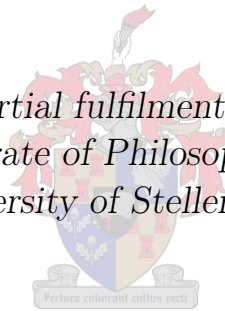


# Large-Scale Clustering of Acoustic Segments for Sub-word Acoustic Modelling

by

Lerato Lerato

*Thesis presented in partial fulfilment of the requirements for  
the degree of Doctorate of Philosophy in Engineering at  
University of Stellenbosch*



Supervisor: Professor Thomas Niesler

April 2019

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2019

Copyright © 2019 University of Stellenbosch  
All rights reserved.

# Abstract

## Large-Scale Clustering of Acoustic Segments for Sub-word Acoustic Modelling

L. Lerato

Thesis: PhD

April 2019

A pronunciation dictionary is one of the key building blocks in automatic speech recognition (ASR) systems. However, pronunciation dictionaries used in state-of-the-art ASR systems are hand-crafted by linguists. This process requires expertise, time and funding and as a consequence is not realised for many under-resourced languages. To address this, we develop a new unsupervised agglomerative hierarchical clustering (AHC) algorithm that can be used to discover sub-word units that can in turn be used for the automatic induction of a pronunciation dictionary.

The new algorithm, named multi-stage agglomerative hierarchical clustering (MAHC), addresses the  $O(N^2)$  memory and computation complexity observed when classical AHC is applied to large datasets. MAHC splits the data into independent subsets and applies AHC to each. The resultant clusters are merged, re-divided into subsets, and passed to a following iteration. Results show that MAHC can match and even surpass the performance of classical AHC. Furthermore, MAHC can automatically determine the optimal number of clusters which is a feature not offered by most other approaches. A further refinement of MAHC, termed MAHC with memory size management (MAHC+M), addresses the case where some subsets may exhibit excessive growth during iterative clustering. MAHC+M is able to adhere to maximum memory constraints, which improves efficiency and is practically useful when using parallel computing resources.

The input to MAHC is a matrix of pairwise distances computed with dynamic time warping (DTW). A modified form of DTW, named feature trajectory DTW (FTDTW), is introduced and shown to generally lead to better performance for both MAHC and MAHC+M.

It is shown that clusters obtained using the MAHC algorithm can be used as sub-word units (SWUs) for acoustic modelling. Pronunciations in terms

*ABSTRACT*

iii

of these SWUs were obtained by alignment with the orthography. Speech recognition experiments show that dictionaries induced using clusters obtained by FTDTW-based MAHC+M consistently outperform those obtained using DTW-based MAHC.

# Acknowledgements

I would like to express my deepest gratitude and appreciation to the following people:

- My supervisor, Professor Thomas Niesler, for your persistent encouragement and guidance.
- My DSP colleagues, especially Dr. Lehlohonolo Mohasi and Dr. Ewald van der Westhuizen for consistently making sure that we kept going.
- My family, bo-Ntate: Tšotleho, Thabo le Thabang, Khethisi le Kabelo. Rakhali 'Manthati, Sis Mongi, bo-Motsoala le bo-Malome kaofela.
- Hanu, Keletso le Nkeletseng le uena Amo, for all the support.
- Thabo Ntitsane and all my friends for your motivation.
- Colleagues from the Department of Maths and Computer Science at NUL, thank you.

# Dedications

*To the memory of my mother, 'Me' 'Matiisetso, and my father, Ntate Tšiu.*

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Dedications</b>	<b>v</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>List of Symbols</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Current state of research . . . . .	2
1.2 Research objectives . . . . .	3
1.3 Project scope and contributions . . . . .	3
1.4 Dissertation overview . . . . .	5
<b>2 Clustering of Acoustic Speech Segments</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.2 A précis of clustering methods . . . . .	6
2.3 Acoustic segments as data objects . . . . .	7
2.4 Clustering methods for acoustic segments . . . . .	9
2.5 Determining the number of clusters . . . . .	13
2.6 Cluster validation methods . . . . .	14
2.7 Summary . . . . .	23
<b>3 Agglomerative Hierarchical Clustering</b>	<b>24</b>
3.1 Introduction . . . . .	24
3.2 Data representation . . . . .	24

3.3	The algorithm . . . . .	25
3.4	Linkage methods . . . . .	26
3.5	Comparative studies on linkage methods . . . . .	32
3.6	Monotonicity in dendrograms . . . . .	33
3.7	AHC variants for large data . . . . .	33
3.8	Summary . . . . .	35
<b>4</b>	<b>Speech Signal Similarity Computation using Dynamic Time Warping</b>	<b>36</b>
4.1	Introduction . . . . .	36
4.2	Dynamic time warping algorithms . . . . .	36
4.3	Experimental evaluation . . . . .	40
4.4	Discussion . . . . .	46
4.5	Summary and conclusion . . . . .	48
<b>5</b>	<b>Multi-Stage Agglomerative Hierarchical Clustering</b>	<b>49</b>
5.1	Introduction . . . . .	49
5.2	The MAHC algorithm . . . . .	49
5.3	Clustering acoustic segments using MAHC . . . . .	52
5.4	Experimental evaluation . . . . .	54
5.5	Summary and conclusion . . . . .	64
<b>6</b>	<b>Cluster Size Management in MAHC of Acoustic Speech Seg- ments</b>	<b>66</b>
6.1	Introduction . . . . .	66
6.2	Limitations of MAHC . . . . .	66
6.3	MAHC with cluster size management . . . . .	68
6.4	Data and evaluation measures . . . . .	69
6.5	Experimental evaluation . . . . .	69
6.6	Summary and conclusion . . . . .	83
<b>7</b>	<b>Pronunciation Dictionary Generation</b>	<b>85</b>
7.1	Introduction . . . . .	85
7.2	Creating a pronunciation dictionary . . . . .	86
7.3	ASR evaluation of MAHC-based pronunciations . . . . .	90
7.4	Summary and conclusion . . . . .	93
<b>8</b>	<b>Summary, Conclusions and Recommendations</b>	<b>95</b>
8.1	Summary and conclusions . . . . .	95
8.2	Recommendations for future work . . . . .	98
	<b>List of References</b>	<b>99</b>



# List of Figures

2.1	Best-fit lines to locate the knee of the graph in the L method. . . .	22
3.1	An example of a dendrogram. . . . .	26
4.1	Alignment of spectral features for the triphone $b-aa+dx$ extracted from the TIMIT corpus [1] for (a) the male speaker mrk0 and (b) the female speaker fdml0. . . . .	39
4.2	Alignment of trajectories of 21 spectral features for instances of triphone $b-aa+dx$ drawn from TIMIT corpus [1] from both (a) a male speaker mrk0 and (b) a female speaker fdml0. . . . .	40
4.3	AHC performance for 8772 TIMIT triphones parameterised as MFCC's in terms of the F-Measure for both Manhattan and Euclidean based DTW when using (a) Complete linkage and (b) Ward linkage. . . .	42
4.4	AHC performance for 8772 TIMIT triphones parameterised as MFCC's in terms of the F-Measure for four linkage methods using Manhattan based DTW. . . . .	43
4.5	Clustering performance for Dataset 1 when using MFCC features in terms of (a) F-Measure and (b) NMI. . . . .	45
4.6	Clustering performance for Dataset 1 when using PLP features in terms of (a) F-Measure and (b) NMI. . . . .	46
4.7	Clustering performance for Dataset 2 in terms of (a) F-Measure and (b) NMI. . . . .	47
4.8	Clustering performance for the 10 independent subsets of Dataset 3 in terms of F-Measure. . . . .	47
5.1	The first stage of MAHC algorithm. . . . .	50
5.2	The second stage of MAHC algorithm. . . . .	51
5.3	The complete MAHC algorithm. . . . .	52
5.4	AHC results of a small experiment with 29 true clusters. The peak in the F-Measure occurs at 24 clusters, while the knee of the L method is found at 22 clusters. . . . .	53
5.5	Distribution of the number of segments per class for the two independent Set A and Set B. . . . .	55

5.6	Performance of MAHC and PSC for the small sets in terms of F-Measure, using F-Measure to determine thresholds in stage 1. (a) F-Measure for Small Set A (b) MAHC optimal number of clusters for Small Set A (c) F-Measure for Small Set B (d) MAHC optimal number of clusters for Small Set B. . . . .	57
5.7	Performance of MAHC for the small sets in terms of F-Measure, using the L method to determine thresholds in stage 1. (a) MAHC and PSC F-Measure for Small Set A (b) MAHC optimal number of clusters for Small Set A (c) MAHC and PSC F-Measure for Small Set B (d) MAHC optimal number of clusters for Small Set B. . . .	58
5.8	Performances for the Medium Set. (a) MAHC and PSC F-Measure (b) MAHC optimal number of clusters (NC). . . . .	59
5.9	Confusion matrix of base phones of the large TIMIT dataset. The degree of shading indicates the strength of the correspondence. . . .	63
5.10	Influence of the number of subsets used by MAHC on the execution time. Classical AHC is included as a baseline. . . . .	64
6.1	Total membership per iteration of the subset containing the largest number of speech segments when applying MAHC to (a) Small Set A and Small Set B in both cases with $P = 4$ subsets and (b) the Medium Set with $P = 6$ subsets and the Large Set with $P = 8$ subsets. . . . .	67
6.2	Multi-stage agglomerative hierarchical clustering with cluster size management (MAHC+M), as also described in Algorithm 1. . . . .	68
6.3	Number of subsets $P_i$ as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to Small Set A with an initial number of subsets of $P_0 = 2$ (a and b) and $P_0 = 6$ (c and d). . . . .	71
6.4	Number of subsets $P_i$ as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to Small Set B with an initial number of subsets of $P_0 = 2$ (a and b) and $P_0 = 6$ (c and d). . . . .	71
6.5	Per-iteration execution time of modified agglomerative hierarchical clustering with (MAHC+M) and without (MAHC) cluster size management with $P_0 = 6$ initial subsets for (a) Small Set A and (b) Small Set B. . . . .	72
6.6	Number of subsets $P_i$ as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to the Medium Set with an initial number of subsets of $P_0 = 6$ (a and b) and $P_0 = 10$ (c and d). . . . .	73

6.7	Number of subsets $P_i$ as well as F-Measure for each iteration when applying modified agglomerative hierarchical clustering (MAHC) and MAHC with cluster size management (MAHC+M) to the Large Set with an initial number of subsets of $P_0 = 8$ (a and b) and $P_0 = 10$ (c and d). . . . .	74
6.8	Number of subsets $P_i$ as well as F-Measure for each iteration when applying modified agglomerative hierarchical clustering (MAHC) and MAHC with cluster size management (MAHC+M) to the Large Set with an initial number of subsets of $P_0 = 15$ (a and b). . . . .	74
6.9	Number of subsets ( $P_i$ ) for each iteration where $P_0$ is initial number of subsets. . . . .	75
6.10	Minimum occupancy per iteration for (a) Medium Set and (b) Large Set. . . . .	75
6.11	Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to Small Set A with an initial number of subsets of $P_0 = 6$ . . . . .	76
6.12	Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to Small Set B with an initial number of subsets of $P_0 = 6$ . . . . .	77
6.13	Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to the Medium Set with an initial number of subsets of $P_0 = 10$ . . . . .	77
6.14	Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of $P_0 = 8$ . . . . .	78
6.15	Triphone labels corresponding to the acoustic segments clustered by MAHC with $P_0 = 8$ and $K = 1220$ for the Large Set. The first two clusters are shown where each cluster consists of a basephone together with its left and right contexts, indicated by the $-$ and $+$ characters respectively. . . . .	79
6.16	TIMIT basephone labels of MAHC output with $P_0 = 8$ and $K = 1220$ for the Large Set. The first two clusters are shown. . . . .	79
6.17	Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when $P_0 = 8$ at iteration 6 using (a) DTW-based MAHC with the number of clusters $K = 1220$ , (b) DTW-based MAHC+M with $K = 1475$ and (c) FTDTW-based MAHC+M with $K = 1386$ . . . . .	80
6.18	Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of $P_0 = 10$ . . . . .	80

6.19	Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when $P_0 = 10$ at iteration 6 using (a) DTW-based MAHC with the number of clusters $K = 1315$ , (b) DTW-based MAHC+M with $K = 1515$ and (c) FTDTW-based MAHC+M with $K = 1560$ .	81
6.20	Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of $P_0 = 15$ . . . . .	82
6.21	Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when $P_0 = 15$ at iteration 7 using (a) DTW-based MAHC with the number of clusters $K = 1554$ , (b) DTW-based MAHC+M with $K = 1810$ and (c) FTDTW-based MAHC+M with $K = 1954$ .	83
7.1	The first two entries of the TIMIT sentence-level dictionary. . . . .	87
7.2	Initial dictionary showing the entries from the first two sentences of the TIMIT training set as indicated in Figure 7.1. . . . .	87
7.3	The trellis structure used to find the optimal alignment between the sequence of SWUs and the sequence of words in a sentence. The locus of red arrows indicates the optimal alignment path. . . . .	88
7.4	Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets $P_0 = 8$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline.	91
7.5	Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets $P_0 = 10$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline. . . . .	92
7.6	Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets $P_0 = 8$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline.	92

# List of Tables

3.1	Parameter values which define the Lance-Williams equation. . . . .	31
4.1	Datasets used for experimental evaluation. . . . .	44
5.1	Composition of experimental data. $N$ indicates the total number of segments, $L$ the total number of classes (unique number of tri-phones), $R$ the frequency of occurrence of each triphone, $V$ the total number of feature vectors in $\mathbb{R}^{39}$ and $M = N(N - 1)/2$ the number of similarities which must be computed for straightforward application of AHC. . . . .	54
5.2	Baseline results when the cutoff is determined via the F-Measure. . . . .	55
5.3	Baseline results when the cutoff is determined via the L method and the output is evaluated with the F-Measure. . . . .	56
5.4	Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Small Set A using the L method. . . . .	60
5.5	Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Small Set B using the L method . . . . .	61
5.6	Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Medium Set using the L method. . . . .	61
5.7	F-Measure performances of the L method based MAHC and the PSC algorithm. . . . .	62
5.8	Performance of the proposed method on the Large Set. . . . .	62
6.1	The F-Measures corresponding to the confusion matrices shown in Figures 6.17, 6.19 and 6.21. All are for the Large Set. . . . .	82
7.1	Average word recognition rate in percentages (%) for three sets of experiments where the number of initial subsets $P_0$ was 8, 10 and 15. . . . .	93

# List of Abbreviations

AHC	Agglomerative Hierarchical Clustering
ASR	Automatic Speech Recognition
DTW	Dynamic Time Warping
FTDTW	Feature Trajectory Dynamic Time Warping
G2P	Grapheme-to-Phoneme
HMM	Hidden Markov Models
MAHC+M	Multi-Stage Agglomerative Hierarchical Clustering with Memory Management
MAHC	Multi-Stage Agglomerative Hierarchical Clustering
MFCCs	Mel-frequency Cepstral Coefficients
NMI	Normalised Mutual Information
PLP	Perceptual Linear Prediction
SADD	Spoken Arabic Digit Dataset speech corpus
SWU	Sub-Word Unit
TIMIT	Texas Instruments and Massachusetts Institute of Technology speech corpus
UPGMA	Unweighted Pair-Group Method using Arithmetic averages
UPGMC	Unweighted Pair-Group Method using Centroids
WPGMA	Weighted Pair-Group Method using Arithmetic averages
WPGMC	Weighted Pair-Group Method using Centroids

# List of Symbols

## Features

$\mathcal{C}$	Set of all clusters
$\mathbf{C}_k$	The $k$ -th cluster
$\mathcal{X}$	Set of all speech segments in a dataset
$\mathbf{X}_i$	Acoustic speech segment feature set
$\mathbf{x}_t$	feature vector at time $t$
$\bar{\mathbf{X}}_p$	Medoid of subset $p$
$\mathcal{X}_i$	A set of acoustic segments at iteration $i$

## Variables

$\mathbf{G}$	A set of $L$ classes
$K$	Number of clusters
$L$	Number of cluster labels
$M$	Total number of HMM observations
$N$	Total number of speech segments to be clustered
$P_i$	Number of subsets in the $i$ -th iteration
$\mathbf{O}$	HMM observation sequence
$uk$	$k$ -th sub-word unit label item $[b_{ij}]$ probability of producing $j$ -th observation from the $i$ -th word

## Metrics

$d(\cdot)$	A similarity measure
$D$	Local distance matrix
$DTW(\cdot)$	Dynamic time warping distance
$FTDTW(\cdot)$	Feature trajectory dynamic time warping distance
$F - Measure$	Recall and precision based cluster evaluation measure
$NMI$	mutual information based cluster evaluation measure

# Chapter 1

## Introduction

Automatic speech recognition (ASR) systems have been designed for many applications, ranging from robotics and software aiding people with disabilities to automated call-centre systems. One of the key building blocks in such state-of-the-art ASR systems is the pronunciation dictionary, which describes how words are decomposed into sub-word units such as phones. These dictionaries are usually hand-crafted, a process which is very time consuming and requires specialist linguistic expertise. For major languages such as English, Chinese, and several other European and Asian languages, pronunciation dictionaries and extensive speech corpora have been prepared and are available for the development of speech technology. However, many of the world's languages, especially those spoken only in developing countries, lack such language resources. In many cases the linguistic expertise required to describe the pronunciation patterns and produce dictionaries may not even be available. Such languages are consequently referred to as under-resourced [2].

To address the development of speech technology in an under-resourced setting, unsupervised approaches have recently attracted increasing attention, with the aim of minimising the need for human linguistic expertise [3]. One particular aspect of this research is aimed at accelerating the generation of pronunciation dictionaries. This can be further broken down into two aspects: the determination of a suitable set of sub-word units, and the subsequent generation of pronunciations in terms of these units. The work presented in this dissertation will focus on the first of these two steps. By the development of a clustering algorithm that can be applied to large audio datasets, a means to automatically locate and group sounds that are similar is proposed. These groups of sounds can subsequently be used to generate pronunciations for the words of the language. Since the methods do not employ linguistic knowledge, they are language-independent and can therefore be applied to under-resourced languages for which speech technology could not yet otherwise be developed.



## 1.1 Current state of research

When considering the development of pronunciation dictionaries with minimal human intervention, one recent approach has been to employ bootstrapping by extracting robust grapheme-to-phoneme rules from a small seed set of pronunciations [4; 5; 6]. New pronunciations are then generated from the given orthography using the extracted rules. In some cases a non-expert human verifier assesses the pronunciations produced by the rules in an ongoing basis by listening to reconstructed audio segments. This human-in-the-loop approach allows the rules to be corrected if necessary, thereby re-introducing a measure of supervision to the learning process.

The above approach however still assumes the availability of a high quality seed dictionary. It also assumes that the set of sub-word units (usually phones) in terms of which the pronunciations will be described are known. In this dissertation we will consider the more extreme case in which no knowledge of a suitable sub-word representation for the language is available, but only some speech audio and corresponding orthographic transcriptions [2; 7]. Also this scenario has been the subject of recent research [3; 7; 8]. One proposed solution is the so-called segment-and-cluster approach, in which speech audio is first divided into segments, and subsequently these segments are clustered using an appropriate similarity measure [3]. Segmentation and clustering can also be attempted jointly, although this raises the computational complexity especially when the audio dataset is large [7; 9]. Since the segment-and-cluster approach assumes no knowledge of word or sub-word boundaries, both segmentation and clustering must be based exclusively on the properties of the acoustic data. Each resultant cluster can then be considered a sub-word unit for which an acoustic model can be trained.

An early approach to segmentation of the speech signal without additional information is to break it down into voiced and unvoiced regions. This has been investigated by several authors for a variety of applications, including speech coding [10; 11] and speech recognition [12]. More generally, segment boundaries in unlabelled audio can be hypothesised at instances where clear spectral changes occur [13]. Such discontinuities in speech spectra have been detected by critical-band analysis [14], or by sub-band analysis which employs a group-delay function for representing their locations [15; 16]. Furthermore, ten Bosch and Cranen [17] detect word-like fragments from the speech signal by a statistical word discovery method which exploits the acoustic similarity between multiple acoustic tokens of the fragments.

A different family of segmentation algorithms tries to identify recurring phrases in unlabelled audio. These techniques are based on an alternative implementation of a dynamic time warping (DTW) algorithm, which allow it to detect local sub-matches between two audio segments [18; 19; 20]. These techniques are particularly suited to detect the frequently recurring words or phrases in unlabelled audio from a single speaker and within a stable acoustic

environment. However they do not attempt to segment all the audio, but only to find frequently recurring sub-portions.

Once the speech has been segmented, the segments must be clustered. This is a challenging task due to the very large number of segments that will be present in a typical speech corpus. It is also complicated by the fact that most efficient clustering algorithms assume prior knowledge of the number of clusters. For a new and understudied language, this number will not be known. In the following chapter, a review of the literature dealing with the clustering of speech segments will be presented. The following chapters then describe the development and evaluation of a parallelisable clustering algorithm that can be applied to very large speech corpora. A key feature of this algorithm is that it automatically determines an appropriate number of clusters, and hence number of sub-word units that should be used for later acoustic modelling. This algorithm can play a key role in the automatic generation of pronunciation dictionaries based on the segment-and-cluster approach.

## 1.2 Research objectives

The overall aim of this research is to develop a clustering method that can be applied to a very large pool of speech audio segments in order to automatically determine a set of sub-word units that are suitable for acoustic modelling in ASR without prior linguistic information. The following sub-objectives will be considered.

- Determine a suitable distance measure with which to compare speech segments of variable length.
- Consider and develop a clustering algorithm which can be used to place such speech segments into groups of similar sounds.
- Develop a means of automatically determining the number of clusters the speech segments should be divided into.
- Develop a means of allowing the clustering algorithm to be applied to very large speech datasets.
- Provide a baseline indication of the effectiveness of the automatically determined clusters when used as sub-word units for acoustic modelling purposes in ASR.

## 1.3 Project scope and contributions

The task of generating pronunciations for the words in a language without any prior linguistic knowledge other than the audio and orthography is a complex

one since it encompasses three sub-tasks: segmentation, clustering and dictionary induction. Each of these sub-tasks is a challenging research field on its own. For this reason, this dissertation will focus on clustering, and will assume segmentation to have been achieved. Furthermore, only a simple dictionary induction scheme will be considered, as a means of obtaining a first indication of the effectiveness of the automatically-determined sub-word units.

Major contributions of this dissertation are:

- The development of a new iterative hierarchical clustering strategy targeted at large speech datasets for which existing approaches are computationally infeasible due to  $O(N^2)$  storage and runtime complexity. This new algorithm is named multi-stage agglomerative hierarchical clustering (MAHC) and is shown to perform well in comparison with classical approaches.
- A feature incorporated into the MAHC algorithm is a means to automatically determine the number of clusters into which the audio segments should be grouped.
- The development of an improved version of MAHC algorithm called MAHC+M to manage cluster sizes and the  $O(N^2)$  complexity.
- An improved variation of the dynamic time warping (DTW) is proposed for the computation of similarities between speech segments. This feature-trajectory DTW is shown to improve on classical DTW in terms of cluster quality.
- The automatically-determined clusters are used to induce a pronunciation dictionary for the purpose of ASR experiments.

Furthermore, the work presented in this dissertation has led to the following publications:

1. Lerato, L., Niesler, T., " Investigating parameters for unsupervised clustering of speech segments using TIMIT", In: Proceedings of *Twenty-Third Annual Symposium of the Pattern Recognition of South Africa*, pp. 83–88, 2012,
2. Lerato, L., Niesler, T., "Clustering acoustic segments using multi-stage agglomerative hierarchical clustering", *PLoS ONE* 2015;**10**(10):e0141756.
3. Lerato, L., Niesler, T., "Feature trajectory dynamic time warping for clustering of speech segments", *EURASIP Journal on Audio, Speech, and Music Processing*, Submitted, November 2018. A pre-print can be accessed from the arXiv.org website: <https://arxiv.org/abs/1810.12722.pdf>.

4. Lerato, L., Niesler, T., "Cluster size management in multi-stage agglomerative hierarchical clustering of acoustic speech segments", In final preparation stages before submission. The manuscript can be accessed from the arXiv.org website: <https://arxiv.org/abs/1810.12744>.

## 1.4 Dissertation overview

Chapter 2 begins with a literature survey of cluster analysis applied to acoustic speech segments. Clustering methods are categorically described and cluster evaluation metrics are considered. Hierarchical clustering algorithms based on agglomerative hierarchical clustering (AHC) are surveyed in Chapter 3. Descriptions of similarity measures and linkage methods are also provided here. Preliminary experiments and the development of an improved new variant of the dynamic time warping algorithm, named feature trajectory dynamic time warping (FTDTW) are presented in Chapter 4. Chapter 5 introduces a new algorithm developed as part of this dissertation named multi-stage agglomerative hierarchical clustering (MAHC). The parameters required by this algorithm are outlined and its implementation is described in detail. Evaluations highlight that in some cases MAHC does not scale well enough for large data. This leads to the development of an improved variant of MAHC in Chapter 6, with supporting experimental evaluation. The resultant clusters are used to automatically induce pronunciation dictionaries in Chapter 7. The dictionaries are used in an automatic speech recognition system. Chapter 8 concludes the dissertation by providing an overall summary, conclusion and recommendations.

## Chapter 2

# Clustering of Acoustic Speech Segments

### 2.1 Introduction

This chapter is the survey of the literature concerned with the cluster analysis of acoustic speech segments, hereafter simply referred to as acoustic segments. The clustering process discussed in this chapter does not refer to the context clustering applied during acoustic model training for speech recognition [21]. Instead, it refers to the discovery of acoustically similar groups of acoustic segments without the availability of a transcription. The intention is to allow these automatically discovered segments to ultimately be used for acoustic sub-word modelling [3].

### 2.2 A précis of clustering methods

Clustering can be described as the process of finding natural grouping(s) of a set of patterns or objects based on their similarity [22]. There are many clustering methods that can be used in the clustering of data objects such as acoustic segments. Such algorithms can be broadly classified into two groups: *hierarchical* and *partitional* [23; 24; 22]. Partitional clustering algorithms are based on the optimisation of an appropriate objective function that quantifies how well the clusters represent their members. A very common example of a partitional method is the k-means algorithm. Fuzzy c-means clustering [25] is another example of a partitional algorithm which searches for a group of fuzzy clusters together with corresponding centres that represent data formation as best as possible. Other algorithms include kernel clustering, spectral clustering and self-organising maps [26]. For partitional approaches, the number of clusters must be known beforehand, and this can present major challenges when this number is difficult to determine [27].

When the number of clusters is not known beforehand, hierarchical clus-

tering methods are a favourable choice [28; 29; 30]. In contrast to partitional approaches, these methods consider how clusters can be subdivided into sub-clusters or be grouped into super-clusters. This provides a hierarchical assignment of objects into groups. Among hierarchical methods, one can further distinguish between divisive and agglomerative approaches. The former are based on a succession of data splits that continues until each data object occupies its own cluster [31; 32]. Divisive hierarchical clustering algorithms are not commonly used in practice due to their high computational cost [29; 33]. Agglomerative hierarchical clustering (AHC), on the other hand, is a bottom-up approach that initially treats each data object as a singleton cluster and successively merges pairs of clusters until a single group remains [23].

The implementation of some of the clustering algorithms mentioned above can be either probabilistic or non-probabilistic [34; 23]. Probabilistic clustering algorithms are also called model-based clustering methods. In probabilistic approaches the assumption is that data originates from a mixture of probability distributions such that each distribution represents a cluster [31]. For hierarchical clustering algorithms in the model-based setting, a maximum-likelihood criterion is commonly used to merge clusters. In partitional clustering, expectation maximization (EM) algorithm is often used to relocate data points until convergence.

Choosing the most suitable clustering method is a challenge [22]. Furthermore for any chosen method, a prerequisite for data analysis is the choice of data representation in the form of features and the definition of a similarity measure between data objects [35]. Determining the number of clusters present in the data and the cluster validity are other important challenges in the implementation of a clustering method.

## 2.3 Acoustic segments as data objects

Acoustic segments are temporally bounded intervals of speech data that correspond to potentially meaningful sound classes, such as phonemes or sequences thereof [36]. They are vector time series of variable length representing a short period of the speech audio signal. This is mathematically represented in Equation 2.1:

$$\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \dots, \mathbf{X}_N\} \quad (2.1)$$

where  $N$  is the total number of acoustic segments (data objects) to be clustered, and  $\mathbf{X}_i$  is the  $i$ -th acoustic segment such that:

- $\mathbf{X}_i = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{n_i}\}$  where  $\mathbf{x}_t$  represents an acoustic frame as a  $v$ -dimensional feature vector in Euclidean space  $\mathbb{R}^v$ ,
- $t = 1, 2, \dots, n_i$ , and

- $n_i$  is the arbitrary length of the  $i$ -th acoustic segment  $\mathbf{X}_i$ .

At times the acoustic features of the segment  $\mathbf{X}_i$  are represented by their centroid  $\bar{\mathbf{x}}_i$  such that the entire dataset in Equation 2.1 is replaced by a sequence of  $N$  feature vectors  $\bar{\mathbf{x}}_i$  in  $\mathbb{R}^v$ , as shown in Equations 2.2 and 2.3.

$$\mathcal{X} = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3, \dots, \bar{\mathbf{x}}_N\} \quad (2.2)$$

$$\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{i=1}^{n_i} \mathbf{X}_i \quad (2.3)$$

This centroid representation is evident in various research outputs such as those of Svendsen *et al* [37], Holter and Svendsen [38], Paliwal [39] and, Mak and Barnard [40].

The representation of each acoustic segment by a set of  $n$  feature vectors ( $\mathbf{X}_i$ ) or by a centroid  $\bar{\mathbf{x}}_i$ , forms the first step from which the process of clustering commences. The most commonly used features for these representations are the Mel-frequency cepstral coefficients (MFCCs) [41]. To represent each feature vector  $\mathbf{x}_t$  as a column or row vector of MFCCs, the sampled acoustic segment signal is divided into frames of 10-30 milliseconds duration. The MFCC feature extraction algorithm generates  $v$  attribute values of  $\mathbf{x}_t$  for each frame. There are other popular feature extraction algorithms such as linear predictive coding (LPC) and perceptual linear prediction (PLP).

So far a standard feature extraction algorithm for clustering of acoustic segments has not clearly emerged from the literature. In the broad area of ASR research, however, MFCCs are a popular choice which in turn motivates their use in cluster analysis. Wang *et al* [8; 42] use 39-dimensional MFCCs to represent objects to be clustered. The same features are employed by Bacchiani and Ostendorf [9]. LPC coefficients are used in the work of Svendsen *et al* [37] when clustering acoustic segments for application to ASR. The same representation is seen in Paliwal's work on lexicon-building methods [39]. Mak and Barnard [40] also utilise 36-dimensional LPC coefficients to represent the syllable-like acoustic segments. Kamper *et al* [43] use LPC coefficients to represent unsupervised training data and PLP for supervised training data, where they perform dimensional mapping on the acoustic feature space followed by probabilistic clustering.

Clustering algorithms compute a similarity between data objects  $d(\mathbf{X}_i, \mathbf{X}_j)$  and  $i \neq j$ . A common distance measure is the Euclidean distance which belongs to the family called Minkowski distances [44], which are described in Chapter 3. For acoustic segments that are represented as centroids,  $d(\mathbf{X}_i, \mathbf{X}_j)$  is obtained with conventional similarity measures such as the Euclidean distance. When comparing the similarity between acoustic segments of variable length, the dynamic time warping (DTW) algorithm is a popular choice [45; 46]. DTW recursively determines the best alignment between the two



segments by minimizing a cumulative cost that is commonly based on the Euclidean distance between time aligned time-series vectors. DTW is described in greater detail in Chapter 4.

## 2.4 Clustering methods for acoustic segments

Literature surveys on clustering algorithms [35; 27; 47; 29; 48], show many possible algorithms, some of which have already been discussed in Section 2.2. This section will focus on those that have been applied to the specific case of sub-word modelling.

For both probabilistic and non-probabilistic clustering approaches let the set of clusters be  $\mathcal{C}$ , such that Equation 2.4 represents the set of all clusters produced by the clustering algorithm.

$$\mathcal{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K\} \quad (2.4)$$

Here  $\mathbf{C}_k$  is a subset or a cluster whose membership ideally comprises similar objects and  $K$  is the total number of clusters. In addition, there are two requirements.

1.  $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$  for  $i, j = 1, 2, \dots, K$  where  $i \neq j$ .
2.  $\bigcup_{i=1}^K \mathbf{C}_i = \mathcal{X}$  (see Equation 2.1).

The symbols  $\cap$ ,  $\cup$  and  $\emptyset$  indicate set intersection, set union and the empty set respectively.

### 2.4.1 Non-probabilistic partitional clustering

Partitional clustering methods seek to divide the data without considering how the final clusters may themselves be combined into larger groups, or be subdivided into smaller groups. They are based on the optimisation of an appropriate objective function that quantifies how well the clusters represent their members [22]. Generally, partitional clustering attempts to seek  $K$  partitions from the data  $\mathcal{X}$ . Several authors have clustered acoustic segments with partitional clustering algorithms such as k-means and spectral clustering. We describe these approaches in a little more detail in the paragraphs to follow.

Codebooks can also be employed in the partitional clustering exercise. This is evident in the work of Svendsen *et al* [37] where, upon segmentation of speech data, a pre-defined number of clusters is chosen for the clustering process. The segment quantization (SQ) algorithm is used to partition acoustic segments by representing them with their centroids. In this process a codebook of  $K$  code-vectors,  $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_K\}$  is designed such that the distortion in Equation



2.5 is minimised.

$$\text{Distortion} = \sum_{i=1}^N \min_{k \in \{1, \dots, K\}} d(\mathbf{x}_i, \mathbf{q}_k) \quad (2.5)$$

In this case  $d(\bar{\mathbf{x}}_i, \mathbf{q}_k)$  is the distortion between the centroid  $\bar{\mathbf{x}}_i$  and the code-book vector  $\mathbf{q}_k$ . The minimisation of Equation 2.5 is similar to the vector quantization problem that is often solved by the Linde-Buzo-Gray (LBG) algorithm [49]. The clusters containing the acoustic segments are then labelled to represent each of the  $K$  unique sub-word units. A similar procedure is followed in the work of Holter and Svendsen [38].

#### 2.4.1.1 The k-means algorithm

Starting from an initial partition, the k-means algorithm minimises the squared error between empirical mean of each cluster and the points in the cluster. This algorithm assumes that  $N$  data objects  $\mathbf{x}_i \in \mathbb{R}^v$  will be clustered into a known number of clusters  $K$ . This means that each cluster  $\mathbf{C}_k$ ,  $k = 1, \dots, K$  contains objects  $\mathbf{x}_i$ ,  $i = 1, \dots, N_k$ . Letting the mean of cluster  $\mathbf{C}_k$  to be  $\boldsymbol{\mu}_k$ , the sum of the squared error (SSE) between this mean and the points in the same cluster is calculated and the result is summed over  $K$  clusters as given in Equation 2.6.

$$SSE = \sum_{k=1}^K \sum_{\mathbf{x}_i \in \mathbf{C}_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \quad (2.6)$$

Here  $SSE$  is the objective function which the k-means algorithm minimizes [22]. The algorithm itself starts by partitioning data into  $K$  clusters. This is followed by generating a new partition by assigning each pattern to its closest cluster centre. Finally, new cluster centres are determined. The latter two steps are repeated until the clusters stabilise.

Paliwal [39] uses k-means to cluster acoustic segments generated by a maximum likelihood segmentation algorithm. Each cluster corresponds to one acoustic sub-word unit that is later used in training a hidden Markov model (HMM) for ASR. Segments from a Norwegian alphabet and digit corpora are represented by centroids. The k-means algorithm is applied to these centroids. A similar process is reported by Lee *et al* [50].

The k-means algorithm has several extensions and variations [22]. An example of such variation is known as embedded segmental k-means (ES-KMeans) as proposed by Kamper *et al* [51] to cluster acoustic word segments for under-resourced languages. Features of words to be clustered are represented in an embedded fixed-dimensional space, thereby allowing a direct similarity calculation without alignment. Subsequently this approach allows k-means to be applied to the embedded word features. The ES-KMeans algorithm introduces a new objective function which includes a weighting dependent on the number of frames used in the embedding of a word segment. The

ES-KMeans algorithm objective function is similar to Equation 2.7.

$$SSE_{emb} = \sum_{k=1}^K \sum_{\mathbf{x}_i \in \mathbf{C}_k \cap \mathbf{X}} \text{len}(\mathbf{x}_i) \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \quad (2.7)$$

Here  $\text{len}(\mathbf{x}_i)$  indicates the number of frames of the embedded  $\mathbf{x}_i$  and  $\mathbf{X}$  the embedding under the current segmentation. ES-KMeans minimizes  $SSE_{emb}$  by alternating between segmentation, cluster assignment and optimisation of the means. This method exhibits competitive performance when applied to large speech corpora [51].

The k-means algorithm can be used as a sub-module in other clustering methods. For example k-means is used in divisive hierarchical clustering and applied to acoustic segments by Bacchiani and Ostendorf [9] in their work on joint learning of acoustic units and a corresponding lexicon. In this case the k-means algorithm clusters the means that were obtained via divisive clustering. As another example, spectral clustering (described below) utilises the k-means as a final step of the clustering process.

#### 2.4.1.2 Spectral clustering

A typical spectral clustering algorithm [52; 53] acquires pairwise distances from  $N$   $v$ -dimensional data points located in a Euclidean space,  $\mathbb{R}^v$ , and constructs a dense similarity matrix  $Y \in \mathbb{R}^{N \times N}$ . In some cases  $Y$  can be modified to be a sparse matrix. Subsequently, the Laplacian matrix,  $L = B - Y$ , is computed where  $B$  is a diagonal matrix whose entries are row/column sums of  $Y$ . Spectral clustering requires the number of clusters,  $K$ , to be specified so that the first  $K$  eigenvectors of  $L$  can be computed and stored as the columns of a new matrix,  $A \in \mathbb{R}^{N \times K}$ . Finally the k-means algorithm is used to cluster the  $N$  rows of the matrix  $A$  into  $K$  groups.

Wang *et al* [8] have considered the clustering of speech segments using spectral clustering. A speech signal is first divided into non-overlapping segments using an Euclidean-based distortion measure. The dataset is represented in terms of a distance matrix whose rows represent the number of Gaussians while the columns correspond to the number of acoustic segments. Gaussian component clustering (GCC) and segment clustering (SC) are applied, where GCC applies spectral clustering to a set of Gaussian components and SC applies spectral clustering to a large number of speech segments. The final step employs multiview segment clustering (MSC), which takes a linear combination of the Laplacian matrices obtained from different posterior representations and derives a single spectral embedding representation for each segment. The OGI-MT2012 corpora were used for experimentation. Clusterings were evaluated using both purity and normalised mutual information (NMI) [54]. The authors had previously reported similar work also utilising GCC and SC [42] where data was converted into segment-level Gaussian posteriors (SGP's)

and then consolidated into distance matrix of size  $M$  Gaussians by  $N$  segments. In this case clustering is carried out using the normalized cut [55] approach with a pre-determined number of clusters.

### 2.4.2 Hierarchical clustering

Hierarchical clustering, which includes agglomerative and divisive variants, is not a popular choice for the acoustic modelling of sub-word units. However, this dissertation includes a strong focus on this approach. Literature points to the research by Mak and Barnard [40] where clustering of biphones is carried out using the Bhattacharyya distance. Although this distance is probabilistically measured, the singleton biphones at the top of the dendrogram are each represented by a Gaussian acoustic model. Agglomerative hierarchical clustering (AHC) is used to merge similar biphones using the Bhattacharyya distance until only one cluster is left. Building Gaussian models for a single biphone leads to a possibility of insufficient data and incomplete biphone coverage. This is solved by a proposed two-level clustering algorithm. The first step is to cluster monophones using conventional AHC until a fair amount of data enough to create a model is obtained. Acoustic models are then re-computed following which a final AHC step is performed. The OGI\_TS corpus is used to evaluate the results of this method.

### 2.4.3 Probabilistic clustering methods

Probabilistic (Model-based) clustering methods are most commonly based on Gaussian mixture models (GMMs) [31; 56]. The data objects are presented as a set of  $N$   $v$ -dimensional points in Euclidean space such that  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . The assumption is that  $\mathbf{x}_i$  is drawn from the  $k$ -th mixture component of a GMM. A GMM is defined by a set of three parameters:  $\lambda = \{\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ . The distribution of the data points according to a GMM is given in Equation 2.8.

$$p(\mathbf{x}) = \sum_{k=1}^K \boldsymbol{\pi}_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (2.8)$$

Here  $\boldsymbol{\pi}_k$ ,  $\boldsymbol{\mu}_k$ , and  $\boldsymbol{\Sigma}_k$  are the mixture weights, the means and the covariance matrices respectively. These parameters are usually estimated using the expectation maximization (EM) algorithm [27; 47]. In general, the maximum-likelihood criterion is used to select the parameters  $\lambda$  that maximise the log-likelihood given by Equation 2.9.

$$\lambda = \underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^n p(\mathbf{x}_i | \lambda) \quad (2.9)$$

Variations of GMM-based clustering have been applied by several authors. The use of Equations 2.8 and 2.9 is reported by Kamper *et al* [43] where the

parameters  $\lambda$  are estimated using the EM algorithm. A second variation of the GMM known as the finite Bayesian Gaussian mixture model (FBGMM) is also considered. In the FBGMM, parameters  $\lambda$  are treated as random variables whose prior distributions are specified. This leads to a GMM being defined by using conjugate priors: a symmetric Dirichlet prior for  $\boldsymbol{\pi}$  and a Normal-inverse-Wishart (NIW) prior for  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$ . The infinite GMM is subsequently introduced by the same authors where the Dirichlet process prior is utilised as a modification in defining the mixture weights  $\boldsymbol{\pi}$  thereby enabling an automatic inference of  $K$ . Of the three clustering approaches, it is found that the IGMM performs better than the others in terms of purity, adjusted rand index (ARI) and one-to-one cluster validity measures (see Section 2.6) when applied to word segments obtained from the Switchboard English corpus. Further work by Kamper *et al* [57] introduces a means of joint segmentation and clustering for word-like segments using the unsupervised Bayesian model; this time evaluating the result in terms of speech recognition.

A probabilistic approach to divisive hierarchical clustering of acoustic segments is also possible. This is for example proposed in the work of Bacchiani and Ostendorf [9; 58; 59] concerning the joint learning of a unit inventory and corresponding lexicon from data. In this strategy, a segmentation criterion is applied to acoustic data where acoustic segments with fixed lengths are obtained via dynamic programming. A statistical model is obtained containing the parameters mean  $\boldsymbol{\mu}_i$ , covariance  $\boldsymbol{\Sigma}_i$  and total segments length  $n_i$ . The log negative likelihood is used to compute the distance between the data and the given model, which enables the assignment of an observation to a cluster. Binary divisive clustering is applied to the data where the lowest average likelihood per frame selects the split. After the split two new clusters are defined by obtaining the cluster mean and applying binary k-means clustering. A pre-determined number of clusters triggers a final application of the k-means algorithm over all the data. The final partitioned clusters are considered as the lexicon and are used for the automatic speech recognition.

## 2.5 Determining the number of clusters

The number of clusters corresponds to the number of sub-word units that will later be used to model the speech of the language in question. For many under-resourced languages, this number may not be known. It would therefore be a great advantage if the clustering algorithm was able to determine the appropriate number of clusters automatically. Very little attention has yet been paid to this aspect in the literature. Example of a probabilistic clustering where the algorithm assumes no prior knowledge of clustering is that of Kamper *et al* [43] where the infinite Gaussian mixture model (IGMM) is employed. Another notable exception is agglomerative hierarchical clustering described in Chapter 3, since it provides a natural mechanism to automatically

determine the number of partitions. A bulk of research in clustering is based on the assumption that the number of clusters is known.

Wang *et al* assume that the number of phonemes is known beforehand from manual transcriptions, and defer the automatic determination of this number of clusters to future work [42; 8]. Svendsen assumes a known fixed number of clusters when applying the segment quantization algorithm to clustering speech segments [37]. Later, Holter and Svendsen make the same assumption when applying the LBG-algorithm [38]. Paliwal [39] deploys the k-means algorithm, which also assumes a pre-determined number of clusters. Bacchiani and Ostendorf [9] cluster data with known boundaries, also assuming a pre-determined number of clusters. The assumption of prior knowledge of the number of clusters is a consequence of the limited availability of clustering algorithms that do not require this as input [31].

## 2.6 Cluster validation methods

According to Jain [22] cluster validity refers to a formal criterion used for the quantitative evaluation of results obtained after a process of clustering. Clustering results can be evaluated on the application itself [60]. For example the clustered speech segments can be used to create acoustic models which are evaluated on the ASR application.

Literature suggests that there is no single metric that always suits a particular application [61; 60; 31]. For example Paliwal [39] evaluates clustering results by applying the automatically obtained acoustic sub-word units from varying number of clusters to automatic word recognition.

When ground truth is available, external evaluation metrics [61] can be used to evaluate the quality of the clusters. External metrics use the prior knowledge about the data; usually in the form of labels, to assess the quality of the experimentally determined clusterings [61]. However, since the aim in this project is to extend the work to speech datasets under zero-resource assumption where such labels are not available, internal metrics will also be considered [62]. Internal metrics are based only on the information intrinsic to clustered data and do not require ground truth labels.

### 2.6.1 External clustering validation

Several external clustering evaluation methods have been proposed in the literature. These methods compute a quality score for an automatically generated partition by comparing it with the ground truth (obtained from human expertise). They can mostly be categorised into those that are entropy based, those that are based on counting pairs and those that use mutual information [61].

Literature surveys and comparative studies list many possible external methods. Jain [35] lists the Rand index (RI), Jaccard Index (JI), Fowlkes

and Mallows and  $\Gamma$  statistic. Desgraupes [63] provides mathematical definitions for the same indices along with many others. There are several other popular cluster evaluation criteria which include purity, normalised mutual information (NMI) and the F-Measure [47]. Amigó *et al* [61] compare some of these methods using constraints which are based on cluster homogeneity and compactness, rag bag and cluster size and quantity. They subject evaluation measures such as purity, RI, JI, NMI and the F-Measure to data and investigate how they perform. They further propose a variant of the F-Measure called BCubed. Vinh *et al* [64] also review the RI and NMI variants and give details regarding a measure termed adjusted Rand index (ARI). A general consensus to use purity and the F-Measure as common metrics is confirmed by Rosenberg and Hirschberg [65] who further propose a new entropy-based index called the V-Measure. This method is based on completeness and homogeneity of a cluster.

In acoustic segment clustering, only a few authors use these extrinsic methods to evaluate their algorithm outputs. A few examples include Wang *et al* [8; 42] and Kamper [43]. Wang *et al* use two external evaluation methods, namely the F-Measure and normalised mutual information (NMI) in [42] whereas in [8] their evaluation is based on purity and NMI. Kamper *et al* evaluate the output of the model-based clustering algorithm using purity, adjusted Rand index, one-to-one mapping and standard deviation of cluster size.

The following paragraphs will provide a detailed explanation of the external methods that are both common and also used for evaluation in cluster analysis of acoustic segments. Throughout this text, it is assumed that a dataset of size  $N$  objects is to be partitioned into  $K$  clusters and that there are  $L$  different classes, which correspond to the number of unique labels among all data objects. With this assumption, the mathematical description of the indices is formulated around the following notation.

- $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\}$  where  $\mathbf{C}$  is the set of  $K$  clusters.
- $\mathbf{G} = \{\mathbf{G}_1, \dots, \mathbf{G}_L\}$  where  $\mathbf{G}$  is the set of  $L$  classes.
- $\mathbf{G}_l$  is a set of segments with the same label. The name of the label is the same as the name of the class.
- $|\mathbf{C}_k \cap \mathbf{G}_l|$  represents the number of data points in class  $\mathbf{G}_l$  present in cluster  $\mathbf{C}_k$
- $|\mathbf{G}_l|$  represents the number of data points of class  $\mathbf{G}_l$ .
- $|\mathbf{C}_k|$  represents the cardinality of cluster  $\mathbf{C}_k$ .

### 2.6.1.1 Purity

Purity finds the dominant class in each cluster and assigns that class to such a cluster. Its value is obtained by counting the frequently occurring classes and

dividing it by the total number of objects in the whole data as described by Equation 2.10.

$$Purity(\mathbf{C}, \mathbf{G}) = \frac{1}{N} \sum_{k=1}^K \max_l |\mathbf{C}_k \cap \mathbf{G}_l| \quad (2.10)$$

Purity values range from 0 for bad clustering to 1 when clustering is perfect. One disadvantage of purity is that if each data object occupies its own cluster, purity will be equal to 1. High purity can be achieved as the number of clusters increases even if they are bad [47]. Nevertheless, the NMI is one of the measures that tries to address this problem. This measure is still valuable as indicated in [43] and [8].

### 2.6.1.2 Adjusted Rand index

The adjusted Rand index (ARI) proposed by Hubert and Arabie [66] is a very popular external validation method. It is a variant of the Rand index (RI), which measures the percentage of clustering decisions that are correct [67; 68]. The type of decisions considered are: (1) a true positive (TP) where two similar segments are assigned to the same cluster, (2) a true negative (TN) which assigns two dissimilar segments to different clusters. The sum of TP and TN are the correct decisions. In addition, a false positive (FP) occurs when two dissimilar segments are assigned to the same cluster and a false negative (FN) places two similar segments into different clusters.

The RI is quantitatively the number of correct decisions divided by the total number of decisions made, as given by Equation 2.11.

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.11)$$

Here  $TP + FP + FN + TN = \binom{N}{2}$ ,  $TP + FP = \sum_{i=1}^K \binom{|\mathbf{C}_i|}{2}$  and  $TP = \sum_{i=1}^M \binom{Q_i}{2} + 1$ .  $Q_i = \max_j |\mathbf{C}_i \cap \mathbf{G}_j|$ .  $FN$  and  $TN$  are computed in a similar fashion.

The Rand index weighs false positives and false negatives equally and it is hard to achieve a trade-off between putting dissimilar segments together and separating similar data points. This is addressed by the adjusted rand index [64] which is given in Equation 2.12.

$$ARI = \frac{N(TP + TN) - [(TP + FP)(TP + FN) + (FN + TN)(FP + TN)]}{N^2 - [(TP + FP)(TP + FN) + (FN + TN)(FP + TN)]} \quad (2.12)$$

The ARI picks the cluster  $\mathbf{C}$  and the class  $\mathbf{G}$  partitions at random such that the cardinality of each partition is fixed. The two partitions are compared using the contingency table with rows representing classes and columns clusters. This ensures that each entry corresponds to the number of class objects that



appear in the  $i$ -th cluster  $|\mathbf{C}_i \cap \mathbf{G}_j|$ . With individual row sums and column sums the values in Equation 2.12 can easily be determined as illustrated in [67]. The ARI is also known as the adjusted-for-chance version of the RI. It is 0 for poor clustering and 1 when clusters are well partitioned.

### 2.6.1.3 Normalised mutual information

Normalised mutual information (NMI) is based on the mutual information,  $I(\mathbf{C}, \mathbf{G})$  between classes and clusters [69; 64; 47]. The mutual information, which is not sensitive to a varying number of clusters, is normalised by a factor based on the cluster entropy  $H(\mathbf{C})$  and class entropy  $H(\mathbf{G})$ . These entropies measure cluster and class cohesiveness respectively. The NMI criterion is given in Equation 2.13.

$$NMI(\mathbf{C}, \mathbf{G}) = \frac{I(\mathbf{C}, \mathbf{G})}{\frac{1}{2} [H(\mathbf{C}) + H(\mathbf{G})]} \quad (2.13)$$

The mutual information  $I(\mathbf{C}, \mathbf{G})$  and the entropies  $H(\mathbf{C})$  and  $H(\mathbf{G})$  are given in Equations 2.14, 2.15 and 2.16 respectively.

$$I(\mathbf{C}, \mathbf{G}) = \sum_{k \in \mathbf{C}} \sum_{l \in \mathbf{G}} P(\mathbf{C}_k) P(\mathbf{G}_l) \log \frac{P(\mathbf{C}_k \cap \mathbf{G}_l)}{P(\mathbf{C}_k) P(\mathbf{G}_l)} \quad (2.14)$$

In Equation 2.14,  $P(\mathbf{C}_k)$ ,  $P(\mathbf{G}_l)$  and  $P(\mathbf{C}_k \cap \mathbf{G}_l)$  are the probabilities of a segment belonging to cluster  $\mathbf{C}_k$ , class  $\mathbf{G}_l$  and their intersection respectively.

$$H(\mathbf{C}) = - \sum_{k \in \mathbf{C}} P(\mathbf{C}_k) \log P(\mathbf{C}_k) \quad (2.15)$$

$$H(\mathbf{G}) = - \sum_{l \in \mathbf{G}} P(\mathbf{G}_l) \log P(\mathbf{G}_l) \quad (2.16)$$

It can be shown that  $I(\mathbf{C}, \mathbf{G})$  is zero when the clustering is random with respect to class membership and that it achieves a maximum of 1 for perfect clustering [47].

### 2.6.1.4 The F-Measure

One of the common external validation measures is the F-Measure attributed to Larsen and Aone [70]. It assumes that each data object,  $\mathbf{X}$ , has a known label (class) representing the ground truth [47; 30]. Like other external measures the F-Measure can be used to quantify the quality of a division of the acoustic segments in the dataset into one of  $K$  clusters. The F-Measure is based on the measures recall and precision for each cluster with respect to each class in the dataset. In describing this method we will use "cluster  $k$ " to mean  $\mathbf{C}_k$  and "class  $l$ " to mean  $\mathbf{G}_l$ .



Assume that, for class  $l$  and cluster  $k$ , we know (a) the number of objects of class  $l$  that are in cluster  $k$  (b) the total number of objects in cluster  $k$  and (c) number of objects in class  $l$ . Now precision and recall are calculated by Equations 2.17 and 2.18 respectively.

$$Precision(k, l) = \frac{|\mathbf{C}_k \cap \mathbf{G}_l|}{|\mathbf{C}_k|} \quad (2.17)$$

$$Recall(k, l) = \frac{|\mathbf{C}_k \cap \mathbf{G}_l|}{|\mathbf{G}_l|} \quad (2.18)$$

Precision indicates the degree to which a cluster is dominated by a particular class, while recall indicates the degree to which a particular class is concentrated in a specific cluster. The F-Measure,  $F$ , is calculated as follows:

$$F(k, l) = \frac{2 \times Recall(k, l) \times Precision(k, l)}{Recall(k, l) + Precision(k, l)} \quad (2.19)$$

where  $k = 1, 2, \dots, K$  and  $l = 1, 2, \dots, L$ . An F-Measure of unity indicates that each class occurs exclusively in exactly one cluster; a perfect clustering result. When computing the F-Measure,  $K \times L$  iterations are required within each of which each cluster is searched for objects of class  $l$ .

## 2.6.2 Internal clustering validation

Internal clustering validation validate cluster quality without use of data labels and hence without prior knowledge of the expected number of partitions. These methods are optimised for a certain value of  $K$ . This value of  $K$  at the optimum level can be regarded as a number of clusters. Researchers over the years have tried to address the challenge of producing a suitable method wherein an optimal number of clusters is automatically determined from the clustering process itself. A common starting point for investigating internal clustering methods is the study of Milligan and Cooper [71] where 30 such methods are compared based on well-posed simulated data. In their findings the Caliński and Harabasz's index (CH) [72] performed better than the rest of other methods in terms of getting the number of clusters from well separated data.

A recent survey of internal clustering validation measures was carried out by Liu *et al* [62] where eleven of them are considered as the widely used ones. In their review, Liu *et al* present the 11 measures along with the proposal of a method called clustering validation index based on nearest neighbours (CVNN) which outperforms all of the others. It is also shown under this study that these methods are based on compactness and separation criteria. This notion is confirmed in another survey of internal methods by Halkidi *et al* [73]. Compactness measures the closeness of objects in a cluster using measures such as variance or other distance measures. Separation measures how

separated or distinct different clusters are. Example of the internal validation cluster measures listed by Liu *et al* include Caliński and Harabasz's (CH) index, Silhouette index (Sil) [74], Dunn's index (Dunn) [75] and Davies-Bouldin index (DB) [76]. There are too many other internal cluster validation methods whose descriptions can be found in [63].

To avoid clutter in this presentation only a few internal methods are described. This choice is based on their common references mentioned above. Other methods included in this report have been chosen because of their application in acoustic segment clustering. Another point to consider in choosing the methods is those that can deal with arbitrary shapes of data. Recently Starczewski [77] in the implementation of the new validity index called the STR index describes among others the Dunn, DB and the silhouette (Sil) validity indexes as the most commonly used. When proposing a new index called the jump method, Sugar and James [78] also suggest the CH and Sil indexes as one of the popular strategies. The other popular method is the gap statistic method proposed by Tibshirani *et al* [79]. This is strengthened by Yan and Ye [80] who propose the weighted version of the same procedure. In the same investigation, Yan and Ye highlight that CH and SIL are among others the most popular. They also include Hartigan's rule and Krzanowski and Lai's indexes as other common methods.

When clustering with hierarchical methods detailed in Chapter 3, a knee shaped plot of inter-cluster similarity values versus the number of clusters is produced. It is hypothesised that the optimum number of clusters occurs at the knee of this plot [62]. Hence the location of the knee can be used to estimate the optimal number of clusters even when no ground truth is available. One method that tests this hypothesis is the L method [81] which is described in more details along with a few common internal indexes in the paragraphs to follow. In general the usage of internal validation methods is not common in cluster analysis of speech segments literature.

The general formulation for internal cluster validation methods makes use of the following parameters and variables:

- $k$  represents a variable for number of clusters.
- $K$  is the optimal number of clusters.
- $W(k)$  is the within-cluster sum of square errors.
- $B(k)$  is the inter-cluster sum of square errors.
- $N$  is the number of data objects.

### 2.6.2.1 Caliński and Harabasz's index

The Caliński and Harabasz index (CH) [72] is calculated from the formulation that data is made up of  $N$   $v$ -dimensional data points such that data  $\mathbf{X} =$

$\{\mathbf{x}_i\}_i^N$ . The data matrix  $\mathbf{X}$  has  $v$  rows and  $N$  columns. The most important parameters are the traces of dispersion matrices  $\mathbf{B}$  and  $\mathbf{W}$  which represent  $B(k)$  and  $W(k)$  respectively. The dispersion-matrices  $\mathbf{B}$  and  $\mathbf{W}$  of each group are defined in Equations 2.20 and 2.21 respectively with the assumption that the similarity measure between data objects  $\mathbf{x}_i$  and  $\mathbf{x}_j$  is Euclidean distance:

$$\mathbf{B} = \sum_{r=1}^K N_r (\bar{\mathbf{x}}_r - \bar{\mathbf{x}}) (\bar{\mathbf{x}}_r - \bar{\mathbf{x}})' \quad (2.20)$$

$$\mathbf{W} = \sum_{r=1}^K \sum_{l=1}^{N_r} (\mathbf{x}_{rl} - \bar{\mathbf{x}}_r) (\mathbf{x}_{rl} - \bar{\mathbf{x}}_r)' \quad (2.21)$$

where  $r = 1, \dots, K$ ,  $N_r = |\mathbf{C}_r|$  is the cardinality of cluster  $r$ ,  $\bar{\mathbf{x}}_r$  is the centroid of cluster  $r$  and  $\bar{\mathbf{x}}$  is the mean over all  $N$  data points.

The optimal number of clusters  $K$  is obtained by finding the value of  $k$  that maximises the index  $CH(k)$  given in Equation 2.22.

$$CH(k) = \frac{B(k)}{W(k)} \times \frac{N - k}{k - 1}, \forall k > 1. \quad (2.22)$$

### 2.6.2.2 Dunn's index

This index was introduced by Dunn [75]. It first measures the compactness of a cluster by assessing the maximum diameter of all other groups. This approach further calculates minimum pairwise distances between data elements in different clusters to quantify their separation [62]. A more compact presentation of this index is provided in [77] as shown in Equation 2.23.

$$Dunn = \min_{1 \leq i \leq k} \left( \min_{1 \leq j \leq k, i \neq j} \left( \frac{d(\mathbf{C}_i, \mathbf{C}_j)}{\max_{1 \leq r \leq k} (\delta(\mathbf{C}_r))} \right) \right) \quad (2.23)$$

Here  $\delta(\mathbf{C}_r)$  is the diameter of a cluster and  $d(\mathbf{C}_i, \mathbf{C}_j)$  is the smallest distance between two clusters,  $\mathbf{C}_i$  and  $\mathbf{C}_j$ . This distance is obtained using a nearest neighbour method. The ideal is to obtain small intra-cluster distances amongst objects in one cluster and large inter-cluster distances which indicates that optimal value of the number of clusters  $K$  is achieved at maximum value of the Dunn index in Equation 2.23.

### 2.6.2.3 Silhouette index

The Silhouette index (Sil) is ascribed to the research by Kaufman and Rousseeuw [24] after it was earlier introduced by Rousseeuw [74]. In this approach, the pairwise difference in inter-cluster and intra-cluster distances is used to measure the performance of the clustering algorithm. Silhouettes of the  $k^{th}$  cluster ( $\mathbf{C}_k$ ) among  $K$  clusters are computed by using:

1.  $a(\mathbf{x}_{ki})$  - the average distance between point  $\mathbf{x}_{ki}$  and the remainder of the points which belong to  $\mathbf{C}_k$  where  $i = 1, \dots, |\mathbf{C}_k|$ ,
2.  $b(\mathbf{x}_{ki})$  - the minimum average distance between the point  $\mathbf{x}_{ki}$  and all other points in any cluster  $\mathbf{C}$  where  $\mathbf{C} \neq \mathbf{C}_k$ .

The silhouette for each object in  $\mathbf{C}_k$  is computed by Equation 2.24.

$$Sil(\mathbf{x}_i) = \frac{a(\mathbf{x}_{ki}) - b(\mathbf{x}_{ki})}{\max(a(\mathbf{x}_{ki}), b(\mathbf{x}_{ki}))} \quad (2.24)$$

Using the result in Equation 2.24, the average silhouette for each cluster  $Sil(\mathbf{C}_k)$  and over all the data  $Sil(\mathbf{X})$  are computed using Equations 2.25 and 2.26 respectively [77].

$$Sil(\mathbf{C}_k) = \frac{1}{|\mathbf{C}_k|} \sum_{\mathbf{x}_{ki} \in \mathbf{C}_k} Sil(\mathbf{x}_{ki}) \quad (2.25)$$

$$Sil(\mathbf{X}_k) = \frac{1}{K} \sum_{k=1}^K Sil(\mathbf{C}_k) \quad (2.26)$$

Here  $K$  is the number of possible clusters. The optimal number of clusters are obtained according to Equation 2.27:

$$K = \underset{k}{\operatorname{argmax}} Sil(\mathbf{X}_k) \quad (2.27)$$

#### 2.6.2.4 The gap statistic

The gap statistic proposed by Tibshirani *et al* [79] can be used to estimate the number of clusters for both partitional and hierarchical clustering methods. The authors however evaluate it only for the k-means algorithm where they measure the within-cluster dispersion  $W$  versus the number of clusters  $k$  and producing the knee graph as the number of clusters increases. Given  $v$ -dimensional data  $\mathbf{X} = \{\mathbf{x}\}_i^N$  let  $D_r$  be the sum of pairwise distances between all points in cluster  $r$ . The pairwise distance  $d(\mathbf{x}_i, \mathbf{x}_j)$  can be a squared Euclidean, Manhattan or any other measure. The within-cluster sum of square errors is calculated in Equation 2.28.

$$W(k) = \sum_{r=1}^K \frac{1}{2|\mathbf{C}_r|} D_r \quad (2.28)$$

$W(k)$  is the pooled within-cluster sum of squares around the cluster means when Euclidean distance used. The graph of  $\log(W(k))$  is standardised by comparing it with its expectation under the null distribution of data. From this step the value of  $K$  is computed by locating the point where  $\log(W(k))$

falls the farthest below the reference curve. This leads to the definition of a Gap statistic in Equation 2.29.

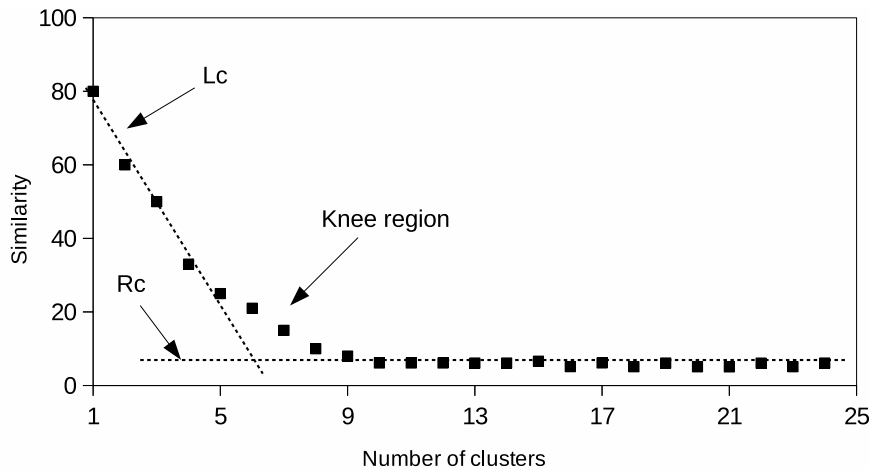
$$Gap_N(k) = E_N^*\{\log(W(k))\} - \log(W(k)) \quad (2.29)$$

Here  $E_N^*$  is the expected value under a sample of size  $N$  from the null distribution. The value of  $K$  is then given by Equation 2.30. The computational implementation of the gap statistic is discussed in [79] and [80].

$$K = \underset{k}{\operatorname{argmax}} Gap_N(k) \quad (2.30)$$

### 2.6.2.5 The L method

The L method was proposed by Salvador and Chan [81] for detecting the knee of the plot of a similarity measures versus number of clusters graph. The L method is computationally cheap, and it has received considerable attention by the research community [82]. The sketch in Figure 2.1 demonstrates one method by means of which the location of the knee may be determined.



**Figure 2.1:** Best-fit lines to locate the knee of the graph in the L method.

The similarity in the y-axis is the inter-cluster proximity or distance between clusters which decreases with increase in clusters. Example of such similarity quantities are the linkage distances from hierarchical methods described in Chapter 3.

The L method estimates the number of clusters by locating the knee region. This implementation separates regions of the curve into two parts, namely  $Lc$  and  $Rc$ . These are left ( $Lc$ ) and right ( $Rc$ ) sequences of data points partitioned at a point where  $x = c$  and  $x$  represents a number along the x-axis.  $Lc$  ranges from  $x = 2$  to  $x = c$ , with  $x = 1$  normally ignored because one cluster is not a

useful result.  $Rc$  includes points with  $x = c+1, \dots, b$ , where  $c = 3, \dots, b-2$ . The location  $c$  of the knee is found by minimising  $RMSE(c)$  as defined in Equation 2.31 in Equation 2.31:

$$RMSE(c) = \frac{c-1}{b-1} \times RMSE(Lc) + \frac{b-c}{b-1} \times RMSE(Rc) \quad (2.31)$$

The quantity  $RMSE(Lc)$  is the root mean square error of the best-fit line on the left of the knee while  $RMSE(Rc)$  is the corresponding figure to the right of the knee. The lines  $Lc$  and  $Rc$  shown in Figure 2.1 intersect at  $c$  which is considered to be at the optimal number of clusters. Since  $R$  can have a very long tail, it is suggested that the data is truncated.

## 2.7 Summary

This chapter has briefly introduced different clustering methods most of which assume a fixed dimensional data point on a Euclidean space. Acoustic segments have been presented as data objects, thereby enabling the investigation of how the existing clustering algorithms can partition them. The literature on clustering of acoustic segments has also been presented where it is found that most authors with the exception of two do not perform cluster analysis using the typical clustering validation methods. External clustering validation methods have been included. These methods use data labels for validation and they can be used for clustering evaluations in the cluster analysis of acoustic segments. Finally the internal clustering validation methods have been presented. These methods do not require data labels. It has been evident that most of them are not popular amongst researchers in the area of acoustic segments cluster analysis. The usefulness of some of these evaluation methods will be demonstrated in the following chapters during the evaluation of a hierarchical clustering method tailored for large data.

## Chapter 3

# Agglomerative Hierarchical Clustering

### 3.1 Introduction

This chapter describes the agglomerative hierarchical clustering (AHC) algorithm and discusses some of the major issues associated with it. First the AHC algorithm itself is introduced, and then linkage methods that calculate the inter-cluster dissimilarity/similarity are described. One of the major challenges of AHC is dealing with large data due to its  $O(N^2)$  complexity. Solutions to this problem suggested in the literature are also reported in this chapter.

### 3.2 Data representation

In the cluster analysis literature, a dataset,  $\mathbf{X}$ , is generally depicted as a composition of  $N$  objects that must be partitioned into  $K$  clusters. Object data are generally represented in the form shown in Equation 3.1:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathbb{R}^v \quad (3.1)$$

where each data point is represented by a  $v$ -dimensional feature vector such that the complete dataset is viewed in the form of an  $N \times v$  *pattern matrix*, [35; 47; 83]. Similar data points are clustered according to a similarity function  $d(\mathbf{x}_i, \mathbf{x}_j)$  which has the following properties:

- (a)  $d(\mathbf{x}_i, \mathbf{x}_i) = 0, \forall i$
- (b)  $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i), \forall i, j$
- (c)  $d(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ .

The input to the AHC algorithm is the list of pairwise similarities between  $N$  data objects. These pairwise similarity values are stored in a proximity ma-

trix  $\mathcal{D}$ . The properties of  $d(\cdot)$  listed above lead to a lower or upper triangular similarity matrix with  $N(N - 1)/2$  independent entries.

Distance between data objects  $d(\mathbf{x}_i, \mathbf{x}_j)$  is commonly calculated using classical distance measures of metric spaces used for quantifying the closeness of objects in a given domain. For example Minkowski distances, the Mahalanobis distance, the Hamming distance, the cosine distance and others [44; 29] can all be used to compute the value of  $d(\cdot)$ . A commonly used distance is the Euclidean distance which is a special case of the Minkowski distance given in Equation 3.2.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{l=1}^v |\mathbf{x}_{il} - \mathbf{x}_{jl}|^p \right)^{\frac{1}{p}} \quad (3.2)$$

Using Equation 3.2, the Euclidean distance is obtained by setting  $p = 2$  and the Manhattan distance is obtained by choosing  $p = 1$ . For values higher than  $p = 2$ , the metric is called the Chebyshev distance. Although the Euclidean distance is popularly used in computing the pairwise distances, some studies suggest alternatives that may perform better. Shirkorshidi *et al* [84] have recently compared the influence of different distance measures on the clustering of both low and high dimensional data. In their findings the best results for hierarchical clustering in terms of the Rand index are achieved using the Manhattan distance and the mean character difference [67]. Bouguettaya *et al* [85] use both Euclidean and Canberra distances for AHC experiments in order to compare the influence of data size and distribution in clustering. They obtain comparable performance with both metrics. Cobo *et al* [86] confirm that the Euclidean and other Minkowski distances are popular in hierarchical clustering and further show that the cosine distance [44] is also commonly applied. In a general sense, there is no one-fits-all proximity measure in agglomerative hierarchical clustering. Instead, the choice of distance measure depends on the problem at hand [29].

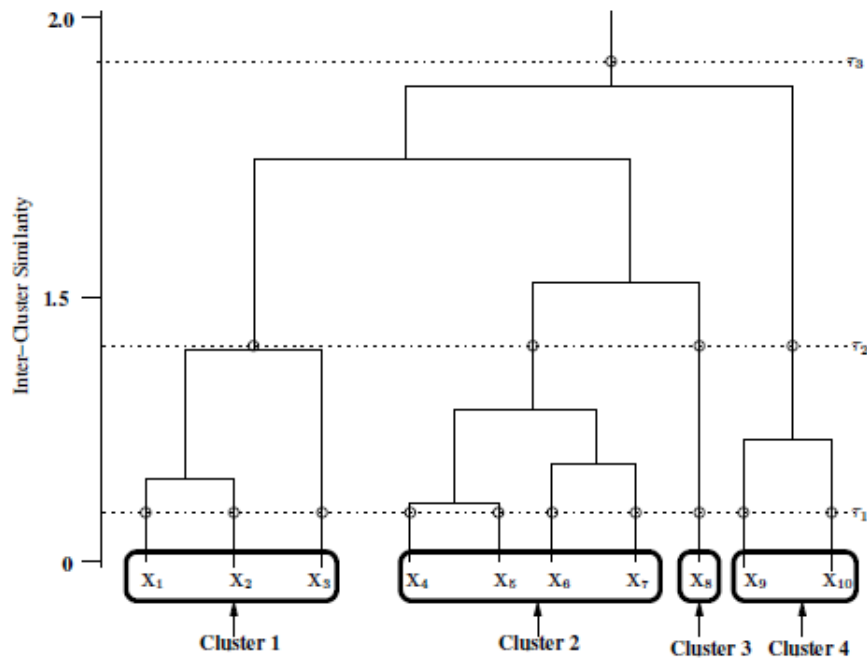
In this dissertation acoustic speech segments are used as data objects as described in Chapter 2. The generic similarity distances such as Minkowski and its derivatives would not directly apply in this case since they compare two fixed-dimension vectors. In this case vector series of differing lengths are clustered, and methods such as the dynamic time warping [45] are used to populate the proximity matrix  $\mathcal{D}$ .

### 3.3 The algorithm

A generic formulation of the clustering problem is that of  $N$  data objects being grouped into  $K$  partitions [35]. The idea is to ensure that each of the  $K$  clusters has increased intra-cluster homogeneity and high inter-cluster heterogeneity [87].



In AHC, the agglomeration of data objects is initialised by the assumption that each object is the sole occupant of its own cluster. Starting from this initial single-occupancy scenario, a binary tree structure referred to as a *dendrogram* is created by successively merging the closest cluster pairs until a single cluster remains [28]. Illustrated in Figure 3.1, the dendrogram is a structure consisting of many *U*-shaped lines that connect data points into a hierarchical tree.



**Figure 3.1:** An example of a dendrogram.

The height of each *U* represents the similarity measure or proximity between the two clusters being merged. To cluster the data, the dendrogram is *cut*, by placing a threshold on this proximity. In Figure 3.1, three possible thresholds  $\tau_1$ ,  $\tau_2$  and  $\tau_3$  are shown. This threshold is often referred to as the *cutoff* of the dendrogram. The inter-cluster proximities are normally calculated using linkage methods [35] as elaborated in Section 3.4.

### 3.4 Linkage methods

Implementations of AHC algorithms are differentiated by the way in which the measure of inter-cluster proximity is computed. Measures of inter-cluster proximity are values along the y-axis of the dendrogram, sometimes also simply called its height. Among other scholars, Müllner [88], Murtagh *et al* [89; 28] and Jain [35] categorise the inter-cluster distances as: single, complete,

average (UPGMA) and weighted (WPGMA), Ward, centroid (UPGMC) and median (WPGMC) linkage methods. The acronyms are abbreviations for the following:

- UPGMA – unweighted pair-group method using arithmetic averages.
- WPGMA – weighted pair-group method using arithmetic averages.
- UPGMC – unweighted pair-group method using centroids.
- WPGMC – weighted pair-group method using centroids.

An unweighted method assumes that each object in a cluster should be treated equally regardless of the structure of the dendrogram. A weighted method weights objects in small clusters more than those in large clusters [35].

Sneath and Sokal [90] categorise linkage methods as (a) graph-based methods and (b) geometric methods, of which the latter require cluster centres to be specified. The former category includes single, complete, UPGMA and WPGMA linkage methods while the latter includes UPGMC, WPGMC and Ward methods [91].

The fundamental mathematical descriptions of these linkage methods are presented in the following paragraphs. In Subsection 3.4.7 all linkage methods are described in terms of the Lance-Williams update formula [92] which is a popular approach in AHC algorithm implementations. These mathematical formulations have been formally presented by various authors, including Müllner [88], Legendre and Legendre [34] and Manning and Raghavan [47].

### 3.4.1 Single linkage

Single linkage was first proposed in the early 1950's [28]. Later it was developed by McQuitty [93] and again by Sneath [94]. Rohlf [95] also presents different approaches of single linkage and argues that the relevance of this approach can depend on data size. For single linkage, the distance between two clusters is the similarity of the two closest data objects in terms of a distance measure,  $d(\cdot)$  as shown in Equation 3.3.

$$d(\mathbf{C}_i, \mathbf{C}_j) = \min_{\mathbf{x}_i \in \mathbf{C}_i, \mathbf{x}_j \in \mathbf{C}_j} d(\mathbf{x}_i, \mathbf{x}_j) \quad (3.3)$$

The dendrogram is grown by progressively amalgamating the merged clusters  $\mathbf{C}_i \cup \mathbf{C}_j$  with another cluster  $\mathbf{C}_k$ . This is achieved by utilising the distance update formula [88] shown in Equation 3.4:

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \min(d(\mathbf{C}_i, \mathbf{C}_k), d(\mathbf{C}_j, \mathbf{C}_k)) \quad (3.4)$$

The major disadvantage of this linkage method is its tendency to form big and straggly clusters since the objects in the nearest neighbourhood of the two clusters to be merged may be far from the other objects. As a result, many objects can become chained together [96; 24]. Such non-compact clusters can lead to poor classification [23]. However the chaining effect can be advantageous in cases where elongated clusters are in fact required [24]. Although the single linkage method is one of the oldest criteria [97], work by Bouguettaya *et al* [85] compares the influence of data size and distribution to demonstrate its continued usefulness in AHC clustering. Computationally, single linkage presents time and space complexities of  $O(N^2)$  and  $O(N)$  respectively [98], although Murtagh reports  $O(N^2)$  complexity for both time and space [97].

### 3.4.2 Complete linkage

Like single linkage, complete linkage is one of the earliest clustering methods and was first proposed in the late 1940's. This approach is attributed to, among others, Lance and Williams [92]. Complete linkage considers the proximity between two clusters to be the distance between the furthest two objects  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in each cluster. It is calculated using Equation 3.5.

$$d(\mathbf{C}_i, \mathbf{C}_j) = \max_{\mathbf{x}_i \in \mathbf{C}_i, \mathbf{x}_j \in \mathbf{C}_j} d(\mathbf{x}_i, \mathbf{x}_j) \quad (3.5)$$

The distance update formula [88] for the dendrogram is given in Equation 3.6.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \max(d(\mathbf{C}_i, \mathbf{C}_k), d(\mathbf{C}_j, \mathbf{C}_k)) \quad (3.6)$$

Due to its maximum distance criterion, the complete linkage method is vulnerable to outliers because such anomalies will often be the most distant. It however has the advantage of normally producing very compact clusters in which members are very close to one another [24; 47]. In particular, an object joins an existing cluster only after the similarity with all other members of this cluster have been considered. This leads to the tendency for large clusters not to allow new membership [96]. Complete linkage exhibits time and space complexities of  $O(N^2)$  according to the implementation by Defays [99] and also reported by Manning and Raghavan [47] and Murtagh [89].

### 3.4.3 Average linkage

Average-linkage methods are ascribed to Sokal and Michener [100] in their work on the systematic hierarchical grouping of solitary bee species. Average linkage methods are described in the literature by Jain [35] and also by Müllner [88] in his investigation into modern AHC algorithms. Two variants of these linkage criteria are the UPGMA and the WPGMA (unweighted/weighted pair-group method using arithmetic average).

In the UPGMA and WPGMA linkage methods, the distance between two clusters  $\mathbf{C}_i$  and  $\mathbf{C}_j$  each of cardinality  $|\mathbf{C}_i|$  and  $|\mathbf{C}_j|$  is the mean of all pairwise distances between the two clusters as depicted in Equation 3.7.

$$d(\mathbf{C}_i, \mathbf{C}_j) = \frac{1}{|\mathbf{C}_i||\mathbf{C}_j|} \sum_{\mathbf{x}_i \in \mathbf{C}_i} \sum_{\mathbf{x}_j \in \mathbf{C}_j} d(\mathbf{x}_i, \mathbf{x}_j) \quad (3.7)$$

As the hierarchical tree grows, the UPGMA method uses the proportional average between the joined clusters  $\mathbf{C}_i \cup \mathbf{C}_j$  and a new cluster  $\mathbf{C}_k$  according to Equation 3.8.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \frac{|\mathbf{C}_i| \cdot d(\mathbf{C}_i, \mathbf{C}_k) + |\mathbf{C}_j| \cdot d(\mathbf{C}_j, \mathbf{C}_k)}{|\mathbf{C}_i| + |\mathbf{C}_j|} \quad (3.8)$$

For WPGMA, the merged clusters  $\mathbf{C}_i \cup \mathbf{C}_j$  are combined with a new cluster  $\mathbf{C}_k$  by computing the mean of the distances between members of  $\mathbf{C}_k$  and those of  $\mathbf{C}_i \cup \mathbf{C}_j$ , as shown in Equation 3.9.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \frac{d(\mathbf{C}_i, \mathbf{C}_k) + d(\mathbf{C}_j, \mathbf{C}_k)}{2} \quad (3.9)$$

UPGMA gives equal weights to initial pairwise similarities. It produces clusters with high variation in cardinality regardless of the true class distribution [30]. The WPGMA gives more weight to smaller clusters and less to larger ones [34]. For both implementations the complexity is  $O(N^2)$  for both time and space [97]. In general, the two variations of average linkage are sensitive to the shape and size of clusters [101].

### 3.4.4 Centroid linkage (UPGMC)

Centroid linkage considers the distance between two clusters  $\mathbf{C}_i$  and  $\mathbf{C}_j$  as the distances between their centroids [85] according to Equation 3.10:

$$d(\mathbf{C}_i, \mathbf{C}_j) = d(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) \quad (3.10)$$

where  $\bar{\mathbf{x}}_i$  and  $\bar{\mathbf{x}}_j$  are the centroids of the two clusters and  $d(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)$  is the Euclidean distance between them [88]. The earliest work using UPGMC (unweighted pair-group method using centroids) is attributed to Sokal and Michener [100]. With this linkage method, the new cluster formed after the merge is again represented by a centroid to be used in further agglomeration [34]. The distance along the vertical axis of a dendrogram is updated according to

Equation 3.11 [102]. In other implementations the square root of the right hand side of 3.11 is used [88].

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \frac{|\mathbf{C}_i| d(\mathbf{C}_i, \mathbf{C}_k) + |\mathbf{C}_j| d(\mathbf{C}_j, \mathbf{C}_k)}{|\mathbf{C}_i| + |\mathbf{C}_j|} - \frac{|\mathbf{C}_i| |\mathbf{C}_j| d(\mathbf{C}_i, \mathbf{C}_j)}{(|\mathbf{C}_i| + |\mathbf{C}_j|)^2} \quad (3.11)$$

The UPGMC method can lead to reversals where the proximity of newly merged clusters is less than their inter-cluster distance before the merge. This leads to more complex non-monotonic dendrograms in terms of inter-cluster proximity [34]. UPGMC exhibits  $O(N^2)$  complexity in terms of both time and space [88].

### 3.4.5 Median linkage (WPGMC)

The median linkage or weighted centroid clustering [34] which Sneath and Sokal [90] refer to as WPGMC (weighted pair-group method using centroids) was proposed by Gower [103] in his comparative study of linkage methods. The update formula for this method is shown in Equation 3.12. Müllner [88] presents a variation of this formula by computing the square root of the right hand side of Equation 3.12.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \frac{d(\mathbf{C}_i, \mathbf{C}_k) + d(\mathbf{C}_j, \mathbf{C}_k)}{2} - \frac{d(\mathbf{C}_i, \mathbf{C}_j)}{4} \quad (3.12)$$

Here the proximity between two clusters  $\mathbf{C}_i$  and  $\mathbf{C}_j$  is given by the Euclidean distance between their weighted centroids as indicated by Equation 3.13.

$$d(\mathbf{C}_i, \mathbf{C}_j) = d(\bar{\mathbf{x}}_{wi}, \bar{\mathbf{x}}_{wj}) \quad (3.13)$$

In Equation 3.13,  $\bar{\mathbf{x}}_{wi}$  and  $\bar{\mathbf{x}}_{wj}$  are weighted centroids of clusters  $\mathbf{C}_i$  and  $\mathbf{C}_j$ . In this case, if cluster  $\mathbf{C}_l$  is formed by the merge  $\mathbf{C}_i \cup \mathbf{C}_j$ , then  $\bar{\mathbf{x}}_{wl}$  is defined as the midpoint  $\frac{1}{2}(\bar{\mathbf{x}}_{wi} + \bar{\mathbf{x}}_{wj})$  [88].

As for UPGMC, WPGMC exhibits quadratic complexity in both time and space. It also can exhibit inversions (reversals) in the inter-cluster proximities [88].

### 3.4.6 Ward linkage

The Ward linkage criterion is a minimum-variance method proposed by Ward [104]. As for UPGMC, the Ward method calculates the proximity of two clusters using the Euclidean distance between their centroids. In contrast, however, this distance is multiplied by the ratio of cardinalities as indicated

by Equation 3.14. This equation is described by Kaufman and Rousseeuw [24] and also provided by Müllner [88].

$$d(\mathbf{C}_i, \mathbf{C}_j) = \sqrt{\frac{2|\mathbf{C}_i||\mathbf{C}_j|}{|\mathbf{C}_i| + |\mathbf{C}_j|}} d(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) \quad (3.14)$$

The Ward linkage computes the sum of squared distances between the centroid of a cluster and the members of the same cluster. This sum is similar to the error in ANOVA. Some authors propose using a mean of all possible squared pairwise distances of objects in a cluster [34]. This makes it possible to apply it to data like the acoustic segments presented in Chapter 2, where the objects to be clustered do not have a centroid since the segments have different lengths. The update formula for the Ward method is given in Equation 3.15.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \frac{1}{S}[(|\mathbf{C}_i| + |\mathbf{C}_k|)d(\mathbf{C}_i, \mathbf{C}_k) + (|\mathbf{C}_j| + |\mathbf{C}_k|)d(\mathbf{C}_j, \mathbf{C}_k) - |\mathbf{C}_k|d(\mathbf{C}_i, \mathbf{C}_j)]^{\frac{1}{2}} \quad (3.15)$$

Here  $S = |\mathbf{C}_i| + |\mathbf{C}_j| + |\mathbf{C}_k|$ .

Murtagh and Legendre [105] refer to Equation 3.15 as the Ward2 case. They present a modified equation referred to as the Ward1 case where the square root on the right hand side of Equation 3.15 is removed. Both these implementations minimize the change in variance or the sum of square errors. The update formula in Equation 3.15 was presented by Müllner [88]. Legendre and Legendre [34] show that even though the update formulas may differ, the clustering topology stays the same. Several publications that will be referred to in Section 3.5 maintain that the Ward method is one of the best performing linkage methods.

### 3.4.7 The Lance-Williams formulation of linkage methods

A generic formulation that incorporates all the linkage methods described in the preceding subsections is the Lance-Williams dissimilarity update formula given in Equation 3.16. This can be used as a merging criteria in agglomerative hierarchical clustering [28; 29; 92]. Because Equation 3.16 covers all possible linkage methods, it allows flexibility in terms of the software implementation.

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) = \alpha_i d(\mathbf{C}_i, \mathbf{C}_k) + \alpha_j d(\mathbf{C}_j, \mathbf{C}_k) + \beta d(\mathbf{C}_i, \mathbf{C}_j) + \gamma |d(\mathbf{C}_i, \mathbf{C}_k) - d(\mathbf{C}_j, \mathbf{C}_k)| \quad (3.16)$$

The values of  $\alpha_i$ ,  $\alpha_j$ ,  $\beta$  and  $\gamma$  define the agglomeration criterion used. The cardinalities of  $\mathbf{C}_i$ ,  $\mathbf{C}_j$  and  $\mathbf{C}_k$  are used in the calculations of the parameters in Equation 3.16. The Lance-Williams update formula uses the parameters  $\alpha_i$ ,

$\alpha_j$ ,  $\beta$  and  $\gamma$  to determine the clustering strategy [92; 34]. The values of these parameters are consistently presented by many authors [89; 35; 88; 28; 105] where the Lance-Williams formula parameter values are listed in Table 3.1 below.

Linkage method	$\alpha_i$	$\alpha_j$	$\beta$	$\gamma$
Single	0.5	0.5	0	-0.5
Complete	0.5	0.5	0	0.5
UPGMA	$\frac{ C_i }{ C_i + C_j }$	$\frac{ C_j }{ C_i + C_j }$	0	0
WPGMA	0.5	0.5	0	0
UPGMC	$\frac{ C_i }{ C_i + C_j }$	$\frac{ C_j }{ C_i + C_j }$	$\frac{- C_i  C_j }{( C_i + C_j )^2}$	0
WPGMC	0.5	0.5	-0.25	0
Ward	$\frac{ C_i + C_j }{ C_i + C_j + C_k }$	$\frac{ C_j + C_k }{ C_i + C_j + C_k }$	$\frac{- C_k }{ C_i + C_j + C_k }$	0

**Table 3.1:** Parameter values which define the Lance-Williams equation.

### 3.5 Comparative studies on linkage methods

Jain *et al* [27] indicate that there is no clustering technique that can uncover arbitrary structures present in multidimensional datasets and that all clustering algorithms contain implicit assumptions about cluster shapes. This means it might be challenging to appropriately choose a suitable linkage method if the true cluster shapes are not known. Despite this challenge, this section summarises some studies that recommend the choice of a linkage method.

Recently, Yim and Ramdeen [87] compared single, complete and average linkages in the clustering of bilingual language usage by 67 adults, where data objects are user ratings of the usage of Cantonese and/or English based on a scale of 0 to 100. The attributes of a data object include the proficiency in each language. In this study it was found that the average linkage outperforms the other two considered methods. Although they do not experimentally verify this, Wu *et al* [30] argue that UPGMA is a more robust and well performing clustering method than single and complete linkages. In their case UPGMA is applied to the clustering of document datasets. Sun and Korhonen [106] clustered 3000 English verbs according to their shared meaning. In the baseline experiments they used single, average, complete and Ward as linkage methods. They discovered that the Ward method outperformed the others in terms of normalised mutual information (NMI) and F-score [54; 70].

In their study of clustering 184 Caribbean maize accessions, Rincon *et al* [107] compare the performance of single, UPGMA, UPGMC and Ward linkage methods. In all cases the Euclidean distance was used as a between-object

similarity metric. Evaluation was performed using cophenetic correlation coefficient and principal component scores. According to their findings, different linkage methods can outperform each other depending on the particular experimental conditions. However for this particular dataset, UPGMA was consistently the most accurate of the four. Saraçlı *et al* [108] also use the cophenetic correlation coefficient to compare linkage methods. In their experiments datasets of different sizes ( $N = 10$ ,  $N = 50$  and  $N = 100$ ) and variable dimensionality are used. In their findings, UPGMA outperforms other considered methods when  $N = 10$ . However, when  $N = 50$  and  $N = 100$ , complete, WPGMA and UPGMC linkage methods offer improved performance. Morlini and Zani [109] apply the  $Z$  evaluation method to evaluate clustering linkages on simulated data. Using the  $S = 1 - Z$  method, these authors discover that there is no outright superior approach.

In terms of advocacy, literature surveys to a certain extent recommend the Ward linkage in agglomerative hierarchical clustering. According to Landau and Ster [33] Ward's method is popular for continuous data. The literature survey by Jain [35] reflects the consensus that Ward's method outperforms other hierarchical methods. In other studies by Blashfield [96], Hands and Everitt [110], Ferreira and Hitchcock [111] and Milligan and Cooper [112], the Ward linkage method performs better than the others considered. Murtagh and Contreras [28] report that Ward's minimum variance criterion is favoured when hierarchical clustering is applied to bibliographic information retrieval. Kuiper and Fisher [113] report that Ward and complete linkage score highly when data classes are of equal sizes, while UPGMA and UPGMC performed better when subjected to data of unequal cluster sizes. Milligan [114] found that both Ward and complete linkage do not perform well when outliers are introduced.

Hence the literature does not identify a single best linkage method. The choice is usually based on the kind and size of the dataset used. The studies mentioned here nonetheless do indicate that average and Ward linkage methods are often recommended across a variety of experimental applications of agglomerative hierarchical clustering.

### 3.6 Monotonicity in dendrograms

The AHC algorithm described in Section 3.3 produces a structure called a dendrogram shown in Figure 3.1, where the inter-cluster proximity is assumed to be monotonically increasing. Equation 3.17 describes this mathematically by considering the case where AHC merges cluster  $k$  with the previously merged clusters  $i$  and  $j$  to form a new cluster [35].

$$d((\mathbf{C}_i \cup \mathbf{C}_j), \mathbf{C}_k) \geq d(\mathbf{C}_i, \mathbf{C}_j) \quad (3.17)$$

Methods that do not satisfy the monotonicity property in Equation 3.17 are



said to exhibit reversals or inversions. In this case the similarity value in the dendrogram after the merge is lower than before the merge. The disadvantage of reversals is that it makes the interpretation of dendrograms difficult because of the unstable results. Other authors point out that it makes it difficult to draw dendrograms [34]. Median (WPGMC) and centroid (UPGMC) methods sometimes generate dendrograms that have inversions [28]. This may lead to their limited use in some applications.

### 3.7 AHC variants for large data

One of the challenges of AHC is its sensitivity to large data. Jain *et al* [27] note this problem and suggest that the data matrix should be stored in secondary memory from which data items can be transferred one at a time for clustering. They propose using  $p$  blocks of data such that each contains  $N/p$  patterns. Each block of data is clustered to produce  $k$  clusters. Finally the representatives of the clustered patterns per block are calculated and clustered again to produce the clusters.

Other researchers have addressed the problem of clustering large data using agglomerative hierarchical techniques in various ways. The early work by Narasimha and Krishna [115] proposes a multilevel technique for clustering datasets. While this work is in some respects similar to that proposed in Chapter 5, it is not iterative and has been tested only on a fairly small dataset consisting of 50 manually generated samples in a 2-dimensional Euclidean space. This data is split into  $P_1$  sub-groups. AHC is applied to each sub-group; in each case yielding  $C$  clusters (a value of  $C = 5$  was used in all experiments). After this 'first level', a representative cluster from each of the 5  $P_1$  sub-groups is determined and stored as a 'data point' for level two. Subsequently, level-two data is further divided into  $P_2$  sub-groups, and AHC is applied to each. This procedure continues until the predetermined number of levels,  $K$ , has been reached, and it is shown that the optimal value of  $K$  can be mathematically determined. An alternative way of automatically finding  $K$  is reported by Suresh [116]. Here the authors show that standard AHC is computationally more expensive than the technique they propose which is based on a two level process.

Tang *et al* [117] also propose a distributed hierarchical clustering algorithm for large data. Their aim is to improve and minimise the storage requirements of traditional implementations for execution on parallel computing architectures. They use a threshold on the similarity determined by a human expert to classify data items as unrelated, thereby making the similarity matrix sparse. The sparse similarity matrix is used to sequentially create disjoint sets of closely related data items. Each disjoint set is clustered in parallel, forming its own sub-clusters. Finally a single linkage method is used to measure similarity between these sub-clusters, which are subsequently themselves

subjected to AHC to complete the final dendrogram. The technique is tested on the MPC Orbit (MPCORB) dataset which contains approximately 380,000 asteroids, each of which is represented by a 6-dimensional feature vector. The major variable here is the threshold on the similarity and the authors show experimentally how it affects the number of disjoint sets and also the execution time at each step of the algorithm.

Cobo *et al* [118] employ a subspace clustering paradigm [119] which assumes that high dimensional data objects lie around a union of subspaces such that clustering can be performed independently in each subspace. The data used in this work represent the activities performed by learners in an online discussion forum. A total of 3842 written posts were captured from 672 students over a period 333 days. Each student is represented by a multi-dimensional feature vector which represents attributes relating to their participation in online discussions. The features are classified as coming from either a reading or a writing domain. Feature vector attributes relating to the writing domain include the ratio of reply posts written by a learner relative to the total number of reply posts, and the ratio of learners who replied at least once relative to the total number of learners. Feature vector attributes relating to the reading domain include the ratio of posts read by a learner as a fraction of total number of posts and the ratio of threads where a learner reads at least one post relative to the total number of threads. These data are clustered using AHC in a first stage where learners with similar activity patterns are grouped together separately in the two domains. AHC acquires normalised Euclidean distance pairs as input and calculates inter-cluster similarities using the complete linkage method. The second stage of the method entails the grouping together of those learners who belong to the same clusters in both reading and writing domains. The participation profiles of the learners are mapped to the final clusters by observing and comparing the values of the parameters that characterise the learners' activity patterns in each cluster. This work was subsequently advanced by Cobo [120] by including more domains. The algorithm presented by Cobo *et al* [118] depends on the data belonging to separate domains or subspaces. These subspaces are identifiable beforehand so that parallel clustering processes can be applied separately to each one. The challenge with this approach will be its application on the data that is not labelled and not categorised into known domains.

### 3.8 Summary

This chapter has provided a description of agglomerative hierarchical clustering (AHC). Different approaches to determining pairwise distances between data points have been highlighted, and the process by means of which these are used to synthesise a structure called a dendrogram is described. The original formulations of linkage methods required by AHC have been detailed along

with their Lance-Williams formulations. Previous comparative studies on the performance of linkage methods have been consulted and it was concluded that there is no specific linkage method recommended in all situations. There was some consensus, however, that the Ward method is a good general choice, especially for continuous data. Finally the chapter shows that when the dataset becomes large, specialised variants of AHC that are able to process such data become necessary. In the chapters that follow, we will first show how pairwise distances between time sequences of different length can be computed using dynamic time warping algorithm and then propose an iterative multi-stage AHC algorithm that is suitable for processing large data.

## Chapter 4

# Speech Signal Similarity Computation using Dynamic Time Warping

### 4.1 Introduction

Dynamic time warping (DTW) can be used to compute the similarity between time series such as acoustic segments. This chapter will describe the fundamentals of DTW. Subsequently a modification to DTW that aligns individual feature trajectories instead of feature vector sequences is described. This algorithm is termed feature trajectory dynamic time warping (FTDTW). Experiments using MFCC and PLP features extracted from portions of the TIMIT as well as from the Spoken Arabic Digit Dataset (SADD) demonstrate the effectiveness of FTDTW when used to cluster speech signals.

### 4.2 Dynamic time warping algorithms

Dynamic Time Warping (DTW) is a method of optimally aligning two distinct time series of generally different length. In addition to the alignment, DTW computes a score indicating the similarity of the two sequences. This ability to quantify the similarity between time series has led to the application of DTW in automatic speech recognition (ASR) systems several decades ago [121; 45]. It has remained popular in this field, with more recent developments reported in [122] and [123].

DTW has also found application in fields related to ASR. For example, it has been used successfully in keyword spotting and information retrieval (IR) systems [124; 125; 126]. To accomplish IR, sub-sequences in a speech signal that match a template with certain degree of time warping are detected.

In the related task of acoustic pattern discovery, DTW can be allowed to consider multiple local alignments between speech signals during the overall

search [18]. In this way DTW can find similar segment pairs in speech audio, followed by a clustering step [127]. The resulting cluster labels are used to train hidden Markov models (HMMs).

In an effort to improve performance, several variations of DTW in speech processing have been proposed since its inception. For example, a one-against-all index (OAI) for each time series under consideration is proposed in [123]. The OAI is subsequently used to weight the corresponding DTW alignment scores in a speech recognition system. DTW has also been modified to allow the direct matching of points along the best alignment for use in a signature verification system [128]. A stability function is subsequently applied, and the resulting score is used as a similarity measure.

### 4.2.1 Classical dynamic time warping

In describing the classical formulation of DTW, speech segments are considered to be temporal sequences of multidimensional feature vectors in the Euclidean space. Sequences are of arbitrary and generally different length, and all feature vectors are of equal dimension. The DTW algorithm recursively determines the best alignment between two such vector time series by minimizing a cumulative path cost that is commonly based on Euclidean distances between time aligned vectors [45; 129].

Consider  $N$  such sequences  $\mathbf{X}_i$ ,  $i = 1, 2, \dots, N$ , each composed of  $n_i$  feature vectors, as defined in Equation 2.1. Each feature vector  $\mathbf{x}_t$  has  $v$  dimensions. Two sequences  $\mathbf{X}_i$  and  $\mathbf{X}_j$  are aligned by constructing an  $n_i$ -by- $n_j$  local distance matrix  $D_{ij}(p, q)$  whose entries contain the distances  $d(\mathbf{x}_{ip}, \mathbf{x}_{jq})$ . Here  $i$  and  $j$  distinguish feature vectors from the two different sequences  $\mathbf{X}_i$  and  $\mathbf{X}_j$  while  $p$  and  $q$  index the feature vectors themselves.

Typical choices for  $d$  are the Euclidean distance and the Manhattan distance. The Manhattan distance described in Section 3.2 is used throughout all the experiments reported in this dissertation. A matrix of minimum accumulated distances  $\gamma_{ij}(p, q)$  is then constructed by considering all paths from  $D_{ij}(1, 1)$  to  $D_{ij}(p, q)$  up to  $D_{ij}(n_i, n_j)$ .

The DTW algorithm is based on both local and global constraints. The starting point  $D_{ij}(1, 1)$  of the alignment path and its final point  $D_{ij}(n_i, n_j)$  are known as endpoint constraints or also as global constraints [130; 131]. Another global constraint that is often applied is the restriction that  $|n_i - n_j| < \phi$  where  $\phi$  is the maximum number of frames allowed between the alignments of  $\mathbf{X}_i$  and  $\mathbf{X}_j$ . Myers *et al* [45] found that values of  $\phi$  approaching 1 yield better results.

Local constraints may include:

- Monotonicity, which requires that the slope of the alignment path is never negative.
- Allowing the path to the grid point  $(p, q)$  on the accumulated distance matrix to originate only from certain points. For example, the path might

be allowed to originate only from the three points  $(p, q - 1)$ ,  $(p - 1, q - 1)$  and  $(p - 1, q)$ .

- The optimal path may not be flat for two consecutive frames, that is  $(p, q)$  can not be connected to point  $(p - 1, q)$  if this is preceded by  $(p - 2, q)$ .

There are many different local constraints that can be used to influence the alignment [130]. For the classical formulation of DTW, the local imposed constraint is that a path to the point  $(p, q)$  can only originate from  $(p, q - 1)$ ,  $(p - 1, q - 1)$  or  $(p - 1, q)$ . Using this constraint and the global constraints described above with the exception of  $\phi$ ,  $\Gamma_{ij}(p, q)$  is computed recursively according to the principle of dynamic programming, as shown in Equation 4.1 [45].

$$\Gamma_{ij}(p, q) = D_{ij}(\mathbf{x}_{ip}, \mathbf{x}_{jq}) + \min \{ \Gamma_{ij}(p-1, q-1), \Gamma_{ij}(p-1, q), \Gamma_{ij}(p, q-1) \} \quad (4.1)$$

In Equation 4.1,  $\Gamma$  is the matrix of accumulated path scores. We will denote the similarity between two acoustic segment sequences  $\mathbf{X}_i$  and  $\mathbf{X}_j$  computed using DTW by  $DTW(\mathbf{X}_i, \mathbf{X}_j)$  and calculate it according to Equation 4.2.

$$DTW(\mathbf{X}_i, \mathbf{X}_j) = \Gamma_{ij}(n_i, n_j) \quad (4.2)$$

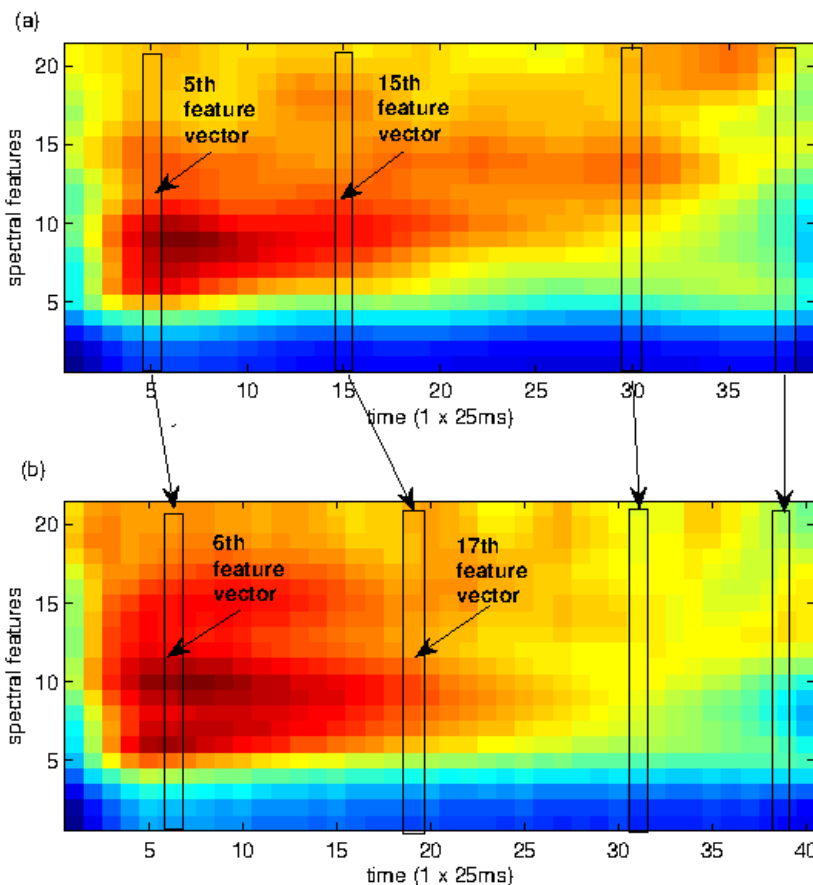
It is common to normalise this similarity by  $\lambda$ , the length of the optimal path from  $D_{ij}(1, 1)$  to  $D_{ij}(n_i, n_j)$ . It has been found that this normalisation improves the performance of the algorithm when used in time series classification such as isolated word recognition and signature recognition [45; 132]. Equations 4.1 and 4.2 represent the standard formulation of dynamic time warping and will be referred to as *classical* DTW in the remainder of this work.

To illustrate classical DTW, Figure 4.1 shows the alignment of 21-dimensional spectral feature vectors representing the same sound uttered by two different speakers. These spectral features were obtained by straightforward binning of the short-time power spectra calculated for each frame. To avoid clutter, the alignment of just four of the feature vectors is shown.

### 4.2.2 Feature trajectory DTW (FTDTW)

We propose feature trajectory DTW (FTDTW) as a modification of classical DTW which exploits the asynchronous temporal structure of features extracted from speech. Related work has considered such feature trajectories by training separate hidden Markov models (HMMs) for each MFCC feature dimension [133].

For FTDTW, a feature trajectory  $X_i^{(l)}$  is defined as the time series obtained when considering only the  $l$ -th element of each feature vector in a



**Figure 4.1:** Alignment of spectral features for the triphone  $b-aa+dx$  extracted from the TIMIT corpus [1] for (a) the male speaker mrk0 and (b) the female speaker fdml0.

sequence  $\mathbf{X}_i$ , as shown in Equation 4.3.

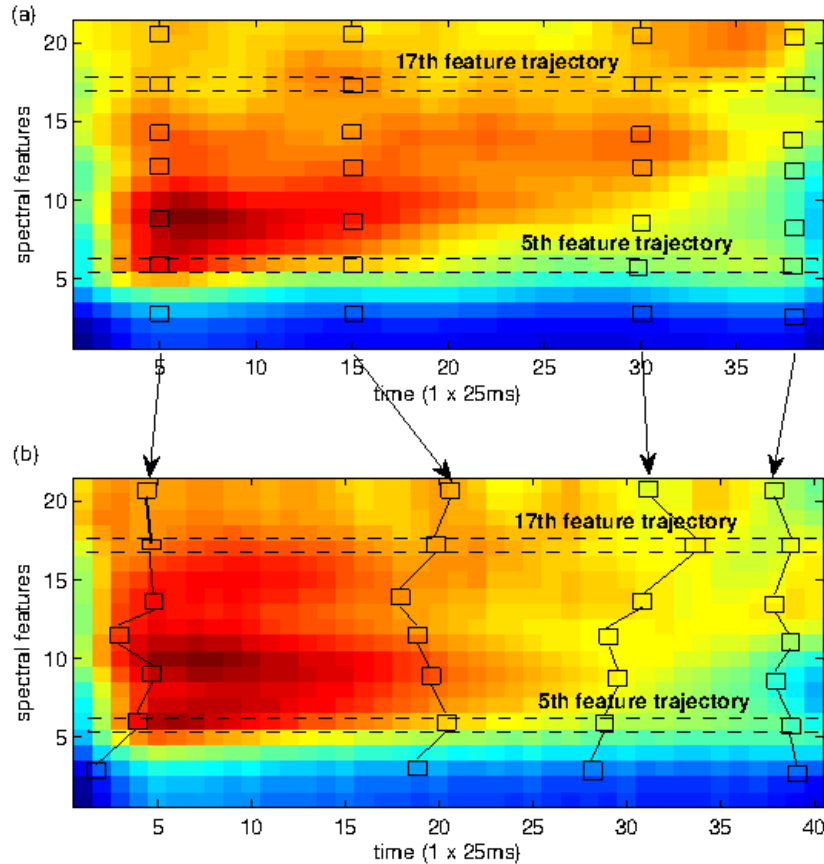
$$X_i^{(l)} = \{x_{i1}^{(l)}, x_{i2}^{(l)}, \dots, x_{in_i}^{(l)}\}, \quad l = 1, 2, \dots, v \quad (4.3)$$

Hence  $X_i^{(l)}$  is a 1-dimensional time series for feature  $l$ . The similarity of two feature vector sequences is calculated by applying classical DTW to each corresponding pair of feature trajectories, and subsequently normalising the sum, as shown in Equation 4.4.

$$FTDTW(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{\beta} \sum_{l=1}^v DTW\{X_i^{(l)}, X_j^{(l)}\} \quad (4.4)$$

where  $\beta = \sqrt{\sum_{l=1}^v \lambda_l^2}$ ,  $\lambda$  is the path length and  $DTW(\cdot)$  indicates non-normalised classical DTW given in Equation 4.2. As illustration, the alignment of the same two acoustic segments shown in Figure 4.1 is repeated with FTDTW. Figure 4.2 (a) identifies seven features from each of the four feature vectors

shown in Figure 4.1 (a). Figure 4.2 (b) demonstrates how each of these seven features aligns with the second speech segment. For the illustrated example, application of Equation 4.4 involves 21 separate alignments, each between corresponding feature trajectories as also indicated in Figure 4.2. The resulting 21 scores are summed and normalised by  $\beta$ . Figure 4.2 illustrates how, in contrast to classical DTW, FTDTW does not require features coincident in time in one segment to align with features in the other segment also coincident in time.



**Figure 4.2:** Alignment of trajectories of 21 spectral features for instances of triphone  $b\text{-aa}+dx$  drawn from TIMIT corpus [1] from both (a) a male speaker mrfk0 and (b) a female speaker fdml0.

### 4.3 Experimental evaluation

The effectiveness of feature trajectory DTW (FTDTW) is evaluated by applying it to agglomerative hierarchical clustering (AHC) of speech segments,



with the Ward method as inter-cluster linkage as described in Section 3.4.6. The objects to be clustered are speech segments corresponding to triphones extracted from the TIMIT corpus as well as the isolated digits taken from the Spoken Arabic Digit Dataset (SADD), both described below. The input to the AHC algorithm is a symmetric  $N \times N$  proximity matrix  $\mathbb{D}$  populated by values computed by  $DTW(\cdot, \cdot)$  or by  $FTDTW(\cdot, \cdot)$  and the output consists of the  $K$  clusters of speech segments. The quality of these clusters is measured by means of the external metrics F-Measure and normalised mutual information (NMI) [61; 70; 54]. The choice of these external measures is based on the literature reported in Chapter 2. Both these measures require the ground truth from both TIMIT and SADD corpora. Since the phonetic alignment is provided in TIMIT and the word alignments in SADD, the ground truth is available.

### 4.3.1 Speech corpora

The acoustic segments used in this chapter are drawn from two different speech corpora. The first is TIMIT [1]. This corpus was prepared by Texas Instruments (TI) in collaboration with Massachusetts Institute of Technology (MIT) in 1993. TIMIT has been extensively used by researchers in the field of automatic speech recognition systems since its release, and has been chosen for our research because it includes time-aligned phonetic transcriptions, meaning that both phonetic labels and their start/end times as determined by phonetic expert are known. To this day, TIMIT remains a unique corpus of continuous speech because of these manual phonetic time-aligned annotations.

TIMIT contains a total of 6300 sentences recorded from 630 speakers 438 of whom are male and the remaining 192 female. Each speaker reads 10 sentences, the first two of which are identical for all speakers in the database. These first two sentences are known as the dialect or SA sentences. To avoid bias, the 2 SA sentences have been excluded in all our experiments. The next 5 sentences are called the SX sentences, and are phonetically compact. The last three sentences are the 3 SI sentences which are phonetically diverse and unique for each speaker.

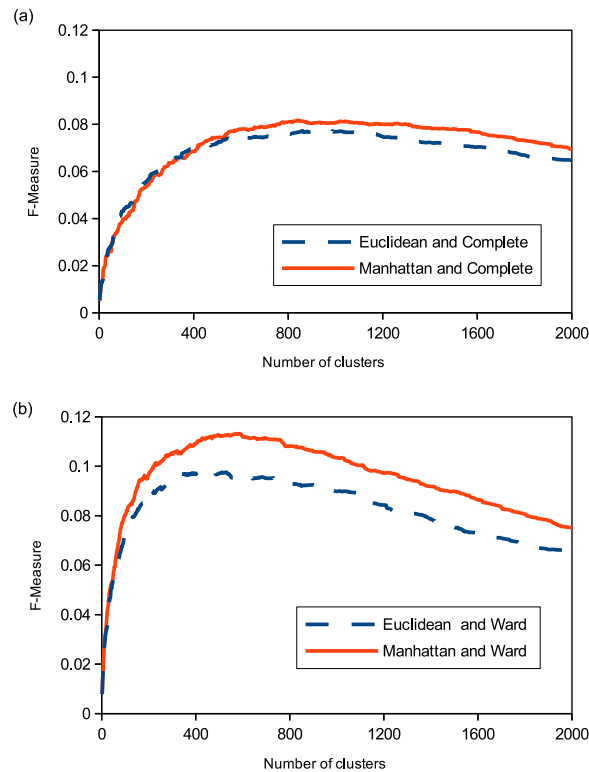
For comparison and confirmation purposes, a second speech corpus, the Spoken Arabic Digit Dataset (SADD) is also used for experimentation [134]. SADD consists of 8800 utterances already parameterised as 13-dimensional MFCCs. The utterances were spoken by 44 male and 44 female Arabic speakers. Each utterance in the SADD corresponds to a single Arabic digit (0 to 9) and was uttered ten times by each speaker.

### 4.3.2 Preliminary optimisation

There are several parameters involved in the implementation of the AHC algorithm. In this section, the choice of these parameters is made based on

common knowledge and experimentation.

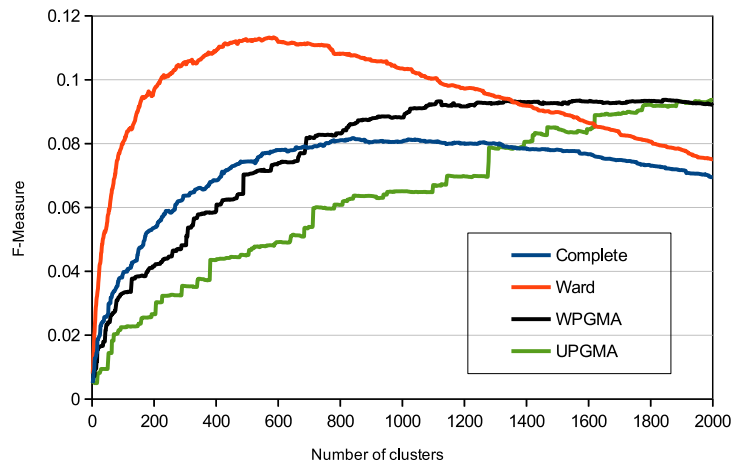
The input to the AHC algorithm is the proximity matrix usually computed using distance measures described in Section 3.2. For time series objects such as speech segments, the DTW algorithm can be used to compute the values populating this matrix. The local distance between vectors used during DTW is commonly Euclidean. An alternative Minkowski distance, as described in Section 3.2, is the Manhattan distance. We initially consider both these metrics when computing local distances  $D_{ij}(\mathbf{x}_{ip}, \mathbf{x}_{jq})$ . In Figure 4.3, AHC is performed using both the complete and Ward linkage methods while DTW scores are calculated using Manhattan and Euclidean distances. Clustering performance is quantified by means of the F-Measure described in Section 2.6.1.



**Figure 4.3:** AHC performance for 8772 TIMIT triphones parameterised as MFCC's in terms of the F-Measure for both Manhattan and Euclidean based DTW when using (a) Complete linkage and (b) Ward linkage.

From Figure 4.3 it is observed that use of the Manhattan distance yields improved performance of the AHC algorithm. Furthermore, substantially better performance is achieved using the Ward linkage method, when compared to complete linkage. It was confirmed that this is true also when considering the other linkage methods described in Chapter 3.

Figure 4.4 presents AHC clustering results that show F-Measure values of 4 linkage methods based on the classical DTW. Results are not shown for single linkage, because these were far inferior to the others considered. UPGMC and WPGMC linkage methods sometimes produce non-monotonic dendrograms as described in Section 3.6 and therefore were not included in the preliminary experiments. These results show that the Ward linkage method performs better than the other three methods considered and hence its choice for the rest of the experiments reported in this dissertation.



**Figure 4.4:** AHC performance for 8772 TIMIT triphones parameterised as MFCC's in terms of the F-Measure for four linkage methods using Manhattan based DTW.

### 4.3.3 Experimental setup

The first set of experiments uses speech segments taken from the TIMIT speech corpus. The segments in question are triphones, which are phones in specific left and right contexts [21]. Only triphones that occur at least 20 times and at most 25 times in the corpus are considered. This leads to an evenly balanced set of 8772 speech segments, which also corresponds approximately to the number of segments in the SADD data used in the second set of experiments.

For comparison and confirmation purposes, a second set of experiments is performed using the Spoken Arabic Digit Dataset (SADD) [134]. The speech segments in this case are isolated spoken digits.

A third set of experiments is based on 10 independent subsets of triphone speech segments drawn from the TIMIT SI and SX utterances, irrespective of occurrence frequency. This better represents the unbalanced distribution of triphones that may be expected in unconstrained speech. Table 4.1 summarises these three datasets.

Dataset	Description
1	8772 TIMIT triphones (evenly balanced).
2	8800 SADD isolated digits (evenly balanced).
3	123182 TIMIT SI and SX triphones divided randomly into 10 subsets (not evenly balanced).

**Table 4.1:** Datasets used for experimental evaluation.

Two feature vector parameterisations popular in the field of speech processing were considered. These are mel frequency cepstral coefficients (MFCCs) [135] and perceptual linear prediction (PLP) coefficients [136]. For the former, log frame energy was appended to the first 12 MFCC's to produce a 13-dimensional feature vector. First and second differentials (velocity and acceleration) were subsequently added to produce the final 39-dimensional MFCC feature vector. For the latter, 13 PLP coefficients were considered, to which velocity and acceleration were added, again resulting in a 39-dimensional feature vector. One such feature vector was extracted for each 10ms frame of speech, where consecutive frames overlapped by 5ms. All TIMIT feature vectors were computed using HTK [137]. The SADD corpus is provided as pre-computed MFCC features, and hence PLP features could not be obtained.

#### 4.3.4 Experimental results

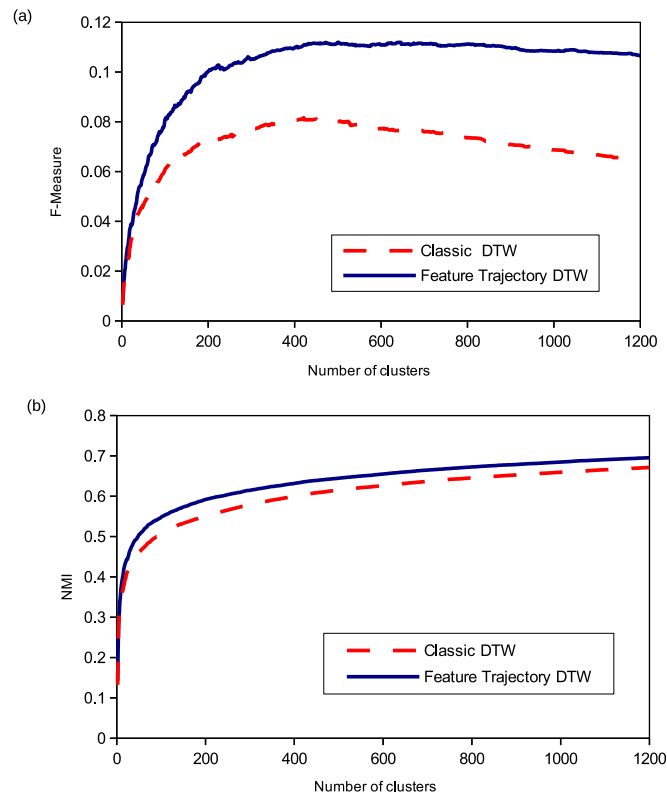
To evaluate the performance benefit of Feature Trajectory DTW (FTDTW) in comparison with classical DTW when used as a similarity measure in AHC, it will be used to cluster the speech segments described in Section 4.3.3. The quality of the automatically-determined clusters will be determined using the F-Measure and in several cases also NMI described in Section 2.6.1.

In a first set of experiments, Dataset 1 (Table 4.1) is clustered. Figure 4.5 reflects the clustering performance in terms of (a) the F-Measure and (b) NMI, when using MFCC features. Both the F-Measure and NMI are plotted as a function of the number of clusters. Note that the F-Measure continues to decline as the number of clusters exceeds 1200.

Figures 4.5(a) and 4.5(b) show that FTDTW improves the performance of classical DTW in this clustering task in terms of both F-Measure and NMI. Especially in terms of F-Measure, this improvement is substantial.

A corresponding set of experiments using PLP features was carried out for Dataset 1, and the results are shown in Figure 4.6. The same trends seen for MFCCs in Figure 4.5 are again observed, with substantial improvements particularly in terms of F-Measure.

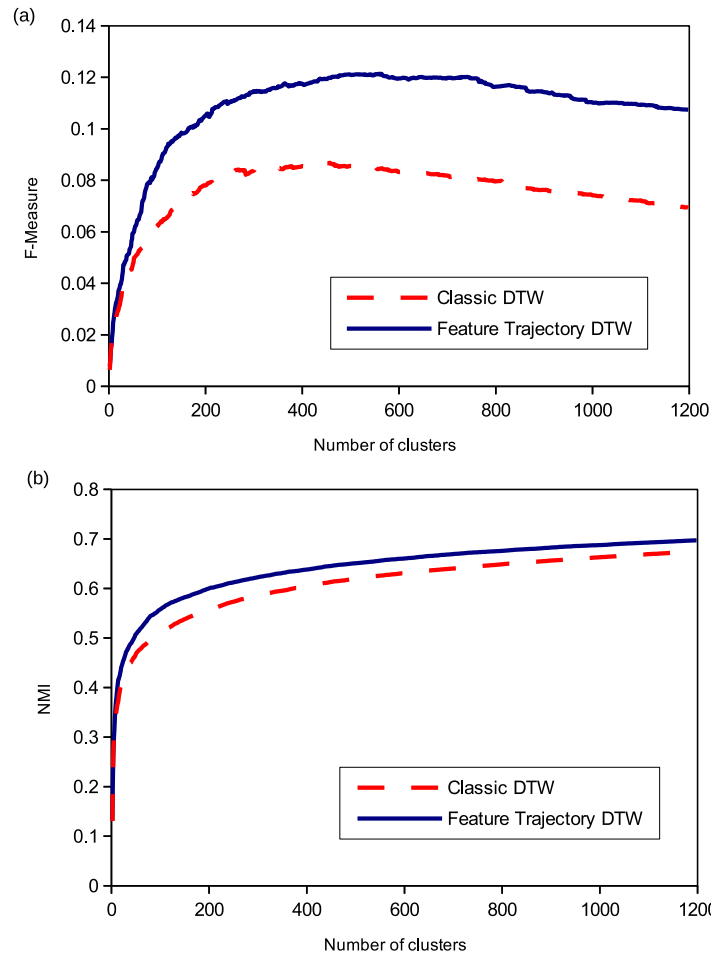
In a second set of experiments, Dataset 2 (Table 4.1) which consists of isolated Arabic digits is clustered. Figure 4.7 indicates the clustering performance, both in terms of F-Measure and NMI for this dataset. Again it is observed that FTDTW outperforms classical DTW in terms of both F-Measure



**Figure 4.5:** Clustering performance for Dataset 1 when using MFCC features in terms of (a) F-Measure and (b) NMI.

and NMI in practically all cases.

In a third and final set of experiments, Dataset 3 (Table 4.1) was considered. The 10 independent subsets of the TIMIT training set each contained between 12034 and 12495 triphone segments. In contrast to the TIMIT experiments for Dataset 1, all triphone tokens were considered irrespective of occurrence frequency. The number of clusters was chosen to be 2394, a figure which corresponds to the number of triphone types with more than 10 occurrences in the data. A single number of clusters, rather than a range as presented in Figures 4.5, 4.6 and 4.7, has been used here in order to make the required computations practical. However, other choices were seen to lead to similar behaviour. Figure 4.8 presents the clustering performance for each of the 10 subsets in terms of F-Measure. It is observed that FTDTW achieves an improvement over classical DTW in all cases. A paired t-test indicated that the improvements are statistically highly significant ( $p < 0.0001$ ). Similar improvements were observed in terms of NMI.

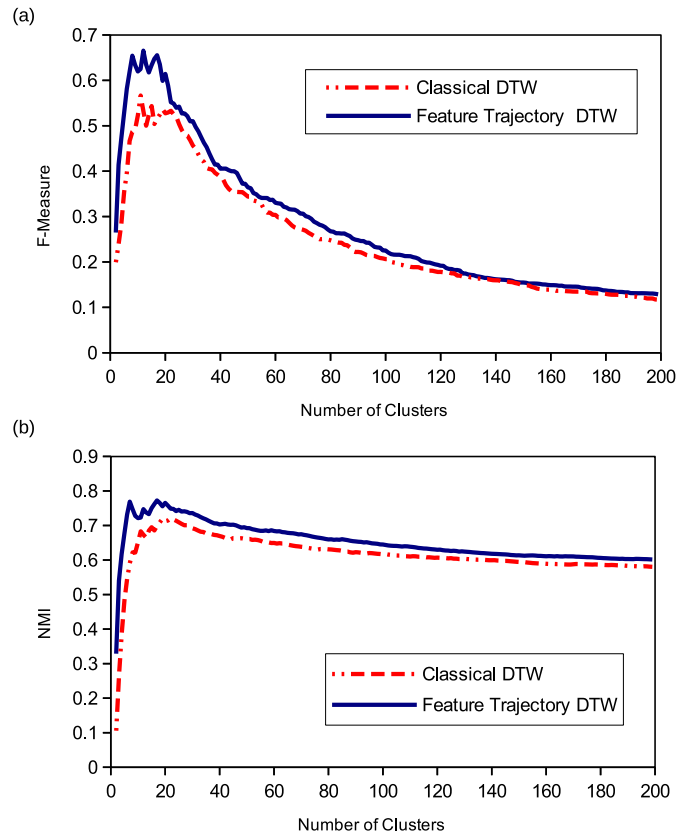


**Figure 4.6:** Clustering performance for Dataset 1 when using PLP features in terms of (a) F-Measure and (b) NMI.

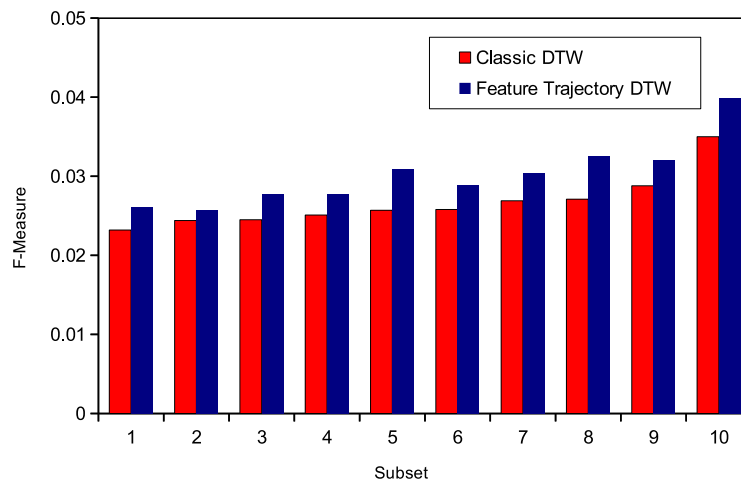
## 4.4 Discussion

Because classical DTW operates on a feature-vector by feature-vector basis, it enforces absolute temporal synchrony between the feature trajectories. In contrast, FTDTW does not impose this synchrony constraint, but aligns feature trajectories independently on a pair-by-pair basis. Since FTDTW is observed to lead to better clusters in these experiments, it is concluded that the strict temporal synchrony imposed by classical DTW is counter-productive in the case of speech signals.

Further, it can be speculated that segments of speech that human listeners would regard as similar also exhibit such differing time-scale warping among the feature trajectories. For the experiments using the MFCC parametrisation of Dataset 1 (Figure 4.6), it is seen that an optimum in terms of F-Measure is reached at 501 and 421 clusters for FTDTW and classical DTW respectively.



**Figure 4.7:** Clustering performance for Dataset 2 in terms of (a) F-Measure and (b) NMI.



**Figure 4.8:** Clustering performance for the 10 independent subsets of Dataset 3 in terms of F-Measure.

The 'true' number of clusters corresponds to the number of triphone types in Dataset 1, which is 404. Hence both DTW formulations over-estimate the number of clusters. A similar tendency is seen for the PLP parametrisations of the same dataset, where the F-Measure peaks at 439 and 559 clusters for classical DTW and FTDTW respectively, and also for Dataset 2 in Figure 4.7.

Although the ground truth is known, the class definitions of triphones for Datasets 1 and 3, and isolated digits for Dataset 2 may be called into question. In particular, although all triphones correspond to acoustic segments from the same phone within the same left and right contexts, there are many other possible sources of systematic variability, such as the accent of the speaker. Hence it may be reasonable to expect that a larger number of clusters is needed to optimally model the data. To determine whether this is the case, the clusters should be used to determine acoustic models for an ASR system as will be demonstrated in Chapter 6. Then the performance of varying clusterings of the data can be compared by comparing the performance of the resulting ASR systems.

## 4.5 Summary and conclusion

In this chapter speech segments extracted from the TIMIT and SADD corpora have been clustered by application of AHC and DTW. Important parameters considered for the optimisation of agglomerative hierarchical clustering have been described. Experiments motivating the choice of the Manhattan distance to measure between-object similarity, as well as the Ward linkage method have been presented. A modified DTW algorithm termed "feature trajectory DTW" (FTDTW) was proposed and shown experimentally to improve clustering for all datasets considered for both MFCC and PLP parameterisations. Hence we conclude that FTDTW can be more effective than classical DTW as a similarity measure for clustering of speech signals. The following chapter will report on AHC when using classical DTW as well as FTDTW, in both cases using Manhattan distances and the Ward linkage method.



## Chapter 5

# Multi-Stage Agglomerative Hierarchical Clustering

### 5.1 Introduction

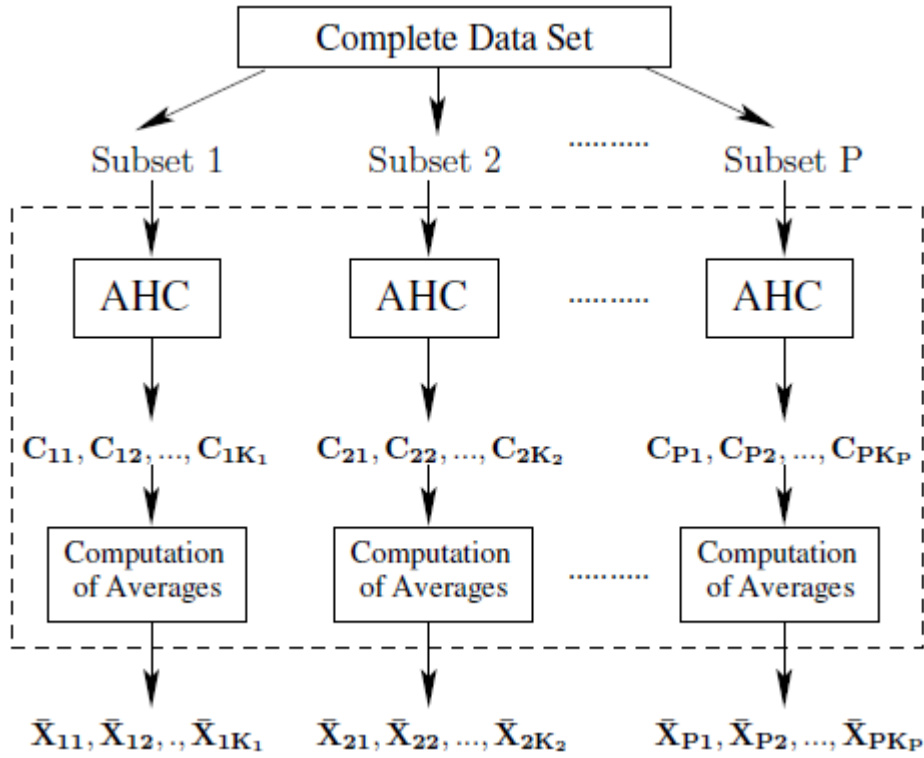
In this chapter, we propose an iterative multi-stage agglomerative hierarchical clustering (MAHC) algorithm which is aimed at the clustering of large datasets of speech segments. This algorithm solves the  $O(N^2)$  complexity of the AHC algorithm reported in Chapter 3. Experimental evaluations reported are based on datasets of varying size. Smaller datasets are used to investigate if the proposed MAHC can approximate the performance of the exact AHC algorithm. Subsequently a method to automatically determine the number of clusters for large data is proposed. MAHC does not require the number of clusters to be specified in advance, and is shown to be comparable in performance to parallel spectral clustering (PSC) which has also been developed specifically for the processing of large datasets. The classical formulation of DTW described in Section 4.2.1 is used for all experiments in this chapter.

### 5.2 The MAHC algorithm

Multi-stage agglomerative hierarchical clustering (MAHC) is based on an iterative divide-and-conquer strategy. The data is first split into independent subsets, each of which is clustered separately. This reduces the storage required for sequential implementations, and allows concurrent computation on parallel computing hardware. The resultant clusters are merged and subsequently re-divided into subsets, which are passed to the following iteration. The algorithm requires only the pairwise distances between objects to be known, and hence makes the clustering of substantial speech databases feasible.

Our proposed method is a two stage iterative process. The first stage divides the complete dataset into  $P$  subsets and applies AHC to each subset. These  $P$  clustering operations can occur sequentially or concurrently in paral-

lel. Figure 5.1 illustrates this first stage.



**Figure 5.1:** The first stage of MAHC algorithm.

The value of  $P$  is heuristically determined by assessing the available memory of the computational resources based on the size of the dataset. In the first stage, each data subset is processed as follows:

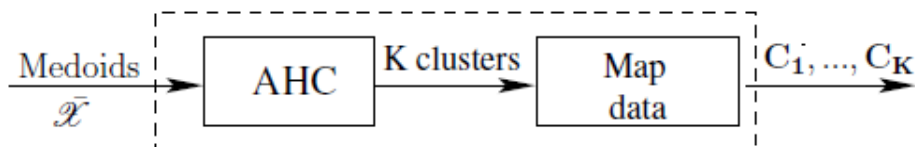
1. AHC is applied to subset  $p$  and generates a set,  $\mathcal{C}_p$ , of  $K_p$  clusters where  $p = 1, 2, \dots, P$  and  $\mathcal{C}_p = \{C_{p1}, C_{p2}, \dots, C_{pK_p}\}$ .
2. An average point,  $\bar{X}_p$ , is determined for each cluster generated from subset  $p$ , where  $\bar{X}_p = \{\bar{X}_{p1}, \bar{X}_{p2}, \dots, \bar{X}_{pK_p}\}$  and  $\bar{X}_{pk} \in C_{pk}$ .

The average,  $\bar{X}_p$ , can be a mean, median, mode, medoid or any other measure of statistical locality which represents all objects in a cluster. From the first stage architecture we can approximate the computational complexity as  $P \times O(N^2/P^2)$ . This is an improvement by a factor  $P$  over the standard AHC algorithm whose complexity is  $O(N^2)$ .

The second stage clusters the averages. At the beginning of the second stage, the total number of average objects that must be clustered is  $S = K_1 + K_2 + \dots + K_P$ . We denote the set of all averages passed to the second

stage as  $\bar{\mathcal{X}} = \{\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \dots, \bar{\mathbf{X}}_S\}$ . We then apply AHC to  $\bar{\mathcal{X}}$  and determine  $K$  clusters of the averages. The purpose of this step is to merge similar clusters resulting from the first stage. Since the data were divided randomly into subsets at the top of Figure 5.1, the  $P$  separate clustering processes performed in parallel may result in some clusters that are similar.

All elements of the complete dataset are then mapped to their corresponding averages, to obtain the final object clusters,  $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K$ , which are the output of the second stage. The second stage process is illustrated in Figure 5.2.



**Figure 5.2:** The second stage of MAHC algorithm.

The final step in MAHC algorithm is the regeneration of the  $P$  subsets shown at the top of Figure 5.1, thereby rendering the MAHC algorithm iterative. This is done by setting  $P = K$  and mapping the data to each of the *new*  $P$  subsets according to the clusters  $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K$  obtained after the first iteration of stages 1 and 2. The motivation for this step is that, by grouping similar clusters in stage 2 and using those to redefine the subsets from which stage 1 proceeds, each independent clustering operation constituting stage 1 will process data that are more self-similar and that are different from the data processed by the other  $P$  clustering operations. If this succeeds, the division into independent clustering operations in stage 1 becomes an increasingly appropriate strategy. During the last iteration of MAHC, stage 2 produces the final  $K$  clusters. The complete process is shown in Figure 5.3.

The parameters of the proposed MAHC algorithm are:

1. The number of subsets,  $P$
2. The number of clusters each subset is divided into during stage 1,  $K_i$
3. The final number of clusters,  $K$
4. The number of iterations of the MAHC algorithm.

The effect of these parameters will be investigated experimentally in the following sections. In particular, it will be shown that the number of clusters  $K$  can be estimated automatically.

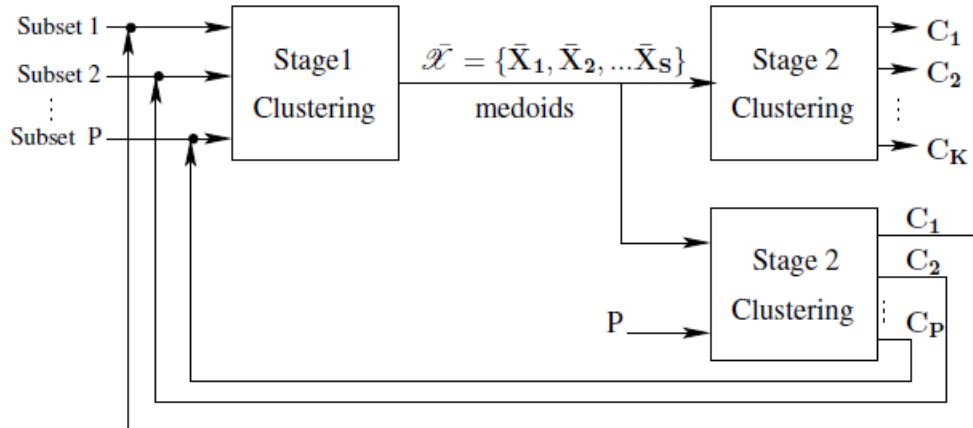


Figure 5.3: The complete MAHC algorithm.

### 5.3 Clustering acoustic segments using MAHC

The set of acoustic segments  $\mathcal{X}$  is divided into  $P$  subsets and a proximity matrix for each subset is calculated using the DTW algorithm described in Chapter 4. The first stage of MAHC is applied to the subsets as described in Section 5.2. The average depicted in Figure 5.1 in this case is a medoid. A medoid is the cluster member,  $\bar{\mathbf{X}}_p$ , which is, on average, closest to all other members, and is computed as follows:

$$\bar{\mathbf{X}}_p = \arg \min_p \sum_{q=1}^{K_p} d(\mathbf{X}_p, \mathbf{X}_q) \quad (5.1)$$

In our implementation, medoids are used as a representation of each cluster because the non-uniform multidimensional time series data objects in a cluster do not lie in the metric space where centroids can be easily determined. We consider the DTW distance between the medoids of two clusters to be a measure of inter-cluster similarity that can be used as input proximity matrix for AHC in the second stage. Finally all acoustic segments are mapped to their corresponding medoids to obtain the final set of clusters.

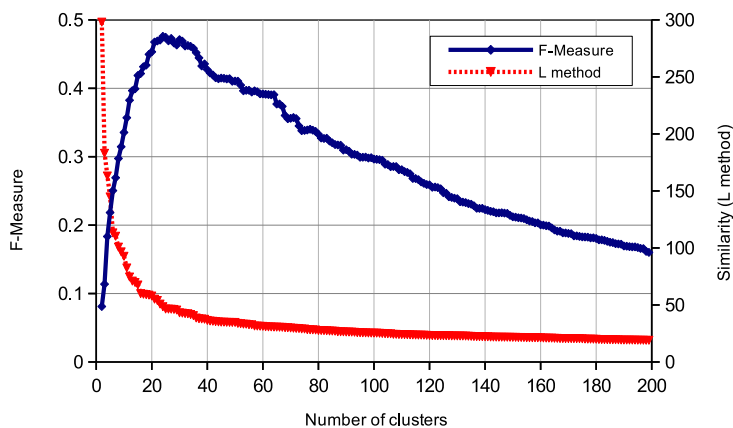
#### 5.3.1 Cluster validity for MAHC

The performance of MAHC is evaluated by applying it to the TIMIT speech corpus. In addition to the F-Measure used for cluster evaluation in Chapter 4, we choose an internal metric called the L method [81] as additional measure to automatically determine the number of clusters. The L method is described in Section 2.6.2 and it is one of the suitable evaluation methods for clustering datasets in which ground truth is not available. This will fulfil our aim of clustering speech segments for which no labels are available. The L method is selected for internal validation because it yields reasonable results in our

experimental evaluation, it is computationally cheap, and it has received considerable attention by the research community [82]. The F-Measure scores provide a benchmark against which the L method can be measured. We maintain the use of the F-Measure because it is widely used for the evaluation of clustering and classification systems [30]. Chapter 7 will provide evaluation of MAHC in an automatic speech recognition system which is the targeted application for the clustering process.

### 5.3.2 Determining a threshold for the dendrogram

One way to determine the best cutoff for a dendrogram and hence the number of clusters is to calculate the F-Measure at all possible threshold values and then determine the number of clusters at the peak. Alternatively, the number of clusters can be estimated by locating the knee of the similarity measure graph (L method). Both are shown in Figure 5.4, which is a result of a small experiment in which 754 acoustic segments were clustered using the classical AHC method. These acoustic segments are triphones from the TIMIT chosen because their duration is roughly the same to avoid bias. Mel-frequency cepstral coefficients are used to represent each segment for clustering. The true number of classes in this case is 29 and the F-Measure peak occurs at 24 clusters.



**Figure 5.4:** AHC results of a small experiment with 29 true clusters. The peak in the F-Measure occurs at 24 clusters, while the knee of the L method is found at 22 clusters.

This experiment demonstrates that the F-Measure increases with the number of clusters, reaches a peak, and then begins to decline as the number of clusters increases. This eventual decline is due to the rise in the number of single occupancy clusters. When applying the L method to the same data, the

knee was located at 22 clusters. We observe that both the F-Measure and the L method produce a comparable number of clusters.

## 5.4 Experimental evaluation

### 5.4.1 Data

All experiments use acoustic segments taken from the TIMIT speech corpus described in Section 4.3.1. The TIMIT corpus is chosen because it includes time-aligned phonetic transcriptions meaning that both phonetic labels and their start/end times are provided. We will consider triphones [36], which are phones in specific left and right contexts, as our desired clusters. We used a maximum of 42 base phones in our experiments.

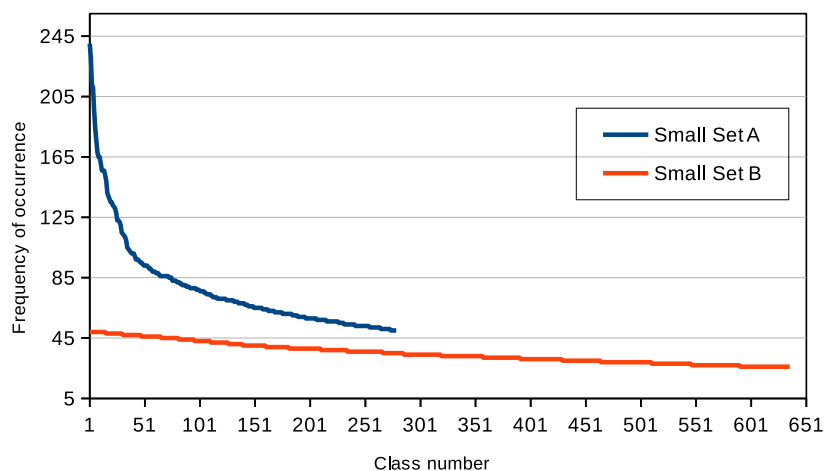
From the TIMIT data we have compiled 4 datasets, varying in size. Table 5.1 shows the number of segments (objects) in each dataset, as well as the true number of classes, the range of class cardinality and the total number of feature vectors.

Dataset	Segments (N)	Classes (L)	Range (R)	Points (V)	Entries (M)
Small Set A	17 611	280	50–373	274 677	$0.16 \times 10^9$
Small Set B	17 640	636	26–49	301 026	$0.16 \times 10^9$
Medium Set	54 787	1 387	20–373	910 189	$1.5 \times 10^9$
Large Set	123 182	19 223	1–373	2 193 793	$7.6 \times 10^9$

**Table 5.1:** Composition of experimental data.  $N$  indicates the total number of segments,  $L$  the total number of classes (unique number of triphones),  $R$  the frequency of occurrence of each triphone,  $V$  the total number of feature vectors in  $\mathbb{R}^{39}$  and  $M = N(N - 1)/2$  the number of similarities which must be computed for straightforward application of AHC.

Small Set A and Small Set B differ in their class distribution as depicted in Figure 5.5. Small Set A is more skewed compared to Small Set B. This means that in Small Set A, some classes have many more members than others. The Medium Set and the Large Set are skewed in the same fashion as the Small Set A, since this is the type of distribution one may expect in unconstrained speech.

During data preparation, each acoustic segment is represented as a series of 39 dimensional feature vectors consisting of 12 Mel frequency cepstral coefficients (MFCCs), log frame energy, and their first and second differentials. The MFCC's were chosen on the basis of their well-established popularity in speech processing systems [21]. Feature vectors are extracted from data frames



**Figure 5.5:** Distribution of the number of segments per class for the two independent Set A and Set B.

that are 10ms in length, and consecutive frames overlap by 5ms (50%). The MFCC's were computed using HTK [138].

#### 5.4.2 AHC baseline

Baseline results were obtained by applying classical AHC to the small and medium datasets described in Table 5.1. In each case the dendrogram was cut so as to optimise the F-Measure as shown in Table 5.2. Subsequently the L method was applied to determine the dendrogram thresholds, and these results are reflected in Table 5.3. Even when the number of clusters is obtained using the L method, we can still apply the F-Measure for comparison. In the case of the large dataset, the excessive size of the similarity matrix did not allow the application of classical AHC. For this case, results will only be shown for the MAHC method.

Dataset	Optimal no. of clusters	AHC:F-Measure	PSC: F-Measure
Small Set A	144	0.1104	0.1198
Small Set B	577	0.0662	0.0655
Medium Set	717	0.0476	0.0265

**Table 5.2:** Baseline results when the cutoff is determined via the F-Measure.

Dataset	Optimal no. of clusters	AHC: F-Measure	PSC: F-Measure
Small Set A	162	0.1074	0.1290
Small Set B	163	0.0529	0.0546
Medium Set	503	0.0456	0.0317

**Table 5.3:** Baseline results when the cutoff is determined via the L method and the output is evaluated with the F-Measure.

### 5.4.3 Parallel spectral clustering benchmark

In order to benchmark our results, we have also applied parallel spectral clustering (PSC) as proposed by Chen *et al* [139] and described in Section 2.4.1 to our datasets. Spectral clustering can also be applied in situations in which only the similarities between objects are known. Furthermore, in contrast to other variants of spectral clustering, PSC can be applied to large datasets. PSC does however require the number of clusters  $K$  to be specified. In benchmark comparisons we will therefore always employ the number of clusters used in the corresponding MAHC experiment. We employ 20 nearest neighbours for small and medium sets, and 100 for the Large Set, as suggested by the experiments in [139]. Additionally, we also show how parallel spectral clustering (PSC) performs at the baseline conditions in Tables 5.2 and 5.3. We observe that PSC delivers better performance than classical AHC for Small Set A, while the two approaches exhibit similar performances for Small Set B. AHC offers improved performance on the Medium Set.

### 5.4.4 MAHC of the small datasets

Since the clustering experiments are computationally demanding, we begin experimentation with the small sets (A and B). Subsequently we extend the investigation to the larger datasets. The first experiments applied classical AHC, to the 17,611 segments of Small Set A and to the 17,640 segments of Small Set B. Subsequently, we split these datasets into 2, 4 and 6 subsets and in each case performed 10 iterations of the MAHC algorithm. In each case the number of clusters was chosen by maximising the F-Measure both after stage 1 and stage 2. This number of clusters was also used as input to parallel spectral clustering (PSC) to provide a benchmark.

The results are shown in Figure 5.6 for each successive iteration both in terms of F-Measure and the optimal number of clusters.

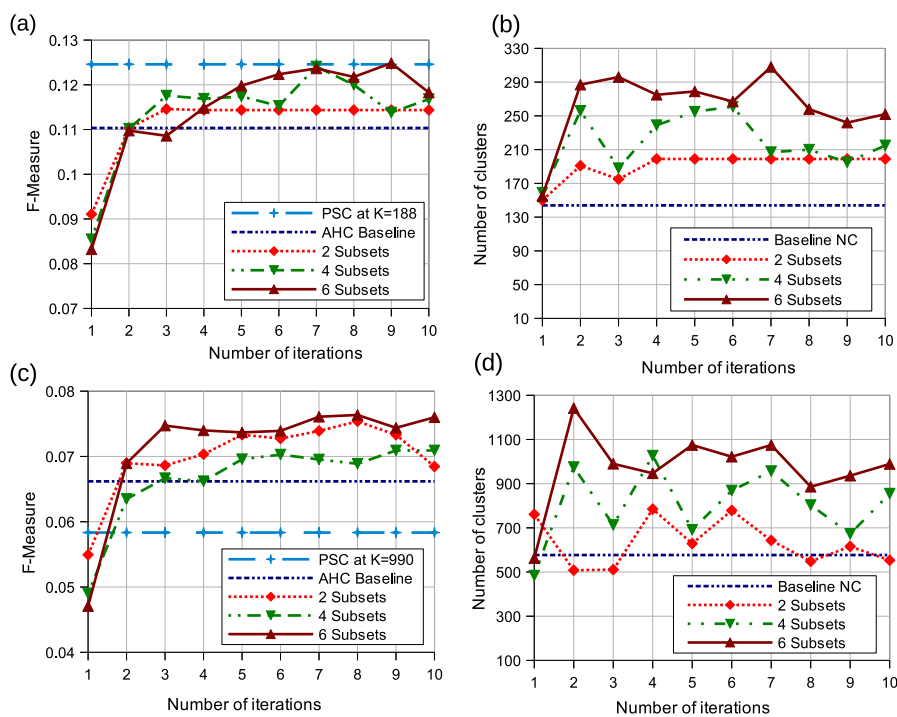
Figure 5.6(a) shows that, for Small Set A, the F-Measure for MAHC generally increases with each iteration, and that it exceeds the performance achieved by the AHC baseline at the third iteration. Figure 5.6(b) shows the number of clusters produced after each iteration of the MAHC algorithm. As mentioned earlier, these clusters are obtained by optimising the F-Measure both



at stage 1 and at stage 2. As a baseline, the number of clusters obtained when optimising the F-Measure for the classical AHC method is also shown (see Table 5.2). Figure 5.6(b) shows that the number of clusters obtained by application of MAHC is larger than that obtained when using AHC, and that it varies somewhat from iteration to iteration. The performance of PSC at  $K = 188$  in terms of F-Measure is also shown in Figure 5.6(a) and generally outperforms MAHC. Details on how the value of  $K$  was chosen are discussed in later sections.

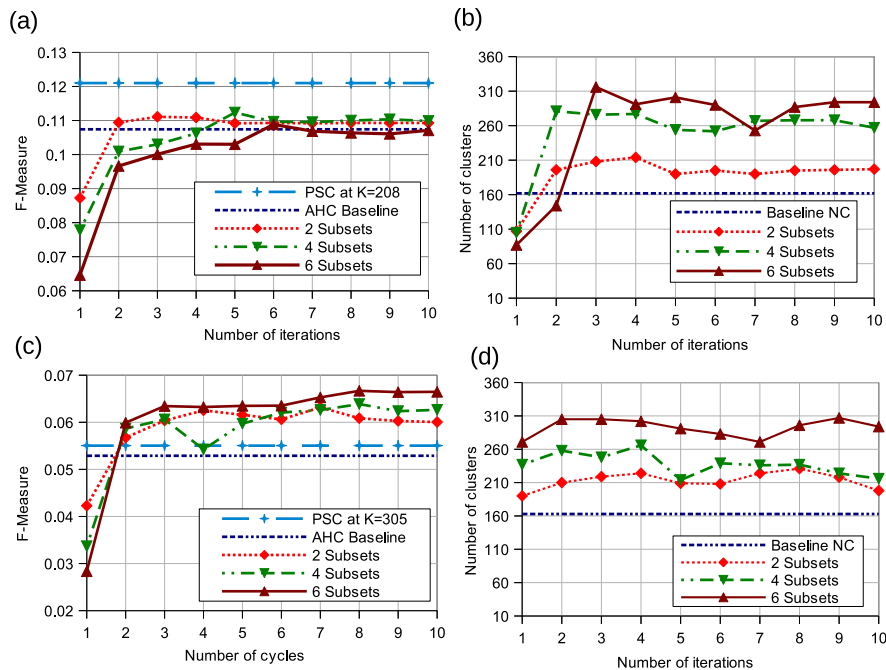
The same trends are observed for Small Set B in Figures 5.6(c) and 5.6(d). However, Figure 5.6(d) shows that the number of clusters determined by MAHC fluctuates more widely. This may be owing to the number of classes relative to the distribution of Small Set B data. Despite this fluctuation, the quality of the clusters, in terms of F-Measure, is consistently better than the baseline from the third iteration onwards. Furthermore, the performance of PSC with  $K = 990$  in Figure 5.6(c) for Small Set B is surpassed by MAHC from the second iteration onwards.

The experiments shown in Figure 5.6 were repeated, this time using the L method to determine the threshold for the dendrogram in stage 1. Thresholds in stage 2 continued to be chosen by optimising the F-Measure. The results



**Figure 5.6:** Performance of MAHC and PSC for the small sets in terms of F-Measure, using F-Measure to determine thresholds in stage 1. (a) F-Measure for Small Set A (b) MAHC optimal number of clusters for Small Set A (c) F-Measure for Small Set B (d) MAHC optimal number of clusters for Small Set B.

are presented in Figure 5.7.

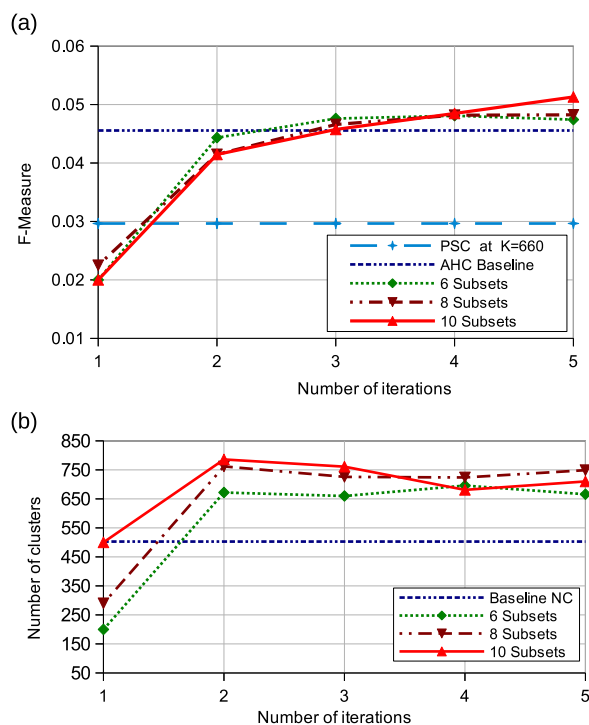


**Figure 5.7:** Performance of MAHC for the small sets in terms of F-Measure, using the L method to determine thresholds in stage 1. (a) MAHC and PSC F-Measure for Small Set A (b) MAHC optimal number of clusters for Small Set A (c) MAHC and PSC F-Measure for Small Set B (d) MAHC optimal number of clusters for Small Set B.

From Figure 5.7(a) we observe that the MAHC surpasses the baseline AHC for Small Set A at the fourth iteration except in the case of 6 subsets. The F-Measure achieved by PSC for Small Set A at  $K = 208$  in Figure 5.7(a) remains the best overall. Figure 5.7(c) mirrors the performance trends observed in Figure 5.6(c) for Small Set B in which the MAHC performance is equal to or better than the AHC baseline from the second or third iteration. Furthermore, MAHC exhibits better performance than PSC after the first iteration. Figures 5.7(b) and 5.7(d) also indicate a much more stable number of clusters than observed in Figure 5.6. These results continue to show that the number of clusters generally increases with the number of subsets. This may again be due to the the distribution of the data. In general we observe from both Figure 5.6 and 5.7 that, with small datasets, MAHC matches or even surpasses the performance of AHC after 3 or 4 iterations. As shown in Figure 5.6, MAHC is able to improve on PSC in one experiment (Small Set B), but not in the other (Small Set A).

### 5.4.5 MAHC of the medium dataset

The Medium Set, which is approximately three times larger than the small sets, is still small enough for classical AHC to be applied on available computing equipment. We used this set to verify and support our findings with the small datasets. The L method was used to determine the dendrogram cutoff in stage 1 for computational reasons. However we continued to use the F-Measure in stage 2 of the MAHC as way of objectively evaluating cluster quality. We also show the performance of parallel spectral clustering (PSC) for the Medium Set at the same value of  $K$ . A set of experiments, similar to those reported in Figure 5.7, was performed for the medium set and the results are displayed in Figure 5.8.



**Figure 5.8:** Performances for the Medium Set. (a) MAHC and PSC F-Measure (b) MAHC optimal number of clusters (NC).

These results show that the performance of the MAHC method closely approximates that of the AHC baseline from the third iteration onwards. This is consistent with our findings for the two smaller sets. At the third iteration, MAHC produces 660 clusters, and we therefore use this value for the PSC benchmark. For the Medium Set, we observe in Figure 5.8(a) that MAHC improves on the performance of PSC in terms of the F-Measure.

Figure 5.8(b) shows that, as observed with Small Set A and Small Set B, the number of clusters (NC) produced by MAHC exceeds that produced by

classical AHC. The latter finds  $K = 503$  while the former suggests between 660 and 786 clusters after 2 iterations. However, in contrast to the smaller sets, the number of clusters determined by MAHC is fairly stable. This may be due to the higher cluster occupancy which is in turn due to the larger volume of data.

Another important observation made after the experiments with the small and medium sets is that the number of clusters produced by stage 1 coincides closely with the number of clusters in stage 2 after a second iteration. Using the notation introduced in Figure 5.1, we can express this observation as:

$$\sum_{i=1}^P K_i \approx K \quad (5.2)$$

where  $K$  is the number of clusters (NC) produced by MAHC. This observation is further substantiated in Tables 5.4, 5.5 and 5.6.

	Number of clusters (NC) per subset in each iteration				
Subset(i)	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1	46	81	67	85	113
2	47	82	62	57	48
3	54	40	59	54	30
4	44	14	32	25	40
5	53	49	71	34	26
6	50	56	25	36	44
$\sum(\mathbf{K}_i)$	294	322	316	291	301
$\mathbf{K}$	87	144	316	291	301

**Table 5.4:** Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Small Set A using the L method.

Here we have verified that the relation in Equation 5.2 holds for all the results shown in Figures 5.6, 5.7 and 5.8. Although this observation should be thoroughly investigated on other datasets, it indicates that the number of clusters at each level of the MAHC algorithm can be chosen in an unsupervised manner; using the L method in stage 1 and Equation 5.2 in stage 2. Our benchmark PSC results are consequently obtained using this observation to determine the required value of  $K$ .

Subset(i)	Number of clusters (NC) per subset in each iteration				
	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1	44	45	91	55	61
2	51	37	36	35	52
3	51	67	27	67	36
4	39	75	49	57	69
5	48	50	48	58	44
6	41	33	54	30	29
$\sum(\mathbf{K}_i)$	274	305	305	302	291
$\mathbf{K}$	271	305	305	302	291

**Table 5.5:** Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Small Set B using the L method

Subset(i)	Number of clusters (NC) per subset per iteration				
	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1	110	189	87	171	134
2	104	144	165	142	137
3	96	121	131	137	84
4	104	48	98	88	131
5	100	99	105	79	83
6	108	71	74	79	97
$\sum(\mathbf{K}_i)$	622	672	660	696	666
$\mathbf{K}$	200	672	660	696	666

**Table 5.6:** Relation between experimental number of clusters ( $K$ ) and the sum of NC's from each subset of Medium Set using the L method.

#### 5.4.6 MAHC of the large dataset

From Table 5.1, we see that the application of classical AHC to the large dataset would require the computation and storage of  $7.6 \times 10^9$  similarities. This was infeasible both from a storage and computational point of view on the computing hardware available. We apply the MAHC algorithm to this dataset, splitting it into 10 subsets. As before, the L method is used to determine the number of clusters in stage 1, while the number of clusters in stage 2 is chosen using Equation 5.2. PSC is again provided as a benchmark with  $K = 1427$ , which corresponds to the number of clusters produced by MAHC at the third iteration.

Table 5.7 summarises the performance of MAHC with PSC as a baseline for the four datasets considered. Since spectral clustering commonly requires the number of clusters  $K$  to be known in advance, and since we have found

that MAHC approximates classical AHC performance after the third iteration, the PSC benchmark in Table 5.7 uses values of  $K$  from the third iteration.

Dataset	No. of Clusters	MAHC: F-Measure	PSC: F-Measure
Small Set A	208	0.1109	0.1210
Small Set B	305	0.06344	0.05504
Medium Set	660	0.04761	0.02966
Large Set	1427	0.01663	0.01039

**Table 5.7:** F-Measure performances of the L method based MAHC and the PSC algorithm.

From these benchmark results we observe that, for Small Set A, PSC delivers better performance than MAHC. For Small Set B, the Medium Set and the Large Set, the MAHC reflects better performance. It should be borne in mind that, for a particular dataset, spectral clustering requires the correct user-determined value of  $K$ . As an example for the large dataset we used  $K = 1427$  for PSC which gave the F-Measure value of 0.01039 shown in Table 5.7.

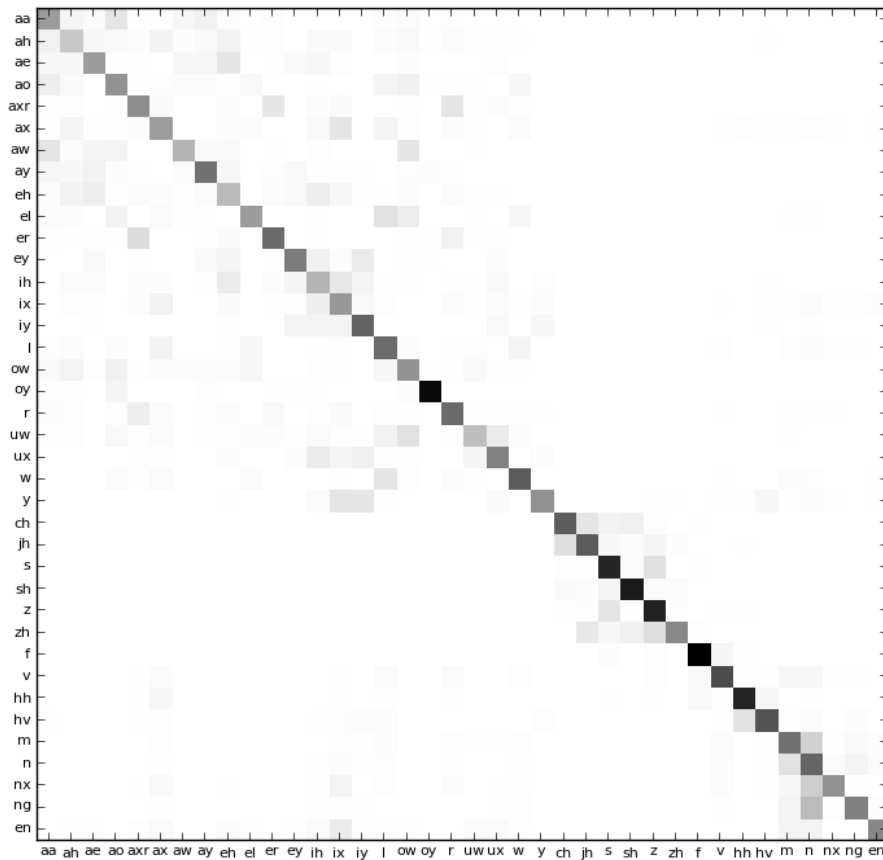
Since a comparison of the results in Table 5.8 with the AHC baseline is not feasible, we make use of a confusion matrix to visualise the similarities of the clustered acoustic segments. Figure 5.9 shows how often the derived clusters coincide with the known phone labels present in the TIMIT reference transcriptions.

No. of iterations	F-Measure	No. of clusters
1	0.007113	1264
2	0.01508	1450
3	0.01663	1427
4	0.01713	1395
5	0.01822	1423

**Table 5.8:** Performance of the proposed method on the Large Set.

To obtain the confusion matrix, we considered only the centre phone of the triphone, that is, the triphone without its context. Clusters were considered to be associated with a phone when that phone was the dominant member of the cluster. Four phones *'em'*, *'eng'*, *'h'* and *'uh'* were not dominant in any of the 1423 clusters. For this reason the matrix has only 38 dimensions.

Figure 5.9 clearly shows a dominant diagonal, indicating good correspondence between the clusters and the known phonetic labels. This is an indication that MAHC successfully determined groups of audio segments that show a high correspondence with the ground truth phonetic labels. Where phones



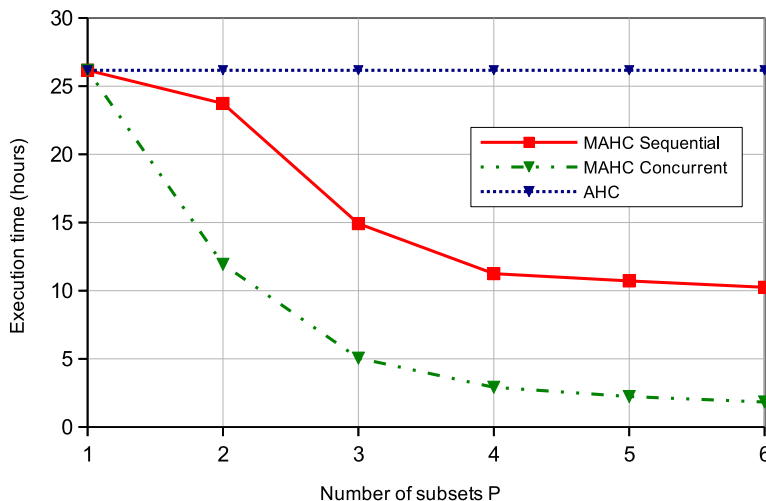
**Figure 5.9:** Confusion matrix of base phones of the large TIMIT dataset. The degree of shading indicates the strength of the correspondence.

are confused, they are usually between similar sounds such as *'n'* and *'ng'* or between *'m'* and *'n'*. These results indicate that the use of the L method together with the empirically observed relationship expressed by Equation 5.2 approximately allow a completely unsupervised application of MAHC for large datasets.

### 5.4.7 Computational efficiency

Our focus has been on the reduction of the storage complexity of AHC in order to make its application to large datasets feasible, and on the performance implications of the proposed approximations. However, we have performed a small test using Small Set B to give an indication of the impact of the proposed method on execution time. We measured the execution time of the classical AHC process, which entails the generation of a full triangular similarity matrix, the Ward linkage computation, the creation of a dendrogram data structure and the L method computation for determining the cutoff. We also measured the execution time of one iteration of the MAHC algorithm, both

when executed on a single processor and when executed concurrently on  $P$  processors, where  $P$  is the number of subsets used in stage 1 of the algorithm. In the former case, each of the  $P$  clustering steps constituting stage 1 of the algorithm are executed sequentially. The results are shown in Figure 5.10.



**Figure 5.10:** Influence of the number of subsets used by MAHC on the execution time. Classical AHC is included as a baseline.

From Figure 5.10 we observe that the execution time of each iteration of the MAHC algorithm is less than that of the classical AHC even when run on a single processor. By taking advantage of parallel computation, the execution time is further reduced to just 2 hours when data is split into 6 subsets. We observe the indication that the MAHC computational complexity at  $P \times O(N^2/P^2)$  per iteration practically leads to a reduction in the execution time when compared with the classical AHC algorithm. Generally, several iterations of MAHC will be needed for a good clustering result to be achieved. Despite this, an overall reduction in execution time might still be achieved.

## 5.5 Summary and conclusion

This chapter has proposed a multi-stage agglomerative hierarchical clustering (MAHC) algorithm that is better suited to large datasets than classical agglomerative hierarchical clustering (AHC). The algorithm is based on a split of the dataset into a number of subsets that are clustered separately using AHC. Subsequently, the results of these separate clustering operations are merged and then used to obtain a new split of the dataset into independent subsets. Experiments show that the iteration of these steps leads to a convergence in the clusters and clustering performance within a small number of iterations. When using speech segments from the TIMIT corpus, experiments show that



the performance of MAHC matches and often surpasses that of AHC. Furthermore, MAHC was also demonstrated to offer some improvement over parallel spectral clustering under matching experimental conditions, and that this improvement was greatest for the largest dataset. Due to its iterative nature, it is possible that some subsets of the MAHC algorithm will grow excessively. Such dominant subsets would diminish the improved space and computational complexity offered by MAHC. Chapter 6 will address this and offer an improvement to MAHC to guarantee maximum memory usage.

## Chapter 6

# Cluster Size Management in MAHC of Acoustic Speech Segments

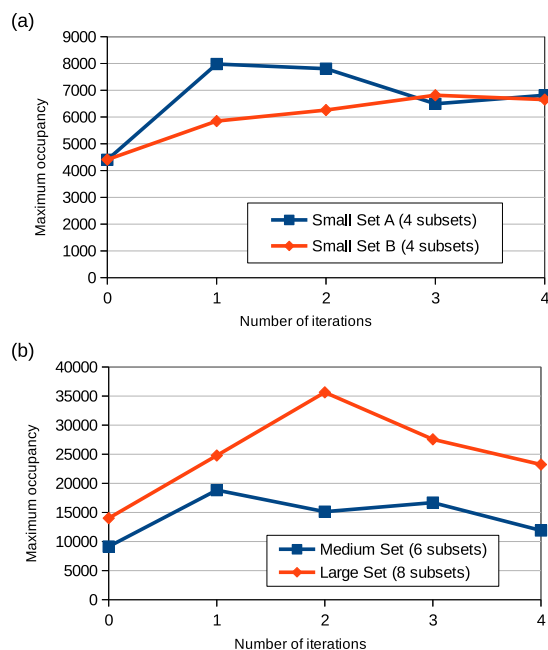
### 6.1 Introduction

Agglomerative hierarchical clustering (AHC) is characterised by  $O(N^2)$  space and time complexity, making it infeasible for partitioning large sets. This problem has been addressed in Chapter 5 by the implementation of Multi-stage hierarchical clustering (MAHC) based on the iterative re-clustering of independent subsets of the larger dataset. We have observed, however, that in some cases individual clusters grow during the iterations and eventually dominate, thereby exceeding available memory and strongly slowing the clustering process. This chapter proposes refinement of MAHC that can be used to remedy this. By monitoring the occupancy of the clusters in each subset and iteratively subdividing them when a threshold size is exceeded, maximum memory usage can be guaranteed. The experiments show that the proposed method leads to no loss in performance in terms of F-Measure while guaranteeing that a threshold space complexity is not breached. Furthermore, feature trajectory dynamic time warping (FTDTW) introduced in Chapter 4 is applied to the clustering experiments and it is found that FTDTW increases performance of DTW.

### 6.2 Limitations of MAHC

We showed in Chapter 5 that MAHC exhibits  $O(\frac{N^2}{P^2})$  space and computational complexity for each subset. However, due to the iterative nature of the algorithm, it is possible that one or more of the  $P$  subsets grows to contain substantially more than  $\frac{N}{P}$  objects. These oversized clusters dominate the computational capacity and storage requirements of MAHC, whose complexity can in the

worst case again approach  $O(N^2)$  thereby bringing back the same computational problem presented by the classical AHC. Figure 6.1 illustrates how the number of occupants of the largest subset evolves during 5 iterations of the MAHC algorithm in an example application to the four datasets described in Section 5.4.1.

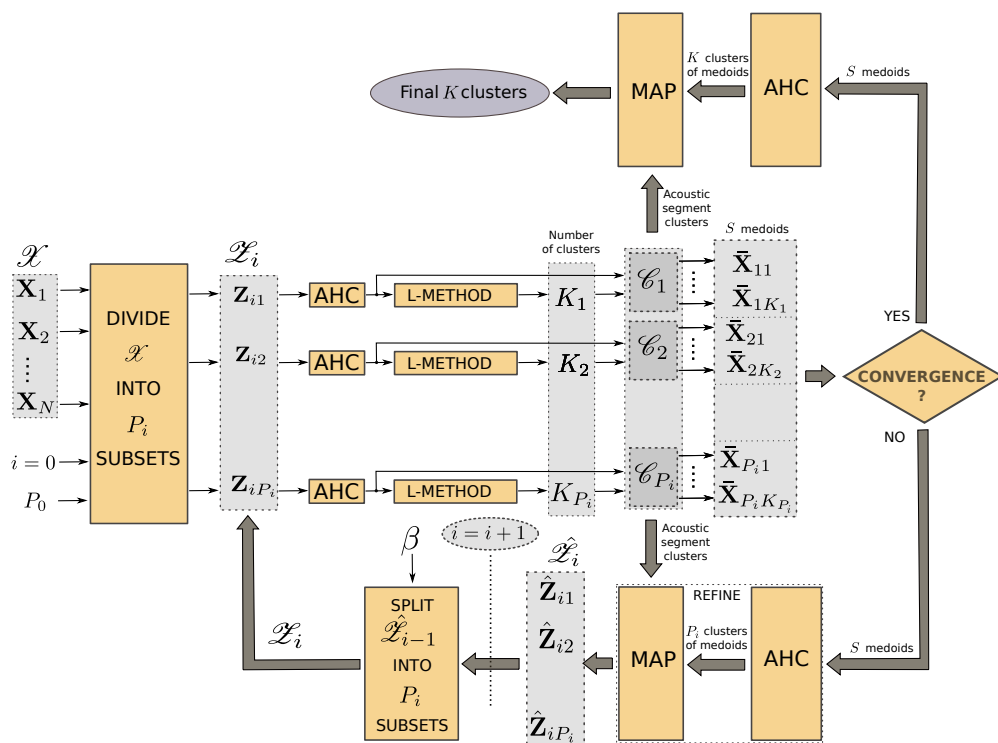


**Figure 6.1:** Total membership per iteration of the subset containing the largest number of speech segments when applying MAHC to (a) Small Set A and Small Set B in both cases with  $P = 4$  subsets and (b) the Medium Set with  $P = 6$  subsets and the Large Set with  $P = 8$  subsets.

At iteration 0, the occupancy corresponds to evenly-divided subsets, i.e.  $\frac{P}{N}$ . For all four datasets, the occupancy of the largest cluster grows during at least the first two iterations. This increase can be mild, as it is for the Medium Set in Figure 6.1 (b). However for Small Set A in Figure 6.1 (a) as well as for the Large Set in Figure 6.1 (b) the occupancy of the largest cluster grows to approximately twice its initial value at some point in the iterative process. A chief objective of the MAHC algorithm was to ensure that the similarity matrices that must be computed remain manageable in size, so that they can be stored in memory and do not have to be relegated to disk, for example. However Figure 6.1 shows that the occupancy of individual subsets may grow substantially. From this observation, there is no guarantee that the practically available memory will not be exceeded and hence a modification to MAHC that ensures no drastic growth for some subset sizes is proposed.

### 6.3 MAHC with cluster size management

The runaway growth in occupancy of certain clusters during MAHC is addressed by repeatedly subdividing the offending clusters at each iteration of the algorithm and also considering the appropriateness of merging clusters when they become too small. This new approach is called MAHC with cluster size management (MAHC+M). MAHC+M seeks to maintain the advantages offered by MAHC while guaranteeing that no subset grows too large for the available computational and storage resources. This process requires the number of subsets  $P$  to be allowed to vary between iterations, as illustrated in Figure 6.2.



**Figure 6.2:** Multi-stage agglomerative hierarchical clustering with cluster size management (MAHC+M), as also described in Algorithm 1.

The parameters of MAHC+M are:

1. The number of initial subsets,  $P_0$ .
2. The final number of clusters,  $K$ .
3. An integer threshold  $\beta$  indicating the largest number of objects any subset is allowed to contain.

The steps of the clustering procedure are given in Algorithm 1. The first step is to determine the threshold  $\beta$  which is usually dictated directly by available memory and processors.  $\beta$  is related to the initial number of subsets by  $P_0 \approx N/\beta$ . In the MAHC+M algorithm the number of subsets is allowed to vary as the algorithm iterates. This varying number of subsets is denoted by  $P_i$  for the  $i^{\text{th}}$  iteration. At each iteration, a split step uses  $\beta$  to subdivide subsets with membership exceeding  $\beta$  and ensures that all subsets delivered to the next iteration of the algorithm are within this limit such that  $\beta \geq N/P_i$ .

Following the split step shown in Figure 6.2, the AHC algorithm is applied to each split subset  $\mathbf{Z}_{ip}$  as defined by Equation 6.1:

$$\mathcal{Z}_i = \{\mathbf{Z}_{i1}, \mathbf{Z}_{i2}, \dots, \mathbf{Z}_{iP_i}\}. \quad (6.1)$$

where  $p = 1, \dots, P_i$ . Equation 2.1 in Chapter 2 defines  $\mathcal{X}$  as set of all speech segments. Equation 6.2 shows the relation between subset elements  $\mathbf{Z}_{ip}$  and  $\mathcal{X}$ .

$$\bigcup_{p=1}^{P_i} \mathbf{Z}_{ip} = \mathcal{X} \quad (6.2)$$

The AHC algorithm produces clusters  $\mathcal{C}_p$  for each subset. The medoids  $\bar{\mathbf{X}}_s$ ,  $s = 1, \dots, S$  of all the subsets are then determined and subsequently themselves clustered using AHC, as already described in Section 5.2. These subsets are fed back into the algorithm, rendering MAHC+M iterative. A stopping criterion for this algorithm is determined at the convergence step illustrated in Figure 6.2.

Convergence can be decided on the basis of a settling in the number of subsets  $P_i$ , or simply by terminating the clustering procedure after a fixed number of iterations. For MAHC in Chapter 5 it has been empirically demonstrated that the final number of clusters is well approximated by the total number of clusters resulting from the first stage of the algorithm. For MAHC+M we verify that this approximation  $K = \sum_{j=1}^{P_i} K_j$  remains valid after the introduction of cluster size management and can therefore again be used to automatically determine a suitable value for the final number of clusters  $K$ . The steps of the clustering procedure are described in Algorithm 1.

## 6.4 Data and evaluation measures

Data used in the experiments is of the same composition as that used in Section 5.4.1 including the MFCC's features. All experiments use a set of TIMIT basephones corresponding to triphones that are at least 5 milliseconds long. Pauses were excluded from our dataset. We will use the F-Measure to quantify the quality of a division of the acoustic segments in the dataset into one of  $K$  clusters while the L method will be used to automatically determine the value of  $K_i$  for each subset.

## 6.5 Experimental evaluation

### 6.5.1 DTW-based experiments

The algorithm described in Section 6.3 is applied to the datasets described in Section 5.4.1. For comparison, results without cluster size management as described in Section 5.4 are also shown. Furthermore all experiments in this

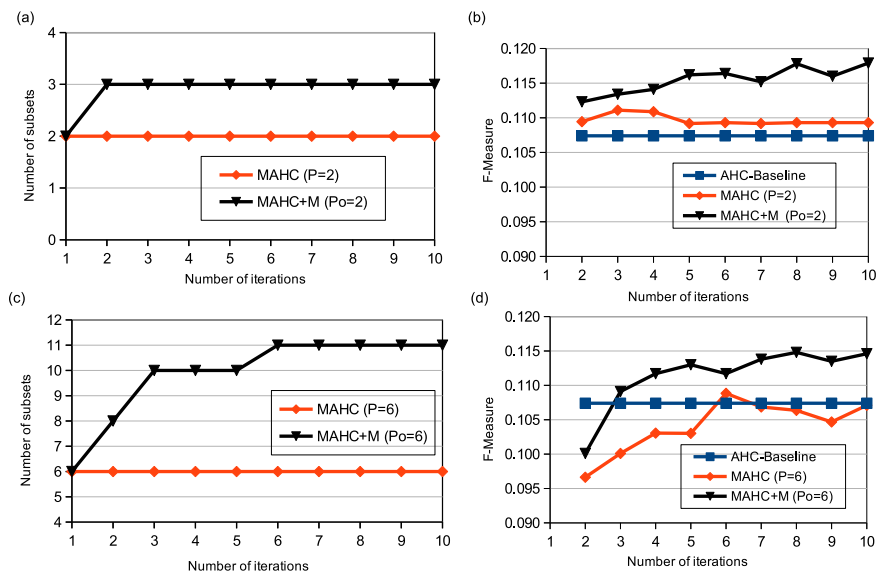
---

**Algorithm 1:** Modified agglomerative hierarchical clustering (MAHC) with cluster size management, as also described in Figure 6.2

---

- Input:**  $N$  acoustic segments  $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \dots, \mathbf{X}_N\}$ ; initial number of subsets  $P_0$ ; integer threshold  $\beta$ ; .
- Output:**  $K$  clusters  $\mathcal{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K\}$
- 1  $i = 0$  ;
  - 2 Divide  $N$  acoustic segments into  $P_i$  subsets  $\mathcal{Z}_i = \{\mathbf{Z}_{i1}, \mathbf{Z}_{i2}, \dots, \mathbf{Z}_{iP_i}\}$  ;
  - 3 Independently apply AHC to each subset, resulting in  $P_i$  dendrograms ;
  - 4 Use the L method to determine the optimal number of clusters  $K_p$ ,  $p = 1, 2, \dots, P_i$ , for each of the  $P_i$  dendrograms in Step 3. This results in  $P_i$  sets of clusters  $\mathcal{C}_p = \{\mathbf{C}_{p1}, \mathbf{C}_{p2}, \dots, \mathbf{C}_{pK_p}\}$  with  $p = 1, 2, \dots, P_i$ ;
  - 5 Find the medoid  $\bar{\mathbf{X}}_{pk}$  of each cluster  $\mathbf{C}_{pk}$  where,  $p = 1, 2, \dots, P_i$  and  $k = 1, 2, \dots, K_p$ . This results in a set of  $S$  medoids  $\bar{\mathcal{X}}$ , where  $S = \sum_{p=1}^{P_i} K_p$  ;
  - 6 If  $i > 2$  and convergence has been achieved go to Step 11 (conclude) ;
  - 7 Divide the  $S$  medoids obtained in the Step 5 into  $P_i$  clusters using AHC ;
  - 8 Map the members of each cluster  $\mathbf{C}_{pk}$  to one of  $P_i$  new subsets  $\hat{\mathcal{Z}}_i = \{\hat{\mathbf{Z}}_{i1}, \hat{\mathbf{Z}}_{i2}, \dots, \hat{\mathbf{Z}}_{iP_i}\}$  according to the result of the previous step (refine);
  - 9 Consider each new subset  $\hat{\mathbf{Z}}_{ij}$   $j = 1, 2, \dots, P_i$  and if it contains more than  $\beta$  acoustic segments, subdivide it evenly to ensure that the limit  $\beta$  is not exceeded (split) ;
  - 10 Let the total number of subsets resulting from the previous step be  $P_{i+1}$  and the subsets themselves be denoted by  $\mathcal{Z}_{i+1}$  ;
  - 11  $i = i + 1$  ;
  - 12 Go to Step 3 (iterate);
  - 13 Divide the  $S$  medoids obtained in the previous step into  $K = \sum_{j=1}^{P_i} K_j$  clusters using AHC ;
  - 14 Map the members of each cluster  $\mathbf{C}_{pk}$  to one of  $K$  new subsets according to the result of the previous step ;
  - 15 The  $K$  subsets obtained in the previous step are the final clustering result.
-

section use the classical formulation of DTW.

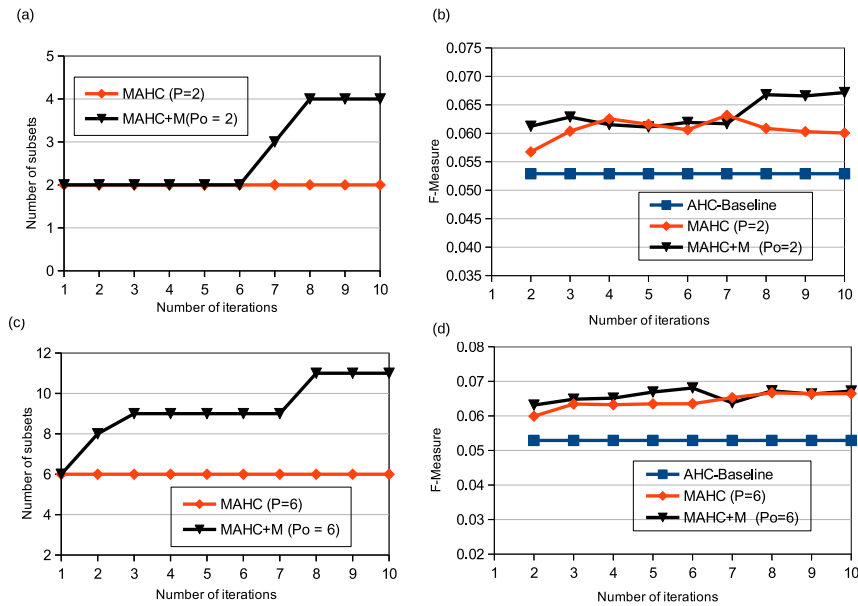


**Figure 6.3:** Number of subsets  $P_i$  as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to Small Set A with an initial number of subsets of  $P_0 = 2$  (a and b) and  $P_0 = 6$  (c and d).

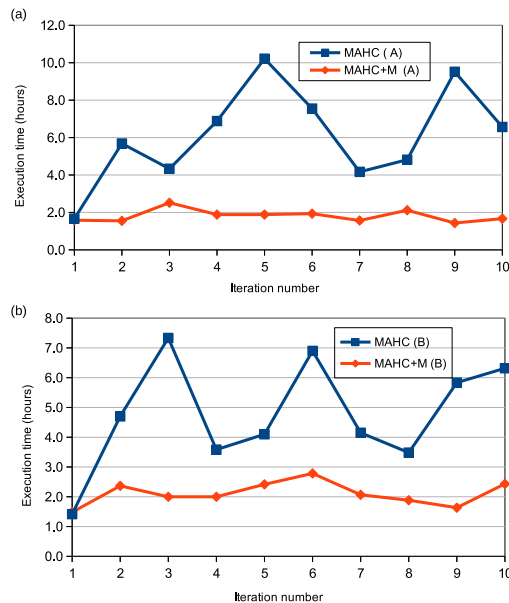
Figures 6.3 (a) and (c) show the number of subsets per iteration  $P_i$  when the Small Set A data are initially divided into  $P_0 = 2$  and  $P_0 = 6$  subsets respectively. The corresponding F-Measure plots are shown in Figures 6.3 (b) and (d). For Small Set A, the introduction of cluster size management has led to some improvement in terms of F-Measure for both  $P_0 = 2$  and  $P_0 = 6$ . Figure 6.4 shows the results of a corresponding set of experiments for Small Set B, which is similar in size to Small Set A but not as skewed. We see that also in this case cluster size management has resulted in no deterioration in terms of F-Measure.

To obtain an indication of the practical impact on processing time afforded by the introduction of cluster size management, Figure 6.5 shows the measured time (in hours) taken per iteration to cluster Small Set A and Small Set B with  $P_0 = 6$ . These small datasets were chosen because they allow clustering to be performed on a normal stand-alone workstation, in our case an Intel Core i7 with four cores, running at 3.40 GHz and with 32 GB of RAM. Figure 6.5 indicates a reduction in processing time of up to a factor of five, while Figures 6.3 (d) and 6.4 (d) have already indicated that these savings does not incur a penalty in terms of F-Measure.

Next, we present results for the Medium Set, in this case also explicitly observing the occupancy of the largest subset. Figure 6.6 shows the number



**Figure 6.4:** Number of subsets  $P_i$  as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to Small Set B with an initial number of subsets of  $P_0 = 2$  (a and b) and  $P_0 = 6$  (c and d).

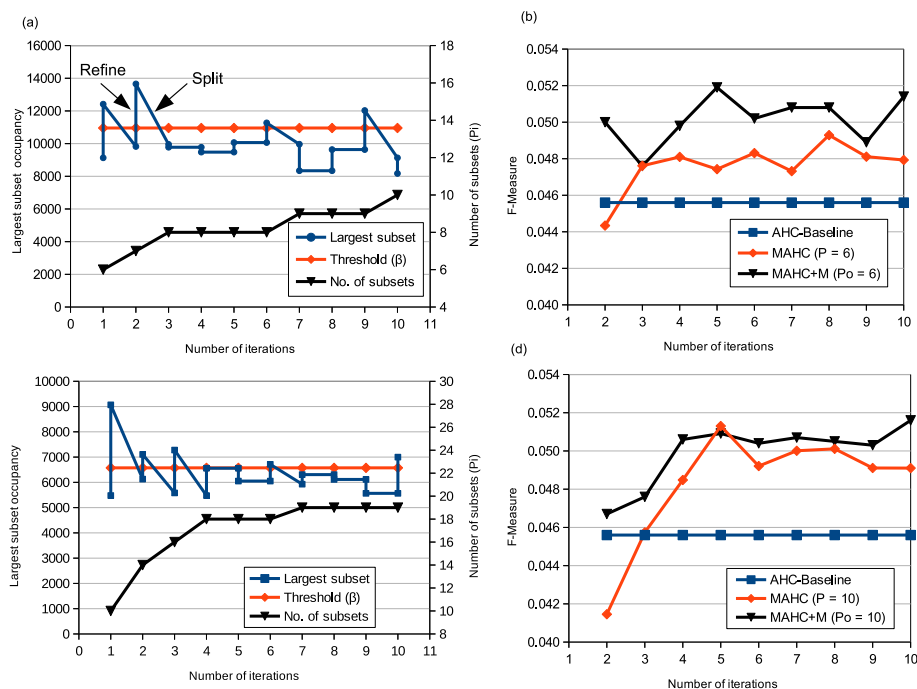


**Figure 6.5:** Per-iteration execution time of modified agglomerative hierarchical clustering with (MAHC+M) and without (MAHC) cluster size management with  $P_0 = 6$  initial subsets for (a) Small Set A and (b) Small Set B.



of subsets  $P_i$ , the occupancy of the largest subset, and the F-Measure when clustering the Medium Set with  $P_0 = 6$  and  $P_0 = 10$  initial subsets, as well as example points at which the *split* and *refine* steps in Algorithm 1 occur.

During the *refine* stage, clusters from previous iterations are regrouped, which may lead to greater imbalance in the membership of the clusters and hence an increase in the size of the largest subset. The *split* stage subdivides any overly large subsets, ensuring that the threshold  $\beta$  is not exceeded.

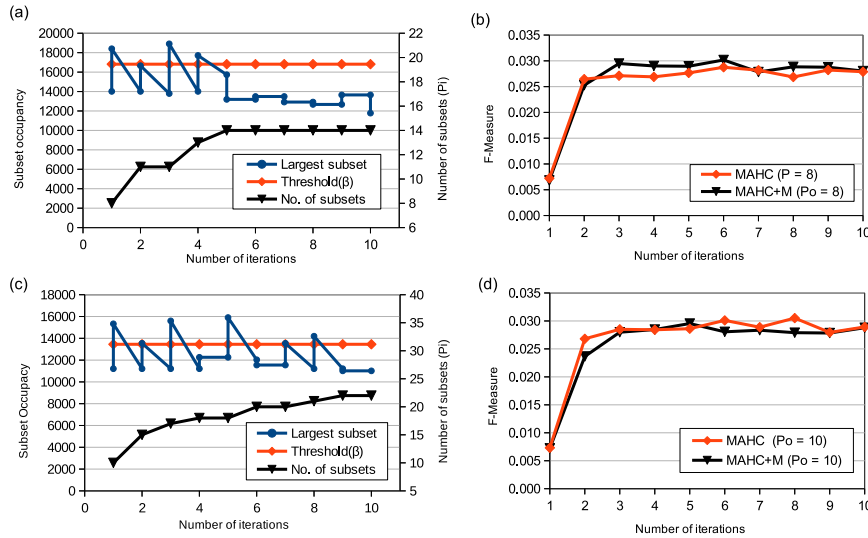


**Figure 6.6:** Number of subsets  $P_i$  as well as F-Measure for each iteration when applying classical agglomerative hierarchical clustering (AHC), modified AHC (MAHC) and MAHC with cluster size management (MAHC+M) to the Medium Set with an initial number of subsets of  $P_0 = 6$  (a and b) and  $P_0 = 10$  (c and d).

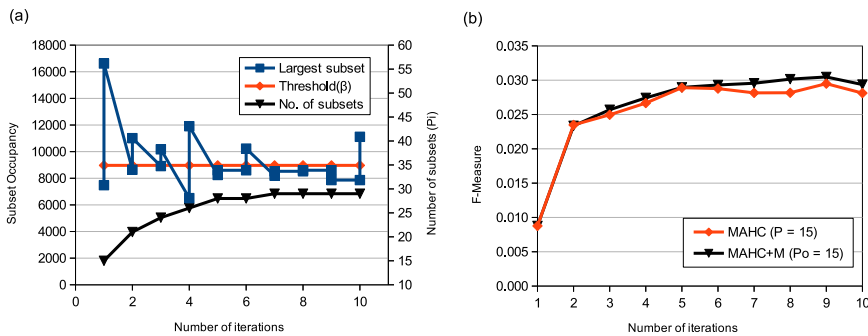
Consider for illustration Figure 6.6(a), where each subset is occupied by 9,131 segments at the start of the first iteration. We observe that the first 2 iterations lead to maximum occupancies that are higher than  $\beta$ . In each case the split step subsequently brings these occupancies below the threshold  $\beta$ . This also leads to an increase in the number of subsets  $P_i$ . Similar behaviour is seen in Figure 6.6(c).

Figures 6.6(b) and (d) show that, as for Small Sets A and B, the introduction of cluster size management has not led to a degradation in clustering performance in terms of F-Measure for the Medium Set.

Finally, results for the Large Set are presented in Figure 6.7. The number of subsets in Figure 6.7(a) where  $P_0 = 8$  reaches a plateau at the fifth iteration,



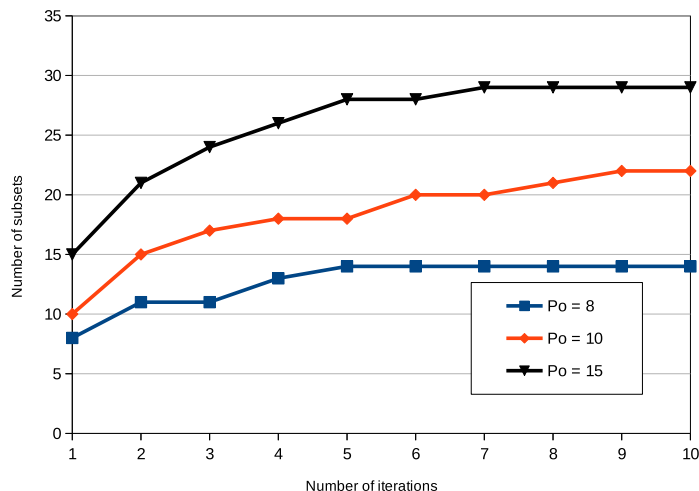
**Figure 6.7:** Number of subsets  $P_i$  as well as F-Measure for each iteration when applying modified agglomerative hierarchical clustering (MAHC) and MAHC with cluster size management (MAHC+M) to the Large Set with an initial number of subsets of  $P_0 = 8$  (a and b) and  $P_0 = 10$  (c and d).



**Figure 6.8:** Number of subsets  $P_i$  as well as F-Measure for each iteration when applying modified agglomerative hierarchical clustering (MAHC) and MAHC with cluster size management (MAHC+M) to the Large Set with an initial number of subsets of  $P_0 = 15$  (a and b).

while in Figure 6.7(c) we see that the number of subsets is still increasing after 8 iterations. In both cases, however, the corresponding F-Measure has settled after 3 iterations, indicating that good clustering has been achieved. In terms of the F-Measure the results for MAHC and MAHC+M are relatively stable and very close in comparison. Figure 6.8 investigates what happens when the number of subsets is further increased to  $P_0 = 15$ . It is observed in Figure 6.8 (a) that the number of subsets remains constant from the seventh iteration onwards. The F-Measure in Figure 6.8 (b) keeps increasing for both MAHC

and MAHC+M but settles after the fifth iteration.



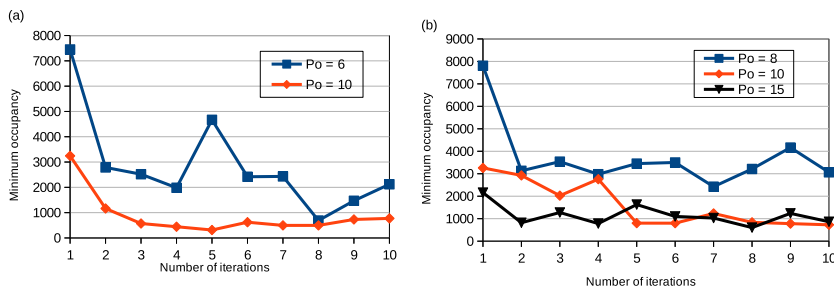
**Figure 6.9:** Number of subsets ( $P_i$ ) for each iteration where  $P_0$  is initial number of subsets.

Figure 6.9 shows how the split step increases the number of subsets used by the clustering algorithm when applied to the Large Set. We see that the number of subsets seems to settle as the iterations progress. Referring back to Figure 6.7, we are reminded that the F-Measure settles even when the number of subsets continues to increase. Hence it appears to be reasonable to terminate the clustering algorithm after a fixed number of iterations, and it is not necessary to wait for example until the number of subsets no longer changes.

Additionally, we would like to consider the merit of introducing a *merge* step to complement the *split* step into Algorithm 1. The motivation for a merge step would be to re-absorb subsets whose membership vanishes during the algorithm due to the repeated iterative application of the split step. Figure 6.10 shows the size of the smallest subset at each iteration for the Medium and Large Sets. We see that for both datasets the subset membership never vanishes. This behaviour was observed consistently in all our experiments. From this we conclude that the addition of a merge step is not necessary for the effective functioning of the algorithm.

### 6.5.2 FTDTW Experiments

Most of the experiments in both Chapter 5 and this chapter utilised the classical formulation of DTW described in Chapter 4 as a similarity measure between segments. Although the experimental results in Chapter 4 presented feature trajectory DTW (FTDTW) as a potentially better formulation of DTW for



**Figure 6.10:** Minimum occupancy per iteration for (a) Medium Set and (b) Large Set.

speech signals, it has up to this point not been fully evaluated in conjunction with the MAHC algorithm. In this section we compare the performance of FTDTW and DTW when used by MAHC+M to cluster the four datasets presented in Section 5.4.1 and also used in the previous section to evaluate MAHC+M. The emphasis of the analysis will be placed on the Large Set and we will consider whether the resulting clusters might be suitable for generating pronunciation dictionaries to be used in automatic speech recognition. Experiments using the Small and Medium sets will be used to highlight the influence of FTDTW in terms of F-Measure.

### 6.5.2.1 Small and medium datasets

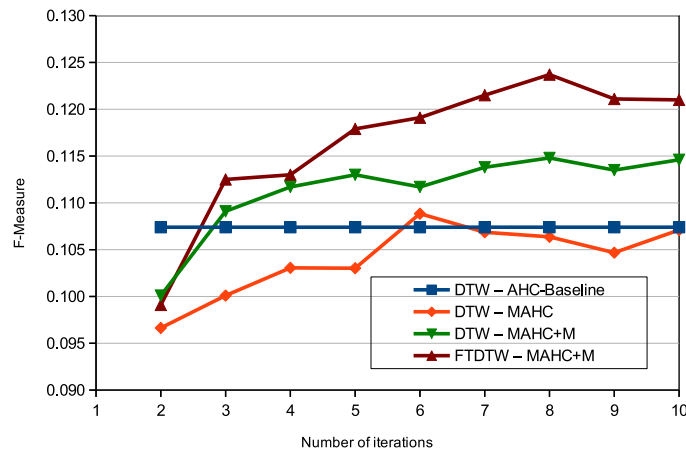
In the following experiments, the FTDTW and DTW results are compared. Specifically, we will consider the cases when  $D(\mathbf{X}_i, \mathbf{X}_j) = FTDTW(\mathbf{X}_i, \mathbf{X}_j)$  and  $D(\mathbf{X}_i, \mathbf{X}_j) = DTW(\mathbf{X}_i, \mathbf{X}_j)$  as depicted in Equation 4.4.

Figure 6.11 shows substantially better performance from the second iteration onwards for Small Set A when using FTDTW instead of DTW. Figure 6.12 shows gains also for Small Set B, although smaller, from iteration 4 onwards. These results affirm the findings already reported in Chapter 4.

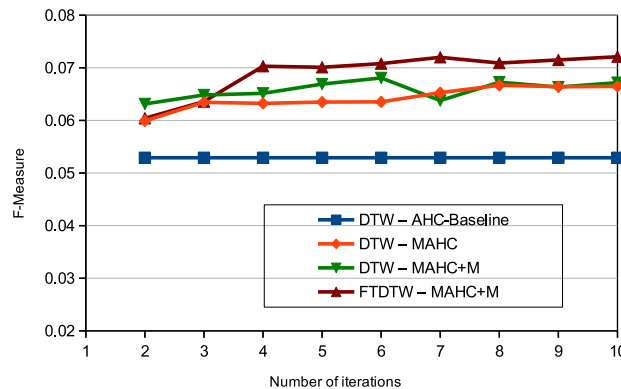
Corresponding results are shown in Figure 6.13 for the Medium Set with an initial number of subsets  $P_0 = 10$ . We see here that FTDTW-based MAHC+M improves on DTW-based MAHC after the fifth iteration. Since neither the small datasets nor the medium sets are going to be used in creating pronunciation dictionaries for speech recognition, these FTDTW results can be considered as additional findings to those reported in Chapter 4.

### 6.5.2.2 Large dataset

The positive results for the Small and Medium datasets led to experimentation using the Large Set, with a view to using the discovered clusters for sub-word modelling in ASR. As in previous experiments, the FTDTW-based MAHC+M results are benchmarked against DTW-based MAHC and MAHC+M in terms of the F-Measure. In these experiments, however, confusion matrices are also



**Figure 6.11:** Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to Small Set A with an initial number of subsets of  $P_0 = 6$ .



**Figure 6.12:** Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to Small Set B with an initial number of subsets of  $P_0 = 6$ .

presented as a means of investigating how well similar sounding acoustic segments assemble in the same cluster.

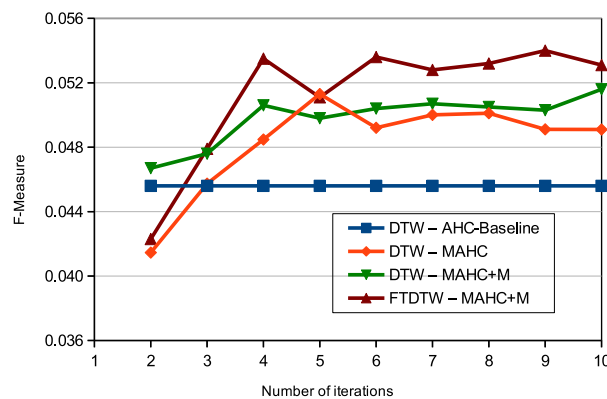
Figure 6.14 shows the performance of DTW-based MAHC and MAHC+M in comparison to FTDTW-based MAHC+M when  $P_0 = 8$ . We observe that FTDTW-based MAHC+M performance in terms of the F-Measure is slightly better than that of the other formulations of MAHC.

To assess the suitability of the  $K$  clusters produced by these MAHC algorithm configurations to sub-word modelling, they are visualised as confusion matrices. A confusion matrix is normally used in classification problems to

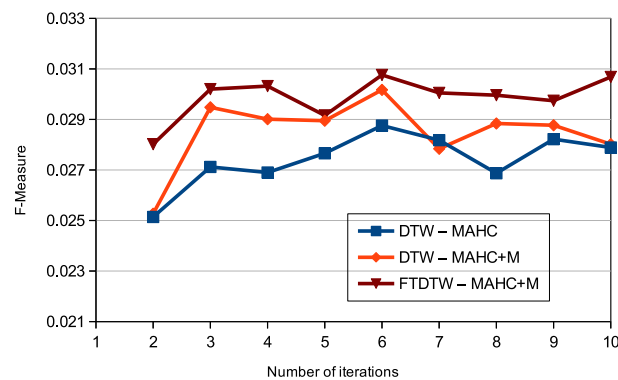
indicate the degree to which the classification result corresponds to the reference class labels [140]. Since clustering is not a classification problem, we must identify "reference" class labels by alternative means. We recall that the speech segments being clustered are triphones. Figure 6.15 shows the triphone labels associated with the segments of two example clusters. A triphone element in a cluster consists of a basephone together with its left and right contexts, indicated by  $-$  and  $+$  characters respectively.

As a first step, we remove the left and right triphone contexts from all members of all clusters. This reduces the example clusters shown in Figure 6.15 to the basephone clusters shown in Figure 6.16.

The second step is to consider these  $K$  basephone clusters and for each identify the single TIMIT basephone which dominates the cluster. This domi-



**Figure 6.13:** Cluster quality in terms of F-Measure when applying DTW-based classical AHC, MAHC, MAHC+M and FTDTW-based MAHC+M to the Medium Set with an initial number of subsets of  $P_0 = 10$ .



**Figure 6.14:** Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of  $P_0 = 8$ .

```

1. f-s+kcl ix-s+tcl ix-z+ix ih-s+tcl ix-z+dcl ao-s+tcl ay-z+ix ux-z+dcl ah-s+bcl
uh-z+epi ux-z+dcl iy-s+kcl ix-s+kcl ux-s+epi iy-s+kcl ix-s+kcl uh-s+gcl er-s+tcl
ux-z+tcl ix-s+f iy-s+tcl ix-z+h# ix-z+tcl oy-z+en ay-s+tcl n-s+tcl ey-s+ix
el-z+h# ih-s+tcl n-s+q r-s+tcl ix-sh+s er-s+tcl ih-sh+z r-s+tcl ih-s+q ao-s+f
ah-s+epi ae-s+bcl v-z+bcl ow-s+tcl n-s+tcl axr-s+pcl r-s+epi ux-z+f ow-z+epi
ih-s+pcl ix-s+tcl ih-s+tcl ix-sh+tcl ah-s+tcl n-s+tcl ix-s+tcl ix-z+epi er-s+tcl
ix-s+tcl ix-z+dcl ix-s+tcl iy-s+tcl ay-s+dcl ix-z+dcl ix-s+tcl ux-s+tcl ih-z+dh
ix-s+dcl ix-s+pau ih-s+tcl ay-s+dcl ix-s+tcl eh-s+dh ih-s+tcl ax-s+tcl ax-z+dcl
ix-z+tcl eh-s+tcl ow-s+tcl ix-s+tcl ey-s+tcl aa-s+tcl ih-s+dcl ix-z+epi pau-s+tcl
uh-s+tcl ao-s+dh n-s+tcl ay-s+dcl n-s+dcl n-s+tcl ix-s+tcl ah-s+tcl ih-s+th
ih-s+tcl ih-s+tcl ih-s+tcl ix-s+tcl ih-s+kcl ix-s+epi ih-s+tcl ux-z+dcl ah-s+tcl
iy-z+epi ih-s+tcl iy-z+tcl dcl-jh+ah n-z+q n-z+ow ow-s+tcl ng-z+bcl n-z+q ax-s+ay
ix-z+hh g-z+ae ih-s+pcl ux-z+f uw-s+ix ux-s+bcl axr-s+ix ix-s+ay ey-z+epi ix-s+ix
ey-sh+ih eh-s+tcl eh-s+tcl ih-s+pcl ah-s+tcl iy-s+tcl iy-z+sh ih-s+ix iy-s+kcl
ow-z+hh ih-s+pcl ih-z+thv iy-z+ax iy-z+kcl iy-z+kcl ix-s+pcl ih-z+pcl m-z+kcl
ux-th+pcl ae-s+ix ih-z+pcl ay-s+ix ih-z+pau eh-s+ax aa-s+bcl eh-z+f ae-z+ax
aa-s+eh ih-s+ix eh-s+ey ih-s+ix ix-s+tcl ih-s+kcl ae-z+epi ax-s+tcl el-s+kcl
aa-s+pcl ih-s+kcl t-s+ay ix-s+ih iy-s+eh l-s+epi ax-s+epi l-s+ax ix-s+pcl
ng-z+tcl ix-s+epi n-s+ux ax-s+pcl ax-s+pcl axr-s+pcl axr-z+tcl ih-s+ax-h ow-s+tcl
ax-s+tcl ng-s+tcl n-s+tcl eh-s+tcl eh-s+tcl ae-s+pcl ae-s+kcl ae-s+tcl ae-z+tcl
ah-s+tcl eh-s+tcl uh-s+tcl ae-s+pcl ix-s+dcl ae-z+hh ey-z+hh eh-z+hh ix-z+f
oy-s+tcl ay-z+bcl ow-z+gcl ih-s+pcl m-z+v ix-z+f ix-z+f d-z+th ax-h-s+pcl n-z+ix
ax-s+ey ay-z+dh ay-s+ax-h ix-z+kcl b-z+ey k-s+dcl ao-s+tcl er-s+tcl n-jh+dcl
ow-s+kcl ix-s+kcl iy-z+pcl iy-z+kcl oy-s+tcl ey-z+tcl k-s+tcl t-s+tcl iy-s+tcl
ey-s+f ow-s+tcl ch-s+tcl

2. z-q+ih zh-p+l sh-th+er s-th+ux s-dh+iy s-q+ix q-dh+ix k-dh+ix z-dh+iy s-dh+ix
sh-th+er z-q+ix tcl-f+ae t-q+ax s-th+ih v-dh+iy jh-dh+ih f-dh+eh s-th+ix p-dh+ax
ch-dh+eh z-dh+eh s-dh+ae h#-dh+ix p-th+ax k-q+ae ch-dh+ax s-dh+eh h#-dh+ix
h#-dh+ix sh-th+er h#-dh+ix tcl-dh+ax k-dh+iy s-dh+ax z-f+r h#-dh+iy h#-q+ao
h#-q+eh t-dh+ix k-dh+ae b-f+iy ch-th+ih z-dh+ix h#-q+aa d-f+ax h#-f+ay pau-f+ih
pau-f+ax s-f+ae m-f+ax dcl-v+ae v-hh+ih pau-f+ih th-v+axr kcl-f+ax k-th+r pau-f+r
kcl-f+eh k-f+ix n-f+ao t-f+axr h#-f+ay k-f+ix tcl-f+ah kcl-th+r tcl-t+ix h#-dh+iy
ao-th+r kcl-k+y dcl-g+r epi-f+aa kcl-t+w tcl-v+er kcl-dh+ix kcl-k+q kcl-k+m
kcl-k+dh h#-dh+eh tcl-t+ih kcl-t+n k-dh+iy kcl-th+eh kcl-k+h# pcl-p+ih kcl-k+dh
bcl-v+ae kcl-k+n s-dh+ih

```

**Figure 6.15:** Triphone labels corresponding to the acoustic segments clustered by MAHC with  $P_0 = 8$  and  $K = 1220$  for the Large Set. The first two clusters are shown where each cluster consists of a basephone together with its left and right contexts, indicated by the  $-$  and  $+$  characters respectively.

nant cluster member will be considered to be the reference basephone. Figure 6.16 for example shows that  $\frac{149}{223} \times 100\% = 66.8\%$  of the members of cluster 1 correspond to the basephone "s". In this example, the basephone "s" would be considered the reference phone for that cluster and will be called *dominant basephone* in confusion matrix plots. However, the number of clusters  $K$  is much greater than the number of different TIMIT basephones. For the purpose of the confusion matrix, the memberships of all clusters with the same dominant basephone are merged.

```

1. s s z s z s z z s z z s s s s s s z s s z z z s s s z s s s sh s sh
s s s s s z s s s s z z s s s sh s s s z s s z s s s z s s s s s s
s s z z s s s s s s z s s s s s s s s s s s s s s s s z s z s jh
z z s z z s z z s s s s s z s sh s s s s s z s s z z z z s z z th
s z s z s z z s s s s s z s s s s s s s s s s s s z s s s s z s s s
s s s s s s z s s s s s z z z z s z z z z z h z h z s z z s s s
jh s s z z s z s s s s s s

2. q p th th dh q dh dh dh th q f q th dh dh th dh dh dh th q
dh dh dh th dh dh dh f dh q q dh dh f th dh q f f f f v hh f
v f th f f f f f f f th t dh th k g f t v dh k k k dh t t dh th k p k
v k dh

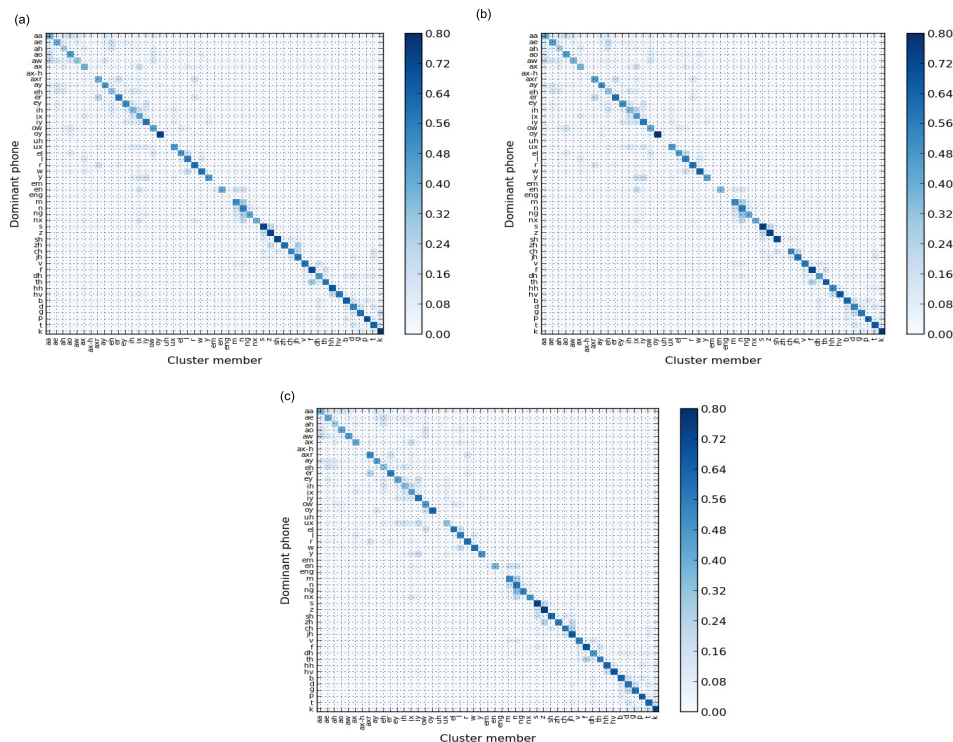
```

**Figure 6.16:** TIMIT basephone labels of MAHC output with  $P_0 = 8$  and  $K = 1220$  for the Large Set. The first two clusters are shown.

Using the dominant basephones as "reference" labels, a confusion matrix is populated. This matrix indicates the degree to which the clusters domi-



nated by a particular basephone are also occupied by other phones. A perfect confusion matrix would have non-zero entries only on the diagonal. Figure 6.17 shows such confusion matrices for the Large Set with  $P_0 = 8$  for (a) DTW-based MAHC and (b) the DTW-based MAHC+M and (c) FTDTW-based MAHC+M. The corresponding F-Measures have already been shown in Figure 6.14 and are summarised again in Table 6.1. The ordering of TIMIT basephones on the axes is based on the categorisation of TIMIT phonemes suggested by Halberstadt and Glass [36] and also Lopes and Perdigão [141].

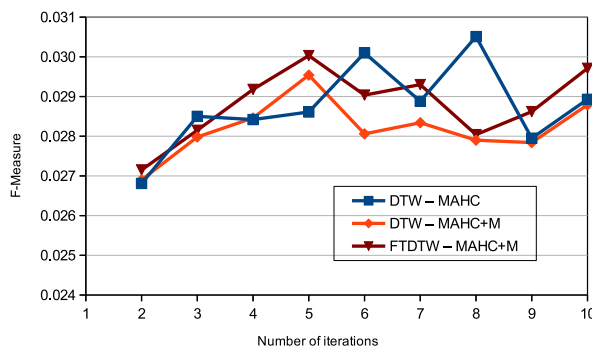


**Figure 6.17:** Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when  $P_0 = 8$  at iteration 6 using (a) DTW-based MAHC with the number of clusters  $K = 1220$ , (b) DTW-based MAHC+M with  $K = 1475$  and (c) FTDTW-based MAHC+M with  $K = 1386$ .

In Table 6.1, we observe that performance in terms of F-Measure of the MAHC algorithms closely match each other at the sixth iteration when  $P_0 = 8$ . DTW-based MAHC has the lowest F-Measure value followed by the DTW-based MAHC+M. The best F-Measure is achieved by the FTDTW-based MAHC+M algorithm. Looking at these closely matched values it is indeed visually difficult to differentiate the quality of basephone alignments from confusion matrices in Figure 6.17.



A second set of experiments was performed for the Large Set, in which the initial number of subsets was chosen to be  $P_0 = 10$ . Figure 6.18 shows that the introduction of MAHC+M and also the use of FTDTW does not always outperform DTW-based MAHC. It is however observed that FTDTW-based MAHC+M usually performs better than its DTW-based counterparts. Table 6.1 also indicates a greater variation in F-Measure than for  $P_0 = 8$ .

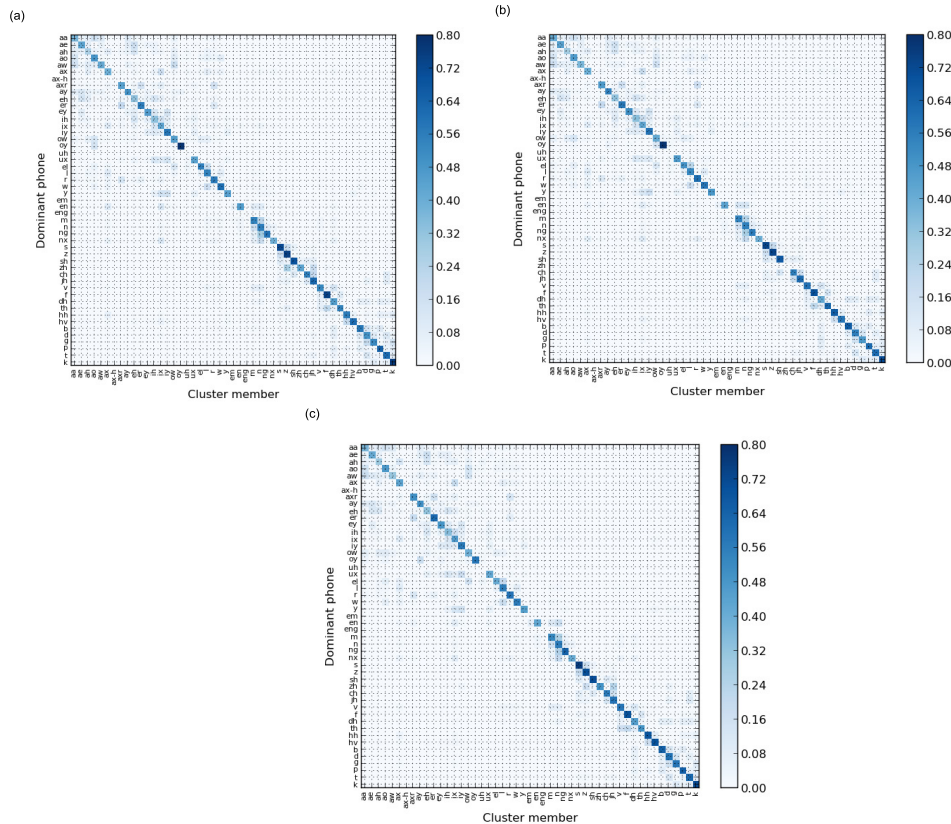


**Figure 6.18:** Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of  $P_0 = 10$ .

The confusion matrices in Figure 6.19 suggest, however, that for  $P_0 = 10$  there is no significant difference in terms of basephone clusters to those reported above for  $P_0 = 8$ . Figure 6.19 shows that DTW-based MAHC and MAHC+M as well as FTDTW-based MAHC+M produce distinctive groupings of basephones which could potentially be used for sub-word unit modelling in ASR.

A final set of experiments used  $P_0 = 15$  as the number of initial subsets. Figure 6.20 shows that the F-Measure for MAHC+M increases until iteration 7 and then starts to fall gradually. The F-Measure for FTDTW-based MAHC+M generally shows improved performance relative to the two DTW-based configurations except in iteration 6 where it is outperformed by DTW-based MAHC+M.

Confusion matrices for the experiments with  $P_0 = 15$  are shown in Figure 6.21 and are not substantially different from those of  $P_0 = 8$  and  $P_0 = 10$  reported above. The F-Measure values for  $P_0 = 15$  shown in Table 6.1 reinforce the trend observed for the other experiments and show the usually better performance achieved by MAHC+M with FTDTW. The confusion matrices for  $P_0 = 15$  also indicate that the clusters obtained could potentially be used for sub-word modelling in ASR.



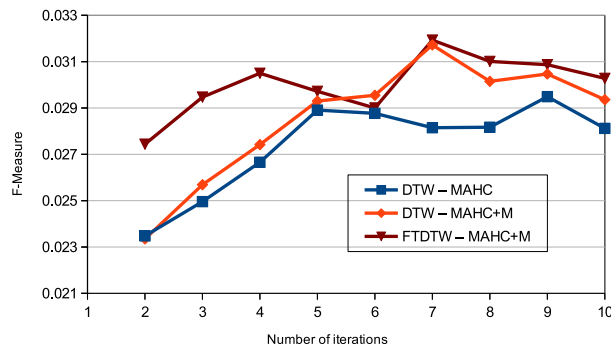
**Figure 6.19:** Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when  $P_0 = 10$  at iteration 6 using (a) DTW-based MAHC with the number of clusters  $K = 1315$ , (b) DTW-based MAHC+M with  $K = 1515$  and (c) FTDTW-based MAHC+M with  $K = 1560$ .

Clustering Algorithm	$P_0 = 8$ at iteration 6	$P_0 = 10$ at iteration 6	$P_0 = 15$ at iteration 7
DTW-based MAHC	0.3649	0.3718	0.3835
DTW-based MAHC+M	0.3744	0.3858	0.3907
FTDTW-based MAHC+M	0.3847	0.3938	0.3921

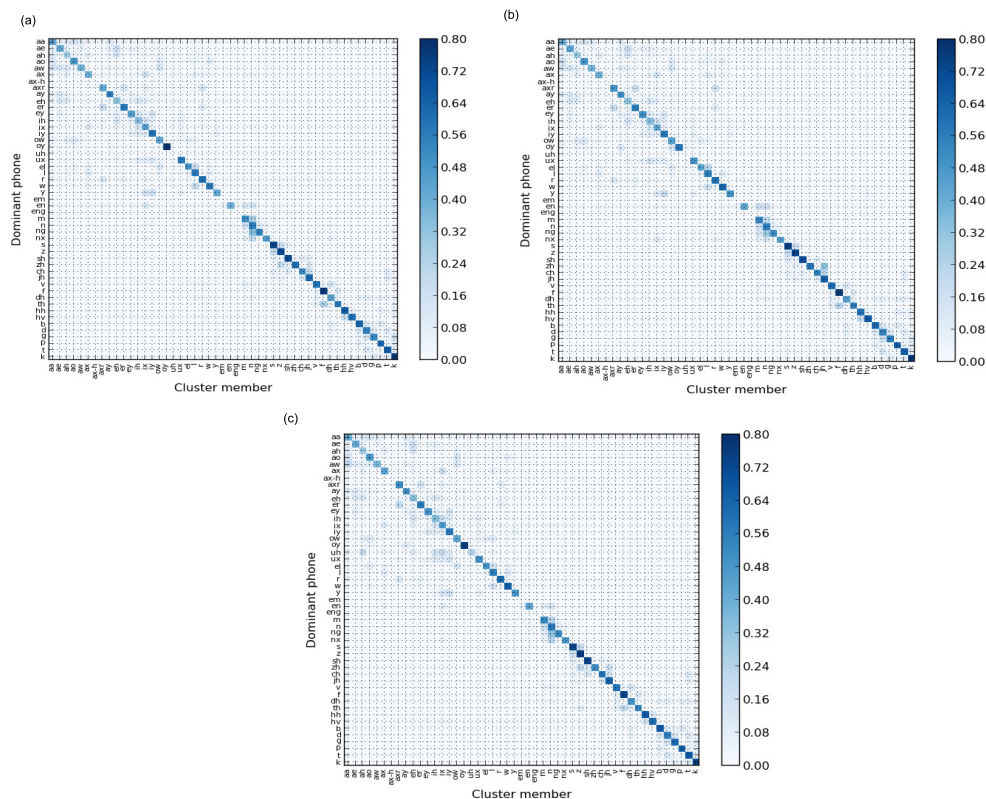
**Table 6.1:** The F-Measures corresponding to the confusion matrices shown in Figures 6.17, 6.19 and 6.21. All are for the Large Set.

## 6.6 Summary and conclusion

In this chapter we have extended the MAHC algorithm by iteratively enforcing a hard limit on the size of each of the internally-generated data subsets. This MAHC with cluster size management (MAHC+M) allows maximum space constraints to be guaranteed, and makes MAHC more reliably useful for the hierarchical agglomerative clustering of large datasets. We have shown that



**Figure 6.20:** Cluster quality in terms of F-Measure when applying DTW-based MAHC, DTW-based MAHC+M and FTDTW-based MAHC+M to the Large Set with an initial number of subsets of  $P_0 = 15$ .



**Figure 6.21:** Confusion matrix showing how strongly the experimentally obtained clusters are dominated by a single TIMIT basephone for the Large Set when  $P_0 = 15$  at iteration 7 using (a) DTW-based MAHC with the number of clusters  $K = 1554$ , (b) DTW-based MAHC+M with  $K = 1810$  and (c) FTDTW-based MAHC+M with  $K = 1954$ .

the proposed modification does not affect the algorithm's performance in terms of F-Measure when applied to a number of datasets of varying size compiled from the TIMIT speech corpus. Furthermore, we have demonstrated how the performance of MAHC+M improves when used in conjunction with feature trajectory DTW instead of with classical DTW. Finally, confusion matrices for sets of clusters obtained when applying MAHC and MAHC+M to the Large Set indicate that the produced clusters strongly correspond to the base-phones of TIMIT. This is promising with respect to their use as sub-word units and associated pronunciations for application in automatic speech recognition (ASR) considered in Chapter 7.

# Chapter 7

## Pronunciation Dictionary Generation

### 7.1 Introduction

As indicated in Chapter 1, the over-arching aim of this research is the development of automatic speech recognition (ASR) for the under-resourced languages for which hand-crafted pronunciation dictionaries are not available. The clustering algorithms proposed in Chapter 6 produce segment clusters that can be used to represent sub-word units for the purpose of acoustic modelling in ASR. In this chapter we will use the automatically obtained clusters to induce pronunciation dictionaries and evaluate their effectiveness in ASR. While in Chapters 5 and 6 cluster quality has already been evaluated in terms of F-Measure, the ASR experiments presented here may be viewed as an additional evaluation that is focussed directly on the application of the clusters to ASR. To obtain a pronunciation dictionary, clustered speech segments are first aligned with each word in the orthographic transcription in order to obtain initial pronunciation(s). Since only the acoustic data and the corresponding orthography are available, the word boundaries are unknown. Deriving pronunciation representations for words not seen at all in training poses a further challenge, as already pointed out by other researchers such as Livescu *et al* [3]. In general, automatically induced dictionaries yield multiple pronunciations for each word. Research has found that excessive pronunciation variability can cause a degradation in the performance of ASR system [142; 143; 144]. Livescu *et al* [3] report that data sparseness is another obstacle in the determination of sub-word models since there are usually too many triphones relative to the training data available. To address these challenges, thorough research into the induction of a good pronunciation dictionary from the results of the acoustic clustering is necessary. This was however not the focus of this work, and hence we will adopt the following straightforward procedure inspired by the method proposed in [145].

- Initial word alignments between words and the sub-word units are approximated.
- The Viterbi algorithm is used to refine these initial word alignments to obtain sub-word unit alignments.
- A list of pronunciations is extracted from these alignments for each word in the training set transcriptions.
- A heuristic pruning algorithm is applied to reduce the number of pronunciation variants for each word in the dictionary.
- Grapheme-to-phoneme (G2P) conversion is used to obtain pronunciations for words that appear in the test data but not in the training set [146].
- The HTK toolkit [138] is used to train hidden Markov models of sub-word units and perform speech recognition.

All experiments are carried out on the TIMIT corpus as described in Section 4.3. TIMIT is not well suited for word-based ASR, and in future other datasets should be considered. TIMIT does however allow a comparison with a system using manually-produced phone transcriptions, which are typically not available for other corpora. In the following section, the procedure described above will be applied.

## 7.2 Creating a pronunciation dictionary

### 7.2.1 Initial SWU alignments

In Chapter 6 it was reported that  $K$  clusters are generated by the multi-stage agglomerative hierarchical clustering algorithm. Each of these clusters is assumed to represent a speech segment called a sub-word unit (SWU). Clusters were labelled  $u_1, u_2, \dots, u_K$  so that  $u_k$  is a distinct sub-word unit. All 3696 SI and SX training sentences of the TIMIT were considered for creating the pronunciation dictionaries.

Although the TIMIT corpus has labelled segments, we will employ clusters that are automatically determined and do not use this available ground truth. We assume that only the sentence boundaries of the acoustic data and the corresponding orthography are known. This information will be used to automatically obtain cluster labels aligned with each sentence. This alignment of orthography and SWUs constitutes a sentence-level dictionary. Figure 7.1 shows an example of SWUs that have been aligned with the first two SI TIMIT training sentences.

Each of the 3696 SI and SX TIMIT training sentences will now have a corresponding series of sub-word units. A simple strategy is used to guess the



```

EVEN THEN IF SHE TOOK ONE STEP FORWARD HE COULD CATCH HER:u437 u673 u21 u644 u120
u667 u252 u457 u80 u698 u897 u247 u411 u211 u416 u273 u204 u22 u490 u776 u210 u835
u267 u221 u100 u798 u441 u247 u764 u95 u517 u524 u475 u543
OR BORROW SOME MONEY FROM SOMEONE AND GO HOME BY BUS: u7 u273 u278 u779 u298 u996
u395 u847 u131 u1009 u66 u644 u421 u4 u859 u88 u316 u29 u192 u459 u21 u160 u743
u339 u401 u370 u72 u684 u98 u777 u1014 u871

```

**Figure 7.1:** The first two entries of the TIMIT sentence-level dictionary.

word boundaries from these data. Each word is aligned with a certain number of SWUs, where this number is determined by the length (in graphemes) of the word. Equation 7.1 indicates the number of SWUs that were aligned with each word.

$$N_{SWU}(w_t) = \frac{length(w_t)}{length(sentence)} \times N_{SWU}(sentence) \quad (7.1)$$

In Equation 7.1,  $N_{SWU}(w_t)$  is the number of sub-word units allocated to word ( $w_t$ ) at position  $t$  in a sentence,  $length(w_t)$  and  $length(sentence)$  are the number of graphemes in word  $w_t$  and in the sentence as a whole (excluding spaces) respectively.  $N_{SWU}(sentence)$  denotes the total number of sub-word units for the sentence. Once the whole number proportions of SWUs have been allocated, the remainder are distributed according to word lengths. This results in an initial dictionary as illustrated in Figure 7.2.

```

EVEN u437 u673 u21
THEN u644 u120 u667
IF u252
SHE u457 u80
TOOK u698 u897 u247
ONE u411 u211
STEP u416 u273 u204
FORWARD u22 u490 u776 u210 u835
HE u267 u221
COULD u100 u798 u441 u247
CATCH u764 u95 u517 u524
HER u475 u543
OR u7
BORROW u273 u278 u779 u298 u996
SOME u395 u847 u131
MONEY u1009 u66 u644 u421
FROM u4 u859 u88
SOMEONE u316 u29 u192 u459 u21
AND u160 u743
GO u339 u401
HOME u370 u72 u684
BY u98 u777
BUS u1014 u871

```

**Figure 7.2:** Initial dictionary showing the entries from the first two sentences of the TIMIT training set as indicated in Figure 7.1.

## 7.2.2 Realignment of initial dictionary

The initial dictionary obtained above is a coarse representation of word pronunciations. In addition, many words are repeated throughout the training data, thereby rendering multiple pronunciation variants for each word. This repetition can be taken advantage of by using the Viterbi algorithm to realign sub-word units with the words in the orthographic transcriptions using hidden Markov models. In this approach, the word sequence forming a sentence is considered as a hidden Markov model (HMM) where each word is represented by a state. The SWUs aligned with the sentence correspond to the observation sequence  $\mathbf{O} = o_1, o_2, \dots, o_M$  where  $M$  is the number of SWUs in the sentence in question. The sequence of sub-word units aligned with each word in the initial alignment is assumed to be the observation sequence generated by the state corresponding to the word in question.

The HMM observation matrix  $\mathbf{B}$  is obtained from the initial dictionary according to Equation 7.2:

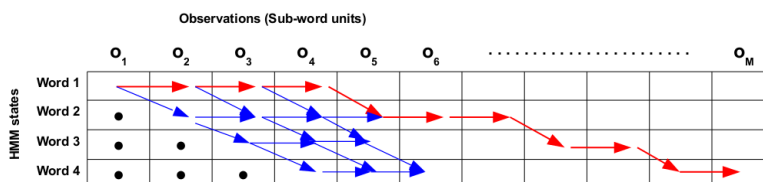
$$b_{ij} = Pr(o_j|w_i) = \frac{\text{Frequency of } o_j \text{ aligning with } w_i}{\text{Number of SWUs aligned with } w_i} \quad (7.2)$$

where  $b_{ij}$  is the probability of producing observation  $o_j$  from word  $w_i$ .

Equation 7.2 indicates that, to obtain values  $b_{ij}$ , the number of times a particular SWU is aligned with each word is counted and divided by the total number of SWUs aligned with the same word across the whole dictionary.

Using these observation probabilities, the Viterbi algorithm is used to realign the sub-word units with the words in the orthographic transcription. This updated alignment is used to obtain an updated dictionary, which is used to update  $\mathbf{B}$  and again align the sub-word units with the orthography. The process is repeated until the dictionary no longer changes.

Figure 7.3 illustrates such an alignment between a sentence and the sub-word units. In this example the sentence contains 4 words, each of which is represented by a single HMM state. The sentence is also represented by a sequence of  $M$  sub-word units, which constitute the observations  $o_1, \dots, o_M$ .



**Figure 7.3:** The trellis structure used to find the optimal alignment between the sequence of SWUs and the sequence of words in a sentence. The locus of red arrows indicates the optimal alignment path.



### 7.2.3 Pruning the dictionary

The dictionary obtained in the previous step will in general contain multiple pronunciations per word. Pronunciations may be as many as the frequency of the word in the training data. For example, the word "catch" appears 8 times in the TIMIT training sentences. When using the clusters from iteration 4 of DTW-based MAHC with  $P = 8$  subsets (Section 6.5), this word is found to have 8 different pronunciations after the Viterbi alignment. As already mentioned in the introduction of this chapter, variation in pronunciation for automatic speech recognition can reduce its performance. For this reason a pruning process is required to reduce pronunciation variants of a word.

Different heuristic approaches can be used to prune a dictionary to fewer pronunciations per word. Hernández-Ábrego *et al* [143] propose a consensus method to prune unneeded pronunciations. In their method, plausible pronunciations to be kept in the dictionary are determined using consensus for each SWU. Hain [147] also describes a method of reducing pronunciation variants in a pronunciation dictionary. In this approach, frequency of occurrence of a pronunciation in the training data is used and the entries of highest frequencies are retained.

The approaches described above have some similarity to the one proposed by Goussard [145] in terms of trying to find pronunciations of a particular word that occur frequently. We chose Goussard's pruning method because the data used by this author also results from the TIMIT corpus. The probability of each pronunciation is computed with respect to the total number of pronunciations seen for the word in question. The pronunciations are sorted in descending order of probability while computing the cumulative probability sum. When this sum reaches a threshold value, the remaining pronunciations are discarded. We used a threshold of 0.7 since this was found to be optimal in [145]. However, in practice many pronunciation sequences are different and occur only once. In such cases a single pronunciation is selected at random. In all other cases the top three pronunciation variants are selected.

### 7.2.4 Adding missing words to the dictionary

The TIMIT test set has words that do not appear in the training set and therefore do not have SWU transcriptions. We estimate the pronunciation of the missing words using a trainable grapheme-to-phoneme converter [146]. Grapheme-to-phoneme (G2P) conversion is a process of finding a pronunciation directly from the word orthography. The G2P converter employs statistical models to learn the joint-sequence models using the alignments between graphemes and phonemes (in our case SWUs). A pruned dictionary from the previous step is used to train the G2P models. The missing word pronunciation is estimated by computing the most likely pronunciation given the G2P models and the word orthography. The missing words and their newly deter-

mined pronunciations are appended to the pruned dictionary to yield the final dictionary used in the SWU acoustic model training.

### 7.2.5 TIMIT baseline dictionary

For benchmarking, the TIMIT phonetic transcriptions were aligned with each training sentence. The transcriptions exclude closures and pauses. Subsequently a pronunciation dictionary was induced using the strategy described in Subsections 7.2.1 through to 7.2.4. This resulted in a pronunciation dictionary in terms of 41 TIMIT phones that will be used as a benchmark during ASR evaluation.

## 7.3 ASR evaluation of MAHC-based pronunciations

The final dictionary was evaluated in terms of automatic speech recognition (ASR) performance by using the scheme suggested in the HTK book tutorial on creating monophone hidden Markov models (HMMs) and performing recognition [137].

The language model used in our ASR experiments was trained on the Brown Corpus and all TIMIT SI and SX training sentences. The bigram model was developed using the SRILM language modelling toolkit [148].

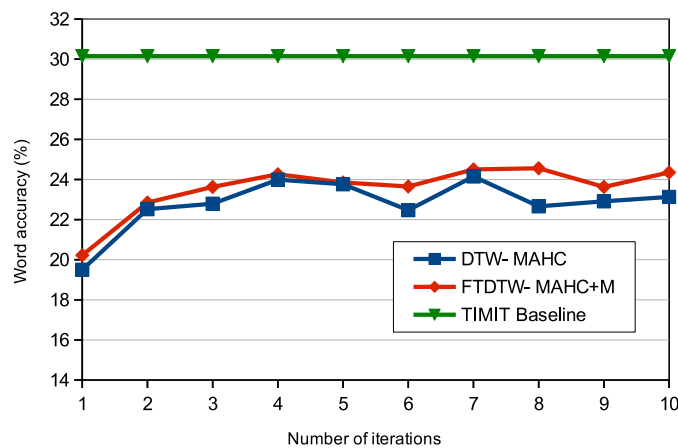
Acoustic data for training the HMMs was obtained from the 3696 SI and SX TIMIT training sentences where Mel-frequency cepstral coefficients (MFCCs) were used as features. Frames overlap by 10 milliseconds and for each frame a 13-dimensional feature vector is obtained consisting of 12 MFCCs and appended energy attribute. Finally, delta and acceleration coefficients are computed appended to obtain the final 39-dimensional feature vector. These feature sets are used for training the HMMs as well as during recognition. One monophone HMMs is created for each distinct SWU label. A dictionary obtained in Section 7.2 is used in the training of the HMMs.

Once the language model and HMMs have been trained, recognition is carried out using the 1344 sentences in the TIMIT SI and SX test set. Since speech recognition accuracy is still poor for the dictionaries we will induce, word accuracy will be used as a performance measure. As performance will hopefully improve in future, the somewhat more severe and generally accepted measure of word error rate (WER) can be used instead.

### 7.3.1 Recognition results

We considered three sets of experiments, (1) recognition associated with pronunciations derived from clusters obtained by MAHC with 8 subsets ( $P_0 = 8$ ),

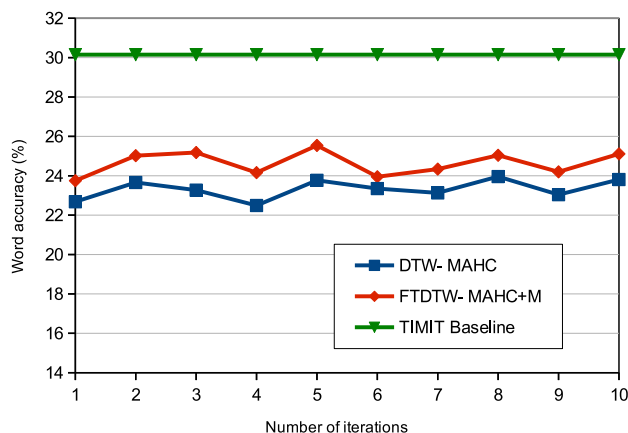
(2) MAHC with  $P_0 = 10$  and (3) MAHC with  $P_0 = 15$ . For each set of experiments, dictionaries were created using the clusters resulting from 10 iterations of multi-stage agglomerative hierarchical clustering (MAHC or MAHC+M). Since it was seen in Chapter 6 that the performance of DTW-based MAHC+M and FTDTW-based MAHC+M are very close, we chose not to include the former in ASR evaluations. Recognition performance when using the TIMIT baseline dictionary are included as a benchmark in all experiments. The first set of ASR experiments used dictionaries induced from the clusters obtained by MAHC with  $P = 8$  subsets and MAHC+M with  $P_0 = 8$  subsets, as shown in Figure 7.4.



**Figure 7.4:** Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets  $P_0 = 8$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline.

Figure 7.4 shows that the word accuracy when using the TIMIT baseline dictionary is substantially higher than the accuracy achieved using the automatically induced dictionaries. The FTDTW-based MAHC+M algorithm provides slightly better performance than DTW-based MAHC. A comparison between Figure 7.4 and Figure 6.14 in Chapter 6, reveals that this is consistent with the F-Measures. In terms of clustering evaluation, one can therefore observe consistency in terms of performance of the two algorithms. In terms of useful ASR results, the performance of both MAHC systems is still fairly poor compared to the TIMIT baseline.

Figure 7.5 presents ASR performance when using dictionaries induced from the clusters obtained with DTW-based MAHC and  $P_0 = 10$  initial subsets. We see that these results are consistent with those obtained for  $P_0 = 8$  subsets in Figure 7.4 .

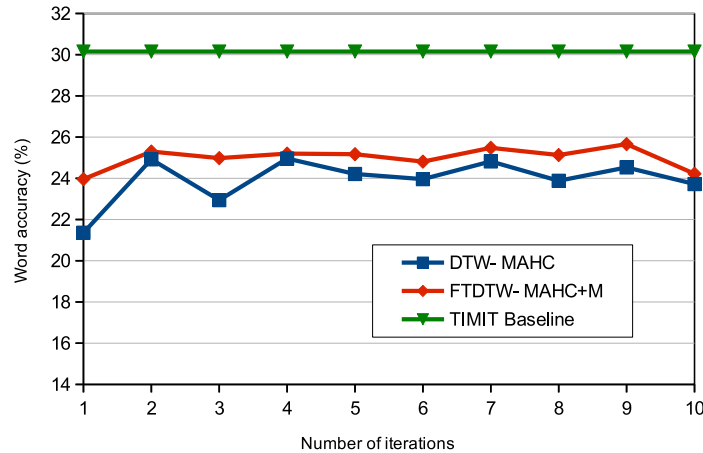


**Figure 7.5:** Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets  $P_0 = 10$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline.

The MAHC results are again substantially below those of the TIMIT baseline in terms of word accuracy. It is observed that the performance achieved with dictionaries induced from clusters obtained by FTDTW-based MAHC+M is better than that achieved with DTW-based MAHC. Also Figure 7.6 shows that FTDTW-based MAHC+M consistently leads to better word accuracies than DTW-based MAHC. Again, comparing it with Figure 6.20 in Chapter 6, we observe that the F-Measure for FTDTW-based MAHC+M almost always dominated the performance.

The highest word accuracy of 25.7% is achieved when using the clusters achieved at iteration 9 when employing the FTDTW-based MAHC+M algorithm in Figure 7.6. The highest accuracy in Figure 7.5 is 25.5% in iteration 5 while the highest accuracy in Figure 7.4 is 24.5% in iteration 7. While all these values are substantially lower than the TIMIT baseline of 30.2%, they reflect a promising start for the segment-and-cluster paradigm for automatic dictionary induction. Many research questions relating to SWU dictionary induction, such as the amount of training data, the choice of acoustic features, language modelling and word boundary determination could not be investigated as part of this project and remain the subject of future work.

Table 7.1 shows that word accuracy increases gradually as the number of initial subsets increases. On average, the FTDTW-based MAHC+M performs better than the DTW-based MAHC. Since the MAHC+M algorithm was introduced as a modification to guarantee the avoidance of potential  $O(N^2)$  memory complexity, the results in Table 7.1 indicate that introducing memory size management does not only solve the complexity but also slightly improves word accuracy.



**Figure 7.6:** Word accuracy achieved for systems trained using dictionaries induced automatically from the clusters obtained with DTW-based MAHC and FTDTW-based MAHC+M with an initial number of subsets  $P_0 = 8$ . Performance when using a dictionary induced from the TIMIT reference phone transcriptions is included as a baseline.

No. of subsets	DTW-based MAHC	FTDTW-based MAHC+M
8	22.9	23.6
10	23.3	24.6
15	23.9	25.0

**Table 7.1:** Average word recognition rate in percentages (%) for three sets of experiments where the number of initial subsets  $P_0$  was 8, 10 and 15.

## 7.4 Summary and conclusion

In this chapter a basic process of dictionary induction from clusters generated by MAHC algorithms has been described. The induction process starts with the rough estimation of word boundaries given sequences of sub-word units (SWUs) per sentence. This process produces an initial dictionary which is further refined using the Viterbi algorithm. The resulting updated dictionary has multiple pronunciations and is heuristically pruned to allow no more than 3 pronunciations per word. Grapheme-to-phoneme (G2P) conversion is used to estimate pronunciations of missing words from this pruned dictionary. Acoustic models were trained using HTK and the induced dictionary. Dictionaries were induced from the clusters obtained by DTW-based MAHC and FTDTW-based MAHC+M algorithms. Recognition performance in terms of word accuracy show that pronunciations induced from clusters obtained by MAHC+M led to slightly better performance than clusters obtained by MAHC. Although the word accuracies are low, we have shown that dictionaries induced from au-

tomatically clustered audio can potentially be used in ASR applications. Many research questions emanating from our results remain for future investigation.

## Chapter 8

# Summary, Conclusions and Recommendations

### 8.1 Summary and conclusions

The main objective of this dissertation has been to develop a clustering method suitable for partitioning a very large pool of acoustic speech segments into groups of similar sounds to be used in acoustic modelling for application in automatic speech recognition (ASR). Because the targeted speech application is for under-resourced languages, this was achieved by employing unsupervised clustering.

Clustering algorithms can be classified as hierarchical or partitional. Hierarchical clustering methods can be further divided into agglomerative and divisive algorithms. Agglomerative hierarchical clustering (AHC) was found to be a suitable approach to the clustering of speech segments because the number of clusters can be automatically determined. This is important because, for under-resourced languages, we generally have no linguistic information with which to motivate the number of distinct sounds used.

AHC determines pairwise distance between data objects to synthesise a hierarchical clustering structure called a dendrogram. This structure uses linkage distances to gradually create more and more refined clusters. Cluster merging is performed using linkage distances that are summarised by the Lance-Williams formulations. The choice of pairwise distance measure and also the linkage methods were experimentally determined in Chapter 4.

For the evaluation of clustering results, the F-Measure and in some cases the normalised mutual information (NMI) was chosen as an external validation metric. The L method was chosen to automatically determine the number of clusters from the dendrogram.

Experimental evaluation was performed using speech segments extracted from the TIMIT and SADD corpora. Mel-frequency cepstral coefficients (MFCCs) and perceptual linear prediction (PLP) parameterisations were used

as features. Dynamic time warping (DTW) was identified as a suitable measure of pairwise similarity between speech segments. Using the F-Measure it was found that:

- The Manhattan distance was a preferable local distance measure for integration into DTW.
- The Ward linkage method was the best choice for AHC.

In Chapter 4, a new variant of DTW called feature trajectory DTW (FT-DTW) was proposed. It was shown experimentally that FTDTW improved clustering performance for both TIMIT and SADD datasets and for both MFCC and PLP parameterisations.

Because AHC has a storage and runtime complexity of  $O(N^2)$ , it quickly becomes impractical for large datasets. To address this, a new algorithm named multi-stage agglomerative hierarchical clustering (MAHC) was proposed in Chapter 5. The MAHC algorithm splits data into subsets and applies AHC to each subset. Following this, clusters are merged and re-split, making the process iterative. This process is repeated until convergence. Experimental evaluation in Chapter 5 revealed the following:

- The performance of MAHC matches and sometimes improves on that of AHC. Hence there is no performance penalty for the data splits used by MAHC.
- After convergence, the final number of MAHC clusters approximately equates to the sum of clusters obtained from each subset. This relation permitted automatic determination of the number of clusters, which is for example very useful when clustering speech segments in under-resourced languages.
- MAHC performs better than spectral clustering under the same experimental conditions.
- MAHC is suitable for execution on parallel computing hardware, and as such can be much faster than AHC.

It was however also observed that sometimes one or more subsets can grow and become too large during MAHC. In this case the  $O(N^2)$  complexity which is a characteristic of conventional AHC resurfaces. To address this problem, a control mechanism termed memory size management was added to MAHC in Chapter 6. This extended algorithm, referred to as MAHC with memory size management (MAHC+M), ensured that subsets that grow more than a predefined threshold were split further during the iterative re-clustering process of MAHC. Experiments in Chapter 6 compared the performance of MAHC and MAHC+M using both DTW and FTDTW as pairwise similarity measures. The results revealed the following:



- The execution time of MAHC+M is potentially shorter than that of MAHC. This can be attributed to the guaranteed maximum subset size.
- The performance of MAHC+M generally matches that of MAHC for small and large datasets considered. There is therefore no performance penalty for the introduction of the memory size management.
- FTDTW-based MAHC+M generally performs better than DTW-based MAHC+M, MAHC and AHC. Therefore, FTDTW offers a superior alternative to DTW for this task.
- Confusion matrices strongly indicate that MAHC and MAHC+M produce clusters that correspond to the basephones of TIMIT which suggests that clusters can be used in sub-word unit modelling.

Chapter 7 presented a process of automatically inducing a pronunciation dictionary from the clusters produced by the MAHC algorithms. Cluster labels obtained in Chapter 7 were used as sub-word units (SWUs) in a Viterbi alignment that was used to refine a rather coarse initial word and pronunciation alignment. A large number of pronunciations per word was reduced by heuristically pruning the resulting dictionary. The pronunciations of missing words were extrapolated by using a grapheme-to-phoneme (G2P) conversion.

Dictionaries were evaluated on a HTK-based ASR application. The results showed that, although recognition accuracies were very low, FTDTW-based MAHC+M consistently performs slightly better than DTW-based MAHC in terms of word accuracy in all reported experiments.

### 8.1.1 Contributions

In conclusion, the contributions of the presented work are:

- A new iterative hierarchical clustering algorithm called MAHC was developed. This algorithm has the particular advantage of being applied to very large datasets due to its divide-and-conquer approach, and its ability to take advantage of parallel computing hardware.
- An improved MAHC algorithm called MAHC+M was also developed to ensure better management of cluster sizes so that the  $O(N^2)$  complexity problem is contained.
- The MAHC+M algorithm was applied to datasets of audio speech segments to automatically determine the number of clusters rendering it suitable for the under-resourced languages sub-word modelling.
- A variant of dynamic time warping, termed feature trajectory dynamic time warping (FTDTW) was proposed and shown to outperform standard DTW when applied to the clustering of speech segments.

- Analyses of the clusters produced by MAHC+M have demonstrated that they strongly resemble human-labelled phonetic classes. Furthermore, the MAHC+M clusters could be used to automatically induce a pronunciation dictionary as required by an automatic speech recognition system. This indicates that the segment-and-cluster approach has potential subject to further investigation.

## 8.2 Recommendations for future work

This study can be continued by considering the following further directions:

- More suitable features for representing the speech segments should be investigated in depth. For example, features extracted by auto-encoder deep neural networks, which can be trained in an unsupervised way and are therefore applicable to under-resourced languages, have shown promise in other ASR research.
- More computationally efficient pairwise distances suitable for speech segments should be considered. For example, word embedding models use a high dimensional but fixed vector space and have been successful in representing similarities in some natural language processing (NLP) applications.
- The FTDTW algorithm should be evaluated more rigorously, for example using different corpora and different tasks such as spoken term detection, before it can be regarded as a stable variant of the DTW.
- Now that the MAHC+M algorithm has been developed, speech corpora other than TIMIT should be used to verify its robustness.
- The induction of a pronunciation dictionary from the clusters was given a straightforward and brief consideration in Chapter 7. However this topic deserves consideration in much greater depth, and improvements in this regard may lead to improved ASR performance.

# List of References

- [1] Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G. and Pallett, D.S.: DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.
- [2] Besacier, L., Barnard, E., Karpov, A. and Schultz, T.: Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, vol. 56, pp. 85–100, 2014.
- [3] Livescu, K., Fosler-Lussier, E. and F.Metze: Subword modeling for automatic speech recognition: Past, present, and emerging approaches. *IEEE Signal Processing Magazine*, pp. 44–57, 2012.
- [4] Davel, M. and Barnard, E.: Bootstrapping pronunciation dictionaries: practical issues. In: *Proceedings of Interspeech*. Lisbon, Portugal, September 2005.
- [5] Davel, M., Heerden, C.V., Kleyhans, N. and Barnard, E.: Efficient harvesting of internet audio for resource-scarce ASR. In: *Proceedings of Interspeech*. Florence, Italy, August 2011.
- [6] Davel, M. and Barnard, E.: Pronunciation prediction with Default&Refine. *Computer Speech and Language*, vol. 22, no. 4, pp. 374–393, 2008.
- [7] Singh, R., Raj, B. and Stern, R.: Automatic generation of subword units for speech recognition systems. *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 89–99, 2002.
- [8] Wang, H., Lee, T., Leung, C.-C., Ma, B. and Li, H.: Acoustic segment modeling with spectral clustering methods. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 2, pp. 264–277, 2015.
- [9] Bacchiani, M. and Ostendorf, M.: Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, vol. 18, no. 4, pp. 375–395, 1999.
- [10] Tolba, H. and O’Shaughnessy, D.: Robust automatic continuous speech recognition based on a voiced-unvoiced decision. In: *Proceedings of ICSLP 1998*. Sydney, 1998.
- [11] Ahmadi, S. and Spanias, A.: Cepstrum-based pitch detection using a new statistical V/UV classification algorithm. *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333–338, 1999.

- [12] Zolnay, A., Schluter, R. and Ney, H.: Extraction methods of voicing feature for robust speech recognition. In: *Proceedings of Eurospeech*. 2003.
- [13] van Vuuren, V. and Niesler, T.: Automatic segmentation of TIMIT by dynamic programming. In: *Proceedings of Pattern Recognition Association of South Africa*. Pretoria, South Africa, November 2012.
- [14] Aversano, G., Esposito, A. and Marinaro, M.: A new text-independent method for phoneme segmentation. In: *Proceedings of 44th IEEE Midwest Symposium on Circuits Systems*. 2001.
- [15] Murthy, H. and Gadde, V.: The modified group delay function and its application to phoneme recognition. In: *Proceedings of ICASSP*, vol. 1, pp. 68–71. 2003.
- [16] Golipour, L. and O’Shaughnessy, D.: A new approach for phoneme segmentation of speech signals. In: *Proceedings of Interspeech*. Antwerp, Belgium, 2007.
- [17] Bosch, L. and Cranen, B.: A computational model for unsupervised word discovery. In: *Proceedings of Interspeech*. Antwerp, Belgium, 2007.
- [18] Park, A. and Glass, J.: Towards unsupervised pattern discovery in speech. In: *Proceedings of ASRU*. 2005.
- [19] Gajjar, M., Govindarajan, R. and Sreenivas, T.: Online unsupervised pattern discovery in speech using parallelization. In: *Proceedings of Interspeech*. Brisbane, Australia, 2008.
- [20] Muscariello, A., Gravier, G. and Bimbot, F.: Audio keyword extraction by unsupervised word discovery. In: *Proceedings of Interspeech*. Brighton, United Kingdom, 2009.
- [21] Imperl, B., Kacic, Z., Horvat, B. and Zgank, A.: Clustering of triphones using phoneme similarity estimation for the definition of a multilingual set of triphones. *Speech Communication*, vol. 39, no. 4, pp. 353–366, 2003.
- [22] Jain, A.: Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [23] Fung, G.: A comprehensive overview of basic clustering algorithms. 2001.
- [24] Legendre, P. and Legendre, L.: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
- [25] Bezdek, J.C., Ehrlich, R. and Full, W.: FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [26] Filippone, M., Camastra, F., Masulli, F. and Rovetta, S.: A survey of kernel and spectral methods for clustering. *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.

- [27] Jain, A.K., Murty, M.N. and Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [28] Murtagh, F. and Contreras, P.: Methods of hierarchical clustering. *arXiv preprint arXiv:1105.0121*, 2011.
- [29] Xu, R. and Wunsch, D.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [30] Wu, J., Xiong, H. and Chen, J.: Towards understanding hierarchical clustering: A data distribution perspective. *Neurocomputing*, vol. 72, no. 10-12, pp. 2319–2330, June 2009.
- [31] Fraley, C. and Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, vol. 41, pp. 578–588, 1998.
- [32] Varshavsky, R., Horn, D. and Linial, M.: Global considerations in hierarchical clustering reveal meaningful patterns in data. *PloS one*, vol. 3, no. 5, p. e2247, 2008.
- [33] Landau, S. and Ster, I.C.: *Cluster analysis: overview*. Elsevier, 2010.
- [34] Legendre, P. and Legendre, L.: *Numerical Ecology, Volume 24, (Developments in Environmental Modelling)*. Elsevier Science, 1998.
- [35] Jain, A.K. and Dubes, R.C.: *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [36] K.Halberstadt, A. and Glass, J.: Heterogeneous acoustic measurements for phonetic classification. In: *Proceedings of Eurospeech 97*. 1997.
- [37] Svendsen, T., Paliwal, K., Harborg, E. and Husoy, P.: An improved sub-word based speech recognizer. In: *Proceedings of ICASSP*, pp. 729–732. 1989.
- [38] Holter, T. and Svendsen, T.: Incorporating linguistic knowledge and automatic baseform generation in acoustic subword unit based speech recognition. In: *Proceedings of Eurospeech*, pp. 1159–1162. 1997.
- [39] Paliwal, K.: Lexicon-building methods for an acoustic sub-word based speech recognizer. In: *Proceedings of ICASSP*, pp. 108–111. 1990.
- [40] Mak, B. and Barnard, E.: Phone clustering using the Bhattacharyya distance. In: *Proceedings of ICSLP*, pp. 2005–2008. 1996.
- [41] Davis, S. and Mermelstein, P.: Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions of Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

- [42] Wang, H., Lee, T., Leung, C., Ma, B. and Li, H.: Unsupervised mining of acoustic subword units with segment-level Gaussian posteriors. In: *Proceedings of Interspeech*, pp. 2297–2301. 2013.
- [43] Kamper, H., Jansen, A., King, S. and Goldwater, S.: Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings. In: *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pp. 100–105. 2014.
- [44] Zezula, P., Amato, G., Dohnal, V. and Batko, M.: *Similarity Search: The Metric Space Approach*. Springer Science + Business Media, Inc., 233 Spring Street, New York, NY 10013, USA, 2006.
- [45] Myers, C., Rabiner, L. and Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 623–635, 1980.
- [46] Yu, F., Dong, K., Chen, F., Jiang, Y. and Zeng, W.: Clustering time series with granular dynamic time warping method. In: *Proceedings of the 2007 IEEE International Conference on Granular Computing*, GRC '07, pp. 393–398. Washington, DC, USA, 2007.
- [47] Manning, C.D. and Raghavan, P.: *Introduction to Information Retrieval*. Cambridge University Press, New York, USA, 2008.
- [48] Kopat, S. and M., E.: Review and comparative study of clustering techniques. *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 805–812, 2014.
- [49] Linde, Y., Buzo, A. and Gray, R.M.: An algorithm for vector quantizer design. *IEEE Transactions on Communications*, vol. 28, pp. 84–95, 1980.
- [50] Lee, C., Soong, F. and Juang, B.: A segment model based approach to speech recognition. In: *Proceedings of ICASSP*, pp. 501–504. 1988.
- [51] Kamper, H., Livescu, K. and Goldwater, S.: An embedded segmental k-means model for unsupervised segmentation and clustering of speech. *arXiv preprint arXiv:1703.08135*, 2017.
- [52] Von Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [53] Chen, X. and Cai, D.: Large scale spectral clustering with landmark-based representation. In: *Proceedings of AAAI*, vol. 5, p. 14. 2011.
- [54] X.Vinh, N., Epps, J. and Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalisation and correction for chance. *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

- [55] Shi, J. and J. Malik: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.
- [56] Sahbi, H.: A particular Gaussian mixture model for clustering and its application to image retrieval. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 12, no. 7, pp. 667–676, 2008.
- [57] Kamper, H., Jansen, A. and Goldwater, S.: Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 4, pp. 669–679, 2016.
- [58] Bacchiani, M. and Ostendorf, M.: Using automatically-derived acoustic subword units in large vocabulary speech recognition. In: *Fifth International Conference on Spoken Language Processing*. 1998.
- [59] Bacchiani, M., Ostendorf, M., Sagisaka, Y. and Paliwal, K.: Design of a speech recognition system based on non-uniform segmental units. In: *Proceedings of ICASSP*, vol. 1, pp. 443–446. 1996.
- [60] Kleinberg, J.: An impossibility theorem for clustering. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 446–453. 2002.
- [61] Amigo, E., Gonzalo, J., Artiles, J. and Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.
- [62] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. and Wu, S.: Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.
- [63] Desgraupes, B.: Clustering indices. *University of Paris Ouest-Lab Modal’X*, vol. 1, p. 34, 2013.
- [64] Vinh, N.X., Epps, J. and Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
- [65] Rosenberg, A. and Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: *Proceedings of EMNLP-CoNLL*, pp. 410–420. 2007.
- [66] Hubert, L. and Arabie, P.: Comparing partitions. *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [67] Yeung, K. and Ruzzo, W.: Details of the adjusted Rand index and clustering algorithms. *Bioinformatics*, vol. 17, pp. 763–774, 2001.



- [68] Santos, J. and Embrechts, M.: On the use of the adjusted rand index as a metric for evaluating supervised classification. In: *Proceedings of the 19th of the 19th International Conference on Artificial Neural Networks*, pp. 175–184. Berlin, 2009.
- [69] Strehl, A. and Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, vol. 3, no. Dec, pp. 583–617, 2002.
- [70] Larsen, B. and Aone, C.: Fast and effective text mining using linear-time document clustering. In: *Proceedings of the fifth ACM SIGKDD*, pp. 16–22. New York, USA, 1999.
- [71] Milligan, G.W. and Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.
- [72] Caliński, T. and Harabasz, J.: A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
- [73] Halkidi, M., Batistakis, Y. and Vazirgiannis, M.: Clustering validity checking methods: part ii. *ACM Sigmod Record*, vol. 31, no. 3, pp. 19–27, 2002.
- [74] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [75] Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.
- [76] Davies, D.L. and Bouldin, D.W.: A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 2, pp. 224–227, 1979.
- [77] Starczewski, A.: A new validity index for crisp clusters. *Pattern Analysis and Applications*, vol. 20, no. 3, pp. 687–700, 2017.
- [78] Sugar, C.A. and James, G.M.: Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, 2003.
- [79] Tibshirani, R., Walther, G. and Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.
- [80] Yan, M. and Ye, K.: Determining the number of clusters using the weighted gap statistic. *Biometrics*, vol. 63, no. 4, pp. 1031–1037, 2007.
- [81] Salvador, S. and Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '04*, pp. 576–584. 2004.



- [82] Bombrun, L., Vasile, G., Gay, M. and Totir, F.: Hierarchical segmentation of polarimetric SAR images using heterogeneous clutter models. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 726–737, 2011.
- [83] Zhang, T., Ramakrishnan, R. and Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: *ACM SIGMOD Record*, vol. 25, pp. 103–114. 1996.
- [84] Shirخورshidi, A.S., Aghabozorgi, S. and Wah, T.Y.: A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, vol. 10, no. 12, p. e0144059, 2015.
- [85] Bouguettaya, A., Yu, Q., Liu, X., Zhou, X. and Song, A.: Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.
- [86] Gobo, G., Garcia, D., Santamaria, E., Moran, J.A., Malenchon, J. and Monzo, C.: Modeling students' activity in online discussion forums: a strategy based on time series and agglomerative hierarchical clustering. In: *Educational Data Mining*, pp. 253–258. 2011.
- [87] Yim, O. and Ramdeen, K.T.: Hierarchical cluster analysis: comparison of three linkage measures and application to psychological data. *The Quantitative Methods for Psychology*, vol. 11, pp. 8–21, 2015.
- [88] Müllner, D.: Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.
- [89] Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.
- [90] Sneath, P.H., Sokal, R.R. *et al.*: *Numerical taxonomy. The principles and practice of numerical classification*. Freeman, San Francisco, California, 1973.
- [91] Murtagh, F. and Contreras, P.: Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
- [92] Lance, G. and Williams, W.: A general theory of classificatory sorting strategies 1. hierarchical systems. *The Computer Journal*, vol. 9, no. 4, pp. 373–380, 1967.
- [93] McQuitty, L.L.: Elementary linkage analysis for isolating orthogonal and oblique types and typical relevancies. *Educational and Psychological Measurement*, vol. 17, no. 2, pp. 207–229, 1957.
- [94] Sneath, P.: The application of computers to taxonomy. *Journal of General Microbiology*, vol. 17, pp. 201–226, 1957.
- [95] Rohlf, F.: 12 single link clustering algorithms. *Handbook of Statistics*, vol. 2, pp. 267–284, 1982.

- [96] Blashfield, R.K.: Mixture model tests of cluster analysis: Accuracy of four agglomerative hierarchical methods. *Psychological Bulletin*, vol. 83, no. 3, p. 377, 1976.
- [97] Murtagh, F. and Contreras, P.: Algorithms for hierarchical clustering: an overview II. *WIREs: Data Mining and Knowledge Discovery*, vol. 7, no. 6, pp. e1219–n/a, 2017.
- [98] Day, W. and Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, vol. 1, no. 1, pp. 7–24, 1984.
- [99] Defays, D.: An efficient algorithm for a complete link method. *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 1977.
- [100] Sokal, R.R. and Michener, C.D.: A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.
- [101] Rani, Y. and Rohil, H.: A study of hierarchical clustering algorithm. *International Journal of Information and Computation Technology*, vol. 3, no. 10, pp. 1115–1122, 2013.
- [102] Lee, A. and Willcox, B.: Minkowski generalizations of ward’s method in hierarchical clustering. *Journal of Classification*, vol. 31, no. 2, pp. 194–218, 2014.
- [103] Gower, J.C.: A comparison of some methods of cluster analysis. *Biometrics*, pp. 623–637, 1967.
- [104] Ward, Joe H., J.: Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.
- [105] Murtagh, F. and Legendre, P.: Ward’s hierarchical agglomerative clustering method: Which algorithms implement Ward’s criterion? *Journal of Classification*, vol. 31, no. 3, pp. 274–295, 2014.
- [106] Sun, L. and Korhonen, A.: Hierarchical verb clustering using graph factorization. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1023–1033. 2011.
- [107] Rincon Sanchez, F., Johnson, B., Crossa, J. and Taba, S.: Cluster analysis, an approach to sampling variability in maize accessions. *Maydica*, vol. 41, pp. 307–316, 1996.
- [108] Saraçlı, S., Doğan, N. and Doğan, İ.: Comparison of hierarchical cluster analysis method by cophenetic correlation. *Journal of Inequalities and Applications*, vol. 2013, no. 1, p. 203, 2013.

- [109] Morlini, I. and Zani, S.: Dissimilarity and similarity measures for comparing dendrograms and their applications. *Advances in Data Analysis and Classification*, vol. 6, no. 2, pp. 85–105, 2012.
- [110] Hands, S. and Everitt, B.: A Monte Carlo study of the recovery of cluster structure in binary data by hierarchical clustering techniques. *Multivariate Behavioral Research*, vol. 22, no. 2, pp. 235–243, 1987.
- [111] Ferreira, L. and Hitchcock, D.B.: A comparison of hierarchical methods for clustering functional data. *Communications in Statistics-Simulation and Computation*, vol. 38, no. 9, pp. 1925–1949, 2009.
- [112] Milligan, G.W.: An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, vol. 45, no. 3, pp. 325–342, 1980.
- [113] Kuiper, F.K. and Fisher, L.: 391: A monte carlo comparison of six clustering procedures. *Biometrics*, vol. 31, pp. 777–783, 1975.
- [114] Milligan, G.W. and Cooper, M.C.: A study of standardization of variables in cluster analysis. *Journal of Classification*, vol. 5, no. 2, pp. 181–204, 1988.
- [115] Narasimha Murty, M. and Krishna, G.: A computationally efficient technique for data-clustering. *Pattern Recognition*, vol. 12, no. 3, pp. 153–158, 1980.
- [116] Suresh Babu, V.: Optimal number of levels for a multilevel clustering method. *Pattern Recognition Letters*, vol. 11, no. 9, pp. 595–599, 1990.
- [117] Tang, C.-H., Huang, A.-C., Tsai, M.-F. and Wang, W.-J.: An efficient distributed hierarchical-clustering algorithm for large scale data. In: *IEEE International Computer Symposium (ICS), 2010*, pp. 869–874. 2010.
- [118] Cobo, G., García-Solórzano, D., Morán, J.A., Santamaría, E., Monzo, C. and Melenchón, J.: Using agglomerative hierarchical clustering to model learner participation profiles in online discussion forums. In: *Proceedings of ACM 2Nd International Conference on Learning Analytics and Knowledge*, pp. 248–251. Vancouver, British Columbia, Canada, 2012.
- [119] Soltanolkotabi, M., Candes, E.J. *et al.*: A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.
- [120] Cobo Rodríguez, G.: *Parameter-free agglomerative hierarchical clustering to model learners' activity in online discussion forums*. Ph.D. thesis, Universitat Oberta de Catalunya. Internet Interdisciplinary Institute (IN3), 2014.
- [121] Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

- [122] Muda, L., Begam, M. and Elamvazuthi, I.: Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083*, 2010.
- [123] Zhang, X., Sun, J. and Luo, Z.: One-against-all weighted dynamic time warping for language-independent and speaker-dependent speech recognition in adverse conditions. *PloS ONE*, vol. 9, no. 2, p. e85458, 2014.
- [124] Zhang, Y. and Glass, J.R.: Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In: *Proceedings of IEEE Automatic Speech Recognition & Understanding Workshop (ASRU)*, pp. 398–403. 2009.
- [125] Anguera, X.: Information retrieval-based dynamic time warping. In: *Proceedings of Interspeech*, pp. 1–5. 2013.
- [126] Lee, L.-S., Glass, J., Lee, H.-Y. and Chan, C.-A.: Spoken content retrieval—beyond cascading speech recognition with text retrieval. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 9, pp. 1389–1420, 2015.
- [127] Walter, O., Korthals, T., Haeb-Umbach, R. and Raj, B.: A hierarchical system for word discovery exploiting DTW-based initialization. In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 386–391. 2013.
- [128] Shanker, A.P. and Rajagopalan, A.: Off-line signature verification using DTW. *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1407–1414, 2007.
- [129] Keogh, E.J. and Pazzani, M.J.: Derivative dynamic time warping. In: *Proceedings of SDM*, vol. 1, pp. 5–7. SIAM, 2001.
- [130] Owens, F.J.: *Signal Processing of Speech*. The Macmillan Press Ltd, London, 1993.
- [131] Theodoridis, S. and Koutroumbas, K.: *Pattern Recognition*. 4th edn. Academic Press, 2008.
- [132] Henniger, O. and Müller, S.: Effects of time normalization on the accuracy of dynamic time warping. In: *Proceedings of the 1st IEEE Conference on Biometrics: Theory, Applications and Systems*, pp. 1–6. Washington DC, USA, 2007.
- [133] Sagayama, S., Matsuda, S., Nakai, M. and Shimodaira, H.: Asynchronous-transition HMM for acoustic modeling. In: *International Workshop on Acoustic Speech Recognition and Understanding*, pp. 99–102. 1999.
- [134] Lichman, M.: UCI machine learning repository. 2013. Available at: <http://archive.ics.uci.edu/ml>

- [135] Davis, S.B. and Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [136] Hermansky, H.: Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.
- [137] Young, S.J., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P.C.: *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.
- [138] Young, S.J.: *The HTK hidden Markov model toolkit: Design and philosophy*. Citeseer, 1993.
- [139] Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J. and Chang, E.Y.: Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.
- [140] Zhou, M., Li, H. and Weijnen, M.: *Contemporary Issues in Systems Science and Engineering*. Wiley, 2015.
- [141] Lopes, C. and Perdigão, F.: Phoneme recognition on the TIMIT database. In: *Speech Technologies*, chap. 14. InTech, 2011.
- [142] Strik, H. and Cucchiaroni, C.: Modeling pronunciation variation for ASR: a survey of the literature. *Speech Communication*, vol. 29, no. 2-4, pp. 225–246, 199.
- [143] Hernández-Ábrego, G., Olorenshaw, L., Tato, R. and Schaaf, T.: Dictionary refinements based on phonetic consensus and non-uniform pronunciation reduction. In: *Eighth International Conference on Spoken Language Processing*. 2004.
- [144] Hain, T.: Implicit modelling of pronunciation variation in automatic speech recognition. *Speech Communication*, vol. 46, no. 2, pp. 171–188, 2005.
- [145] Goussard, G.: *Unsupervised clustering of audio data for acoustic modelling in automatic speech recognition systems*. Master's thesis, Stellenbosch University, 2011.
- [146] Bisani, M. and Ney, H.: Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.
- [147] Hain, T.: Implicit pronunciation modelling in ASR. In: *Proceedings of ISCA ITRW PMLA*. 2002.
- [148] Stolcke, A.: SRILM – an extensible language modeling toolkit. In: *Proceeding of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pp. 901–904. 2002.