of non-parametric clustering. The algorithm starts with the tree corresponding to the mixture model chosen by the Bayesian Information Criterion. It then progressively merges clusters that do not appear to correspond to different modes of the data density.

9.1 The Hybrid Clustering Algorithm

Hierarchical Model Based Clustering generates a hierarchy of mixture models: The model with $m - 1$ mixture components is obtained by merging the two clusters of the m component model for which the change leads to the smallest decrease in log-likelihood. The result of this merging process can be represented by a binary tree T . The leaves of the tree are the observations. Each interior node N of the tree is assigned a *generation* between 1 and $n - 1$, indicating where in the sequence of merges it was generated. The interior node corresponding to the i -th merge in the sequence is assigned generation $n - i$; the root node therefore has generation 1. Each node N is also associated with the cluster formed by its descendent leaves.

The merge sequence defines a sequence of trees: T_m is obtained from T by pruning the

offspring of all nodes with generation greater than or equal to m . By construction, T_m has m leaves and corresponds to a mixture model with m mixture components. Let G be the number of mixture components chosen by the BIC, and let T_G be the corresponding tree.

If the distinct groups in the data all have Gaussian distributions with a covariance structure compatible with the fitted model, then we expect roughly a one-to-one correspondence between groups and mixture components associated with the leaves of T_G . Also, the clusters associated with the leaves of T_G will be similar to the groups. (“Roughly” because G , after all, is only an estimate.) If the groups are not Gaussian or the covariance structure of the model is incorrect, however, each group may be modeled by more than one mixture component, and consequently will be the union of several clusters.

The idea of hybrid clustering is to test, for each node of T_G whose daughters are leaves, whether the corresponding clusters are well separated. If they are not, then the clusters probably correspond to the same group, and we merge them. The new cluster is then modeled by the sum of the mixtures modeling the daughters that were merged. This pruning process is repeated until no further clusters can be merged.

9.2 Illustration of hybrid clustering

Before describing its ingredients in more detail, we present the pruning process in action. The upper panel of Figure 9.1 shows the tree whose leaves correspond to the mixture model fit to the data in our running example. The circled node is the one being tested. The lower panel of Figure 9.1 shows the projection of its associated cluster onto the *Fisher discriminant direction*, which is the direction that best separates the projections of the two daughter clusters (Gnanadesikan, Kettenring, and Landwehr 1982; Mardia et al. 1979, Chapter 11.5). The grey curve is the kernel density estimate for the projected data with the smallest bandwidth that yields a unimodal density (Silverman 1986, Chapter 6.3 and 6.4). The black curve is the kernel density estimate with the smallest bandwidth that yields a bimodal density. We shall call this plot a *GKL-Silverman plot* (GKL for Gnanadesikan, Kettenring, and Landwehr (1982)) who first suggested projecting onto discriminant coordinates. The

dot plot of the projected data looks unimodal, and the unimodal and bimodal distributions are almost identical, which indicates that the daughter clusters are not well separated in feature space. A formal test for unimodality of the projected data (Section 9.3) would reject the null hypotheses of unimodality at level $\alpha = 0.988$, meaning that the evidence against unimodality is weak. We therefore prune the daughters. The new tree is the one shown in black in Figure 9.2. The diagnostic plot is qualitatively similar to the one in Figure 9.1; the daughter clusters of the node being tested do not seem to be well separated, with unimodality being rejected at level $\alpha = 0.22$. We therefore prune again and are left with the tree shown in Figure 9.3. Now the picture is different: The diagnostic plot reveals a clear separation between the clusters, and a formal test rejects the hypothesis of unimodality at level $\alpha = 0.002$. We conclude that there appear to be two distinct groups in the data, one modeled by three mixture components, and the other one modeled by one mixture component.

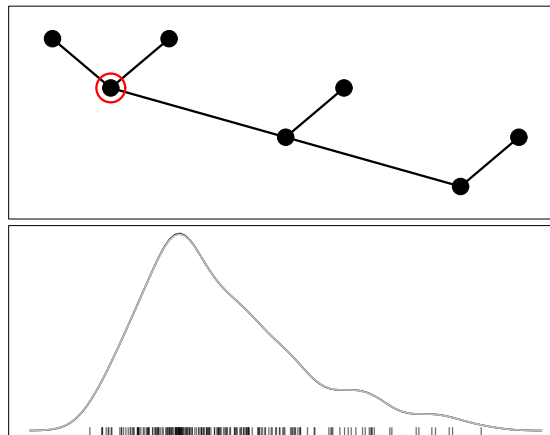


Figure 9.1: Running example: Tree generated by hierarchical model based clustering and diagnostic plot for the circled node.

9.3 Testing for unimodality

In order to automate the pruning process described in Section 9.2 we need a way of measuring the amount of evidence against unimodality for a univariate data set (the projection of a

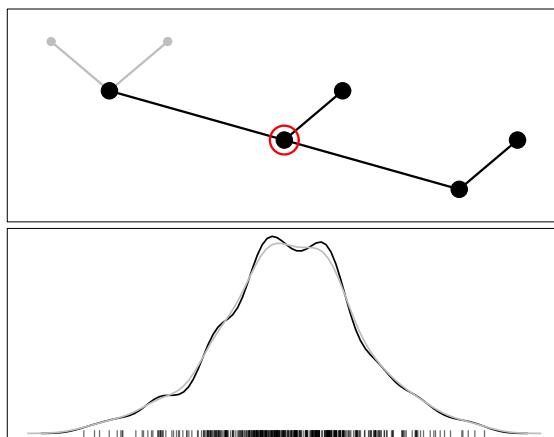


Figure 9.2: Running example: Tree generated by hierarchical model based clustering after first step of pruning, and diagnostic plot for the circled node.

cluster onto the Fisher discriminant direction best separating its daughters). Even if we carry out the pruning process interactively, by looking at diagnostic plots like the ones in Figures 9.1-9.3, such a measure of evidence still provides a useful guideline.

Let x_1, \dots, x_n be a set of (univariate) data sampled from some density $f(x)$, and let $F_n(x)$ be the empirical cdf of the sample. To test the null hypotheses that $f(x)$ is unimodal we use the DIP test of J.A. Hartigan and P.M. Hartigan (1985). The test statistic is the DIP

$$D = \sup_x |F_n(x) - H(x)|,$$

where H is the unimodal cdf closest to F_n . Bickel and Fan (1996) show that the non-parametric maximum likelihood estimate of the closest unimodal cdf, given the mode location m_0 , is the greatest convex minorant of F_n on $(-\infty, m_0]$ and the least concave majorant on $[m_0, -\infty)$. (The greatest convex minorant of F_n on $(-\infty, m_0]$ is the convex function G not exceeding F_n on $(-\infty, m_0]$ that minimizes $\sup_{x \leq m_0} |F_n(x) - G(x)|$. The least concave majorant is defined analogously.)

Bickel and Fan (1996) also show that this estimate is robust against inaccuracy in the estimate of the mode. We could estimate the mode location by minimizing the DIP. However, this would be computationally expensive. Instead we estimate the mode using a kernel smoother, as suggested by Silverman (1986, Chapter 6.3 and 6.4). Figure 9.4 shows the

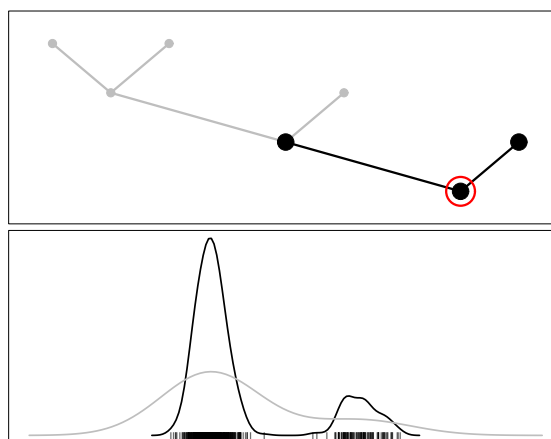


Figure 9.3: Running example: Tree generated by hierarchical model based clustering after second step of pruning, and diagnostic plot for the circled node.

empirical cdf of a sample (black curve), and the closest unimodal cdf (grey curve). The DIP is the maximum absolute difference between the two curves, indicated by the heavy vertical line. The estimated mode location is shown by the grey vertical line.

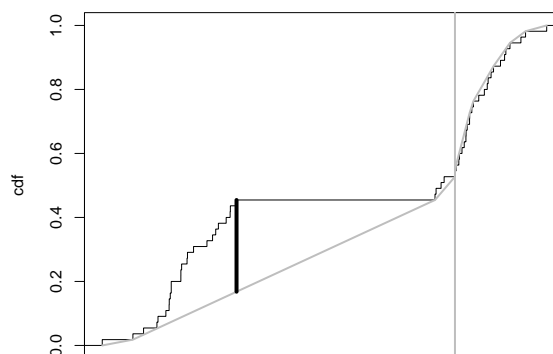


Figure 9.4: Illustration of the DIP statistic.

The distribution of the DIP under the null hypotheses is not available in closed form; it has to be estimated by Monte Carlo. As before, let $H(x)$ be the unimodal cdf closest to $F_n(x)$. We generate M samples of size n from $H(x)$ ($M = 100$, say) and compute the DIPs D_1, \dots, D_M . If the DIP D_{orig} for the original sample is the k -th largest among $\{D_{orig}, D_1, \dots, D_M\}$ then we reject the null hypotheses of unimodality at level $k/(M + 1)$.

9.4 *Remarks*

Hybrid clustering is based on the premise that groups can correspond to collections of mixture components, not just individual components. The purpose of our method is to identify those collections, not to find a better fitting mixture model. This is in contrast to the work by Sand and Moore (2001) on repairing faulty mixture models.

Automatic pruning requires specification of a significance level for the DIP tests; the larger the level, the larger the pruned tree. The significance level should not be taken too literally: the total pruning procedure does not constitute a level α test for unimodality of the multivariate feature distribution. First, there is the problem of multiplicity: If we are carrying out many tests at a given level α , then the probability of erroneously rejecting one or more of the null hypotheses is greater than α .

Second, we are choosing the projection directions to maximize the separation between the clusters. This becomes an issue if the dimensionality of the feature space is large relative to the total number of observations in the two clusters which are under consideration. For example, if we have a total of $p + 1$ observations in a p dimensional feature space then there will always be a direction for which the observations in the two clusters project onto exactly two points, one for each cluster. We deal with this problem by first projecting the combined observations from the two clusters onto their k largest principal components and then finding the Fisher discriminant direction in this lower dimensional subspace. We chose k to be one third of the total number of observations in the two clusters.

Chapter 10

ASSESSMENT AND HYBRID CLUSTERING EXAMPLES

We present examples illustrating both the visualization and assessment methods of Chapter 8 and the hybrid clustering of Chapter 9. We use two real world data sets, the Olive Oil data set (described below) and the TDT dataset. We also consider a uniformly distributed data set for comparison.

10.1 Diagnostics and Hybrid Clustering of the Olive Oil Data

The data for this example consist of measurements of eight chemical concentrations on 572 samples of olive oils from nine different areas of Italy, which are split into 3 regions. Applying hierarchical model-based clustering with diagonal covariance matrices and then using the EM algorithm and BIC to estimate the number of mixture components results in a mixture model with 20 components, corresponding to the 20 leaves of the tree shown in Figure 10.1. The 20 columns of Figure 10.3 are histograms of $P(Y = g|\mathbf{x}_i)$ for $g = 1, \dots, 20$, with the counts encoded as grey levels; the columns thus are a different graphical representation of the rootograms making up the rows of Figure 8.4. The bars in the upper panel of Figure 10.3 encode the observation counts in the clusters. If the clusters were all well separated, then each observation would have posterior probability one for one of the mixture components and zero for all the others, and the plot would have a solid black stripe at the top and be white elsewhere. We are obviously quite far removed from this ideal situation. This impression is confirmed by Figure 10.4. Some of the mixture components are not very isolated; observations generated from mixture component 1, for example, have roughly an 9% probability of being assigned to some other component.

Applying our pruning algorithm with significance level $\alpha = 0.01$ prunes the nodes shown in

grey in Figure 10.1 and results in 7 clusters, four of which are modeled by more than one mixture component. Figure 10.5 shows a typical diagnostic plot for a node whose daughters are pruned ($\alpha = 0.88$), and Figure 10.6 shows a typical plot for a node whose daughters are retained ($\alpha = 0.01$). These two nodes are circled in Figure 10.1. The two-way contingency tables for the pruned and unpruned clusters are shown in Figure 10.2. After pruning, we see that areas 1, 2 and 4 are clustered together into cluster 1, but this makes sense as the first 4 areas are all from the same region part of region 1. The clusters 3 through 7 agree well with areas 5 through 9.

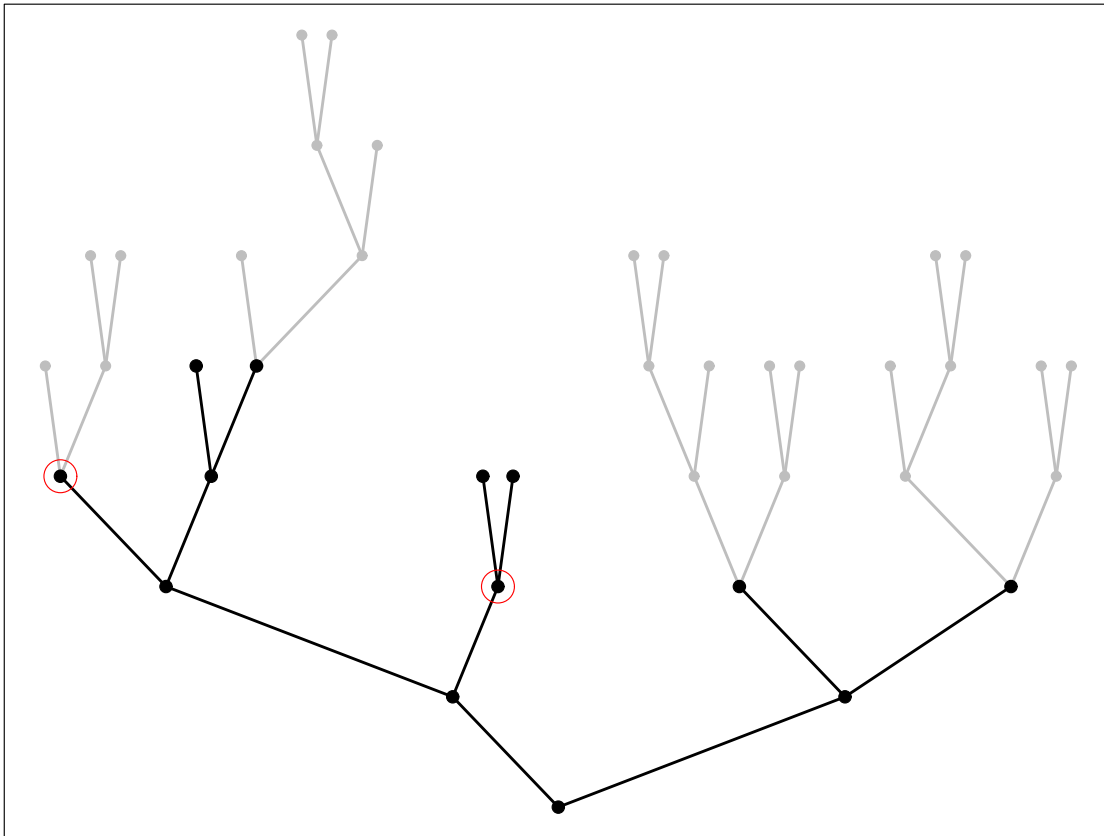


Figure 10.1: Olive oil data: Original tree (all nodes) and pruned tree (dark nodes).

Figure 10.3b is the post-pruning analog to Figure 10.3a. It is much closer to the ideal of “black stripe, white elsewhere”. The misclassification probabilities shown in Figure 10.4

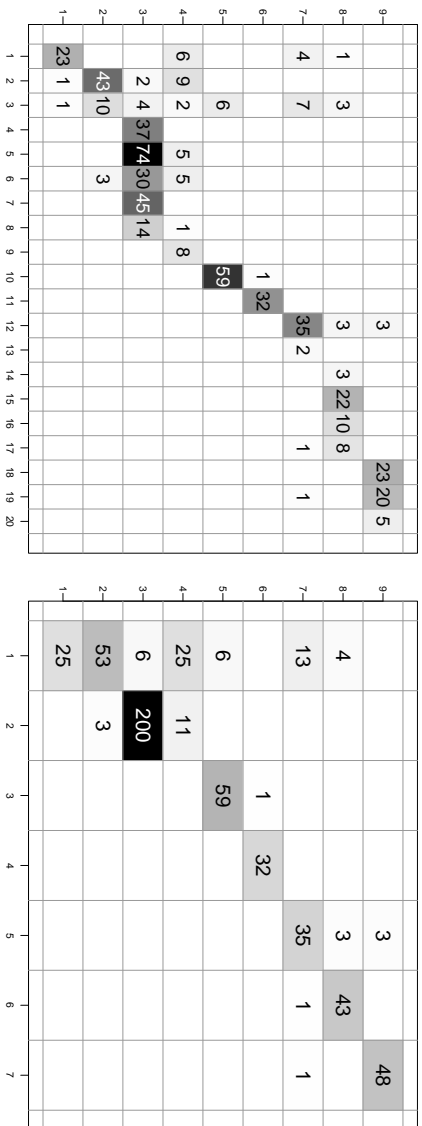


Figure 10.2: Two way contingency table of areas vs pruned clusters for the olive oil data before and after pruning. The 9 areas are shown on the vertical axis and mixture components are shown on the horizontal axis.

also have decreased somewhat; the largest one is now 2% instead of 9%. The overall misclassification rate is 3.0% before pruning and 0.7% after pruning.

Figure 10.7 shows the cdf's of the margins for the two clusterings, pre-pruning in black, post-pruning in grey. If the mixture components were perfectly separated then the cdf of the margin would be a step function with a single step at margin = 1. Pruning brings us closer to this ideal.

In our example we know the group labels of the observations – we know the area of origin for each olive oil and it seems reasonable to assume that any groups in the data reflect the areas of origin. We therefore assess how closely the clusters match the areas. The Fowlkes-Mallows index before pruning is 0.52, compared to an index of 0.81 after pruning. This shows that pruning substantially improved the agreement between groups and clusters.

10.2 Diagnostics and Hybrid Clustering of Simulated “Olive Oil Data”

In the previous example, pruning was successful in that it significantly improved the agreement between clusters and areas. The purpose of the second example is to illustrate how hybrid clustering performs on data which were in fact generated from a Gaussian mixture

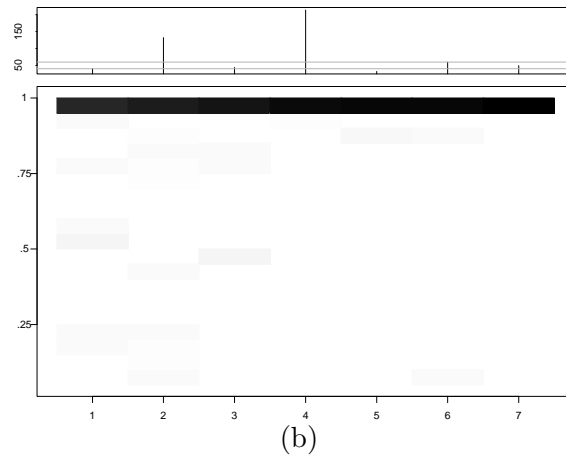
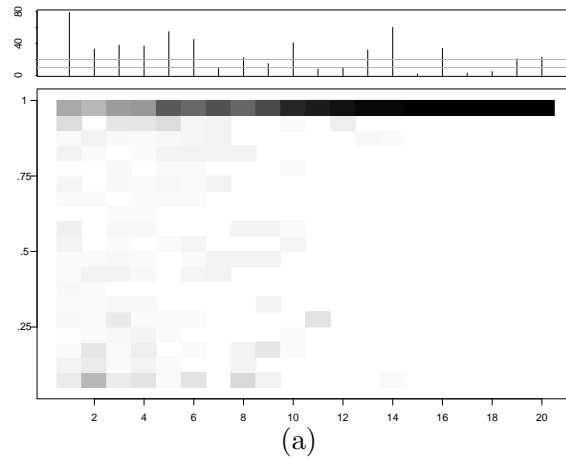


Figure 10.3: Olive oil data: Histograms of posterior probabilities $P(Y = g | \mathbf{x}_i)$ for the data, before (a) and after (b) pruning.

model. We choose a mixture model that mimics the olive oil data: we estimate mean and covariance for each area and then generate a sample of the same size as the olive oil data from the corresponding mixture model.

Applying hierarchical model-based clustering with diagonal covariance matrices and using the BIC to estimate the number of mixture components results in a mixture model with 9 components, corresponding to the 9 leaves of the tree shown in Figure 10.8. Pruning leads to a partition into 7 clusters corresponding to the leaves of the subtree drawn in black, and increases the Fowlkes-Mallows index from 0.71 to 0.86. Figures 10.9 and 10.10 show the

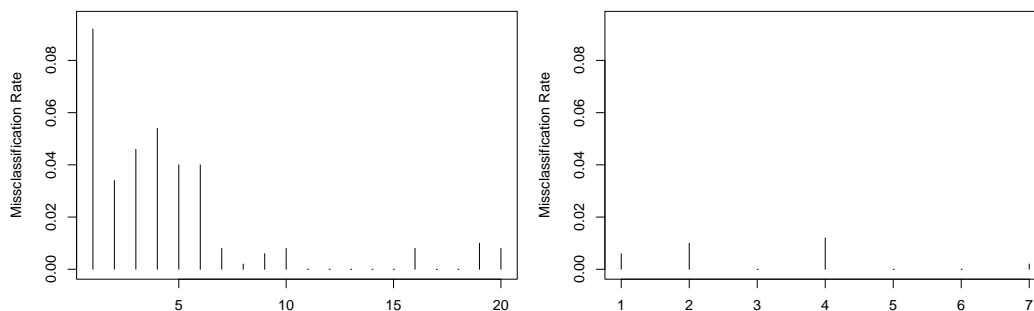


Figure 10.4: Olive oil data: Misclassification probabilities MC_g for the 28 components of the mixture model, before (left) and after (right) pruning.

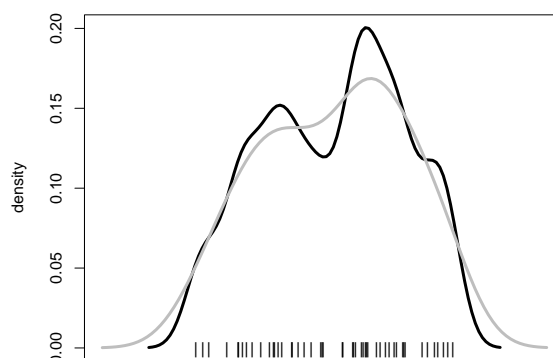


Figure 10.5: Pruned node of olive oil tree

contingency tables of areas versus clusters before and after pruning, respectively. Pruning removes the split of area 3 and merges the two impure clusters 1 and 2.

10.3 Diagnostics and Hybrid Clustering for the “1100 TDT” Collection

Clustering the 1100 TDT data set using model based clustering (Section 4.6), results in 42 mixture components with a Fowlkes-Mallows index of 0.49 and a corresponding Rand index of 0.38. The hierarchical clustering tree whose leaves correspond to the fitted mixture model is shown in Figure 10.11.

A plot of the posterior probabilities (Figure 10.12) shows that the mixture components are

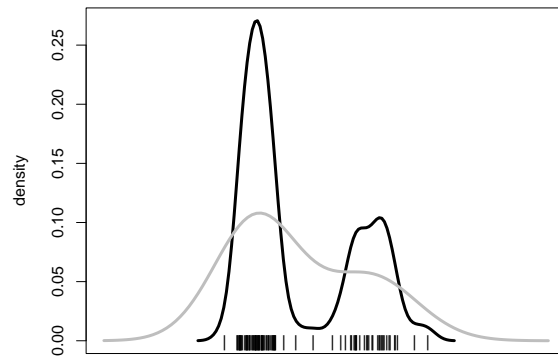


Figure 10.6: Non pruned node of olive oil tree

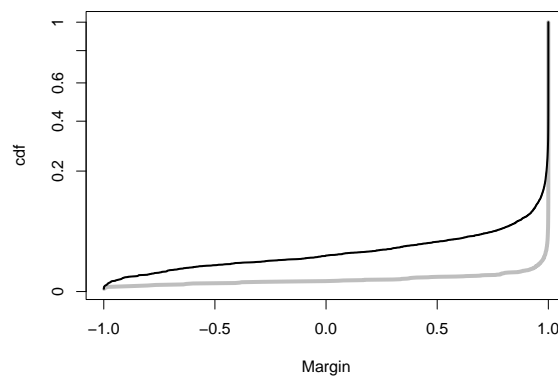


Figure 10.7: Olive oil data: cumulative distribution function of the margins before pruning (black line) and after pruning (grey line).

well separated; there is a black stripe at the top indicating that most of the documents having either probability 0 or 1 of belonging to each cluster. There appears to be almost no overlap of the clusters as the misclassification probability for the entire model is less than 0.00005. The distribution of the margins of the model is shown in Figure 10.13.

Applying the pruning methods of Chapter 8 at significance level 0.01 removes 6 clusters and results in the tree whose edges are drawn in black in Figure 10.11. The distribution of the margins and the posterior probabilities are essentially unchanged. The misclassification rate reduces to 0. The Fowlkes-Mallows index after pruning is 0.54 and the Rand index after pruning is 0.45, a big improvement on the unpruned tree. Pruning doesn't merge all

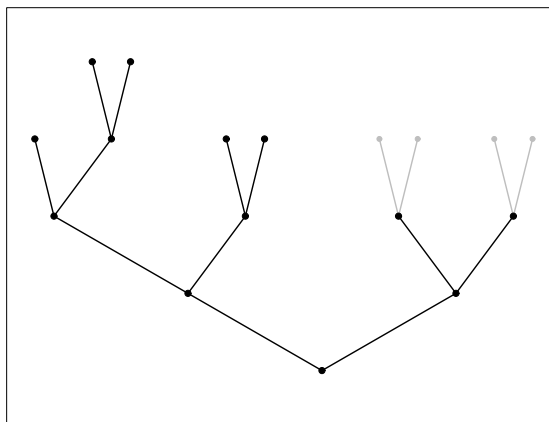


Figure 10.8: Simulated olive oil data: Original tree (all nodes) and pruned tree (black nodes).

the mixture components corresponding to the same topics, and the question is why is this so.

The contingency table in Figure 4.8 shows that many topics are modeled not by one, but by several mixture components. Leaves (\sim mixture components) corresponding to those topics are drawn in color in Figure 10.11. The figure indicates that leaves for the same topic are typically on the same branch of the tree. This means that the pruning procedure in principle could improve the clustering. The fact that no more pruning occurs suggests that these topics might not be homogeneous, i.e. they might be split into subtopics. Reading the documents assigned to the four mixture components corresponding to the “DNA evidence in the OJ Simpson Trial” topic suggests that a split into two subtopics might not be unreasonable, one topic seems to be about the DNA specifically and the other seems to contain other stories which are more about the trial than the DNA.

10.4 Diagnostics and Hybrid Clustering of the Full TDT Data Set

Clustering the full TDT data set using Refractionation (Section 7.5), results in 209 mixture components with a FM1 index of 0.515 and a corresponding Rand index of 0.405. The

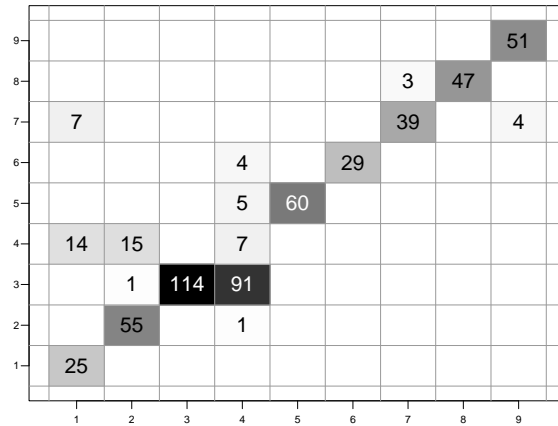


Figure 10.9: Simulated olive oil data: Two way contingency table of areas versus clusters before pruning.

hierarchical clustering tree whose leaves correspond to the fitted mixture model is shown in Figure 10.15.

A plot of the posterior probabilities (Figure 10.16) shows that the mixture components are well separated; there is a black stripe at the top indicating that most of the documents having either probability 0 or 1 of belonging to each cluster. There appears to be almost no overlap of the clusters as the misclassification probability for the entire model is less than 0.005. The distribution of the margins of the model is shown in Figure 10.17.

10.4.1 Pruning

Applying the pruning methods of Chapter 8 at significance level 0.01 removes 56 clusters and results in the tree drawn in black in Figure 10.15. The misclassification rate, the distribution of the margins and posterior probabilities are essentially unchanged. The Fowlkes-Mallows index after pruning is 0.533 and the Rand index after pruning is 0.431, which are a slight improvement from the values before pruning.

Table 10.1: Document Headlines for Cluster 33

Cluster 33 – 69 documents

USAir extends off-peak holiday fare sale

Russian plane hijacker frees most of his hostages

Plane makes emergency landing on Long Island

Turkish plane skips over road to railway line

Kiwi Airlines hopes for full schedule by Wednesday

Jamaica's air traffic controllers strike

Plane crash in Siberia

Air Canada jet evacuated after fire, no serious injuries

Boeing confirms Gulf Air cancellation for jets

U.S. fighter jet crashes in Texas, no fatalities

Alitalia plane makes hard landing, blows tires

Royal Air Maroc flight ATR 630 crash

Midway Airlines planes will feature women-only restrooms

Hijacker holds 16 people hostage on Russian plane

Table 10.2: Document Headlines for Cluster 72

Cluster 72 – 45 documents

Japan's Hitachi claims tiniest video camera
 Japan, U.S. reach agreement on financial services
 US intellectual property rights in China
 US intellectual property rights in China
 Barings, London bankers went bust
 Japan's trade surplus narrowed slightly in March
 Japan's prime minister denies resignation rumors
 Japan PM threatens to break up coalition-report
 Japan business outlook lifts, but caution remains
 Japan airliner hijacked, transport ministry says
 Clinton to press Japan on cars in Halifax
 Japan opposition selects new name
 Japan's trade surplus narrowed slightly in March
 U.S., Japan say still apart after car talks
 US intellectual property rights in China

We then clustered the data using model based Refractionation with a diagonal covariance model. Refractionation stopped after 8 passes and the BIC chose 9 clusters. The resulting clustering tree with 9 leaves is shown in Figure 10.18.

The histograms of posterior probabilities (Figure 10.19) show that the mixture components do not separate the data very well as there are almost no observations with a probability close to 1 of belonging to any cluster. The misclassification probability for the entire model is 0.38 indicating a large overlap between mixture components. This is also borne out by Figure 10.20 showing the individual misclassification probabilities for all the 9 clusters. The margins have almost a uniform distribution (Figure 10.21) reinforcing the conclusion.

Pruning using significance level 0.01 reduces the tree to two clusters. Figure 10.22 shows the GKL-Silverman plot and also the GKL-DIP plot for root node of the tree. Even though the p-value is less than 0.01, the distribution does not appear to be bimodal. So in conclusion, it seems that the Hybrid clustering procedure does not tend to create seemly well separated clusters in situations where there are no groups in the data. This increases our confidence in the clusters found in the TDT data.

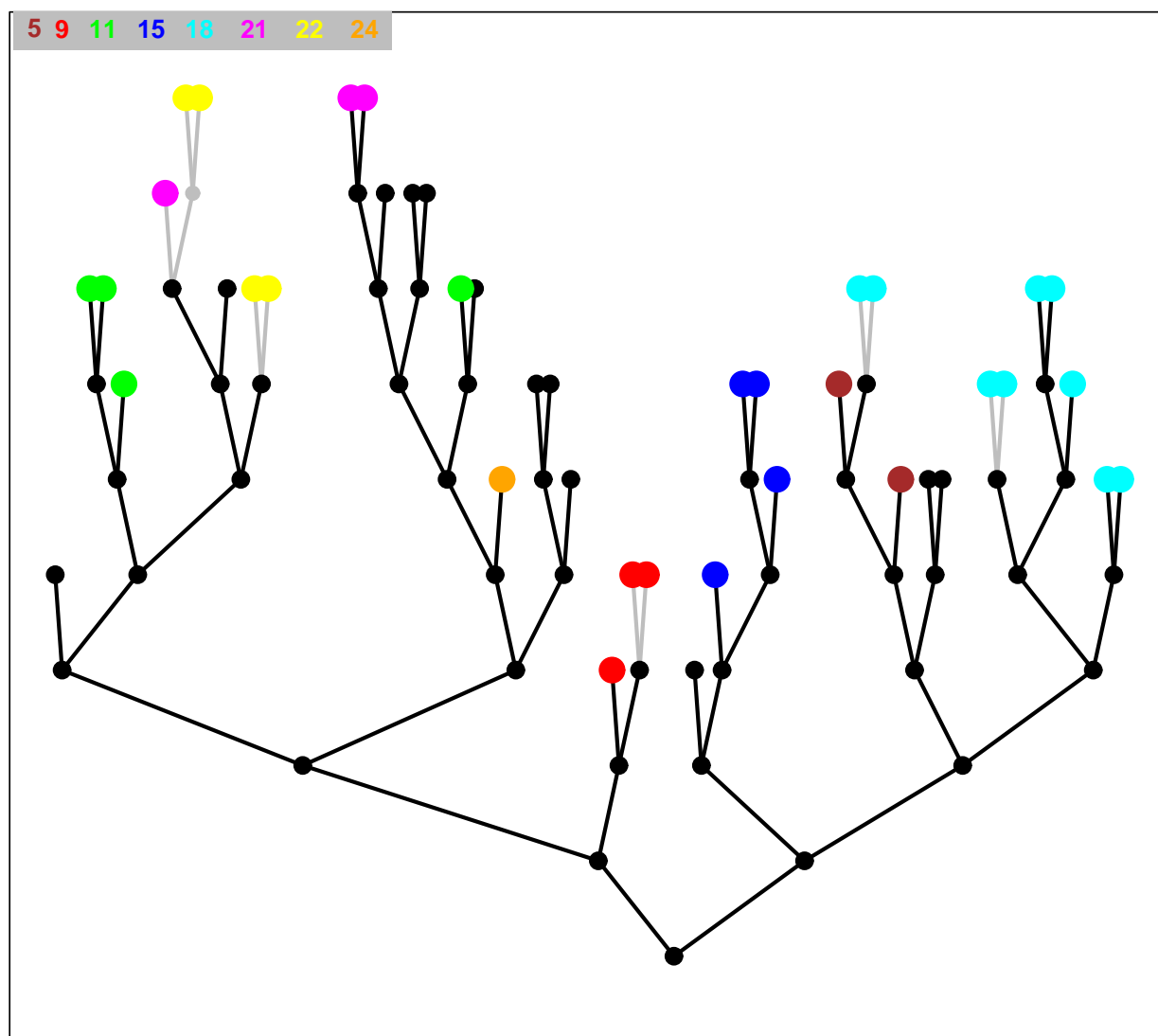


Figure 10.11: Clustering tree for the “1100 TDT” collection. The colored leaves correspond to topics which are split across several mixture components.

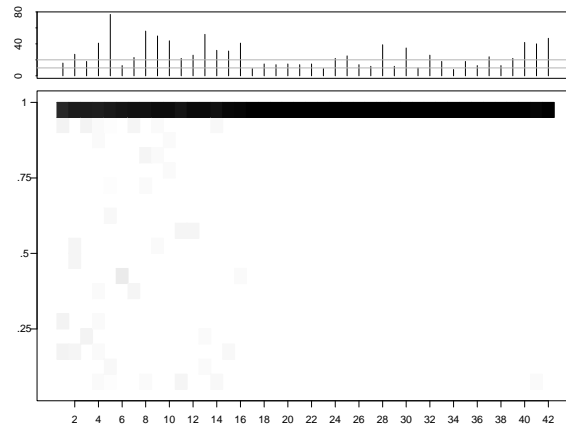


Figure 10.12: Histograms of posterior probabilities $P(Y = g|\mathbf{x}_i)$ for the “1100 TDT” collection, before pruning

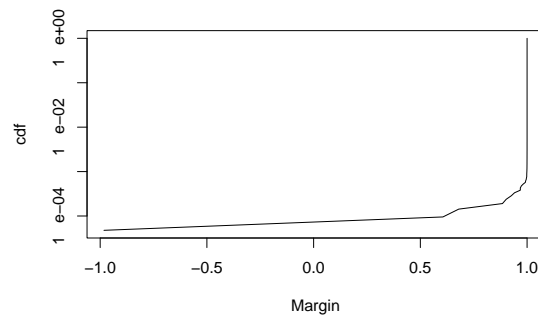


Figure 10.13: Cumulative distribution function of the margin of the model for the “1100 TDT” collection on the log scale.

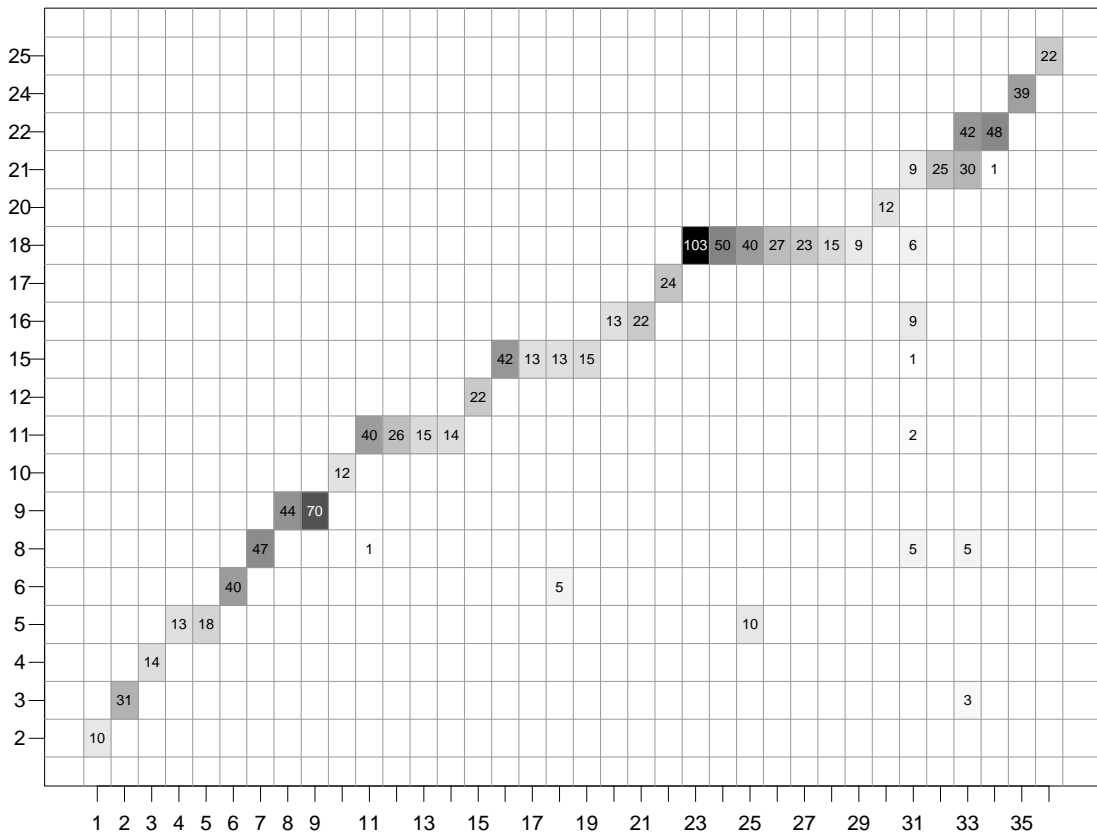


Figure 10.14: Two way contingency table of topics vs pruned clusters for the “1100 TDT” collection. Topic labels are shown on the vertical axis and mixture components are shown on the horizontal axis.

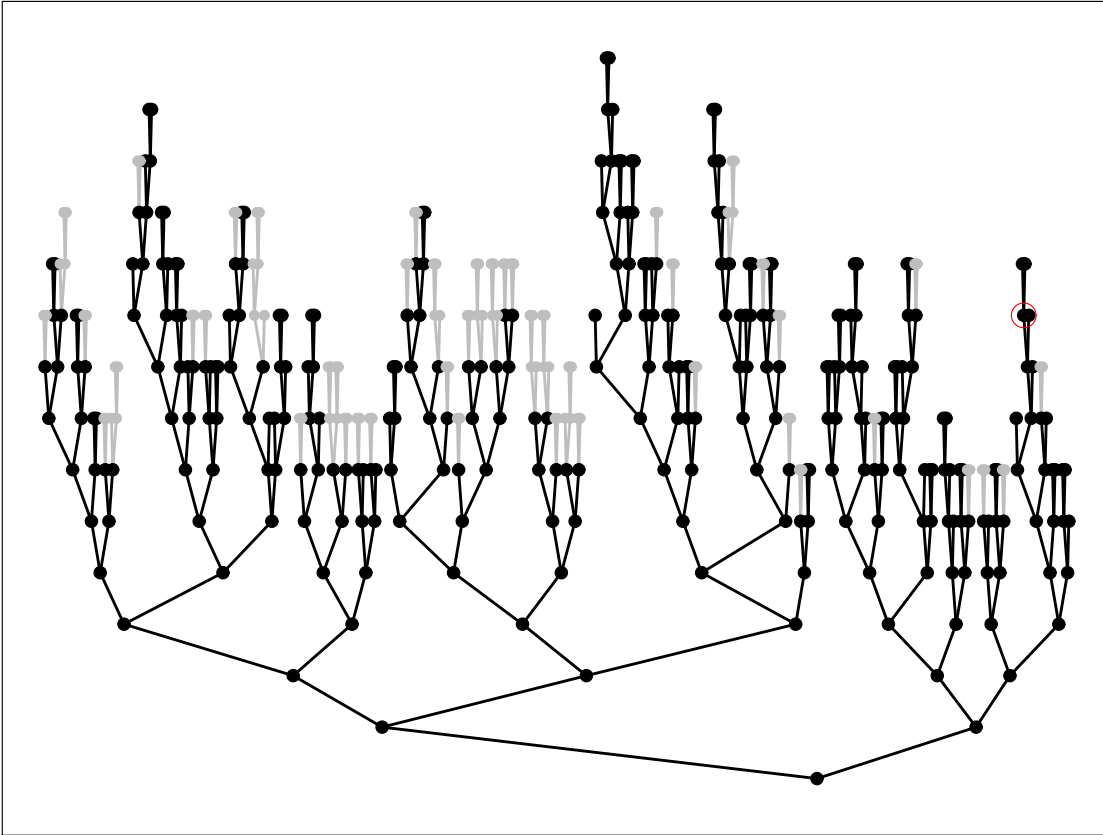


Figure 10.15: Clustering tree for the full TDT data with pruned leaves shown in grey.

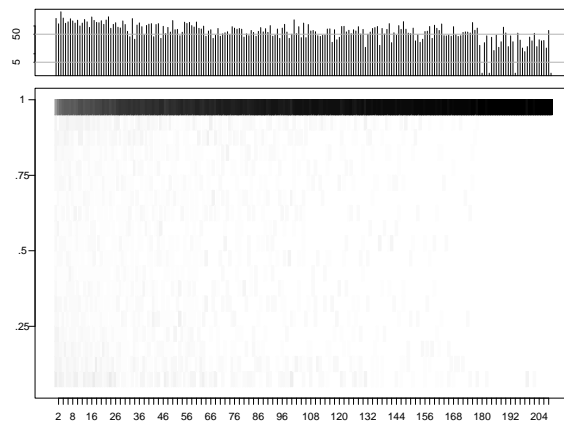


Figure 10.16: Histograms of posterior probabilities $P(Y = g | \mathbf{x}_i)$ for the full TDT data, before pruning.

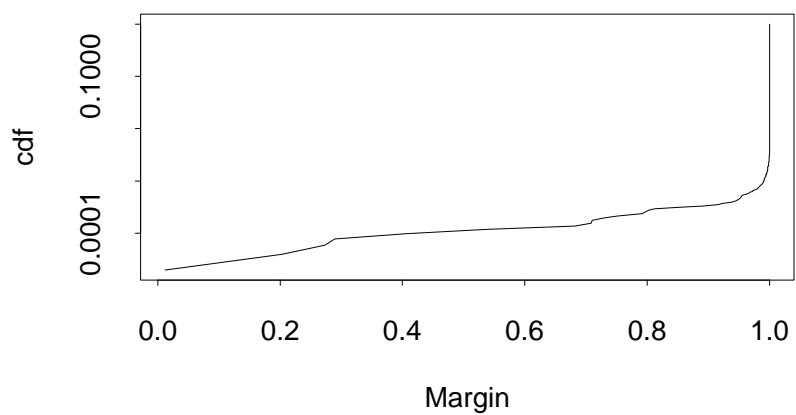


Figure 10.17: Cumulative distribution function of the margin of the model for the full TDT data on the log scale.

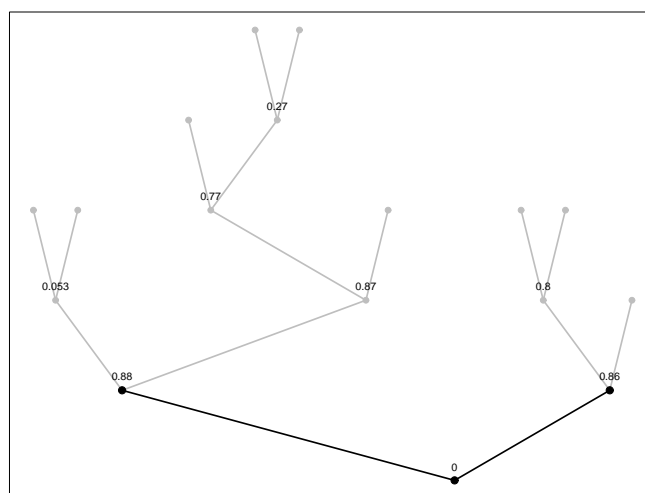


Figure 10.18: Clustering Tree for the Uniform data with pruned leaves shown in grey.

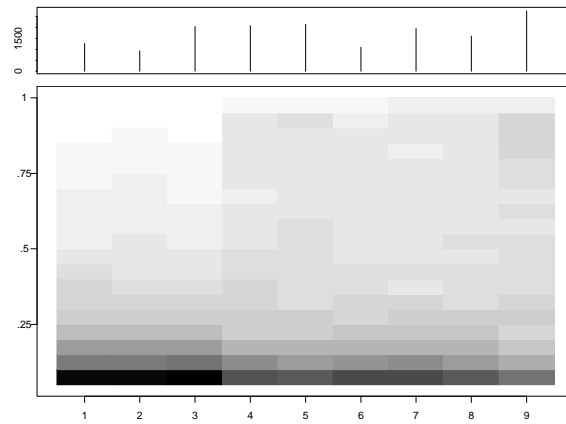


Figure 10.19: Histograms of posterior probabilities $P(Y = g|\mathbf{x}_i)$ for the Uniform data, before pruning

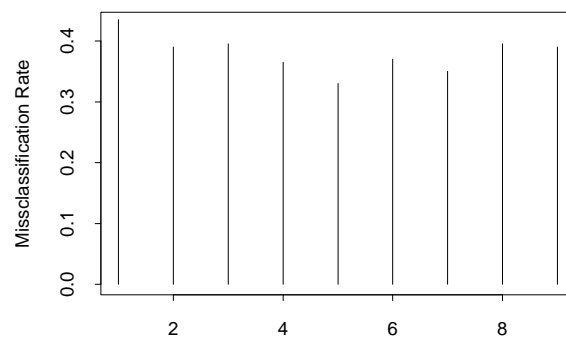


Figure 10.20: Misclassification probabilities for each of the mixture components of the model fitted to the Uniform data.

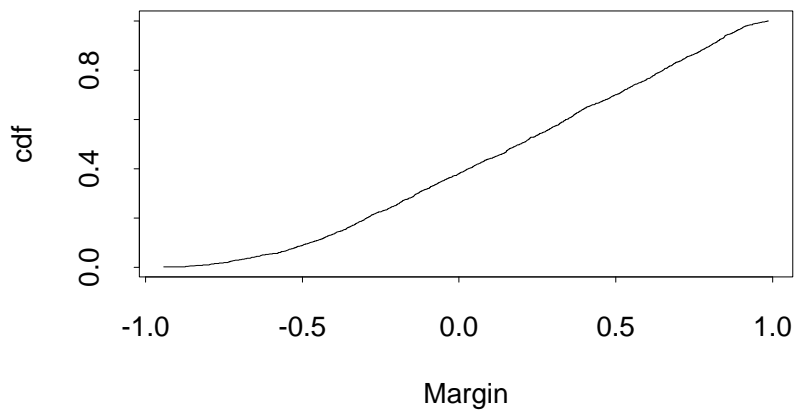


Figure 10.21: Cumulative distribution function of the margin of the models for the Uniform data.

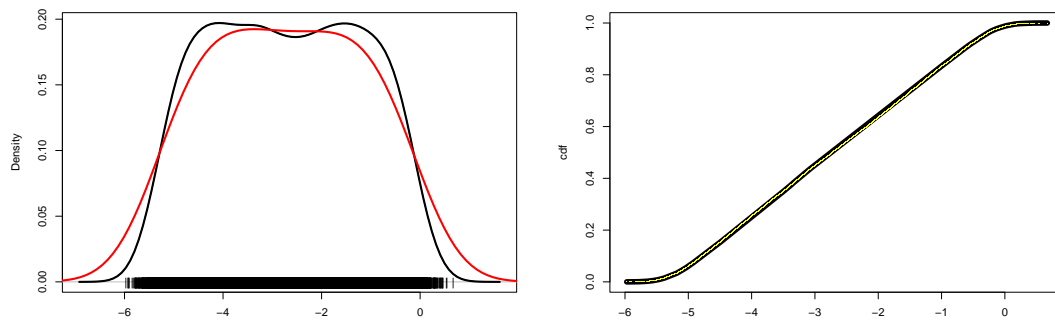


Figure 10.22: GKL-Silverman and GKL-DIP plots for the full TDT data split on the root node.

Chapter 11

DISCUSSION AND FUTURE WORK**11.1 Discussion**

We have proposed model based Fractionation and Refractionation, methods for extending the range of model based hierarchical clustering to data sets with tens of thousands of observations and hundreds of groups. Compared with competing approaches to model based clustering of large datasets (Bradley, Fayyad, and Reina 1998; 1999; Posse 2001; Domingos and Hulten 2002; Wehrens, Buydens, Fraley, and Raftery 2003), model based Refractionation does not require that the number of groups in the data be known a priori; it can be estimated from the data. We have presented experimental evidence that the heuristics underlying our method indeed appear to be valid.

A basic premise of model based clustering is that each distinct group in the data corresponds to a single component of the mixture model. If this premise holds, then the ability to estimate the number of mixture components (equal to the number of groups) is a major strength of model based clustering compared to non-parametric clustering methods. On the other hand, if the premise does not hold, the result of model based clustering can be misleading, because several mixture components may model the same group. Consequently the number of mixture components will over-estimate the number of groups, and the clusters corresponding to individual mixture components will no longer be well separated. It is therefore important to be able to decide whether or not the premise holds and, in case the premise does not hold, to determine which mixture components correspond to the same group.

We have introduced methods for assessing the degree of separation between the components

of a mixture model, and between the corresponding clusters. We have also presented an algorithm for pruning the cluster tree generated by hierarchical model based clustering. The algorithm starts with the tree corresponding to the mixture model chosen by the BIC. It then progressively merges clusters that do not appear to correspond to different modes of the data density.

We have applied model based clustering to a simple synthetic example in which the premise was violated. In this case the method indeed exhibited the deficiencies that we had anticipated. We have also shown that our proposed diagnostic tools reveal the true structure of the data and lead to more accurate clustering. Application of our techniques to real-world examples have also been encouraging. In one of the data sets which we have investigated (the Olive Oil data), our diagnostics have shown that the premise of model based clustering most probably was violated, and our pruning algorithm has significantly improved the quality of the clustering.

11.2 Future Work

We discuss two areas of future work, clustering truly massive datasets with large numbers of clusters, and investigating the use of mixture of factor analyzers models for hierarchical model based clustering.

11.2.1 Massive Data Sets with Many Clusters

Both Fractionation and Refractionation make the implicit assumption that the number of clusters is less than the maximum problem size practical for an $O(n^2 \log n)$ algorithm. If this assumption is violated then modifications are needed.

A simple suggestion is to omit the final iteration in the Fractionation process and let BIC choose the number of clusters in each fraction. The problem is that groups which are split across fractions will be scattered over several mixture components. Refractionation is reducing but not eliminating the possibility that such splits will actually occur. It is not

obvious how to overcome this difficulty.

11.2.2 *Mixtures of Factor Analyzers*

A factor analysis model represents a covariance matrix as the sum of a diagonal matrix and k rank one matrices.

$$\Sigma = \Phi + \Lambda_k \Lambda_k' \quad (11.1)$$

A Gaussian mixture model with estimated covariances matrices of this particular form is called a mixture of factor analyzers model (Hinton, Dayan, and Revow 1997). Factor analysis models for the covariances bridge the gap between low complexity diagonal covariance models and unconstrained covariance models. However, hierarchical model based clustering with factor analysis covariance models is computationally demanding. This is because for each potential merge we have to perform a factor analysis for the new cluster that would be generated by the merge. Hierarchically clustering 1000 observations in 50 dimensions with factor analysis covariance models takes approximately two hours on a 1.4GHz machine.

A simple idea for speed up is to not compute all pairwise distances between clusters using the mixture of factor analyzers model. Instead, restrict consideration to those pairs of clusters which are close by some measure that is cheaper to calculate, such as distance based on a diagonal covariance or perhaps even a variable spherical covariance model.

BIBLIOGRAPHY

- [Akaike, 1973] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. N. and Csaki, F., editors, *Second International Symposium on Information Theory*, pages 267–281. Budapest: Akademiai Kiado.
- [Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19:716–723.
- [Allan et al., 1998] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study final report.
- [Ankerst et al., 1999] Ankerst, M., Breuning, M., Kriegel, H., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. In *Proceedings, ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 49–60.
- [Banfield and Raftery, 1993] Banfield, J. D. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.
- [Bentley and Friedman, 1978] Bentley, J. and Friedman, J. (1978). Fast algorithms for constructing minimal spanning trees in coordinate spaces. *IEEE Transactions on Computers*, C-27(2):97–105.
- [Berry et al., 1999] Berry, M., Drmac, Z., and Jessup, E. (1999). Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362.
- [Berry et al., 1995] Berry, M., Dumais, S., and O'Brien, G. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595.
- [Bickel and Fan, 1996] Bickel, P. and Fan, J. (1996). Some problems of the estimation of unimodal densities. *Statistica Sinica*, 6:23–45.
- [Bradley et al., 1998] Bradley, P., Fayyad, U., and Reina, C. (1998). Scaling clustering algorithms to large datasets. In *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD98)*.
- [Bradley et al., 1999] Bradley, P., Fayyad, U., and Reina, C. (1999). Scaling EM (expectation-maximization) clustering to large databases. Technical Report MSR-TR-98-35, Microsoft Research.
- [Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont: CA.
- [Carmichael et al., 1968] Carmichael, J., George, G., and Julius, R. (1968). Finding natural clusters. *Systematic Zoology*, 17:144–150.
- [Cutting et al., 1992] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *15th Ann Int'l SIGR*, pages 318–329.

- [Dasgupta and Raftery, 1998] Dasgupta, A. and Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93:294–302.
- [Domingos and Hulten, 2002] Domingos, P. and Hulten, G. (2002). Learning from infinite data in finite time. In *Advances in Neural Information Processing Systems 14*.
- [Dumais, 1991] Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments & Computers*, 23(2):229–236.
- [Ester et al., 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231.
- [Fowlkes and Mallows, 1983] Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *J. American Statistical Association*, 78:553–569.
- [Fraley, 1998] Fraley, C. (1998). Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.*, 20:270–281.
- [Fraley and Raftery, 1998] Fraley, C. and Raftery, A. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.
- [Fraley and Raftery, 2002] Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631.
- [Gnanadesikan et al., 1982] Gnanadesikan, R., Kettenring, J., and Landwehr, J. (1982). Projection plots for displaying clusters. In *Statistics and Probability: Essays in Honor of C. R. Rao*, pages 269–280. Elsevier/N.Holland.
- [Griffiths and Steyvers, 2002] Griffiths, T. and Steyvers, M. (2002). A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*.
- [Hartigan, 1981] Hartigan, J. (1981). Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association*, 76:388–394.
- [Hartigan, 1985] Hartigan, J. (1985). Statistical theory in clustering. *Journal of Classification*, 2:63–76.
- [Hartigan and Hartigan, 1985] Hartigan, J. and Hartigan, P. (1985). The dip test of unimodality. *Annals of Statistics*, 13:70–84.
- [Hinton et al., 1997] Hinton, G. E., Dayan, P., and Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74.
- [Hofman, 1999] Hofman, T. (1999). Probabilistic latent semantic indexing. *SIGIR 1999*, pages 50–57.
- [Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classification*, 2:193–218.
- [Jeffreys, 1935] Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. In *Proceedings of the Cambridge Philosophy Society*, volume 31, pages 203–222.

- [Kass and Raftery, 1995] Kass, R. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90:773–795.
- [Koehler and Murphee, 1988] Koehler, A. B. and Murphee, E. H. (1988). A comparison of the akaike and schwarz criteria for selecting model order. *Applied Statistics*, 37:187–195.
- [Mardia et al., 1979] Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate Analysis*. Academic Press.
- [McLachlan and Basford, 1988] McLachlan, G. and Basford, K. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker.
- [McLachlan and Peel, 2000] McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. John Wiley & Sons.
- [Nene and Nayar, 1997] Nene, S. and Nayar, S. (1997). A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:989–1003.
- [Posse, 2001] Posse, C. (2001). Hierarchical model-based clustering for large datasets. *Journal of Computational and Graphical Statistics*, 10(3):464–486.
- [Prim, 1957] Prim, R. (1957). Shortest connection matrix network and some generalizations. *Bell Systems Technical Journal*, 36:1389–1401.
- [R. Wehrens and Raftery, 2003] R. Wehrens, L. Buydens, C. F. and Raftery, A. (2003). Model-based clustering for image segmentation and large datasets via sampling. Technical Report 424, Department of Statistics, University of Washington, Seattle, WA.
- [Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850.
- [Rozal and Hartigan, 1994] Rozal, G. and Hartigan, J. (1994). The map test for multimodality. *Journal of Classification*, 11:5–36.
- [Sand and Moore, 2001] Sand, P. and Moore, A. (2001). Repairing faulty mixture models using density estimation. In *Machine Learning: Proceedings of the eighteenth International Conference*, pages 457–464.
- [Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:497–511.
- [Scott, 1992] Scott, D. (1992). *Multivariate Density Estimation*. Wiley.
- [Silverman, 1986] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.
- [Stuetzle, 2003] Stuetzle, W. (2003). Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *J. Classification*. to appear.
- [Swayne et al., 1998] Swayne, F., Cook, D., and Buja, A. (1998). XGobi: Interactive dynamic data visualization in the X window system. *J. Computational and Graphical Statistics*, 7:113–130.
- [Van Rijsbergen, 1979] Van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London.
- [Vivisimo.com, 2003] Vivisimo.com (2003). <http://www.vivisimo.com/>.
- [Ward, 1963] Ward, J. H. (1963). Hierarchical groupings to optimize an objective function. *J. American Statistical Association*, 58:234–244.

- [Wishart, 1969] Wishart, D. (1969). Mode analysis: A generalization of nearest neighbor which reduces chaining effects. In Cole, A., editor, *Numerical Taxonomy*, pages 282–311. Academic Press.
- [Wong and Lane, 1983] Wong, M. and Lane, T. (1983). A kth nearest neighbor clustering procedure. *Journal of the Royal Statistical Society, Series B*, 45:362–368.

VITA

Jeremy Tantrum was born in Warkworth, New Zealand in 1976. He grew up on a dairy farm, attending Kaiwaka Primary School and Otamatea High School. During his secondary school years, he had an excellent math teacher (Mr McMahon) whose enthusiasm for mathematics rubbed off onto Jeremy. In 1997 Jeremy received a Bachelor of Science with first class honors from the University of Auckland, with a major in Statistics. The Statistics department was excellent in both teaching statistics and encouraging students to pursue it further. In the fall of 1997, Jeremy commenced graduate studies at the University of Washington. He received a M.S. in Statistics in 1999 and a Ph.D. in 2003.