Towards natural speech acquisition: Incremental word learning with limited data

Irene Ayllón Clemente

Thesis submitted to the Faculty of Technology
of the Bielefeld University for obtaining the academic degree

Doctor of Engineering (Dr.-Ing.)

Supervisors: Dr. Ing. Martin Heckmann, HRI Europe GmbH
apl. Prof. Dr. Ing. Britta Wrede, Bielefeld University
Prof. Dr. Ing. Gerhard Sagerer, Bielefeld University

Date of opening of the doctoral examination procedure: 6^{th} November, 2013.
Date of display and acceptance for disputation: 17^{th} June, 2014.
Date of disputation and approval: 5^{th} September, 2014.

Acknowledgments

I would like to express my gratitude to the group of people that have accompanied me during the realization of my thesis for their support and motivation.

Starting with my family, I want to thank my dear grandparents, for their affection and patience with me along this time, my darling sister and father for being always proud of me and interested in the state of my project as well as my mother for reaffirming my decision to do a doctorate and teaching me to be determined and responsible with my work. A special mention is for my beloved husband for his support and daily motivation, especially for discussing my numerous ideas and theories that came to my mind along the day.

Furthermore, I thank my supervisors Martin Heckmann and Britta Wrede, from HRI Europe GmbH and the Bielefeld University respectively, for their interest in new approaches and methodologies, giving me technical facilities when realizing the thesis and the possibility for additional activities as the chance to organize a workshop in my field of research with different international recognized experts and letting me give practical courses about speech recognition in the faculty. Both were wonderful experiences and helped me to exchange ideas and collect experiences with the rest of experts and students. My gratitude to Gerhard Sagerer, although his supervision in the university could not be the whole period, is for his support and engagement in my project.

In HRI, my thanks to Frank Joublin are for inspiring me to consider ideas that seem unrealizable at first sight, not to be content with partial improvements. I also want to express my gratitude to other colleagues of the former CARL group as Tobias Rodermann, Samuel Ngouoko, Rujiao Yan, Claudius Gläser, Holger Brandl, Xavier Domont, Miguel Vaz, among other colleagues for the interesting and fruitful discussions during the research period apart from all the good times we shared from time to time. Additionally, I would like to thank Heiko Wersing, Stephan Hasler, Matthias Franzius, Samuel John and Stephan Kirstein, for the scientific talks and the times of joyful activities as well as the plenty of celebrations we spent together. Several appreciations are also to Christian Görick, for giving me some tips for writing scientific publications, to Herbert Janssen, Michael Gienger and Manuel Mühlig for letting me visit "Asimo", to Bernhard Sendhoff, for his knowledge about the redaction of patents and finally to Jannik Fritsch, Benjamin Dittes, Antonello Ceravola, Burkhard Zittel, Andreas Richter and many other colleagues for their contributions to expand my scientific horizons and for encouraging a good research environment.

In Bielefeld, to the members of the Research Institute for Cognition and Robotics (CoR-Lab) and the working group "Applied Informatics" (AG AI), my deepest appreciation is for the chances of interchanging and contrasting a vast diversity of opinions. Primarily, my mention to Katharina Rohlfing, for all those books about language acquisition in children that I borrowed from her, to Lars Schillingmann and Franz Kummert for their conversations about speech and pattern recognition, likewise to Ruth Moradbakhti because of her constant interest about my progress. In the same way, I would like to express

my gratitude to the Bielefeld University for having been able to participate in the mentoring program coordinated by Ursula Keiper and Kira Driller during my research term, equally I thank my mentor Grit Behrens for the time she devoted me and her advices and for the good climate that we shared all the time with the other mentees: Alexandra Barchunova, Johanna Egetemeir, Andrea Finke, Silke Fischer, Rebecca Förster, Marie-Christine Heinze, Hannah Mormann, Annika Peters, Jeannette Prochnow, Maha Salem, Dominique Schröder, Viktoria Spaiser, Marnie Ann Spiegel and Nazgul Tajibaeva.

For reading the drafts of my thesis, my sincerest thanks are for my husband, my supervisors, Frank Joublin and Katharina Rohlfing. Your suggestions and comments have helped me a lot.

I would also like to express my gratitude to many other friends, colleagues, superiors, professors and teachers, that although I do not mention them directly, they have led me too through this journey motivating me to exert myself until achieving my goals.

Abstract

A strong trend in robotics is the investigation of adaptable machine learning algorithms and frameworks that enhance the skills and application of artificial systems during the interaction with humans. The use of language is one of the most convenient human methods to communicate with artificial agents. In the last decades, the introduction of automatic speech recognition (ASR) systems achieved important advances in the field, however the simple and natural style of how parents teach speech to their children is still an open question under investigation. With the goal of improving the interaction and learning process with artificial agents, these must be able to increase their vocabulary (acquire novel terms) in a satisfying manner (rapid and adequate) for the user/tutor.

In this work, we introduce an incremental word learning system[1] to enhance speech acquisition in artificial agents. Here, different word-models are successively learned in a framework that possesses little prior knowledge and is inspired by the human infants language acquisition process. In order to build a user-friendly system that requires a low tutoring time our approaches are built to cope with a small number of training samples. Relative to the used architecture, we employ a hidden Markov model (HMM) framework similar to most ASR systems. Although HMMs are a powerful tool, for obtaining good recognition scores, special attention is required for the quantity of samples employed, the bootstrapping method and the performance of the discriminative training techniques integrated in the framework. Therefore, we present several procedures to overcome these challenges. A main drawback of employing few training data samples is the overspecialization of the learned models, which complicates the recognition of unseen items. In this context, we propose a novel computation of a parameter, which is adapted according to the amount of provided data samples, and analyze different influences for limited learning data. Afterwards, we describe the proposed initialization technique introduced in the system to properly bootstrap the estimates of the newly created model. In this approach, we combine unsupervised and supervised training methods with the following re-building of the model through a multiple sequence alignment method, which arranges and incorporates the succession of hidden states obtained by the Viterbi decoding algorithm. Next, several large margin (LM) discriminative training methods are analyzed to increase the generalization performance of the previously created models, i.e. improving the classification of the models. Here, we propose different procedures appropriate for employing limited data in discriminative training. Finally, the proposed methods are compared against state-of-the-art techniques during the experimental phase. Similarly, each individual contribution of the introduced approaches is measured in relation to the global yielded improvement of the whole framework. Additionally, we examine a potential decrease of the amount of training samples with the purpose of decreasing the time employed to teach a robot. The evaluation of our approaches is realized on different recognition tasks containing isolated and continuous digits. After all, we demonstrate that the introduced collection of techniques achieve important improvements so that they represent a significant step towards efficient incremental word learning with limited data.

[1] Based on the previous works enumerated in Appendix D.

Acronyms and abbreviations

ACORNS	Acquisition of COmmunication and RecogNition Skills
ADS	Adult Directed Speech
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AM	Acoustic Model
ANN	Artificial Neural Networks
ASIMO	Advanced Step in Innovative MObility
ASR	Automatic Speech Recognition
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BVA	Best Viterbi Alignment
CBS	Confidence Based Selection
CDHMM, CHMM	Continuous (Density) HMM
CDS	Children Directed Speech
CELL	Cross-channel Early Lexical Learning
CSR	Continuous Speech Recognition
D	Deletion, Density
DB	DataBase
DHMM	Discrete HMM
DT	Discriminative Training
EBW	Extended Baum-Welch

EM	Expectation-Maximization
FS	Flat Start
FV	Feature Vectors
GD	Gradient Descent
GMM	Gaussian Mixture Model
HAC	Histograms of Acoustic Co-occurrence
HMM	Hidden Markov Model
HTK	HMM ToolKit
I	Insertion
iCub	(i) Cognitive universal body
IWL	Incremental Word Learning
IWR	Isolated Word Recognition
L-BFGS	Limited memory Broyden-Fletcher-Goldfarb-Shanno
LEX	LEXicon
LM	Language Model, Large Margin
LME	Large Margin Estimation
MAP	Maximum A Posteriori
MCE	Minimum Classification Error
MMI	Maximum Mutual Information
MFCCs	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MSA	Multiple Sequence Alignment
Nao	Not an abbreviation, just a robot name
NMF	Non-negative Matrix Factorization
NW	Number of Words
OOV	Out-Of-Vocabulary

PCA	Principal Component Analysis
PDF	Probability Density Function
PLP	Perceptual Linear Predictive
RASTA	RelAtive SpecTrAl
RLM	Retraining the Last Model
RNN	Recurrent Neural Networks
S	Substitution
SCHMM	Semi-Continuous HMM
SF	Scaling Factor
SRI	Stanford Research Institute
SVM	Support Vector Machines
TIDigits	Texas Instruments (TI), Digits
TIMIT	Texas Instruments (TI), Massachusetts Institute of Technology (MIT)
TDNN	Time Delay Neural Networks
UCLA	Phonological Segment Inventory Database
US	Uniform Segmentation
VAD	Voice Activity Detection
VQ	Vector Quantization
WER	Word Error Rate

Mathematical notation

These are terms and notation used throughout this work.

Variables, symbols and operations

$\approx$	approximately equal to
$\equiv$	equivalent to
x	scalar quantity
$\widehat{x}$	estimate of the true value of x, re-estimation of the parameter x
$\arg\max_x f(x)$	value of x that maximizes $f(x)$
$\max_x f(x)$	value of $f(x)$ when x maximizes $f(x)$
$\arg\min_x f(x)$	value of x that minimizes $f(x)$
$\min_x f(x)$	value of $f(x)$ when x minimizes $f(x)$
$log(x)$	logarithm n of x
$exp(x)$	exponential of x
δ	derivative
$O(\cdot)$	computational cost
$\bigtriangledown f(x)$	gradient of the function $f(x)$

Vectors and matrices

$\boldsymbol{x}$	vector of arbitrary dimensions
$\boldsymbol{A}$	a matrix
$\boldsymbol{A}^{-1}$	inverse of a matrix
$\boldsymbol{A}^{T}$	transpose of a matrix
i	index of an element in a row of a matrix $\boldsymbol{A}$
j	index of an element in a column of a matrix $\boldsymbol{A}$
a_{ij}	element in row i and column j of $\boldsymbol{A}$

Observations

T	number of frames in a sequence of observations
t	time frame index
D	number of dimensions of a vector

d	dimension index
$\boldsymbol{x}_t$	speech feature vector at a determined time frame t composed of static $\boldsymbol{c}_t$, delta $\Delta\boldsymbol{c}_t$ and delta-delta $\Delta\Delta\boldsymbol{c}_t$ coefficients
$\boldsymbol{X}$	sequence of speech feature vectors $[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T]$; a training sample or exemplar
R	number of training samples
$\overline{\boldsymbol{X}}$	set of training samples
$\boldsymbol{x}_{t'}^{t''}$	partial observation $[\boldsymbol{x}_{t'}, \boldsymbol{x}_{t'+1}, ..., \boldsymbol{x}_{t''}]$

Probability and distributions

$P(\cdot)$	probability, probability density function
$P(x, y)$	joint probability density function (probability), i.e. the probability density (probability) of having both x and y
$P(x\|y)$	conditional probability density function (probability) of having x given y
μ, σ^2, σ	mean, variance and standard deviation
$\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\Sigma}$	mean vector (or set of means), variance vector (or set of variances), covariance matrix
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate Gaussian distribution
$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}, \boldsymbol{\Sigma})$	probability of vector $\boldsymbol{x}$ given a multivariate Gaussian distribution
$\sum_m^M cm_m \mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$	probability of vector $\boldsymbol{x}$ given a Gaussian mixture model (GMM)
M	number of components in a GMM
$\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$, cm_m	mean vector, covariance matrix and weight for the m^{th} mixture component of a GMM

ASR

w	acoustic model (label)
$\boldsymbol{w}$	set of acoustic models $[w_1, w_2, ...w_U]$
$P(\boldsymbol{X}\|w_i)$, $P(\boldsymbol{X}\|\lambda_{w_i})$	acoustic probability
$P(w_i\|\boldsymbol{X})$, $P(\lambda_{w_i}\|\boldsymbol{X})$	posterior probability
$P(w_i)$	prior probability (language model)
$P(\boldsymbol{X})$	probability of the occurrence of the observation $\boldsymbol{X}$
$F(\boldsymbol{X}\|w_i)$, $F(\boldsymbol{X}\|\lambda_{w_i})$	discriminant function
$h_\cdot()$	a monotonically increasing function
λ	set of parameters of an acoustic model

HMMs: parameters

S	hidden state
N	number of hidden states S
S_t	hidden state at a determined time frame t
$\boldsymbol{\mathcal{S}}$	sequence of hidden states $[S_1, S_2, ..., S_T]$
L	number of sequences of hidden states $\boldsymbol{\mathcal{S}}$
ϵ	number of skip transitions
a_{ij},$a_{S_iS_j}$	transition probability from state S_i to state S_j
$\boldsymbol{A} = a_{ij}$	matrix of transition probabilities
π_i,π_{S_i}	initialization probability for state S_i
$\boldsymbol{\pi} = \pi_i$	vector of initialization probabilities
$b_j(\boldsymbol{x}), b_{S_j}(\boldsymbol{x})$	observation or emission probability for the state S_j
$\boldsymbol{B} = b_j(\boldsymbol{x})$	set of the observation or emission probabilities
$O_{S,d}$	representation of the emission probabilities in DHMMs through a discrete representative value for each state S and feature dimension d
$\boldsymbol{W}_{S,d}$	representation of the emission probabilities in SCHMMs through a vector of weights to the elements of a codebook for each state S and feature dimension d
$\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$	set of parameters of a Markov chain

HMMs: learning, estimation and decoding

$\alpha_t(i)$	probability that the path is in state S_i having generated the partial observation $\boldsymbol{x}_1^t$
$\beta_t(i)$	probability of generating the partial observation $\boldsymbol{x}_{t+1}^T$ assuming that the HMM is in state S_i at time t
$\xi_t(i,j)$	probability that a path passes through state S_i at time step t and through state S_j at the next time step $t+1$, given the model λ and the observation $\boldsymbol{X}$
$\gamma_t(i)$	probability of being in state S_i at time t, given the model λ and the observation $\boldsymbol{X}$
$Q(\cdot)$	auxiliary function for the Baum-Welch algorithm
S^*	most likely sequence of states
$VIT_t(i)$	probability of the most likely sequence of states $\boldsymbol{\mathcal{S}}^*$ at time t, which has generated the observation $\boldsymbol{x}_1^t$ (until time t) and ends in state S_i
$BTR_t(j)$	pointer to the most likely previous state S_i in the best path sequence $\boldsymbol{\mathcal{S}}^*$ being currently in state S_j

Variance floor

$\overline{Var}(R)$	average over the variances of all GMMs for all feature dimensions for a determined number R of training samples
$\overline{Var}(\infty)$	saturation value of the average of the variances when the number of samples R is large
$G(\cdot)$	Gompertz function
$var(R)$	approximation of the behavior of the variances
$var_{f1}(R)$	normalization of $var(R)$ through $\overline{Var}(\infty)$
$r_f(R)$	reinforcement factor for R training samples
$var_f(R)$	scaling function for the variance floor depending on the number of training samples R used
$V_F(d)$	variance floor for a feature dimension d
K	variance floor scaling constant
$\overline{V}_d$	variance floor without scaling
$V_F^*(d, R)$, V^*	variance floor value depending on the feature dimension d and the number of training samples R
V_G	variance floor based on the average of the means of the GMMs in each iteration
$V_\bigcirc$	variance floor based on the global variance
$V_\%$	variance floor based on a percentile threshold

Bootstrapping

K	number of clusters in the K-means algorithm
$\boldsymbol{\mu}_k$	prototype or representative associated with the k^{th} cluster in the K-means algorithm
$r_{l,k}$	assignment of the vector $\boldsymbol{x}_l$ to the k^{th} cluster in the K-means algorithm
J	distortion or dissimilarity measure in the K-means algorithm
$b_{S_i}^{max}$	maximum value of the emission probabilities associated with a state S_i via Viterbi decoding analyzing all the samples related to the model that contains the state S_i
$\boldsymbol{D}$	cost matrix in the multiple sequence alignment bootstrapping method (MSA)
K_δ	proportionality constant for the computation of the cost matrix $\boldsymbol{D}$
$\delta_{j\to i}$	accumulator of the event that the state S_j is followed by S_i in the computation of the cost matrix $\boldsymbol{D}$
$\boldsymbol{C}$	comparison matrix in MSA
$\boldsymbol{S}$	similarity distance (measure), grid in MSA
$\boldsymbol{v}, \boldsymbol{t}$	Viterbi decoded sequences to merge in an iteration of MSA
$\boldsymbol{v}', \boldsymbol{t}'$	modified $\boldsymbol{v}$ and $\boldsymbol{t}$ sequences including two tokens (start and end)

Large margin discriminative training

$\overline{\boldsymbol{X}}_S$	set of support samples
$d(\cdot)$	distance, margin, also Euclidean distance
$M(\cdot)$	Mahalanobis distance
θ	a non-negative scalar offset
$\boldsymbol{\phi}$	expanded positive semidefinite matrix
$Q(\lambda_{w_m})$	objective function for large margin estimation (LME) of the model λ associated with the class w_m in the iterative localized optimization method
SF	Scaling factor for the strategy of the same name
$\text{LM}_{\boldsymbol{\mu}}$	LM applied to the means of the GMMs
$\text{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$	LM applied to the means and variances of the GMMs
λ_{w_n}	new learned model in the system for the class w_n

Contents

1 Introduction

Nowadays, the use of robotic systems is a very extended practice for the automation of most industrial procedures obtaining several benefits such as the increase of productivity (Gupta and Arora 2007, Sec. 13.7, p. 270). The non-living nature of machines enables them to perform complex tasks at high rates and precision without getting tired or bored as well as to endure extreme or dangerous conditions (Woog 2010; Romanelli 2010). As stated in the Robot Institute of America 1979, the term robot can be defined as:

> "*a reprogrammable, multifunctional manipulator designed to move material, parts, tools or specialized devices through various programmed motions for the performance of a variety of tasks.*"

The above description quoted in Gupta and Arora 2007 (Sec. 13.3, p. 267) points to the well-known employment of robots for repetitive jobs under specific restrained conditions such as the case of the manufacturing scene. However, another dominant intention behind robotics is the creation of a human-like "automatic device that performs functions normally ascribed to humans or a machine in the form of a human" as also defined by Gupta and Arora quoting Webster. This alternative can be characterized by intelligent autonomous agents, which can be denoted as adaptive, self-sustaining and self-governing systems that execute tasks for others (Steels 1995).

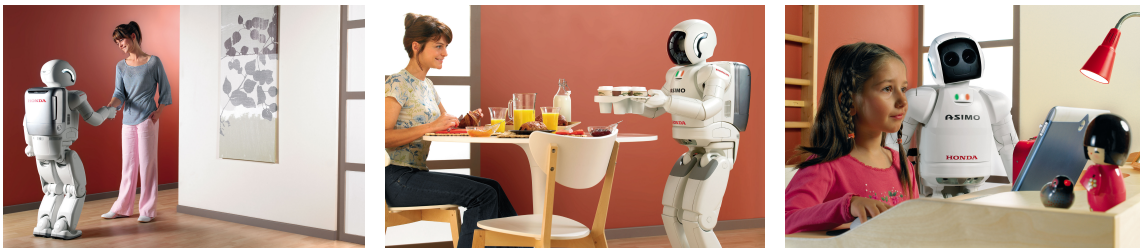
Despite major advances in recent decades, current robots cannot outperform human skills or cover users' necessities yet, thus robotics remains as one of the most actively investigated fields of study with a highly dedicated and diversified research community. The broadly recognized evolution of the use of machines from industrial applications to more personal and social agents in order to enrich even more domains of our everyday life is in particular visible due to the development of humanoid robots (e.g. ASIMO[1], NAO[1], iCub[1]). They resemble human appearance and this resemblance enhances their acceptance among the users[2] with the target to assist them on human level performance with respect to flexibility and mobility (see Kupferberg et al. 2011).

[1] See the list of references at the end of the thesis.

[2] Not always, as is the case of too human-like androids, see Ho et al. 2008.

Possible applications for these autonomous systems are robots employed in security areas to monitor and control the household protecting their owners (e.g. Fernández et al. 2008) or future domestic robots to be introduced in the market as home assistants, products for entertainment as well as health-care services among others (e.g. Sung et al. 2009). Particularly in the last case, there are numerous instances referred below about the purpose of developing such applications. Despite controversial discussions the employment of artificial agents can relieve the caretakers (without replacing them) facilitating a more dedicated assistance to the patients (see Bailey 1992; Lin et al. 2011, Sec. 1.1, pp. 4-6, Ch. 7, pp. 276-278). Lonely people or people with socializing disorders can benefit in some cases from the presence of a robot in the role of a friend or partner (see Dautenhahn and Robins 2015; Dautenhahn et al. 2005), who supports them emotionally, entertains them or also educates the user as suggested by Sung et al. 2009 (see Fig. 1.1). In recent years, several studies, such as Roger et al. 2012, have demonstrated that artificial agents applied in health care can therapeutically make significant progress in the treatment of some mental diseases. In the case of autistic children, playing with artificial agents promotes the development of social skills like interacting with other children and adults and encourages progressive steps aimed towards natural communication with people (Dautenhahn 2007). Similarly, Kubota and Mori 2009 mentioned that senior home-care helpers as conversational robots can be employed for the prevention of dementia and stimulation of the brain of senior people. Kubota and Mori also suggested that such cognitive robots could enhance the concentration, reinforce the mental health and improve the memory skills of these people. From the above mentioned issues other benefits for having a robot at home can be extracted in addition to its employment in recurring domestic tasks.

In this context, this new class of computer-based services raises the question on how to develop robotic prototypes for operating in an unrestricted and uncontrolled environment without the supervision of engineers (Romanelli 2010; see Seabra Lopes 2002). Current machines cannot react to dynamically changing situations as it is not feasible to establish all possible cases in advance (Iwahashi 2007). Hence, the agents have to be adaptable as well as simple to supervise and teach by users without special skills (Seabra Lopes 2002).



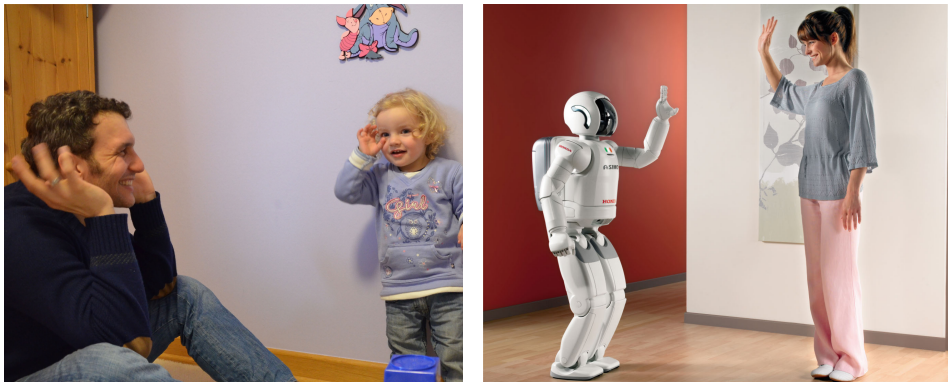
(a) Social interaction (b) Domestic task (c) Children care and tutoring

Figure 1.1: *A snapshot in the future. Evolving from a household robot to a personal humanoid robot for helping and supporting us socially, physically and mentally. Pictures taken from the ASIMO Gallery (see the list of references at the end of the thesis).*

Our universal desire as users is that in the future, machines should adapt to user needs and not the other way round. For all these reasons, the robotics society aims at the investigation and development of suitable mechanisms for human-robot interaction that imitate human abilities, such as to learn from their individual experiences, undertake some tasks autonomously and interact in a natural way with people and their environment (Romanelli 2010). This is also the long-term goal of cognitive sciences and artificial intelligence (Romanelli 2010).

Language is the universal and most powerful method of human communication, with which humans interchange ideas and information (O'Dea and Mukherji 2000, Ch. 1, p. 1; Bailey 1992). One advantage of endowing robots with verbal conversational capabilities besides being able to discourse with them is that "talking" to a robot is clearly faster and more comfortable than pressing a button, typing in a computer, mouse clicks or speaking command language (Theodoridis and Koutroumbas 2003, Sec. 1.1, p. 3; Bailey 1992). This idea of having a talkative agent is not new, just in the last centuries very simple speaking machines as the ones proposed by Wolfgang Von Kempelen (1791), Joseph Faber (1846) and Homer W. Dudley (1939) caused enormous enthusiasm (see historical facts in Huang et al. 2001, Sec. 16.7, p. 826; Juang and Chen 1998). From these first attempts, language processing methods have progressed significantly, for example in the case of automatic speech recognition (ASR) systems being employed in many applications with great success (see examples in Sec. 2.3). However, these techniques do not present the performance required in this type of situations, where these systems are still one to two orders of magnitude worse than humans (Furui 2009). Similarly, although several conversational robots have been developed in recent years, most of them are only able to communicate in predefined scenarios in which each detail of the conversation is already known, so that the conversations with such agents are no longer significant and interesting (Kubota and Mori 2009). A standard user will not accept interacting with a personal robot without communication skills such as learning new words from humans (Iwahashi 2007). Based on the former statements, to acquire and retain an open-ended collection of words as humans do, artificial agents require to be gifted with suitable learning mechanisms, which enable them to learn novel terms.

For this purpose, the interaction with the user is fundamental considering that meanings in language are defined according to shared experiences (Iwahashi 2007) and person-specific features (gender, age, culture, dialect; see Cheshire 2005), i.e. some experiences and novel words are not possible to predict or define previously since meaning is agreed in interaction between the user and the artificial agent (Iwahashi 2007). Hence, personal robots have to be endowed with adaptive perceptual mechanisms to learn the preferred terminology of the user as well as to cope with the changes of the users' vocabulary as one can deduct from Kubota and Mori 2009. Nevertheless, if the system is very dependent on the user, learning from interactions becomes a very challenging task (Nguyen et al. 2011). Thus, one of the main goals of personal robots may include learning with a reasonable



(a) Parent-child interaction (b) Human-robot interaction

Figure 1.2: *Interactive learning scenario between a human and a robot inspired by the language acquisition process in children. The left picture shows a father teaching his daughter the word "fingers" by means of acoustic and visual stimuli. This scenario can be transformed into a human-robot interaction scene, where a human tutor (a woman in the right photo) takes the role of the father of the left image and the robot becomes the learner. The left photo was taken by the author with the consent of the participants and the right picture from the ASIMO Gallery (see the list of references at the end of this thesis).*

human effort, i.e. little interaction (Iwahashi 2007). In order to save tutoring time, the amount of training samples, that the user has to provide the robot, has to be optimized (Ayllón Clemente et al. 2012). The reduction of the number of training samples does not only minimize the tutoring time, but it also obviously decreases the computation learning time and the necessary memory capacity in the robot as both depend on the amount of data needed.

All the above mentioned challenges lead scientists to face different technical problems to build cognitive models and develop new methods and algorithms in order to provide artificial systems with learning capabilities and performance similar to humans. In order to bridge this performance gap, many researchers investigate and take inspiration from how language acquisition in human infants occurs (e.g. Minematsu et al. 2009 and ten Bosch et al. 2009). An illustration of such an interactive scenario is displayed in Fig. 1.2 and emulates the principal scenario that prompts this work.

1.1 Towards incremental speech acquisition

From the last section we can extract and identify two key technologies for future research in speech processing: firstly the possibility of endowing artificial agents with compliant systems that adapt to the users' language and vocabulary and secondly, a flexible acquisition of knowledge that allows artificial systems to add new words to their language representation. This motivates us to aim at the design and implementation of an incre-

mental word learning system suitable for the interaction with artificial agents. Such a framework should enable the acquisition of new words in an incremental way, reducing the dependence on predefined lexicons and qualifying intelligent systems to operate in interactive learning scenarios. This capability of introducing new terms into their vocabulary should serve the open-ended process of language acquisition and allow the personalization of the framework's lexicon. The nature of the speech directed to the infants helps them to quickly acquire new expressions (see further details in Sec. 2.2), what leads us to aim at the construction of a learning framework inspired by the language acquisition process in children.

Furthermore, we focus on the problem of how efficiently artificial systems can learn interacting with human tutors. As referred in the section before, users will only hold a relaxed conversation with an artificial agent if these can dialog in a competent manner (see Iwahashi 2007). Therefore, teaching new words to a system in a short time is an important ability. The reduction of the tutoring time motivates decreasing the number of training samples in order to optimize the learning and interaction procedure while maintaining a good performance[3]. Thus, our second goal is to develop suitable learning algorithms that enable our learning framework to become a user-friendly language acquisition system, which would require little supervision and low tutoring time.

From the above mentioned goals arise several targets to be addressed in our work. The construction of suitable speech models to represent the terms to learn in the embodied conditions indicated before is critical to achieve an appropriate performance for the intended application. The known time series nature of speech has to be taken into account, together with the great variety of spoken language patterns. Reducing tutoring time via limiting the use of speech data requires the further optimization of well suited algorithms for speech recognition and the introduction of novel techniques and strategies for processes like the initialization of the models as well as the use of discriminative learning approaches to enhance the generalization performance of our system.

To differentiate this thesis from previous works, it is worth mentioning what this thesis does not deal with. We do not concentrate on how to develop a robust system against noises. Although some related topics appeared along with this dissertation, we do not aim to solve this special task. Additionally, we do also not handle the problem of speech segmentation. At the beginning of the infant development process, caregivers and parents employ a special speech register with children, called children directed speech (CDS), which characterizes through the stress of the word boundaries and the utterance of most of the words in isolation (e.g. Dominey and Dodane 2004, more references in Sec. 2.2). Following child-like reasoning for our learning process, we assume that the utterances entered in the system are already segmented into words as analogy to the infant language acquisition procedure.

[3]As mentioned in our previous work Ayllón Clemente et al. 2012.

1.1.1 Contributions

This thesis, which comprises our previously presented works enumerated in Appendix D[4], contributes to the fulfillment of the above mentioned goals through the introduction of a hidden Markov model framework able to learn incrementally new words. Inspired by the first development stages of the language acquisition process in infants, our speaker-independent incremental system uses very little prior knowledge and has the capability to recognize words in continuous speech despite the terms were acquired in isolation. The goal of obtaining a user-friendly framework reducing the tutoring time is covered via the efficient acquisition of the terms, so that the acoustic models representing the new terms are trained during the learning phase with a very small amount of training samples while maintaining a good performance in the system. This core contribution was achievable thanks to the proposal of several methods and strategies along this thesis, namely the controlled parameter estimation by means of a dynamic adaptable threshold to avoid overfitting, the initialization of the acoustic models based on the combination of unsupervised and supervised algorithms integrating a novel multiple sequence alignment method as well as the introduction of several strategies to be applied in discriminative training of an incremental learning system with limited data.

The first of these methods is motivated by the fact that the use of few learning exemplars can lead to wrongly estimate the acoustic models representing the words learned by the system. In these cases, the acoustic models are overspecialized to the small set of data used for the training stage, what is the widely known overfitting problem (see Bishop 2006, Sec. 1.1, p. 6; also referred in Sec. 4.2). Consequently, different pronunciations of the same term that are not "heard" during the learning phase are not recognized as such. To gather the exemplars belonging to the class that are not "seen" by the system, one solution is to accommodate the feature distributions via the integration of expert knowledge adapting the parameters intelligently for each situation. Here, we contribute with the reformulation of a well-known threshold called variance floor (Melin et al. 1998), in the context of hidden Markov models, to be dynamically adapted according to the number of available exemplars in each situation to estimate the acoustic models. This threshold has the ability to alleviate the overfitting effect on few training samples. In order to automatize the adaptation of the threshold, we evaluate the behavior of the variances in different limited data training conditions to include this in the modeling of the mentioned variance floor.

[4]In this thesis, several paragraphs (some of them verbatim) are taken from Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012. The use of these own scientific publications is allowed by IEEE© (ICASSP - see `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` for permissions) and ISCA (Interspeech) for this thesis. The reuse or reprint from "Ayllón Clemente, I., M. Heckmann, and B. Wrede (2012). Incremental word learning: efficient HMM initialization and large margin discriminative adaptation. *Speech Communication 54* (9), 1029-1048" is also authorized with permission from Elsevier. In the last case, a license and a confirmation for reuse was also granted.

The second technique targets an appropriate learning of the acoustic models on the basis of the accessible feature distributions. In our system, these models are represented by hidden Markov models (HMMs), which are a widely recognized, very suitable and powerful framework to model speech. HMMs most popular training algorithm is based on an expectation-maximization approach, which success depends on the previous initialization of the estimates (see Huang et al. 2001, Sec. 8.4.1, p. 396). Moreover, the limited use of samples and the little prior knowledge in the learning process emphasize the relevance of making the best use of the available information. Regarding this, we extend an existent bootstrapping method that integrates unsupervised and supervised approaches (see Iwahashi 2007 and Brandl et al. 2008). Here, we present a novel multiple sequence alignment technique inspired by the sequence alignment algorithms used in Bioinformatics, such as the proposed by Needleman and Wunsch 1970 as well as Smith and Waterman 1981. Our technique combines the information contained in the exemplars of each uttered word into a profitable initialized sequence of units (states in HMMs). In other words, our algorithm is able to look for the most fitting topology and initialization of each acoustic model without manual interaction.

Finally, we propose several techniques that can be applied in discriminative incremental learning when the number of available samples for word modeling is limited. In our framework, we apply a combination of generative and discriminative approaches to learn the acoustic models. Discriminative methods improve the separation or classification margin between models trained by means of generative approaches such as HMMs (e.g. Juang et al. 1997). In this context, we employ several methods when the number of exemplars of a term is pretty limited. The first method that we present is based on the artificial softening or blurring of the boundaries of the previously learned models to encourage the existing discriminative algorithms (in our case large margin discriminative training) to further optimize the acoustic models, which already appear to be well separated according to the exemplars used for learning. The second and third strategy, which we employ jointly with the first one, aim to reduce the computation time necessary for discriminative training each time the vocabulary is incremented. When a new term is learned, all models of the classes could be recalculated in order to optimize their boundaries against the other competitive models (see approaches cited in Sec. 6.1.2 and 6.2). The second technique takes advantage of the incremental nature of our framework by only re-estimating the model to which the last training samples were assigned. The third one is more sophisticated and applies confidence-like intervals on discriminant functions to decide which the models to optimize in each iteration are.

Our efficient incremental word learning framework and all the above mentioned approaches are explained in detail along this thesis and have been jointly evaluated with other existing techniques in order to demonstrate their efficiency and good performance in very sparse learning conditions.

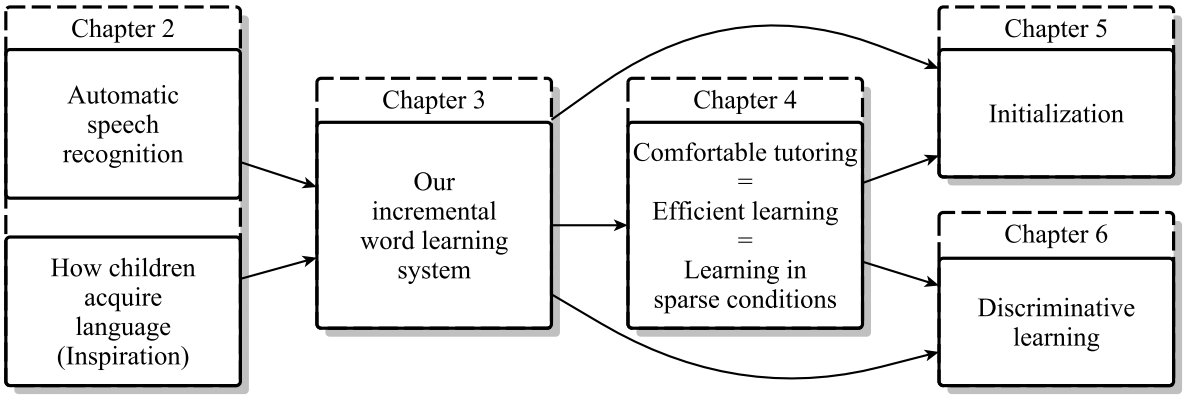


Figure 1.3: *Overview of the presented work. The role of Chapter 2 is to present the basic elements of an automatic speech recognition (ASR) system and the process of how children acquire language. Both topics lead us to introduce our incremental word learning system in Chapter 3 which main aspects are subsequently explained in the following chapters. The challenges of training with sparse training conditions in order to provide the user with a comfortable scenario to interact with an artificial agent are handled in Chapter 4. Here, an efficient use of the model parameters and a novel adaptation of one of those to avoid the overspecialization to the learning data are analyzed. The limited use of training data derives in a continuous optimization of the learning strategies, namely the initialization of the models in Chapter 5 and the improvement of them through discriminative approaches in Chapter 6.*

1.2 Outline of the thesis

Automatic speech recognition (ASR) systems are the most well-known and extended approaches to model the human language perception skills from an engineering viewpoint. Chapter 2 aims to give the reader an overview of these systems, emphasizing the speech signal, its corresponding processing stages in ASR and the pattern classification methods usually applied in this context. Some notions of the language acquisition process in children are also introduced in the chapter to illustrate how children learn speech. The chapter ends with an overview about the main current incremental language acquisition approaches.

After disclosing the different state-of-the-art methods for incremental learning and the fundamentals of ASR systems in Chapter 2, we propose our incremental word learning framework in Chapter 3. The chapter starts by introducing the architecture of our learning system. Next, hidden Markov models (HMMs), currently the most successful and widely recognized statistical technique to construct computational models for language acquisition, are presented as the core component of our framework. Here, we explain the most important and basic HMM algorithms operating in our proposal. In addition, a brief description of the different parts composing the system is provided.

The problematic of learning with limited data is formulated in Chapter 4. Firstly, an

exhaustive analysis of a suitable parameterization in sparse training conditions is realized. This is conducted through the investigation of how to choose a suitable topology and set of parameters of the HMMs. Afterwards, the problem of the overspecialization is discussed and a novel adjustable threshold is proposed to cope with it.

The use of an efficient HMM-based system leads to the necessary optimization of each computing step. In this direction, a method for a suitable initial estimation of the model parameters to obtain a good performance and to operate in an incremental word learning system with few training samples is presented in Chapter 5. In consecutive sections, we describe in more details the different phases of our initialization presented in Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012, which is influenced by the approaches of Iwahashi 2007 and Brandl et al. 2008 and consists of the combination of unsupervised and supervised training, where a transformation of an ergodic HMM into a left-to-right HMM takes place performed by means of a novel multiple sequence alignment (MSA) procedure. In Chapter 6, the improvement of powerful state-of-the-art discriminative learning methods, namely large margin discriminative training, through different efficient strategies is analyzed and described when the number of training samples is very limited.

At the end of the Chapters 4, 5 and 6, we describe experiments in which we compare and evaluate the approaches proposed with different standard techniques[5]. To measure our improvements, we report the recognition scores obtained on an isolated and continuous digit recognition task using well-known benchmarks.

In Chapter 7, we conclude this work going through the different contributions of the thesis. We specially analyze the possible reduction of the training samples by means of our incremental system while maintaining a good performance. Extensions to our work and further improvements for future research and limitations of the framework are also discussed at the end of the chapter.

[5]Partially extended from the realized one in Ayllón Clemente et al. 2012.

2

Computational approaches to language acquisition

As referred in Chapter 1, the intention of having talkative agents has already fascinated humans for centuries. However, the numerous widely known challenges that have to be tackled in speech recognition, e.g. the continuous evolution of language or the speech dependency/specificity to culture and users, make it difficult to find general computational approaches. As a result, the simplest method is to resolve the problem for a particular application and a determined scenario, i.e. to take a state-of-the-art approach and adapt it to the desired environment, achieving the predefined scenarios mentioned in the chapter before. Nevertheless, conventional techniques hold several disadvantages that do not recommend them for changing contexts and practices where a certain degree of adaptability, a characteristic of human nature, is required (see Benzeghiba et al. 2007). In such cases, it is necessary to find new approaches for the design and development of compliant frameworks.

Outline of the chapter

The aim of this chapter is to briefly describe the nature of speech and to review the basics of typical automatic speech recognition (ASR) systems[1]. Firstly, we start giving some hints about speech and its units, namely phonemes, syllables, words, and afterwards we explain how the combination of the last ones results into sentences in continuous speech. This type of speech and the utterance of isolated words are jointly discussed in order to elucidate the advantages and disadvantages of each one. Then, speech variability is also introduced where we emphasize the difference between speaker-dependent and independent recognition. We describe all these features together with the challenges they bring to the recognition system. The reader should note that although the presence of noise is also a big challenge, it is out of the scope of this thesis and will not be reviewed as such. Secondly, we present a brief summary of the language acquisition process in infants.

[1]These issues are fundamentals in the speech recognition community and their definition and description in this thesis are based on Huang et al. 2001, Theodoridis and Koutroumbas 2003 and Bishop 2006.

After the introduction of the speech learning procedure in children, an overview of a classical automatic speech recognition system is provided. Here, we describe its architecture consisting of several processing stages: the signal acquisition and pre-processing step, the feature extraction stage as well as the classification phase. In the latter, we introduce several mathematical notions to explain statistical pattern recognition and classification. Afterwards, we review these approaches and outline their limitations with respect to incremental learning. A survey of different incremental approaches presented in recent years is supplied at this point. Finally, we conclude with an overview of the chapter recollecting unresolved issues.

2.1 The nature of speech

As mentioned in Chapter 1, the most extended and natural way of human communication is speech. However, speech is far away from being simple. Speech is well-known as a time series signal (see Fig. 2.1), which can be decoded into linguistic elements or units explained in Sec. 2.1.1. In the speech signal, a differentiation of these units can be observed in Fig. 2.1 through their temporal progression. Similarly, the manner in which these units are combined lead to the different meanings and messages in language (Huang et al. 2001, Sec. 2.2.1, p. 37).

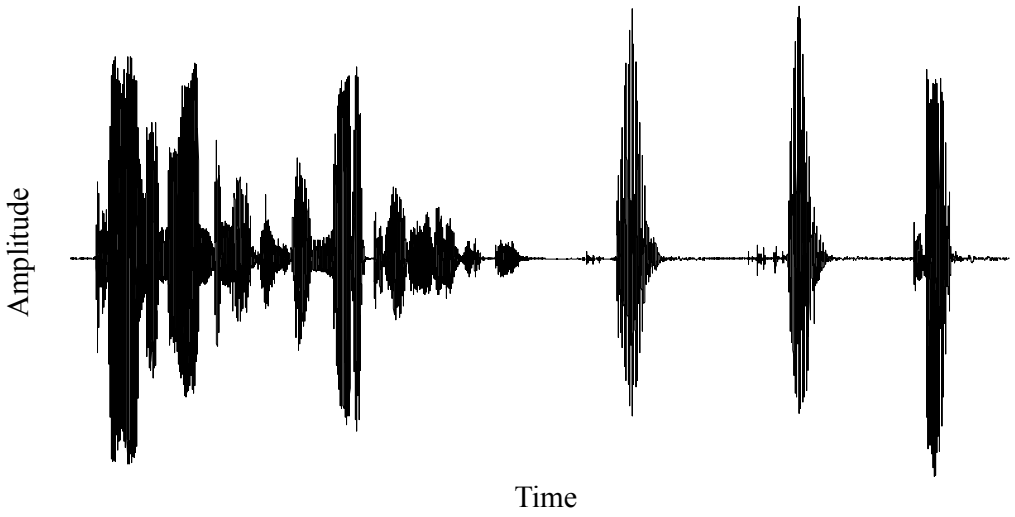


Figure 2.1: *Illustration of a speech signal in the time-domain for the uttered sentence "call an ambulance for a medical assistance one one two". The audio record plotted here is a concatenation of a speech sentence taken from the TIMIT database (Garofolo et al. 1993, see Appendix A.1) and several isolated words from TIDigits database (Leonard and Doddington 1993, see Appendix A.2). This audio record is used as an example for the explanation of various concepts and processing methods in future sections.*

2.1.1 Speech units

In the linguistic hierarchy, phonemes are at the lowest level (Quinlan and Dyson 2008, Ch. 6, p. 219; see Fig. 2.2). Each language employs a set of phonemes (Kuhl 2007), which are elements supposed to have distinct acoustic and articulatory features (Reddy 2001).

Huang et al. 2001 (Sec. 2.2.1, p. 37) stated that the term phoneme is employed to designate any of the minimal speech sound units in a language, which can lead to differentiate among words. According to Huang and colleagues, these building blocks help us to distinguish word meanings, e.g. the phoneme /p/ reveals that the word "pat" is not the same as the word "bat" in spite of being heard as similar sounds. In this context, the term phone is conventionally used to denote a phoneme's acoustic realization, e.g. the phoneme /t/ with the words "sa**t**" and "me**t**er" (Huang et al. 2001, Sec. 2.2.1, p. 37).

When phonemes are linked to build more substantial linguistic elements, the coarticulation effect occurs, where the acoustic features of a phoneme are a little altered according to its neighboring phonetic context due to the movement combinations of the anatomical elements involved in the pronunciation such as the tongue or the vocal cords (Reddy 2001).

The set of phonemes is small; the maximum number of phonemes in the UPSID (UCLA Phonological Segment Inventory Database) for a language is 141 according to Maddieson 1984 (Sec. 1.3, p. 7). American English comprises 16 vowel and 24 consonant sounds (40 phonemes, Reddy 2001). As mentioned in Brandl 2009 (Sec. 3.6.2, p. 44), a strategy used in some early speech recognition systems consists in modeling each phoneme by itself, which seems to be quite efficient due to the fact that we could model most languages with around 50 phonemes. Unfortunately, these early systems did not achieve good performances due to coarticulation effects so that posterior frameworks aim to consider them (Brandl 2009, Sec. 3.6.2, p. 44).

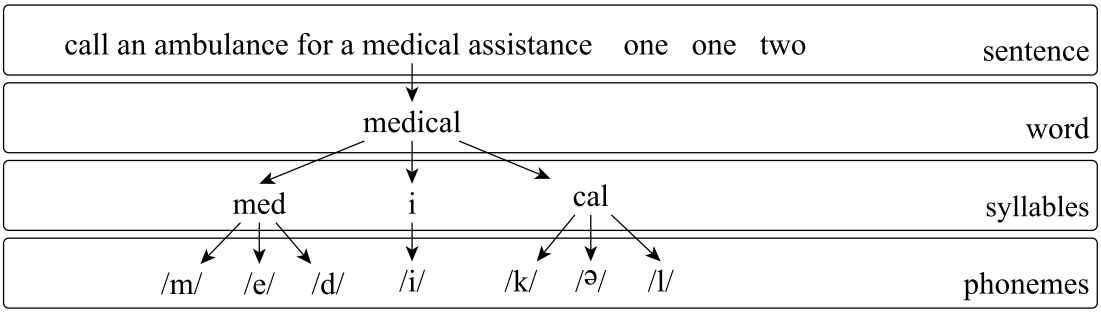


Figure 2.2: *Exemplification of the hierarchy of the different speech building blocks. The combination of different phonemes forms syllables, and these likewise form words. A succession of words constructs a sentence and their strategic combination into sentences creates a message for the listener (see the body of the text).*

Going one step up in the hierarchy, several phonemes are combined to form larger elements such as syllables and words (see Quinlan and Dyson 2008, Ch. 6, p. 210). Words are the main vehicle of meaning and meaningful elements by themselves, hence these facts together with the possibility of catching the phonetic coarticulation effects inside words have favored the widely employment of word-models for many ASR systems (Huang et al. 2001, Sec. 9.4.1, p. 427).

2.1.2 Isolated words vs. continuous speech

In continuous speech, sentences often do not contain pauses between words, what makes segmentation a real challenge without prior language knowledge as it is extremely hard to distinguish where one word finishes and the other starts, in contrast to reading a line of text where the word boundaries are designated by spaces (Ambridge and Lieven 2011, Sec. 2.4, p. 31). This is called the segmentation problem, which would not exist if each word could be distinctively differentiated in the speech signal (Huang et al. 2001, Sec. 2.3.2, p. 53; Ambridge and Lieven 2011, Sec. 2.2, p. 14). This phenomenon is illustrated in Fig. 2.3, where we can observe a pause between the word "one" and "two", no pause appears between "call" and "an".

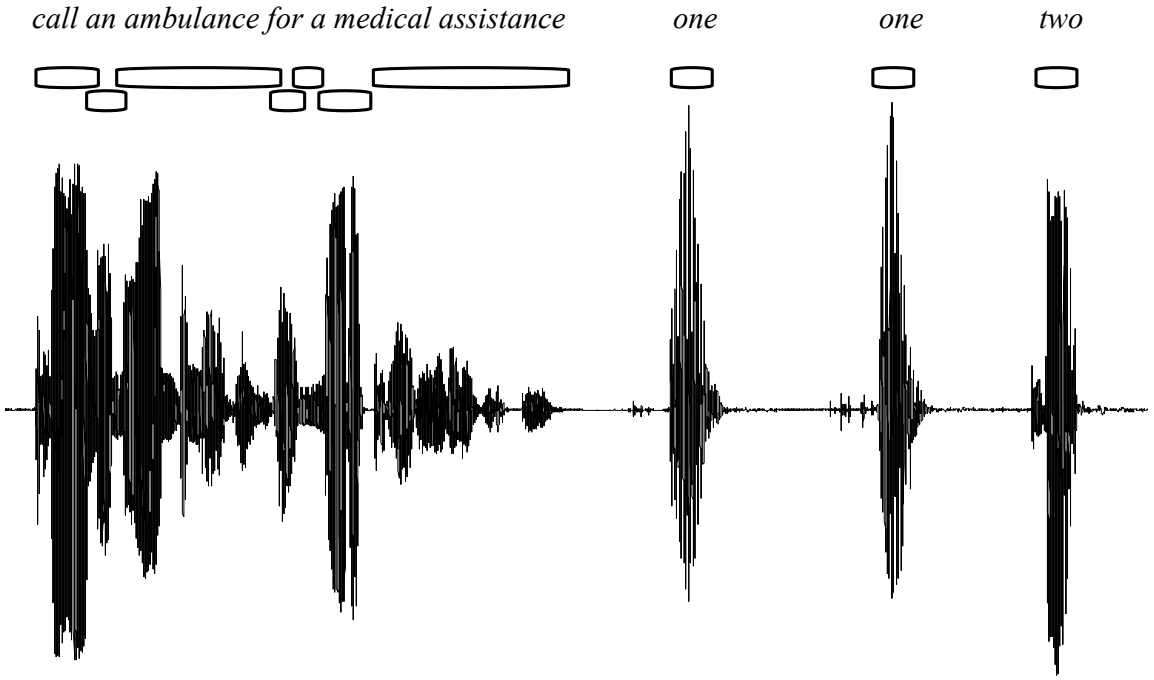


Figure 2.3: *Representation of the temporal progression of the different words of a sentence. A small box matching the duration and position of each word is placed in order to observe the overlap of each of these words with its context. In the sentence, we can distinguish that the first part represents continuous speech, where it is not clear when a word ends and the next starts (coarticulation). On the contrary, different numbers ("one one two") are clearly uttered in isolation (with pauses/silences between the words) at the end of the sentence, in this case to make this segment more comprehensible to the listener.*

Additionally, coarticulation can occur among word boundaries, i.e. starts and/or endings of words in *continuous speech* (Reddy 2001). According to Theodoridis and Koutroumbas 2003 (Sec. 8.2.3, p. 329), this is the case when the users communicate in a natural manner and word boundaries are not well outlined. On the other hand in *isolated speech*, as Theodoridis and Koutroumbas also mentioned, each word is uttered in isolation among silences, what substantially simplifies the decoding of the speech as it is known when a word ends and another one begins. This might be one reason why parents initially focus on isolated words in child directed speech (Broen 1972: qtd.[2] in Dominey and Dodane 2004; see Sec. 2.2). Thus, the increase of silent periods leads to a significant improvement in performance and in a reduction of computational complexity in isolated word recognition (IWR) systems in contrast to the continuous speech recognition (CSR) ones (Huang et al. 2001, Sec. 9.1.2, p. 416).

2.1.3 Speech variability

Apart from the coarticulation problem indicated above, pronunciation is also a challenge in itself. Speech is produced by the coordinated movements of the speech organs and the regulation of the airflow (Ambridge and Lieven 2011, Sec. 2.2, p. 14). So, as reported by Huang et al. 2001 (Sec. 9.1.3, p. 416), speech communication not only provides a message to the hearer but it reflects some anatomical information about the speaker as well as his/her gender, age, health and cultural characteristics, which are also encoded in the signal. As each individual speaker is different, Huang and colleagues also added that a word pronounced by one person can have a completely different signal form as the same word uttered by another person. Moreover, the same speaker cannot pronounce the same utterance twice, even with the best attempt to reproduce it (Ambridge and Lieven 2011, Sec. 2.2, p. 14; Huang et al. 2001, Sec. 9.1.3, p. 416).

Theodoridis and Koutroumbas 2003 (Sec. 8.2.3, p. 329) explained that in cases where the goal is the recognition of words spoken by a single speaker, the recognition task is called speaker-dependent recognition; otherwise a more challenging task is speaker-independent recognition. In the last case, as Theodoridis and Koutroumbas continued, the training samples include different speakers and the framework must have the ability to generalize, i.e. to ignore the phonetic differences, and identify words uttered by speakers not belonging to the ones employed in the training set.

Furthermore, Huang et al. 2001 (Sec. 9.1.2, p. 416) mentioned that speakers may also vary their speaking style according to the situation, e.g. spontaneous or natural speech vs. read-aloud speech[3]. Huang and colleagues also stated that changes involving the

[2]Here, the abbreviation "qtd. in" stands for "as cited/quoted in".

[3]As our long-term goal (Sec. 7.2) should be the natural interaction with an artificial agent, the system should be able to work in the future with spontaneous or natural speech including changes in intonation or speaking rate and unspecific grammar, although now due to our focus on efficient learning procedures we use benchmarks, where the sentences are read (slow rate, careful pronunciation).

speaking rate can affect the word recognition performance so that when the speaking rate increases, the recognition performance drops suggesting that the system should also handle rate variations of the same word.

2.2 Language acquisition in children

Infants are biologically gifted with the ability to learn to "decode speech" (Ambridge and Lieven 2011, Sec. 2.2, p. 17). However, this fact is more fascinating since the process is quite fast and the progressive steps for the continuous acquisition of the first language follow a quite similar time-line across languages, so that when infants are 6 months old, they start babbling and produce complex sentences at the age of 3 (Dominey and Dodane 2004; Kuhl 2007).

Although linguists, psychologists, and neuroscientists have deeply investigated the language acquisition process in order to elucidate how children acquire language and the reason of its similarity across different languages, we do not have a well-documented and uniform understanding about this process yet (Kuhl 2007; Roy 2009). Consequently, many encouraging theories of how this process takes place persist partially articulated, inconsistent and unverified (Roy 2009). The aim of this section is to provide the reader with some hints about this special process without going into too much detail.

Language acquisition happens almost automatically while the child is simply exposed to it (Dominey and Dodane 2004). This circumstance motivates researchers called *nativists* to propose the theory that language is very likely to be innate or pre-programmed in children, e.g. an inborn general grammar independent of the language called "Universal Grammar" (Ambridge and Lieven 2011, Sec. 1.1.1, p. 2; Dominey and Dodane 2004). On the other hand, *constructivist* researchers argue that children have the ability to acquire language but through generalization when listening to speech and not in consequence of some innate grammar as reported by Ambridge and Lieven 2011 (Sec. 1.1.2, p. 2).

While focusing on the mother or native language, children lose their generalization ability to discern speech discriminations in other languages failing to distinguish contrasts among non-native languages, which they originally could discriminate (Werker and Desjardins 1995: qtd. in Tomasello 2003, Sec. 3.2.1, p. 59). Children start learning[4] few words at the age of 10-12 months, about 300 words around 24 months and more than 500 words by 30 months (Bates et al. 2002).

In the first stage of development, it is not simple for children to recognize a word said by a different speaker as children start storing repetitions of early listened terms instead of abstract models (Houston et al. 1998: qtd. in Tomasello 2003, Sec. 3.2.1, p. 60). By increasing the exposure time, children start to generalize and are able to recognize already heard words (Jusczyk and Aslin 1995), although they are uttered by different speakers

[4]Please note that no distinction is made between comprehension and production of words at this point.

(Ambridge and Lieven 2011, Sec. 2.3.1.2, p. 23).

In the last paragraphs, we mentioned the exposition of children to language. How children come into contact with their first language via their caretakers is essential for an efficient acquisition process, it is well-known that children with socializing problems (e.g. autism) present a slower linguistic progress than other infants (Kuhl 2007). Hence, the quality of the speech received and the favored learning conditions that adapt the process to the necessities of the child play an important role (Dominey and Dodane 2004). This rich caretaker-child interaction style is called motherese (e.g. Fernald 1985 and Grieser and Kuhl 1988), fatherese (Shute and Wheldall 1999), parentese (Dominey and Dodane 2004) or child directed speech (CDS, Saxton 2009; Saunders et al. 2011).

In CDS, parents or caretakers highlight important cues to ease the acquisition of speech features present in adult (directed) speech (ADS, Dominey and Dodane 2004). However, these findings are subject to discussion, since some exceptions of CDS can be found in Kaluli in New Guinea, Kwara'ae of the Solomon Islands, Samoa and among some African Americans, where adults do not address children or they do it indirectly until they can speak (Kit 2003; de Boysson-Bardies 1999, pp. 87-88).

According to Kit 2003, CDS compared to ADS is: slower and more repetitive, has higher pitch, uses shorter utterances and limited set of topics as well as more frequent and longer pauses. Nevertheless, these CDS characteristics do not appear at the same time, the caretakers adapt their speech to the developmental progress of the child (Dominey and Dodane 2004). In the early developmental stages, caretakers introduce incrementally isolated words and pauses (Broen 1972: qtd. in Dominey and Dodane 2004). Brent and Siskind 2001 as well as Ninio 1993 investigated the influence of isolated words at the start of children development and discovered that most of the first words uttered by children were produced in isolation by their parents (Tomasello 2003, Sec. 3.2.4, p. 78). Additionally, most children start learning some adult locutions as holophrases or frozen phrases such as: "I-wanna-do-it", "Lemme-see" or "Where-the-bottle" in their first years of life (Pine and Lieven 1993: qtd. in Tomasello 2003, Sec. 2.3.2, p. 38). The reason why children start with isolated words or holistic expressions is not clear yet (Tomasello 2003, Sec. 2.3.2, p. 39).

According to the CDS studies summarized by Dominey and Dodane 2004, when the child is about 14 months-old, pauses are especially longer compared to the used ones in ADS (Broen 1972; Fernald and Simon 1984), novel words are extended and produced with a exaggerated pitch (Fernald and Mazzie 1991) and are often situated at the end of the sentence (Aslin et al. 1996).

Adults usually point to objects or people near to the child (Lacerda et al. 2004), hence the successions of sounds heard by the infants are very probable to jointly appear with objects in the visual field of children (see right image in Fig. 2.4, Hörnstein and Santos-Victor 2010).

Where's your ***mom***? ... she became a princess ... ... orange ... orange

Figure 2.4: *Different language learning scenarios for an infant. The left image shows a girl receiving a telephone call from a relative. The child listens to a familiar voice talking to her trying to contact her mother. The person at the telephone utters the word "mom" slowly and loud (stressed) in order to ease the decoding process for the girl. In the central image, the same child is hearing a tale using a pair of headphones. The understanding of some already known words amuses the girl. The right image displays an interactive learning exercise between the child and their parents. They aim to teach the wording of different colors to their daughter with the help of colorful cubes. The visual stimuli, the isolated repetition of each term and the enjoyable game aid the acquisition of the new terms. All these pictures have been taken by the author with the consent of the participants.*

Related to the segmentation problem, children use mainly two ways to segment novel words according to Bortfeld et al. 2005: bottom-up, e.g. using a collection of features[5] such as word accent when some word knowledge is missing and top-down, employing knowledge of already known words. There are some experiments, which demonstrate that having a learned word before an unknown word can help the segmentation process and consequently, the lexicon of learned isolated words could be used as initial anchors to further segment the rest of the utterance (Bortfeld et al. 2005), see central picture of Fig. 2.4. On the other hand, some researchers as stated before have focused on how children might segment speech from the bottom-up, situating word boundaries by means of prosodic features as infants are receptive to stressed words and the end of the sentences not reacting to words produced without prosody[6] (Dominey and Dodane 2004), see left photo of Fig. 2.4, where the word "mom" is specially stressed.

As closure of this section, our interest in building a user-friendly system with little interaction makes us ask how many repetitions[7] are needed for the infant to achieve a good generalization. Although some authors, as reported by Tomasello 2003 (Sec. 3.2.4, p. 79), have stated that listening to considerable quantities of speech eases the acquisition process, their studies do not make any precise link to how often children should hear a

[5]See examples enumerated in Bortfeld et al. 2005: Jusczyk et al. 1999a, Jusczyk et al. 1999b, Friederici and Wessels 1993, Goodsitt et al. 1993, Mattys and Jusczyk 2001, Mattys et al. 1999, Saffran et al. 1996.

[6]See instances listed in Dominey and Dodane 2004: Menyuk 1977, Eimas 1975, Kagan and Lewis 1965.

[7]It should be understood that the number of repetitions needed for the child are also dependent on the acquisition of meaning, however we are only interested in perceptual aspects in this work.

specific word to learn it. Therefore, specific experiments are needed in which different children hear a novel word in a predefined environment with a predetermined number of exemplars to answer this question (Tomasello 2003, Sec. 3.2.4, p. 79).

In this paragraph, based on the summary and analysis realized by Tomasello 2003 (Sec. 3.2.4, pp. 79-81), we enumerate some experimental studies in this field and their corresponding extracted conclusions. Firstly, comprehension of a word is faster than the production of it. In several studies realized by Goldin-Meadow et al. 1976 as well as Childers and Tomasello 2002, 2 year-old infants understood more words than they uttered. Secondly, children are able to understand some words after only a few repetitions. Carey and Bartlett 1978 detected that many children at the age of 3- and 4-years-old could recognize a new term for a new object after hearing it once. Similarly, Woodward et al. 1994 investigated that 13-month-old children recognized a novel term after one session with 9 samples of the new word. In addition, children can also pronounce new words appropriately after a very limited exposure. In one study to measure exactly how many repetitions are necessary for a child to produce a new word, Schwartz and Terrell 1983 discovered that children between 12 and 18 months needed approximately 10-12 repetitions of a new term to pronounce it correctly.

2.3 Automatic speech recognition systems and their components

After this brief introduction to the speech and language acquisition process in infants, and before explaining the used systems to recognize speech in this section, we give a short overview about learning and recognition in artificial agents.

The question of how machines could learn has been deeply investigated by a scientific discipline called machine learning. In the 1950's, Arthur Samuel defined machine learning as (Samuel 1959: qtd. in Asthana and Khorana 2013):

> "*a field of study that gives computers the ability to learn without being explicitly programmed.*"

The learner builds representations of data employing experience to enhance performance (e.g. reaction to new situations/data) or to realize precise forecasts based on observed data (Mohri et al. 2012, Ch. 1, p. 1). Related to the machine learning, main targets of pattern recognition are to learn to recognize complex regularities or patterns as well as to take intelligent decisions such as the classification of the given data into several groups or categories (Bishop 2006, Ch. 1, p. 1).

Language acquisition in artificial agents is investigated by well-known (automatic) speech recognition (ASR) methods that are pattern recognition techniques well established for numerous applications e.g. the extended use of telephonic customer service in companies/institutions or computer control interfaces, where several software products using

speech recognition technology are broadly available, e.g. Dragon speech recognition software[8]. Recently, speech control interfaces for mobile devices got significant public attention, as is the case of Siri[8], and became one of the key components for new technologies like wearable computing systems[9] (Google Glass[8]).

A standard speech recognition system consists of the following main components shown in Fig. 2.5: signal acquisition, feature extraction and pattern classifier. The function of the first processing unit is to acquire the speech signal (as its name indicates) and to filter possible noises[10] (Huang et al. 2001, Sec. 9.3, p. 419). Next, the input data are usually processed to calculate advantageous features in a rapid way (some kind of dimensionality reduction), but at the same time to maintain valuable discriminatory cues in order to distinguish among classes, because if important information is not considered, the performance of the system can be affected (Bishop 2006, Ch. 1, pp. 2-3). In the last module, the system computes the acoustic models w during the training phase (e.g. word-models) and estimates speech unit probabilities in the decoding or test phase using the features previously calculated as indicated in Fig. 2.5. In this way, the principal challenge of the system is to find the succession of speech units $\boldsymbol{w}= w_1, w_2, ..., w_U$ for a given sequence of observed feature vectors $\boldsymbol{X}=\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_T$ (Huang et al. 2001, Ch. 9, p. 413). In the following sections, we explain the different modules that compound a standard ASR system.

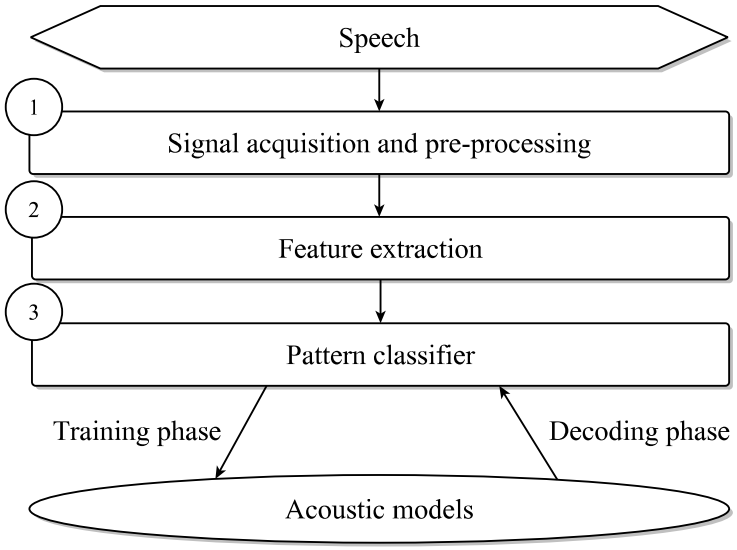


Figure 2.5: *Basic framework of a typical ASR system, where acoustic features are obtained from the entering speech signal and transferred to the pattern classifier (simplified from Huang et al. 2001 (Sec. 1.2.1, p. 5)).*

[8]See the list of references at the end of this thesis.

[9]However, these are examples of recognition systems (mainly programmed off-line by the provider), which have a limited adaptability to the user.

[10]This filtering is out of the scope of this thesis.

2.3.1 Signal acquisition and pre-processing

Speech is generated by the succession of movements of speech organs resulting in a rapidly changing signal (non-stationary) as Fig. 2.1 illustrates (Ambridge and Lieven 2011, Sec. 2.2, p. 14). This acoustic signal, which is a longitudinal pressure wave, is captured by a transducer, e.g. a microphone, and converted into an electric analog signal subsequently digitalized (Huang et al. 2001[11]). According to the sampling theory (Shannon 1949), the digitalization is usually realized with the double sampling rate (16kHz) as the relevant speech bandwidth (8kHz) and an anti-aliasing filter is applied previous to the sampling, where downsampling is also a common practice to reduce computation and the amount of data to process (Huang et al. 2001[12]).

2.3.2 Feature extraction

The feature extraction is one of the most relevant stages in machine learning and classification tasks (Huang et al. 2001, Sec. 9.3.3, p. 423). As mentioned before, the speech signal is converted into a concatenation of feature vectors that contain valuable information to distinguish between the different speech units pronounced (Bishop 2006, Ch. 1, pp.2-3). In order to ease the alignment with the classes the feature extraction module aims at the reduction of the dimensionality, which affects the computation effort of subsequent processing stages as well as the feasibility of the learning problem itself, e.g. see curse of dimensionality (Bishop 2006, Ch. 1, p. 3, Sec. 1.4, pp. 33-38). Furthermore, feature extraction concentrates on specific features for the patterns that can be more easily represented without the variability of the input signal that is not related to the learning problem (Bishop 2006, Ch. 1, p. 2).

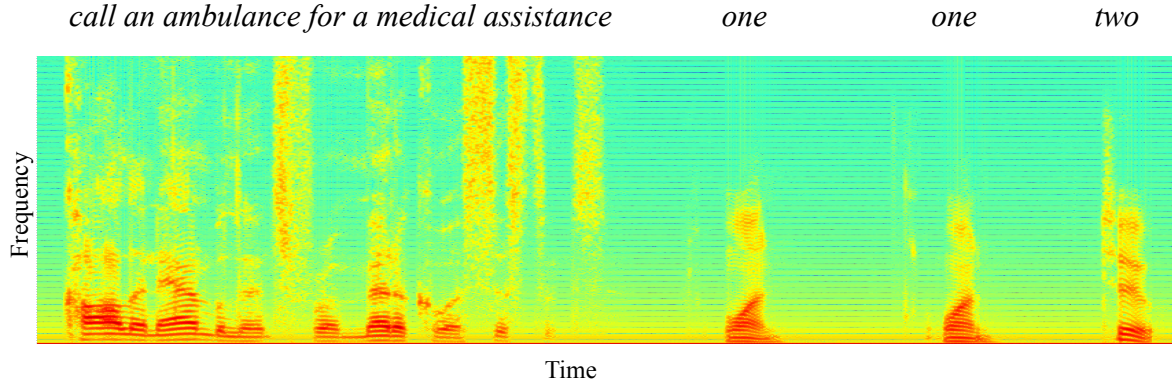


Figure 2.6: *Representation of the speech signal in the time-frequency domain as a spectrogram, where darkest regions of the picture highlight the intensity peaks of the spectrum (adapted from Ambridge and Lieven 2011 (Sec. 2.2, p. 15)).*

[11]Sec. 2.1.1, p. 21; Sec. 5.1, p. 202; Sec. 10.2, p. 482.
[12]Sec. 5.5.2, p. 244; Sec. 6.6, p. 317; Sec. 9.3.1, p. 420.

According to Huang et al. 2001 (Sec. 9.3.3, p. 423), there are a great variety of speech features to employ such as the speech waveform (see Fig. 2.1), although it does not take advantage of all the inherent information. One alternative method of analyzing speech segments is in a spectrogram through a time-frequency plot (Fig. 2.6) where it is displayed how the spectral density of the signal changes over time (Ambridge and Lieven 2011, Sec. 2.2, p. 14). The spectrogram can then be seen as a plot of the magnitude of the short-time Fourier transform of the signal (Rabiner and Schafer 2007, Sec. 4.5, p. 46). In the short-term analysis, this magnitude is estimated on the so called frames in which the parameters to be computed are considered to be roughly constant (Benesty et al. 2008, Sec. 10.2, p. 185).

Apart from the above mentioned ones, the most vastly used features for speech recognition systems are the mel-frequency cepstral coefficients based on cepstral analysis (MFCCs, Davis and Mermelstein 1980) and the relative spectral (RASTA) features, which extend the perceptual linear predictive (PLP) analysis (Hermansky 1990; Hermansky and Morgan 1994). MFCCs usually obtain better recognition scores than RASTA-PLP features on clean signals, however their performance strongly decreases in noisy conditions, letting RASTA-PLP being quite superior in these environments (Domont 2009, Sec. 4.3.2, pp. 52-53).

In order to partially mend the simplification that the signal does not change significantly over a short time span for the analysis and computation of the features, dynamic (temporal) derivatives of the coefficients are usually incorporated (Furui 1981: qtd. in Hermansky 2011; Huang et al. 2001, Sec. 9.3.3, p. 423). These new coefficients are called delta or dynamic and double delta or acceleration features (Hanson and Applebaum 1990). According to Huang et al. 2001 (Sec. 9.3.3, p. 423), the feature vector $\boldsymbol{x}_t$ usually employed in ASR systems is:

$$\boldsymbol{x}_t = \begin{pmatrix} \boldsymbol{c}_t \\ \triangle \boldsymbol{c}_t \\ \triangle \triangle \boldsymbol{c}_t \end{pmatrix} \tag{2.1}$$

where $\boldsymbol{c}_t$ are the feature coefficients, $\triangle \boldsymbol{c}_t$ is the delta features computed from $\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}$ and $\triangle\triangle \boldsymbol{c}_t$ is the double delta features computed from $\triangle \boldsymbol{c}_{t+1} - \triangle \boldsymbol{c}_{t-1}$ (Fig. 2.7).

2.3.3 Classification of speech patterns

As shortly referred in Sec. 2.3, pattern recognition is the scientific field that aims at the classification of data into a number of categories or classes (Theodoridis and Koutroumbas 2003, Ch. 1, p. 1). These data can be normally referred to as patterns or feature vectors, where the vectors that provide the true class employed for designing a recognizer are named training patterns or training feature vectors (Theodoridis and Koutroumbas 2003, Sec. 1.2, pp. 3-5).

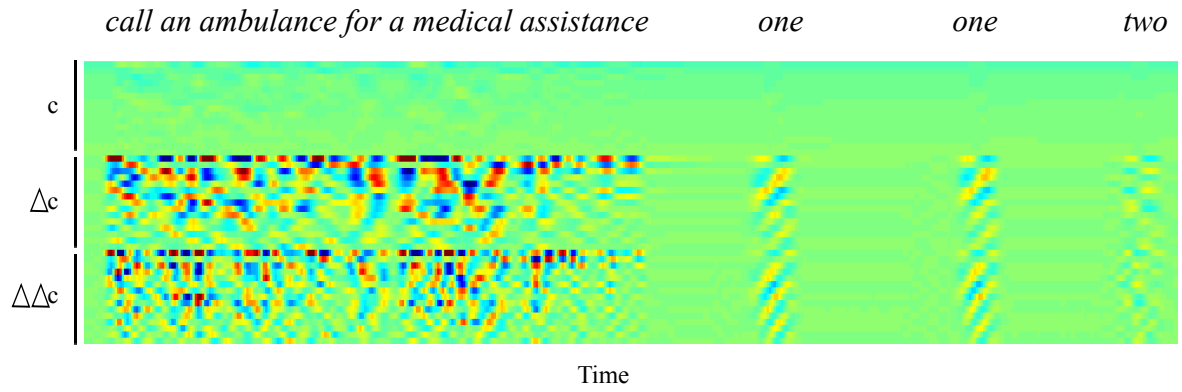


Figure 2.7: *RASTA-PLP speech features (**c**) with their temporal derivatives (△**c**, △△**c**). Temporal variations that appear in the spectra are quite relevant in human perception and are usually incorporated through the computation of delta coefficients (temporal deviation) that determines how the coefficients vary over time (Huang et al. 2001, Sec. 9.3.3, p. 423).*

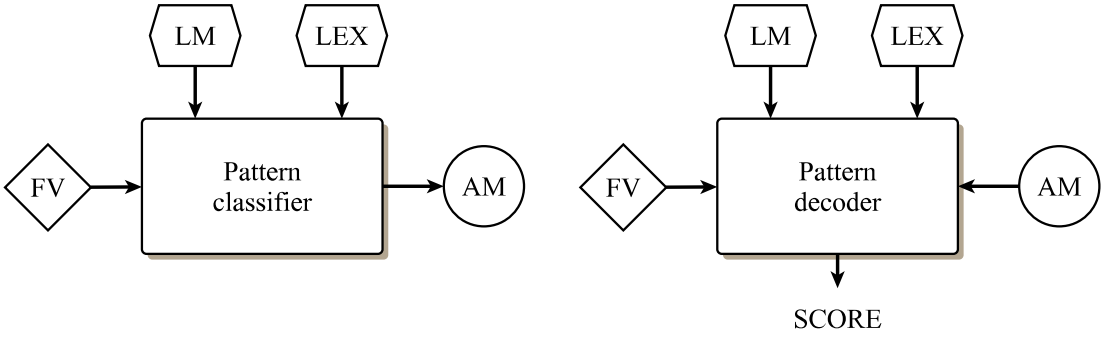
After the model is computed during the training or learning phase, the machine learning algorithm is then prepared to recognize the identity (label) of novel data, which can be part of a test set (including data normally not used in the training phase) pre-processed using the same stages as the training data in order to correctly evaluate the system (Bishop 2006, Ch. 1, pp. 2-6). The challenge of the evaluation phase lies in the fact that, the collection of all possible behaviors given all possible inputs is too large to be comprised by the set of training data (observed examples) and consequently, the learner must acquire the ability, known as generalization, to correctly classify new data, which are different (unseen) from those employed in the learning phase (Bishop 2006, Ch. 1, p. 2).

Bayes decision rule applied to speech recognition

The Bayes decision rule is one of the fundamentals of statistical pattern recognition that enables to evaluate the uncertainty in each class w_i after observing $\boldsymbol{X}$ in the form of the posterior probability $P(w_i|\boldsymbol{X})$ (Huang et al. 2001, Sec. 4.1, pp. 134-135),

$$P(w_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|w_i) \cdot P(w_i)}{P(\boldsymbol{X})}; \quad P(\boldsymbol{X}) = \sum_i P(\boldsymbol{X}|w_i) \cdot P(w_i) \tag{2.2}$$

so that a speech recognizer can select a previously learned class w_i as outcome (greatest probability in the decoding process, see Fig. 2.8(b)) using this theorem (Jiang et al. 1999; Huang et al. 2001, Sec. 4.1, p. 135).



(a) Learning phase (b) Recognition phase

Figure 2.8: *Pattern classifier and decoder for ASR systems (derived from Huang et al. 2001, Sec. 1.2.1, p. 5), being the scheme in (a) the moment when the module operates as learner in the training phase and in (b) when it works in the evaluation stage. AM stands for acoustic models, LM for the language model, FV for the observation or feature vectors* $\boldsymbol{X}$ *and LEX for the lexicon. The statistical relation of all these elements can be described through the Bayes decision rule (Eq. 2.2), where the SCORE is the resulting probability in case of decoding.*

Based on the explanations reported by Huang et al. 2001 and Theodoridis and Koutroumbas 2003, the terms of the equation can be defined and described as:

- $P(\boldsymbol{X}|w_i)$ is a class-conditional probability density function (pdf), which characterizes the feature distribution of each class (Theodoridis and Koutroumbas 2003, Sec. 2.2, p. 14). It is also known as the likelihood function or acoustic probability, because it calculates how likely it is that the acoustic model of the category w_i generates the sample $\boldsymbol{X}$ (Huang et al. 2001, Sec. 4.1, p. 135, Sec. 9.5, p. 437). By means of acoustic models, statistical representations of speech units are built using several training samples (normally labeled) that include information about acoustics, phonetics, the environment and speaker effects among other factors involved in the pronunciation (Huang et al. 2001, Sec. 1.2.1, p. 4). The computation of the acoustic models is explained in Sec. 2.3.5. If the acoustic models do not represent words, a lexicon or dictionary is required to express each word as a concatenation of units (Huang et al. 2001, Sec. 12.2, p. 602) (e.g. medical /m/ /e/ /d/ /i/ /k/ /ə/ /l/).

- $P(w_i)$ is a prior probability known as language model or grammar, which specifies the set of permissible combinations for the language (Huang et al. 2001, Ch. 11, p. 539), i.e. which words are more probable to appear together and then helping to decode utterances in continuous speech (Huang et al. 2001, Sec. 1.2.1, p. 4, Ch. 12, p. 585).

- $P(\boldsymbol{X})$ is the probability that the observation $\boldsymbol{X}$ occurs and as this denominator $P(\boldsymbol{X})$ is generally constant for all categories, the above equation is equivalent to the following one (Huang et al. 2001, Sec. 4.1, p. 135):

$$P(w_i|\boldsymbol{X}) \equiv P(\boldsymbol{X}|w_i) \cdot P(w_i) = F(\boldsymbol{X}|w_i) \tag{2.3}$$

- $F(\boldsymbol{X}|w_i)$ is usually called discriminant function, where a discriminant function $h_{\cdot}(\boldsymbol{X})$ assigns the unknown data $\boldsymbol{X}$ to the category w_i if (Huang et al. 2001, Sec. 4.1.2, p. 138; Theodoridis and Koutroumbas 2003, Sec. 2.3, p. 20):

$$h_i(\boldsymbol{X}) > h_j(\boldsymbol{X}) \quad \forall j \neq i \tag{2.4}$$

being $h_{\cdot}()$ a monotonically increasing function.

Hence, as mentioned before, a natural decision rule would be picking the category w_k with the largest posterior probability $P(w_k|\boldsymbol{X})$ (Huang et al. 2001, Sec. 4.1, p. 135). That is,

$$k = \arg\max_i P(w_i|\boldsymbol{X}) = \arg\max_i P(\boldsymbol{X}|w_i) \cdot P(w_i) \tag{2.5}$$

The rule of the last equation is referred to as Bayes decision rule or minimum-error-rate decision rule (Huang et al. 2001, Sec. 4.1.1, p. 137).

2.3.4 Learning criteria

For the learning or estimation of the acoustic and language models, there are two well-known learning criteria/methods in statistics: maximum likelihood (ML) estimation and maximum a posteriori (MAP) estimation (see references below).

Maximum likelihood (ML) estimation

Maximum likelihood (ML) estimation (e.g. Baum et al. 1970 and Juang 1985) is a statistical estimation method, where it is assumed that the likelihood function $P(\boldsymbol{X}|\lambda_{w_i})$ is defined by a set of parameters λ_{w_i} that are unknown as pointed by Theodoridis and Koutroumbas 2003 (Sec. 2.5.1, p. 28). Based on the further explanations of Theodoridis and Koutroumbas, the target of the approach is to compute these parameters using a set of training samples for each category w_i. With the assumption that samples from one category do not influence the computation of the parameters of the other classes[13], we are able to express and solve this problem for each category independently (Theodoridis and Koutroumbas 2003, Sec. 2.5.1, p. 28).

[13]This assumption identifies the poor discriminative nature of this approach, see Chapter 6.

Being $\overline{\boldsymbol{X}}$=$\boldsymbol{X}_1$, $\boldsymbol{X}_2$, ..., $\boldsymbol{X}_R$ a set of random samples coming from $P(\boldsymbol{X}|\lambda_{w_i})$ and considering statistical independence between the different data, we have (Theodoridis and Koutroumbas 2003, Sec. 2.5.1, p. 28):

$$P(\overline{\boldsymbol{X}}|\lambda_{w_i}) \equiv P(\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_R|\lambda_{w_i}) = \prod_{r=1}^{R} P(\boldsymbol{X}_r|\lambda_{w_i}) \tag{2.6}$$

The maximum likelihood (ML) approach computes the parameters in a way that the likelihood function of Eq. 2.6 achieves its maximum, meaning that (Theodoridis and Koutroumbas 2003, Sec. 2.5.1, p. 28):

$$\widehat{\lambda}_{ML} = \arg\max_{\lambda_{w_i}} \prod_{r=1}^{R} P(\boldsymbol{X}_r|\lambda_{w_i}) \tag{2.7}$$

Maximum a posteriori (MAP) estimation

The parameters $\widehat{\lambda}_{MAP}$, applying the maximum a posteriori probability (MAP) approach, are estimated where $P(\lambda_{w_i}|\boldsymbol{X})$ becomes its maximum value (Theodoridis and Koutroumbas 2003, Sec. 2.5.2, p. 31):

$$P(\lambda_{w_i}|\boldsymbol{X}) = P(\boldsymbol{X}|\lambda_{w_i})P(\lambda_{w_i}) \tag{2.8}$$

$$\widehat{\lambda}_{MAP} : \frac{\delta P(\lambda_{w_i}|\boldsymbol{X})}{\delta \lambda_{w_i}} = 0 \quad or \quad \frac{\delta P(\lambda_{w_i}) \cdot P(\boldsymbol{X}|\lambda_{w_i})}{\delta \lambda_{w_i}} = 0 \tag{2.9}$$

In contrast to the ML, MAP uses the prior $P(\lambda_{w_i})$ for maximization, i.e. some kind of prior knowledge about the distribution (Theodoridis and Koutroumbas 2003, Sec. 2.5.2, p. 31). In the case that $P(\lambda_{w_i})$ is considered to be the uniform distribution (constant for all the classes), both estimates obtain the same results, what can be also roughly extrapolated if $P(\lambda_{w_i})$ shows little deviation (Theodoridis and Koutroumbas 2003, Sec. 2.5.2, p. 31). For the estimation of the parameters of the model, the expectation-maximization (EM) algorithm can be used as in the ML criterion (Huang et al. 2001, Sec. 9.6.1, p. 443).

Additionally, the use of prior information about existing models allow this approach to cope when there is a lack of available training samples (Gauvain and Lee 1991), and in the same way as the amount of training data grows, the prior may lose importance and the parameters estimated via MAP become steadily more similar to the parameters computed by ML (Huang et al. 2001, Sec. 9.6.1, pp. 443-445).

2.3.5 Acoustic modeling of speech

One of the most critical decisions when designing a system for speech acquisition is the selection of a suitable speech representation, what is known as acoustic modeling and it refers to the process of building statistical representations for the speech units using the

introduced speech features[14] (see Huang et al. 2001, Ch. 9, p. 413). Looking inspiration in young infants[15], some researchers have demonstrated that the processing of speech in children is in some degree due to their ability to extract and exploit the statistical attributes of the speech signal (Saffran et al. 1996:qtd. in Versteegh et al. 2011).

According to Bishop 2006 (Sec. 1.5.4, p. 43), these statistical representations can be learned through different approaches such as the generative models, which can learn the distribution of the observations (inputs and outputs) in each class, or the discriminative models, which concentrate on separating one class from another being the models learned by considering the recognition scores (posterior probabilities, see Sec. 2.3.3) or the training samples instead of modeling the data into disconnected distributions as in the case of generative models. As reported by Bishop, the name "generative" comes from the fact that if we extract samples from these models we can generate artificial or synthetic data in the same (input) space as the observations (training data). Examples of generative and discriminative models are hidden Markov models and some neural networks respectively (see Bishop 2006, Sec. 13.2, p. 612).

Apart from the above classification, one can also categorize the algorithms used to learn or train the models according to the wished result in three large categories (summarized from Bishop 2006, Ch. 1, p. 3):

- Supervised learning: This approach computes a model (estimation of its parameters) employing training samples, which are composed of input vectors with their corresponding outputs (also called labels). The most typical supervised algorithms are classification (discrete output) and regression (continuous output).

- Unsupervised learning: These methods also become a set of input data however without their corresponding outputs, i.e. there is no information about the class of each data example. Classical approaches of unsupervised learning are clustering (to find groups/clusters of comparable samples), density estimation (to define the distribution of the examples) or as in visualization, the reduction of the dimensionality.

- Reinforcement learning: This kind of algorithm is related to the problem of learning how to act given an observation with the goal of maximizing a reward (Sutton and Barto 1998). Several states with their corresponding actions are normally employed. In these sequences, it is described how the learning algorithm interacts with the environment. Compared to supervised learning, this method does not operate with optimal outputs, but it must learn them by a procedure of "trial and error".

We are not going to handle the use of feedback or reinforcement learning, but supervised and unsupervised learning. We briefly review the three most extended representations

[14] See Sec. 2.3.3 and Fig. 2.8 for its relation with the Bayes decision rule.
[15] Without aiming at its modeling, see Sec. 3.1.

that use supervised and unsupervised methods (hidden Markov models, e.g. Rabiner 1989; support vector machines, e.g. Ganapathiraju et al. 2004a; artificial neural networks, e.g. Sathya and Abraham 2013) highlighting their advantages and disadvantages for speech recognition.

Support vector machines

Support vector machines (SVMs) are learning methods used for classification, in particular for binary classification problems (Cortes and Vapnik 1995). According to Bishop 2006 (Sec. 7.1, p. 326), the SVM method uses the concept of margin, which is formulated as "the smallest distance between the decision boundary and any of the data samples", so that, to optimize the classification, the decision boundary is computed with the goal of maximizing the margin (minimizing the generalization error). Additional information about the computation of support vector machines can be found, as advised by Bishop, in Vapnik 1998, Burges 1998 and Cristianini and Shawe-Taylor 2000.

One appealing advantage of SVMs is that, the cost function is convex, and so local optima are then global optima (Theodoridis and Koutroumbas 2003, Sec. 3.6.1, p. 81; Bishop 2006, Ch. 7, p. 325). Nevertheless, in general terms, the classical SVM is not very suitable for speech recognition tasks as one can observe in the several aspects described and enumerated by Jiang et al. 2006[16]:

- Firstly, SVMs demands fixed-length feature vectors as samples but speech rate changes and always involves variable-length features. To overcome this obstacle, several authors have investigated the use of suitable and sometimes sophisticated kernels that have the ability to project a variable-length feature vector into a fixed-length one. A subset of kernels studied for this purpose are the proposed by Smith and Gales 2001 and Moreno and Ho 2003.

- Secondly, standard SVMs were originally conceived for binary classification problems as stated above. However, ASR systems have to handle multi-class problems. A heuristic technique is to construct a set of binary SVMs for all categories based on the "one vs. one" or "one vs. rest" methods. During the verification stage, the recognition is realized through the combination of multiple local decisions that come from the sets of binary SVMs using strategies such as "majority-voting" or "winner-take-all". Other proposed techniques to formulate multi-class problems are the so-called k-class SVM as in Crammer and Singer 2001 as well as Arenas-Garcia and Perez-Cruz 2003.

- Finally, SVM classifiers are static. This means that they cannot efficiently deal with the dynamic nature of speech, so it is difficult to decode continuous speech as

[16] Jiang et al. 2006 used these argumentations in another context, but they can be also valid here.

it is unclear where the words start and end. There are hybrid solutions for that explained in Sec. 6.1.

Artificial neural networks

Artificial neural networks (ANNs) are learning algorithms inspired by the way biological frameworks can process information (see McCulloch and Pitts 1943; Widrow and Hoff 1960; Rosenblatt 1962; Rumelhart and McClelland 1986) to be used for statistical pattern recognition and for estimating sophisticated probability distributions as reported by Bishop 2006 (Ch. 5, p. 226). From the practical point of view of pattern recognition, Bishop stated that a strictly biological imitation would lead to needless limitations, so these conditions are often relaxed in benefit of the application. One of the best known and successful ANNs in pattern classification referred by Bishop are the feed-forward neural networks, also called the multilayer perceptrons (see Hinton 1989). Other extensively used ANNs for this purpose are the recurrent neural networks (RNN, see Burrows and Niranjan 1994) and time delay neural networks (TDNN, see Waibel and Lee 1990, Sec. 7.2, p. 393).

In some cases, one significant advantage of ANNs is that the model obtained is more compact and faster to evaluate than a SVM showing the same generalization performance, however the likelihood function of an ANN is no longer a convex function as in SVMs (Bishop 2006, Ch. 5, p. 226). ANNs achieve good performance for small vocabulary ASR systems despite that they cannot compete with the effectiveness of other methods (see HMMs) for coping with non-stationary signals (Huang et al. 2001, Sec. 9.8.1, p. 456). When the time-evolving nature of speech has to be addressed, e.g. alignment of not synchronous sequences, segmentation and classification, the standard ANNs are not properly qualified to deal with these challenges (Huang et al. 2001, Sec. 9.8.1, p. 455).

As reported by Brandl 2009 (Sec. 6.1.2, p. 67), regular ANNs frameworks are based on batch schemes that are not able to ensure model adaptation, incremental addition of new classes or to deal with a great amount of classes without a significant degree of manipulation. Additionally, Brandl stated that the average of frames obtained after feature extraction associated with a word is normally more than 100, a number that is not usually tractable for typical RNN approaches limiting their application range.

Hidden Markov models

Hidden Markov models (HMMs) are a widely used and successful statistical method of "characterizing the observed data samples of discrete time series", e.g. speech, as a parametric random process (Huang et al. 2001, Ch. 8, p. 375). Regarding the model structure, there are some significant similarities to the previously introduced methods, so that HMMs can be considered as a special type of linear neural network (Baldi and Chauvin 1994). Nevertheless, compared to ANNs, HMMs are one of the most prominent

approaches for modeling speech signals[17] (Huang et al. 2001, Ch. 8, p. 375).

According to the reasoning of Brandl 2009 (Sec. 6.1.2, p. 67), HMMs present the following advantages. Firstly, languages own a huge amount of words, which increase over time (the open-ended process of language acquisition). HMMs are able to work in these conditions supporting the scalability required. Secondly, HMM toolboxes constitute effective and simple computational frameworks (e.g. HTK, Young et al. 2009, Appendix B). Finally, the search algorithms employed in HMM frameworks enable the incorporation of new representations even in the course of the decoding phase.

In spite of their flexibility, computational efficiency and success in ASR systems, HMMs present some disadvantages or limitations (Fujimura et al. 2010). For instance, HMMs do not represent suitably the temporal structure of speech, because the probability of being in a hidden state drops rapidly with time as reported by Huang et al. 2001 (Sec. 8.5.1, p. 404). Huang and colleagues also added that due to the conditional independence and the first-order assumptions in standard HMMs, these are not able to consider strong correlations between separated observed variables. Several authors have proposed a number of approaches[18] to cope with these three limitations, e.g. Levinson 1986, Russell and Moore 1985, Mari et al. 1997, Deng et al. 1994, Gish and Ng 1993 as well as Fujimura et al. 2010. Nevertheless, the improvements obtained by these techniques were quite modest for practical applications compared to the significant increment of model parameters, the increased computational complexity and the need of more training samples (Huang et al. 2001, Sec. 8.5, pp. 403-407).

2.3.6 Utterance verification

At the end of the decoding stage, the algorithms provide the most likely succession of models for a given input (see Huang et al. 2001, Ch. 9, p. 413). Nevertheless, as the result may sometimes be wrong, it is crucial to have one suitable confidence measure as component in an ASR system in order to mend recognition errors as well as identify keywords or reject out-of-vocabulary (OOV) units as reported by Huang et al. 2001 (Sec. 9.7, p. 451).

According to Huang and colleagues, one good measure to be used is the posterior probability $P(w|\boldsymbol{X})$ (see Bayes decision rule, Sec. 2.3.3) when the ASR system considers $P(\boldsymbol{X})$[19], which is usually disregarded as it is a constant when estimating $P(\boldsymbol{X}|w) \cdot P(w)/P(\boldsymbol{X})$ for the whole set of words (it does not depend on w).

In the last decade, several methods for computing measures derived from a recognition

[17] A detailed introduction in this approach is subjected to Chapters 3 and 4.

[18] Some researchers have also attempted to combine the advantages of ANNs, SVMs and HMMs in hybrid systems. Some examples are mentioned in Sec. 6.1.

[19] One advantage of the generative approaches (e.g. HMMs) is that they allow the computation of the observation probability $P(\boldsymbol{X})$ (Bishop 2006, Sec. 1.5.4, pp. 43-44).

hypothesis have been reported to improve confidence estimation, where an overview and a review of these methods can be found in Jiang 2005. Other interesting metrics are investigated in Jianjun and Xingfang 2007, Bardideh et al. 2007, Vaisipour et al. 2007 and Huang et al. 2008.

2.3.7 Performance metrics

When choosing the best suitable framework for our application, the evaluation and comparison of the performance of ASR systems is crucial as stated by Huang et al. 2001 (Sec. 9.2, p. 417). Huang and colleagues stated that the test set of the database should contain completely unseen samples with respect to the training and parameter tuning sets and should also comprise more than 500 utterances from 5 to 10 different speakers. The comparison of the output of the recognizer with the labels of the data sets is realized using dynamic programming and the resulting errors can be classified as (Huang et al. 2001, Sec. 9.2, p. 418):

- Substitution: a correct word becomes substituted by an incorrect word.
- Deletion: a correct word was omitted in the recognized utterance.
- Insertion: an additional word was included in the recognized utterance.

The above mentioned errors are integrated into the word error rate (WER), which is usually employed in percent and extensively used as one of the most important performance measures in ASR systems (Huang et al. 2001, Sec. 9.2, p. 418):

$$WER = \frac{Substitutions(S) + Deletions(D) + Insertions(I)}{Number\ of\ words\ (NW)} \cdot 100\% \tag{2.10}$$

By comparing different ASR systems, it is important to evaluate their relative error reduction, so that if a new algorithm outperforms another in more than 10% relative error reduction, we can consider adopting it instead of the current one (Huang et al. 2001, Sec. 9.2, p. 417).

2.4 Limitations of a batch process for incremental learning

In the previous sections, we introduced different criteria for learning approaches so that they can be classified into generative or discriminative, supervised or unsupervised methods. Here, we expand these criteria to the groups of batch and incremental learning.

When teaching human language to an artificial agent as explained in Chapter 1, the framework should be able to learn incrementally new terms. However, the conventional state-of-the-art ASR system presented before in Sec. 2.3 and Fig. 2.5 is conformed to batch

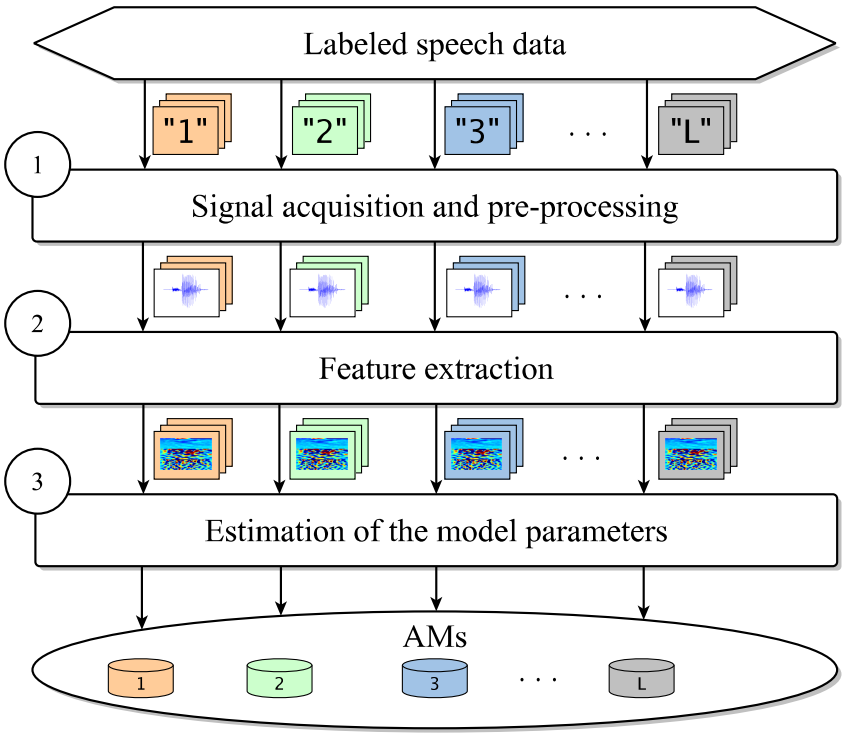


Figure 2.9: *Example of a batch system (based on the simplification pictured in Fig. 2.5, see Ayllón Clemente et al. 2012). Here, the whole set of the labeled training data are introduced into the framework at once and all the models are estimated after each training step of the system as referred in the text. Each model with its corresponding training samples used is represented by a different color. In the example presented, three training data per model are displayed. The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a).*

schemes that do not fulfill the incremental learning property referred above. In batch processes or techniques (illustrated in Fig. 2.9), the system is provided with the whole training set at once and all the models are jointly estimated (Bishop 2006, Sec. 3.1.3, pp. 143-144) in contrast to interactive learning (Fig. 2.10, also mentioned in Ayllón Clemente et al. 2012), where the data samples are provided incrementally and often with some kind of visual stimulus or pseudolabel for the model to learn, e.g. the approaches proposed by Roy 2003, ten Bosch et al. 2009 or Iwahashi 2007.

2.5 Incremental learning systems

In this section, we provide the reader with a survey on incremental learning approaches concentrated on the recognition and detection of words in the speech stream (based on Ayllón Clemente et al. 2012):

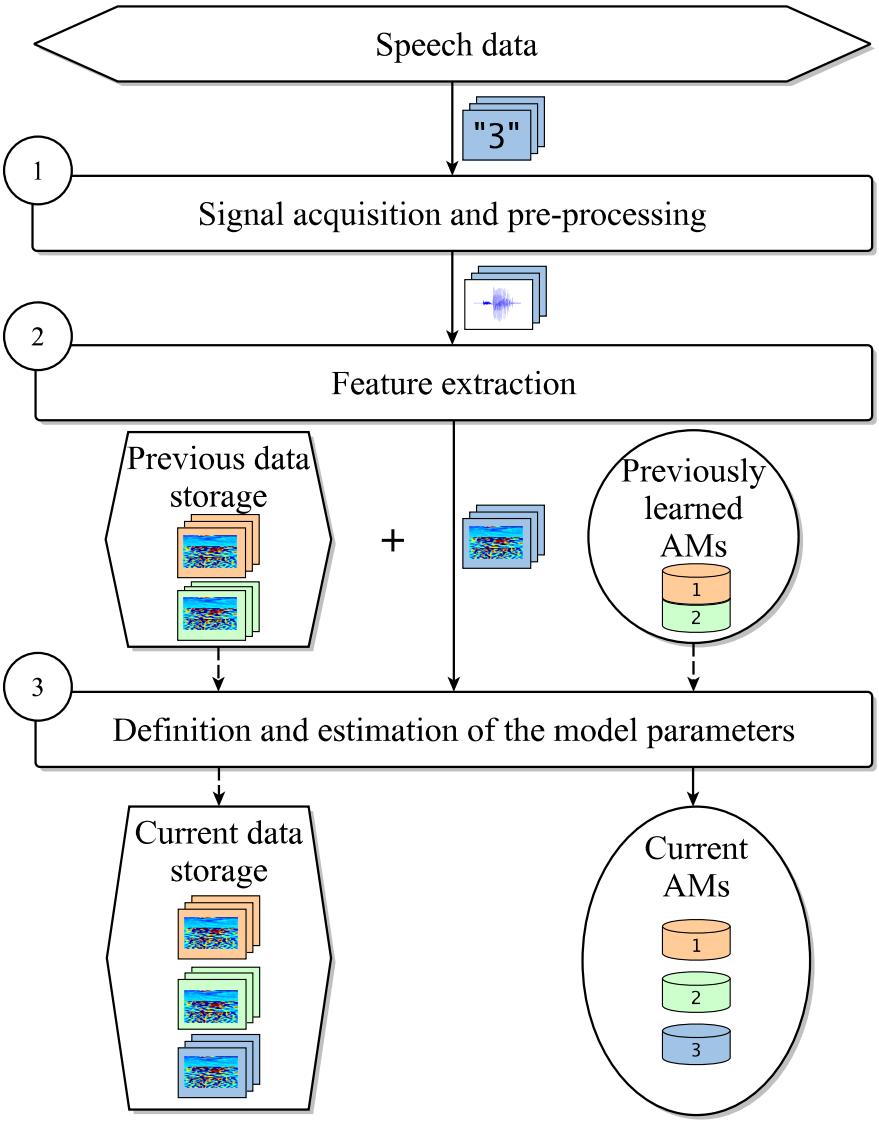


Figure 2.10: *Example of an incremental system for a given iteration (see Ayllón Clemente et al. 2012). Here, only the labeled training data belonging to the category to learn are introduced into the framework and only the model to learn is estimated at a determined time step, although in some refinement stages the other models can be adapted. As in Fig. 2.9, each model with its corresponding training samples used is represented by a different color and three training data per model are displayed. The dashed lines display data that can be used and/or adapted (depending of the particular application) in this determined time step. The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a).*

Roy 2003: The system presented is called CELL (cross-channel early lexical learning), a "computational model of sensor-grounded word learning", which acquired words and associates them visually with objects. This visual input can be also used as a kind of speech labels. This approach addressed the "segmentation of continuous spontaneous speech without a pre-existing lexicon". However, a recurrent neural network (RNN) trained off-line for sub-word units recognition with the TIMIT database (Appendix A.1) was used to facilitate the segmentation process. Additionally, the statistical models to represent spoken words are based on hidden Markov models (HMMs). In CELL, the system is able to produce a spoken response. For the evaluation, a corpus was collected. It contained approximately 7600 utterances uttered by six different speakers. These samples were recorded through experiments with caregivers playing with pre-linguistic infants.

ten Bosch et al. 2009: The ACORNS project[20] aimed to develop computational models that demonstrate the capability to acquire language and communication skills based on the sensory input (multimodal stimuli). One of the challenges of the project was in contrast to CELL, to avoid the use of predefined representations for decoding the information in the speech signals. In this system, the learning agent obtained a feedback from the "caregiver" according to the response to the previous stimulus. Several methods, e.g. non-negative matrix factorization (NMF), were integrated in the incremental learning system to extract patterns (word discovery) from utterances. In the experiments, the speech database contained one proper name and nine words that name common objects. The proper name that the caregivers used when they speak to the learner is one per caregiver. The corpus was recorded in three different languages: Dutch, Finnish and Swedish. The database of each language consisted of sentences from 2 male and 2 female speakers. Each speaker pronounced 1000 sentences in ADS and CDS mode (a total of 2000 sentences per speaker). This set of 1000 utterances also comprised 10 repetitions combining 10 target words and 10 carrier sentences per speaker.

Van hamme 2008: Also in the context of the ACORNS project for language acquisition explained above, a bottom-up (activation-based) representation for continuous speech recognition has been introduced. The framework was built on histograms of acoustic co-occurrence (HAC-models) with their related learning method based on non-negative matrix factorization (NMF). In the presented system, HAC-models were extended to detect not only the words in an utterance, but also their order in the sentence. Additionally, the system introduced a sliding window (activation-driven) decoder. The framework was evaluated with the TIDigits database (see Appendix A.2). Although the proposed methods presented very promising results, HMMs still outperformed it.

[20]More information can be found in ACORNS Project (see the list of references at the end of the thesis).

Markov and Nakamura 2007: The proposed approach consisted of a network composed of hidden Markov states, which could realize unsupervised on-line adaptive learning while keeping the already learned information. Speech patterns were modeled by state sequences in the network. The system could detect new patterns and if it occurred, new states and transitions were attached to the network. The structure of the network varies through the growth (adding new states) and shrinkage (removing states that are hardly visited). The framework was called never-ending learning, because the learning procedure could continuously learn until the network persists. In their experiments, they recognized sentences containing single samples of 22 English letters uttered by several Japanese speakers. The speech data set was composed of 440 speech sentences.

Iwahashi 2007: This work presented a developmental approach for language processing to be used in interaction with artificial agents. The system was able to learn words related to objects and motions of moving objects without being initialized with knowledge associated to these objects, how to move them nor any prior linguistic information. To learn these concepts, the tutor pointed to an object or moved it in a determined way while uttering a word describing the object or the movement. In the proposed approach, the acquisition of words is based on a HMM framework. Next, the system iteratively acquired some simple grammar that enables the agent to understand not complicated sentences. Additionally, the system could detect known and unknown words with the help of speech and visual information. When the agent could not decide if the word was already learned, it asked the user for feedback. To obtain good performance, the system was trained using numerous interaction episodes. Additionally, this system is also able to learn a pragmatic capability. In this learning stage, not only the user but also the agent could ask the tutor to realize an action. In cases of incorrect behavior of the agent, the user provides the system with feedback through a touch sensor.

Brandl et al. 2008: These authors proposed acoustic modeling through unsupervised and supervised techniques inspired by some aspects of language acquisition in children and using as basis the initialization of the models proposed by Iwahashi 2007. The approach suggested by the authors incorporates a regulation scheme assuring asymptotic homeostasis evaluated on a data set containing isolated words. The system, operating in a HMM framework, also incorporated a syllable spotter. When only 5 training samples were used, the system performed quite well in a limited speaker-dependent scenario. In the on-line evaluation stage when integrated with the framework presented by Mikhailova et al. 2008, the tutor presented an object to a robot while uttering an already learned monosyllabic word describing the object. If the word was associated with the object, the robot answered the tutor by means of head nodding.

2.5.1 Comparison of the methods

Most of these systems were trained using a large amount of data besides being helped through visual confidence and feedback. Related to the use of feedback, in the case of Roy 2003, the agent was able to provide a spoken reply. In ten Bosch et al. 2009, the learner obtained feedback from the caregivers according to the response to the previous stimulus. Similarly, Iwahashi 2007 proposed a system where both parties (learner and tutor) interchanged inquiries. This system also enabled a physical feedback, e.g. touch sensors. The integration of the system proposed by Brandl et al. 2008 with the framework presented by Mikhailova et al. 2008 also included a response from a robot in the form of head nodding/shaking.

A similarity between Iwahashi 2007 and Brandl et al. 2008 is the use of an unsupervised (without transcriptions) bootstrapping phase (to build sub-word units). Both works were not focused on speaker-independent recognition and like Markov and Nakamura 2007 as well as Roy 2003, they also implemented their approaches in a hidden Markov framework. On the other hand, Van hamme 2008 investigated novel approaches different from HMMs however still without outperforming them.

One notable difference between the approach proposed by Roy 2003 and the methods applied by the rest of researchers such as ten Bosch et al. 2009, Iwahashi 2007 and Brandl et al. 2008 is the use of prior knowledge. While Roy 2003 employed a previously trained recognizer[21] to ease the segmentation task, ten Bosch et al. 2009, Iwahashi 2007 and Brandl et al. 2008 targeted at scenarios with little prior knowledge (e.g. by avoiding the use of predefined representations or transcriptions, unsupervised bootstrapping).

To sum up, one of the main goals of the approaches enumerated above is to endow a system with some language in an incremental learning fashion. Nevertheless, to our knowledge (also argued in Ayllón Clemente et al. 2012), one key aspect not directly addressed by these techniques is the jointly reduction of the number of repetitions not overspecializing to one individual tutor during the training while maintaining good performance. As stated in our previous work (Ayllón Clemente et al. 2012), a system that achieves this is much more intuitive and user-friendly through the reduction of the required tutoring time.

2.6 Summary and concluding remarks

The aim of this chapter is to provide the reader with a first hint in the terminology and fundamentals of speech recognition and its community before introducing our proposed framework and methods.

We opened the chapter by describing the basic properties of speech, especially its time

[21] As mentioned above, Iwahashi 2007 and Brandl et al. 2008 initialized their systems in an unsupervised manner with sub-word models to build later word-models during the learning procedure.

series nature and the different linguistic units, e.g. phonemes, syllables and words. As referred in Sec. 2.1.1 and 2.1.2, one of the challenges of learning in continuous speech vs. isolated words, or using phonemes vs. words as speech unit models is coarticulation, which occurs when two units are uttered concatenated and their boundaries (when a unit starts and the latter one ends) are not clearly defined (Reddy 2001).

No speech recognition system has achieved human performance until now (Furui 2009). Human infants learn through the interaction with their parents and caregivers, which use a special register when talking to their children, called children directed speech (CDS), aiding the learning of the child (Dominey and Dodane 2004). Related to the number of repetitions that a child needs to learn a word, it seems that the number of exemplars required for understanding a term is lower than for pronouncing it in spite of the quite little research on this topic (Tomasello 2003, Sec. 3.2.4, p. 79).

After the brief account of speech and how children acquire language, we explained the central elements of standard automatic speech recognition (ASR) systems. Firstly, we described how the recording of the speech sounds and their digitalization occur. This digitalization stage eases the extraction of the most salient features of the segments as preparation for the next processing stage: the recognizer, decoder or classifier (Huang et al. 2001, Sec. 1.2.1, p. 5). In this context, the Bayes theorem or decision rule is crucial in the field of pattern recognition and machine learning (Bishop 2006, Sec. 1.2, p. 15). By means of this theorem, acoustic and language models are employed by the decoder to provide a score (posterior probability) for the decision of the assignment of an observation (feature vectors) to a determined class (Huang et al. 2001, Sec. 1.2.1, p. 5). The parameters of the acoustic and language models can be carefully estimated using learning criteria such as maximum likelihood (ML) and maximum a posteriori (MAP) estimation as mentioned previously in Sec. 2.3.4 (see Theodoridis and Koutroumbas 2003, Sec. 2.5.1-2.5.2, pp. 28-31). For the choice of the acoustic models, we distinguished in this chapter several categories of approaches, namely generative vs. discriminative and supervised vs. unsupervised learning techniques. In Sec. 2.3.5, three statistical methods belonging to these classes are shortly described with their advantages and disadvantages: support vector machines (SVMs), artificial neural networks (ANNs) and hidden Markov models (HMMs). After the classifier and previous to the final output, it is possible to apply an utterance verification algorithm employing confidence measures in order to determine the correctness of the output of the classifier and so to improve the quality of the system usually evaluated by means of performance metrics as the word error rate (WER) (see Sec. 2.3.6 and 2.3.7; Huang et al. 2001, Sec. 9.2-9.7, pp.418-451).

Although this conventional ASR systems are well suited for a wide range of applications as mentioned in Sec. 2.3, due to its predominant off-line/batch learning architecture, they do not fully allow the incremental acquisition of new terms. These facts motivated various authors to propose approaches in order overcome this handicap (see Sec. 2.5), however those methods do not directly target at a low tutoring time, i.e. reduction of the

training data, together with a speaker-independent learning system (most of them are based on the interaction with a specific tutor or teacher) as also argued in our previous work (Ayllón Clemente et al. 2012). This existing lack motivates us to look for systems (see the previous works in Appendix D) that aim at a direct handling of these restrictions in order to favor the conception of user-friendly agents.

3
An efficient incremental word learning system

In the preceding chapter, the fundamentals of speech recognition were introduced. In particular, we explained standard ASR systems, which are primarily composed of batch schemes that are well suited for applications trained previously off-line, although they are not appropriate to acquire speech in an interactive learning scenario (see Bishop 2006, Sec. 3.1.3, pp. 143-144). Hence, different authors (enumerated in Sec. 2.5) develop novel systems in order to cope with the needs of such complex scenarios. However, these new methods do not directly target at the user-friendly feature of decreasing the tutoring time (i.e. employing a smaller number of repetitions) and dealing with a speaker-independent context at the same time, thus, as argued in our previous work (Ayllón Clemente et al. 2012), we take into account both aspects in a user-friendly learning framework adequate to incrementally acquire language in a partially unsupervised way and to require a low tutoring time.

Outline of this chapter

In this chapter, we give an overview of our efficient incremental word learning (IWL) system defining each processing stage as well as the characteristics and settings of our framework. Firstly, we describe the speech input data and language model of our learning framework, enumerating which speech units and type of speech we use. The central part of the system is the set of acoustic models. Here, we explain why we choose HMMs as acoustic models, how they are defined and the algorithms employed for the learning, decoding and evaluation procedures of these methods. We continue with the exposition of the training and recognition phases, where the model definition, the initialization and the discriminative training of the models are only briefly introduced. All of them are subjected to succeeding chapters (4, 5 and 6 respectively). Finally, the implementation details of the system are presented together with the evaluation metrics and benchmarks. Some related basic principles and state-of-the-art methods explained as well as the framework proposed in this chapter are based on the previously presented work in Ayllón Clemente

and Heckmann 2009, Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012[1].

3.1 Architecture of the system

As stated by Hörnstein and Santos-Victor 2010, children can acquire remarkable language skills from few repetitions by interaction with their parents in a flexible way. This fact motivates researchers to attempt to imitate the manner infants acquire language, despite that it is hardly known how children do this or if this capability is already partially pre-programmed in our brains as pointed by Hörnstein and Santos-Victor.

This implies that without forgetting the infant language acquisition process, a criterion for developing agents with language skills might be no longer the biological (or psychological) plausibility but the effectiveness of the application (Seabra Lopes 2007). This idea is compatible with a more practical and relaxed vision of artificial intelligence (AI), where a set of efficient and successful approaches can be used to imitate a human capability instead of simulating human intelligence at a whole[2] (Seabra Lopes 2007). Likewise, our framework proposes and combines novel and well-known promising statistical methods to approach the performance achieved by infants. For this purpose, we inspire[3] our system by how children learn their first language without attempting to simulate or model this process.

In Fig. 3.1, a simplified scheme of the design of the system is illustrated. The system described in this work is capable to learn incrementally as it has the ability to add new terms to its vocabulary. In our incremental word learning (IWL) system (see Ayllón Clemente et al. 2012), all the training samples do not enter the system at the same time, but only the samples relative to the word-model to be learned for that time step as instead of computing all the models again, only the estimates of the current word-model are learned not needing to re-estimate all the previous learned models when training samples of a still unseen class are provided to the system.

The framework is composed of four main functional modules in addition to the signal acquisition and feature extraction components common to the standard ASR systems (see Ayllón Clemente et al. 2012): the definition of the word-model, an initialization or model bootstrapping phase, the learning stage or also called re-estimation of the parameters (in the bootstrapping phase we initially estimate these values) and a discriminative adaptation stage, see Sec. 3.4. At the final stage, different large margin discriminative training strategies refine the observation estimates of the current model to separate it from the models learned before (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). In the following sections, we will explain each of these steps that configure our system and a description of the main characteristics of it.

[1] Some passages are extracted verbatim from these sources (see Sec. 1.1.1).
[2] As strong AI aims at (see e.g. Kurzweil 2005).
[3] Based on simplifications and/or assumptions.

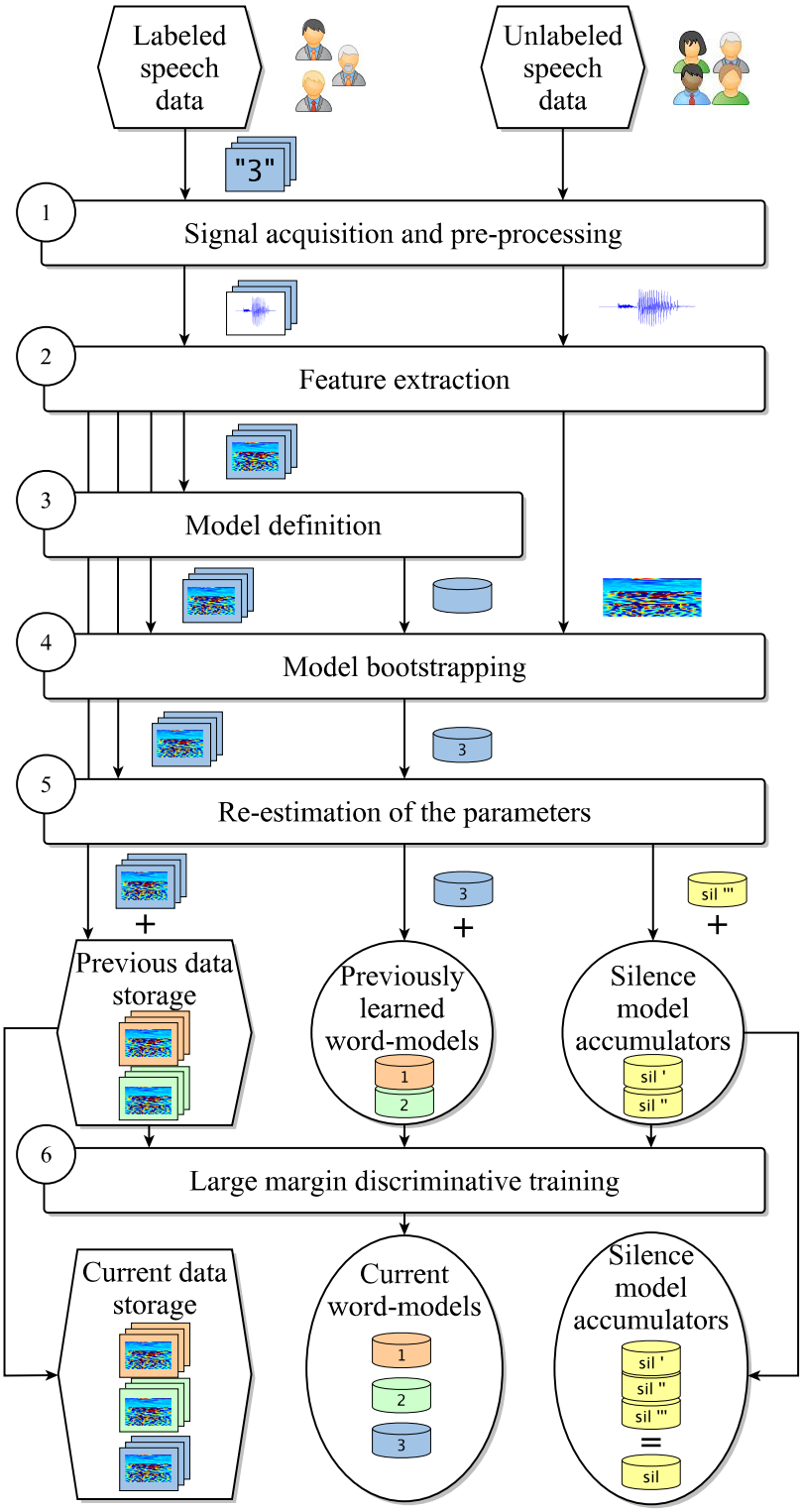


Figure 3.1: *Overview of our IWL framework for one learning iteration (similar to the one displayed in Ayllón Clemente et al. 2012). In this snapshot, some models (1 and 2) have been already learned and a new class (3) is added to the system. A description of each phase will be given in the following sections. The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a).*

3.2 Learning data and language model (LM)

In this section, we start presenting the speech units and different data sources employed in our system. Next, we explain how the general signal processing modules, namely signal acquisition and feature extraction, of standard ASR systems are applied in our framework. Finally, we introduce our language models or grammars for the learning and recognition phases respectively.

Speech units

In our system, the speech units (modeled by AMs) are words as they are the main vehicle of meaning, the smallest unit to be used independently and are more robust against coarticulation (Huang et al. 2001, Sec. 9.4.1, p. 427). The use of words in an interactive learning scenario is also beneficial as they can be employed to name the objects present in the scenario (see examples in some of the systems referred in Sec. 2.5). So, our framework is named incremental word learning system.

Data sources

Actually, the efficiency of infants compared to machines in language acquisition is mainly due to the conditions (the quality of the data) in which the former learn (Hörnstein and Santos-Victor 2010). For this reason, the training samples supplied to our system resemble some aspects of the data that infants perceive at the beginning of their lives[4] (CDS, referenced in Sec. 2.2), e.g. isolated words and optimal infant learning conditions such as quiet environments:

- The training samples entered into the system contain isolated words. As mentioned in Sec. 1.1, we assume that the utterances are already segmented into words. Although our training phase takes place in an isolated word learning scenario, our application framework is quite challenging since we aim at a speaker-independent context and continuous speech recognition (CSR) once the words are learned (recognition phase). The performance of the recognizer is worse in the case of continuous speech, as natural, casual continuous speech is more difficult to recognize than isolated words (Huang et al. 2001, Sec. 18.3.1.4, pp.914-915). Consequently, it cannot be assured that a word that has been learned in isolation will be also recognized in continuous speech.

  Apart from the terms to learn in isolation, the system receives another speech input containing unlabeled continuous speech utterances to be used in an unsupervised learning step in the initialization phase. This input does not contain the words to

[4] As mentioned in Sec. 3.1, we do not aim at modeling this process (still not completely understood as referred in Sec. 2.2), we will take only inspiration in issues that could ease the acquisition of words in our framework ("effectiveness of the application") as mentioned above.

learn and the utterances are produced by speakers from different dialects, ages and genders. More details are given in Sec. 5.2.1 and 5.3.

- The recordings provided to the system are clean of noises. The range of different possible environments is so large that it is not feasible to train our framework for all conditions and moreover, although children are able to learn with the presence of noise, infants learn new words faster in quiet conditions and while growing up, they are progressively able to also recognize speech in challenging conditions (Domont 2009, Sec. 1.1.2, p. 2; see also Nozza et al. 1990; Bronzaft 1997).

Data processing

The first pre-processing steps in order to determine the features or speech characteristics as input for our learning framework are computed similarly to the conventional ASR systems explained in Sec. 2.3.

It is very difficult to have pure clean conditions in an interactive scenario (see Wang and Pols 1997), so in the feature extraction module, we process RASTA-PLP features, which are quite suitable features to represent the speech spectrum as they clearly outperform the MFCC features in almost all conditions and have proven their good performance in ASR systems (see Domont 2009, Sec. 4.3.2, pp. 52-53).

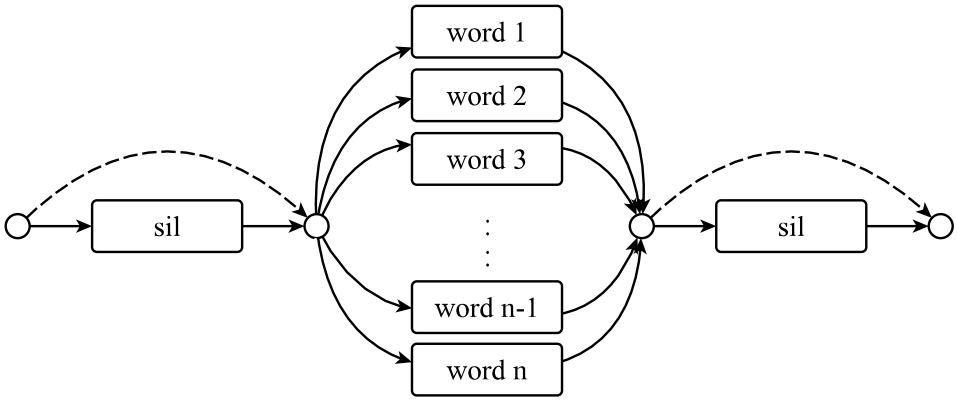
In our experiments[5], we extract 45 dimensional acoustic feature vectors comprising 15 RASTA-PLP coefficients (Hermansky 1990; Hermansky and Morgan 1994) and their first and second order time derivatives, where previously the data is sampled at a rate of 16 kHz (see Ayllón Clemente et al. 2012).

Language model (LM)

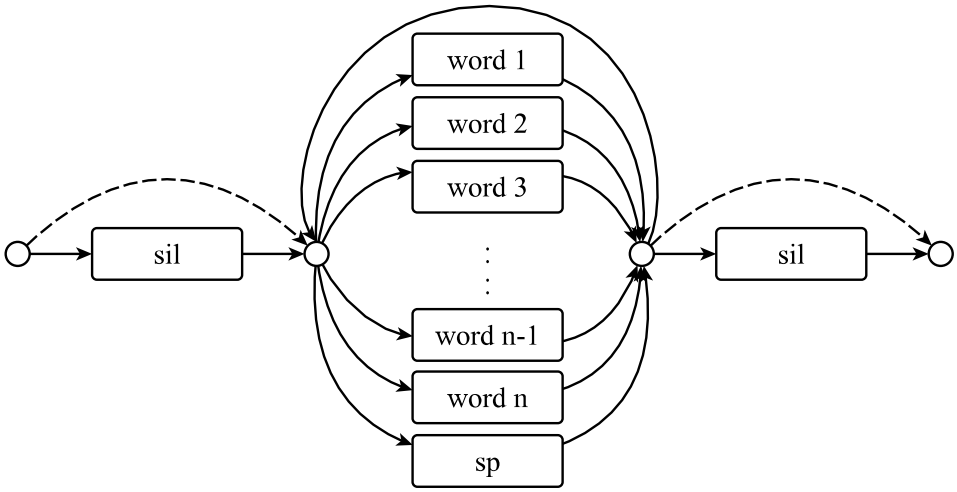
Looking backwards to the Bayes decision rule of Sec. 2.3.3, the language model defined in our architecture is fixed according to the Fig. 3.2(a) for training (isolated words) and Fig. 3.2(b) for testing (continuous speech) respectively. In the testing stage, isolated words can be also recognized.

According to the constructivists, infants do not possess any innate grammar when they are born (see Sec. 2.2; Ambridge and Lieven 2011, Sec. 1.1.1, p. 2). Similarly, our system does not possess prior knowledge of the world, what makes our language model a uniform unigram, which if the framework comprises only two words at a given moment, w_1 and w_2, then: $P(w_1) = P(w_2) = 1/2$ according to Huang et al. 2001 (Sec. 12.2.4, p. 604). Hence, each time a novel word is learned in our system, it incorporates the new term into the language model, which recalculates the distributions.

[5]See also the experiments realized in our previous works (Appendix D).



(a) Learning isolated words



(b) Recognition of continuous speech

Figure 3.2: *Language models or grammars used in our framework (graphical representation inspired in Fig. 12.4 of Young et al. 2009). The LM in (a) is for the training phase where isolated words are presented, e.g. sil - word 2 - sil and the LM in (b) for recognition. In the latter, the grammar is very generic allowing all type of mixtures, e.g. sil - word 2 - sil, word 3 - word 4 - sil, word 5 - sp - word 6. Sil represents the silence (or long pause) model and sp a short pause that can be repeated so many times as required. For each newly learned term, the lexicon and language model are updated.*

3.3 Acoustic models (AMs)

As mentioned in Ayllón Clemente et al. 2012, the process of learning speech units can be explained by the construction of computational models in order to map the speech signals on discrete representations (ten Bosch et al. 2009). In the last decades, hidden Markov models (HMMs) have become one of the most prominent and widely used statistical methods for ASR systems despite that, as Huang et al. 2001 (Sec. 8.4-8.5, pp. 396-407) stated, HMMs present some drawbacks[6] such as the need of a suitable initialization of the parameters, the great required quantity of samples, the duration modeling of the speech units, the first-order Markov chain assumption and the conditional independent assumption (see next section). Therefore, researchers look for modeling novel approaches (e.g. Boves et al. 2007), the chance to build hybrid systems (e.g. Morgan et al. 1993 and Ganapathiraju et al. 2000) or the inclusion of more parameters (degrees of freedom) to resolve the last mentioned drawbacks by increasing the number of training samples required (Bilmes 2006). Thus, the remaining limitations are the need for a considerable quantity of training data and their initialization as we last mentioned in Ayllón Clemente et al. 2012. Regardless of the known limitations, HMMs are one of the most used AMs for speech recognition systems being included in numerous successful applications, e.g. Google Search by Voice in different languages (Schuster and Nakajima 2012; Biadsy et al. 2012). This is the main reason why some authors as Austermann et al. 2010, Iwahashi 2007, Brandl et al. 2008 and Hörnstein and Santos-Victor 2010, employ them in their interaction scenarios.

These facts encourage us to take HMMs as the acoustic models of our system. The choice of the model parameters and the word-model topology for the limited training data learning conditions are explained in Chapter 4. Next, we explain these statistical models with their most relevant algorithms employed in the training and test phase as well as their general schemes. This introduction to HMMs is founded on the concepts explained in Rabiner 1989, Huang et al. 2001, Bishop 2006, as well as Theodoridis and Koutroumbas 2003.

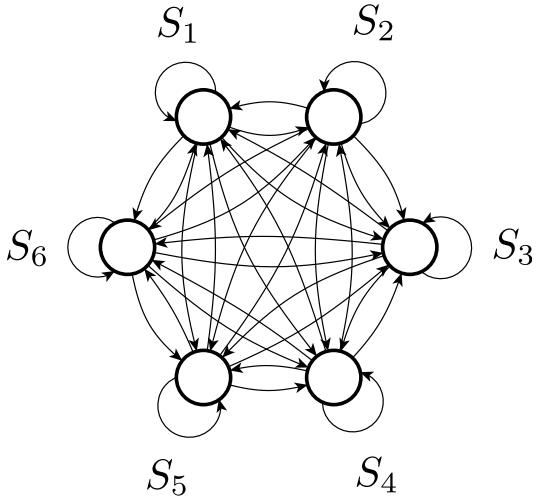
3.3.1 Hidden Markov models

According to Bishop 2006 (Ch. 13, p. 606), speech features are highly correlated in time as there are dependencies between coming observations and the previous occurrences. As stated by Bishop, this characteristic motivates to take a look at Markov chain models[7] in which it is considered that future estimations are independent of all but the latest observations.

With $\boldsymbol{X}= [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ being a succession of random variables, we can formulate the

[6] Also considered in our previous works.

[7] This class of random process includes some memory (Huang et al. 2001, Sec. 8.1, p. 376).



$$a_{ij} = P(S_t = S_j | S_{t-1} = S_i) \quad 1 \leq i, j \leq 6$$

$$\pi_i = P(S_t = S_i);\ t = 1 \qquad 1 \leq i \leq 6$$

$$\sum_{j=1}^{6} a_{ij} = 1 \qquad 1 \leq i \leq 6$$

$$\sum_{i=1}^{6} \pi_i = 1$$

$$0 \leq a_{ij}, \pi_i \leq 1$$

Figure 3.3: *Representation of a Markov chain of 6 states S with their respective parameters, namely the transition probabilities a_{ij} and the initial probabilities π_i (adapted from Huang et al. 2001, Sec. 8.1, pp. 376-377).*

distribution for these variables, based on the Bayes rule, as (Huang et al. 2001, Sec. 8.1, p. 376):

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) = P(\boldsymbol{x}_1) \prod_{i=2}^{n} P(\boldsymbol{x}_i | \boldsymbol{x}_1^{i-1}) \tag{3.1}$$

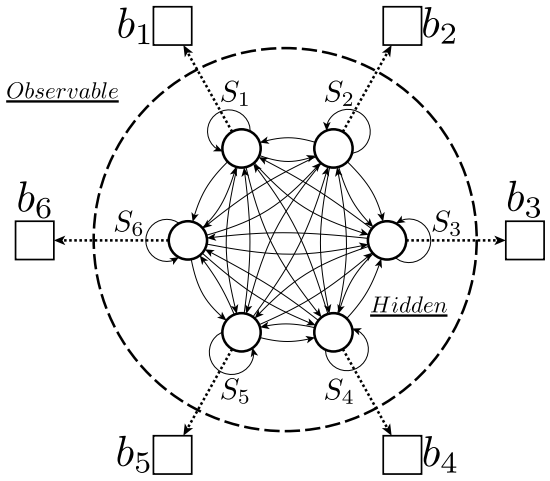
where $\boldsymbol{x}_1^{i-1} = \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{i-1}$.

By assuming that each conditional distribution is independent of all the random variables (also called observations) except the last one $P(\boldsymbol{x}_i | \boldsymbol{x}_1^{i-1}) = P(\boldsymbol{x}_i | \boldsymbol{x}_{i-1})$, we achieve a first-order Markov chain (Bishop 2006, Sec. 13.1, pp. 607-608):

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) = P(\boldsymbol{x}_1) \prod_{i=2}^{n} P(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}) \tag{3.2}$$

This last equation is called the Markov assumption (Huang et al. 2001, Sec. 8.1, p. 376). Additionally, if each random variable or observation $\boldsymbol{x}_i$ is related to a state S, the Markov chain can be formulated by a finite state process or automaton (Theodoridis and Koutroumbas 2003, Sec. 9.6, p. 361) in which the transitions from a state S' to a state S are defined by $P(\boldsymbol{x}_i = S | \boldsymbol{x}_{i-1} = S') = P(S|S')$, (Huang et al. 2001, Sec. 8.1, p. 376). Then, the Markov assumption can be expressed according to Huang and colleagues as: "the probability that the Markov chain will be in a particular state at a given time depends only on the state of the Markov chain at the previous time".

The transition probability from one state S_i to another S_j is determined by a fixed probability P($S_j|S_i$)= a_{ij}, which is considered to be the same for all consecutive time



$$\boldsymbol{A} = a_{ij};\ a_{ij} = P(S_t = S_j | S_{t-1} = S_i)$$

$$\boldsymbol{\pi} = \pi_i;\ \pi_i = P(S_t = S_i);\ t = 1$$

$$\boldsymbol{B} = b_j(\boldsymbol{x});\ b_j(\boldsymbol{x}) = P(\boldsymbol{x} | S_t = S_j)$$

Figure 3.4: *Representation of a hidden Markov model of 6 states S with their respective parameters, namely the transition probabilities a_{ij}, the initial probabilities π_i and the emission probabilities $b_j(\boldsymbol{x})$ (adapted from Huang et al. 2001, Sec. 8.1, pp. 378-379).*

steps and only depends on the state transitions and not on the moment of occurrence (Theodoridis and Koutroumbas 2003, Sec. 9.4, pp. 353-354).

In Fig. 3.3, a Markov chain with 6 states is displayed, where the state at time t is defined as S_t, a_{ij} the transition probability from state S_i to state S_j and π_i the (initial) probability that the Markov chain begins in state S_i (Huang et al. 2001, Sec. 8.1, p. 376). This kind of Markov chain is known as the observable Markov model, because the outputs are the states S at each time step t and each state matches an observation $\boldsymbol{x}_i$, i.e. there is a complete correspondence between the succession of the observations $\boldsymbol{X}$ and the Markov chain sequence of states $\mathcal{S} = [S_1, S_2, ...S_T]$, (Huang et al. 2001, Sec. 8.1, p. 377).

Two important models for sequential data are characterized by the Markov chain (Bishop 2006, Ch. 13, p. 607):

- When the variables are discrete, then we speak of hidden Markov models (HMMs).
- However, if the variables are Gaussians, then we are speaking about a linear dynamical system.

In this context, a hidden Markov model can be defined as the previously explained Markov model, where the (discrete) random variables are the non-observable (hidden) states S and the only observable events are the symbols or observations $\boldsymbol{x}$ that the states emit, see Fig. 3.4 (Huang et al. 2001, Sec. 8.1, pp. 378-379).

HMMs are supposed to fulfill two assumptions (Huang et al. 2001, Sec. 8.2, p. 380):

- Firstly, the hidden process conforms to the (first-order) Markov assumption already enunciated:

$$P(S_t|S_1^{t-1}) = P(S_t|S_{t-1}) \tag{3.3}$$

- Secondly, the output-independence assumption:

$$P(\boldsymbol{x}_t|\boldsymbol{x}_1^{t-1}, S_1^t) = P(\boldsymbol{x}_t|S_t) \tag{3.4}$$

  where $\boldsymbol{x}_1^{t-1}$ represents the output sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{t-1}$ and likewise for S_1^{t-1} and S_1^t.

According to Huang et al. 2001 (Sec. 8.2, p. 380), the output-independence assumption specifies that "the probability that a particular symbol is emitted at time t depends only on the state S_t and is conditionally independent of the past observations".

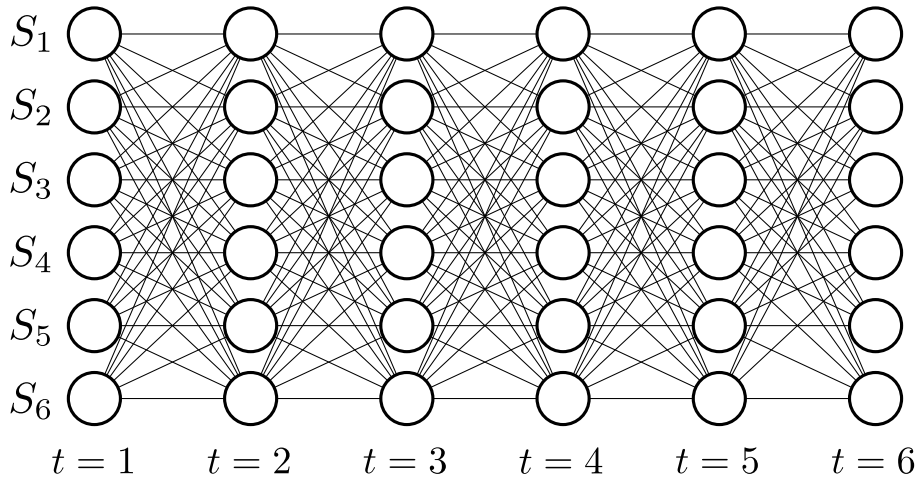
The specification of a hidden Markov model $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ is fully characterized by (Rabiner 1989):

- A number N of hidden states S.
- A matrix $\boldsymbol{A}$ describing the state transition probability distribution.
- The initial state distribution $\boldsymbol{\pi}$.
- The observation probability distribution $\boldsymbol{B}$ dependent on the states S

The observation probability distribution are also known as emission probabilities as they represent the probability of emitting a determined observation $\boldsymbol{x}$ from state S_j, when state S_j is entered (Bishop 2006, Sec. 13.2, p. 611; Huang et al. 2001, Sec. 8.2, p. 379). In speech recognition, these observations can be modeled as discrete, semi-continuous or continuous output distributions (Huang et al. 2001, Sec. 9.5.1, p. 437).

3.3.2 Evaluation: forward algorithm

According to the Bayes decision rule, the HMM λ with the highest posterior probability $P(\lambda|\boldsymbol{X})$ might be chosen as the one representing a given observation $\boldsymbol{X}$, where the posterior probability $P(\lambda|\boldsymbol{X})$ can be calculated through the probability $P(\boldsymbol{X}|\lambda)$, (Huang et al. 2001, Sec. 8.2, p. 381). For this computation, the most insightful way is first to enumerate all (L) successions of states $\mathcal{S}$ that could generate the observation $\boldsymbol{X}$, then to calculate for each succession $\mathcal{S}$ the joint probability of that sequence $\mathcal{S}$ with the observation $\boldsymbol{X}$ given the parameters λ and finally sum all these probabilities (Rabiner 1989):



$$P(\boldsymbol{X}|\lambda) = P(\mathcal{S}^1|\lambda)P(\boldsymbol{X}|\mathcal{S}^1,\lambda) + P(\mathcal{S}^2|\lambda)P(\boldsymbol{X}|\mathcal{S}^2,\lambda) + ... + P(\mathcal{S}^L|\lambda)P(\boldsymbol{X}|\mathcal{S}^L,\lambda)$$

Figure 3.5: *Evaluation of all sequences for the computation of the likelihood $P(\boldsymbol{X}|\lambda)$. This figure shows a trellis (representation of the transition between states) with $T = 6$ dot columns, where each dot symbolizes each of the 6 possible states, $S_1, S_2, ..., S_6$ and the following columns are associated with the consecutive observations $\boldsymbol{x}_t$, where $t = 1, 2, ..., T$ (adapted from Theodoridis and Koutroumbas 2003, Sec. 9.4, p. 353).*

$$P(\boldsymbol{X}|\lambda) = \sum_{L} P(\mathcal{S}|\lambda)P(\boldsymbol{X}|\mathcal{S},\lambda) \tag{3.5}$$

Using one specific succession of states $\mathcal{S} = S_1, S_2, ..., S_T$, the probability of such a sequence in the last equation can be expressed by applying the (first-order) Markov assumption and the output-independence assumption as (Huang et al. 2001, Sec. 8.2.2, pp. 383-384):

$$\begin{gathered} P(\mathcal{S}|\lambda)P(\boldsymbol{X}|\mathcal{S},\lambda) = \\ a_{S_0,S_1} \cdot b_{S_1}(\boldsymbol{x}_1) \cdot a_{S_1,S_2} \cdot b_{S_2}(\boldsymbol{x}_2)...a_{S_{T-1},S_T} \cdot b_{S_T}(\boldsymbol{x}_T) \end{gathered} \tag{3.6}$$

The above equation can be interpreted as we start in state S_1 with probability π_{S_1} or a_{S_0,S_1} and produce the observation $\boldsymbol{x}_1$ with probability $b_{S_1}(\boldsymbol{x}_1)$, then we transitionate from S_{t-1} to S_t with probability $a_{S_{t-1}S_t}$ producing the observation $\boldsymbol{x}_t$ with probability $b_{S_t}(\boldsymbol{x}_t)$ at time t until we finally go from S_{T-1} to S_T in the sequence S (Rabiner 1989).

Nevertheless, this kind of evaluation needs to enumerate all the possible succession of states or paths that leads to an exponential computational complexity[8] $O(N^T)$ according to Huang et al. 2001 (Sec. 8.2.2, p. 384). Luckily, a more efficient algorithm, called

[8]According to Arora and Barak 2009 (Ch. 0, p. 2), the computational complexity theory quantifies the amount of computational resources needed to provide a solution for a determined duty, where the concept "polynomial time" is usually used to characterize fast or feasible algorithms (Arora and Barak 2009, Ch. 1, p. 12).

forward algorithm, has been introduced to compute the last equation based on the accumulation of intermediate results for future calculations to reduce the computational effort (Huang et al. 2001, Sec. 8.2.2, p. 384). This algorithm is also called the "any path method", because all paths are involved in the final cost (Theodoridis and Koutroumbas 2003, Sec. 9.6, p. 365).

Based on Eq. 3.5 and Eq. 3.6, the computation of $P(\boldsymbol{X}|\lambda)$ only depends on the transitions from S_{t-1} to S_t and the generation of the observations $\boldsymbol{x}_t$; so, the likelihood $P(\boldsymbol{X}|\lambda)$ can be calculated recursively on t using the forward probability (Huang et al. 2001, Sec. 8.2.2, p. 384):

$$\alpha_t(i) = P(\boldsymbol{x}_1^t, S_t = S_i|\lambda) \tag{3.7}$$

$\alpha_t(i)$ is the probability of the partial observation sequence $\boldsymbol{x}_1^t$ ($\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_t$) and state S_i at time step t given the model λ, which can be computed inductively (see Eq. 3.7) as illustrated in the diagram or trellis of Fig. 3.6 (Rabiner 1989).

Forward algorithm

Description based on Huang et al. 2001 (Sec. 8.2.2, pp. 384-385); Rabiner 1989.

Step 1: *Initialization*

We start the α computation from $t = 1$, where the implied factors are only the initial and output probabilities.

$$\alpha_1(i) = \pi_i \cdot b_i(\boldsymbol{x}_1) \quad 1 \leq i \leq N$$

Step 2: *Induction*

The next values α are calculated from left to right. The value α for each time step t and each state S is calculated before starting the computations for the next time steps $t+1$.

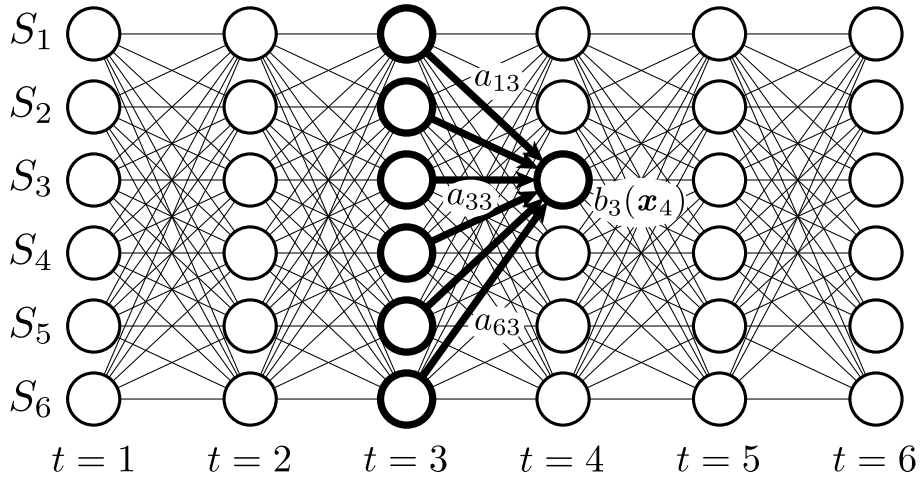
$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij}\right] \cdot b_j(\boldsymbol{x}_t) \quad 2 \leq t \leq T; 1 \leq j \leq N$$

Step 3: *Termination*

When the partial probabilities α are calculated for all time steps and states, the desired computation of $P(\boldsymbol{X}|\lambda)$ is the addition of all probabilities α in the last column.

$$P(\boldsymbol{X}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

The introduction of the variable α reduces the computational cost from being exponential to being polynomial $O(N^2T)$ by taking advantage of the partially computed probability α (Huang et al. 2001, Sec. 8.2.2, p. 385).



$$\alpha_4(3) = \left[\sum_{i=1}^{6} \alpha_3(i) \cdot a_{i3}\right] \cdot b_3(\boldsymbol{x}_4)$$

Figure 3.6: *Forward recursion for the evaluation of α, where we contemplate in this section of the trellis that $\alpha_t(j)$ is computed by first using the elements $\alpha_{t-1}(i)$ and adding them up with the weights given by a_{ij}, and then multiplying by $b_j(\boldsymbol{x}_t)$ (adapted from Bishop 2006, Sec. 13.2.2, p. 621).*

3.3.3 Learning procedure: forward-backward algorithm

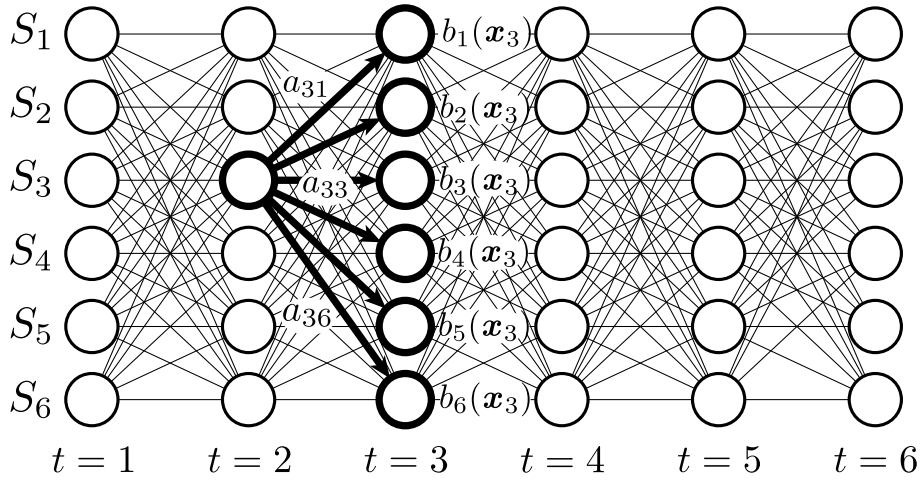
The estimation of the model parameters $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ is a complex procedure, because according to Huang et al. 2001 (Sec. 8.2.4, p. 387): "there is no known analytical method that maximizes the joint probability of the training data in a closed form". The most used approach is the iterative Baum-Welch algorithm[9], also called the forward-backward algorithm, which is based on the same principle employed by the EM algorithm (Dempster et al. 1977) as also stated by Huang and colleagues. The learning problem of a HMM is a standard case of unsupervised learning, where the succession of hidden states is unknown, i.e. there is no information about which state S_t emits the corresponding observation $\boldsymbol{x}_t$, (Huang et al. 2001, Sec. 8.2.4, p. 387).

Previous to the definition of the forward-backward algorithm, we formulate three useful terms employed in this procedure (Huang et al. 2001, Sec. 8.2.4, p. 387):

- $\beta_t(i)$ is the probability of the partial observation sequence $\boldsymbol{x}_{t+1}^T$ $(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2}, ..., \boldsymbol{x}_T)$ given the model λ and the state S_i at time t, which can be computed inductively (Rabiner 1989).

$$\beta_t(i) = P(\boldsymbol{x}_{t+1}^T | S_t = S_i, \lambda) \tag{3.8}$$

[9]See Baum et al. 1970, Juang et al. 1986 and Liporace 1982.



$$\beta_2(3) = \sum_{j=1}^{6} a_{3j} \cdot b_j(\boldsymbol{x}_3) \cdot \beta_3(j)$$

Figure 3.7: *Backward algorithm in the evaluation of the parameter β, where we note in this part of the trellis that $\beta_t(i)$ is calculated by employing the elements $\beta_{t+1}(j)$ at step $t+1$ and adding them with the weights given by a_{ij} and the values of the emission probability $b_j(\boldsymbol{x}_{t+1})$, (adapted from Bishop 2006, Sec. 13.2.2, p. 622). The direction of the arrows can be considered the other way around (backward).*

Backward algorithm

Description based on Huang et al. 2001 (Sec. 8.2.4, pp. 387-388)

Step 1: *Initialization*[10]

$$\beta_T(i) = 1/N \quad 1 \leq i \leq N$$

Step 2: *Induction*

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(\boldsymbol{x}_{t+1}) \cdot \beta_{t+1}(j) \quad t = T-1, ..., 1; 1 \leq i \leq N$$

- $\xi_t(i,j)$ is the probability that we are in state S_i at time step t and state S_j at the next time step $t+1$, given the model λ and the observation $\boldsymbol{X}$ (Rabiner 1989):

$$\xi_t(i,j) = P(S_t = S_i, S_{t+1} = S_j | \boldsymbol{X}, \lambda) \tag{3.9}$$

[10]In Rabiner 1989, this value is set to 1 for all the states S_i.

$$\xi_t(i,j) = \frac{P(S_t = S_i, S_{t+1} = S_j, \boldsymbol{X}|\lambda)}{P(\boldsymbol{X}|\lambda)} = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(\boldsymbol{x}_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(\boldsymbol{x}_{t+1}) \cdot \beta_{t+1}(j)} \tag{3.10}$$

$\sum_{t=1}^{T-1} \xi_t(i,j)$ is the "expected number of transitions from state S_i to state S_j" (Rabiner 1989).

- $\gamma_t(i)$ is the probability that the model is in state S_i at time step t, given the model λ and the observation $\boldsymbol{X}$ (Rabiner 1989), hence:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.11}$$

$$\gamma_t(i) = P(S_t = S_i|\boldsymbol{X}, \lambda) = \frac{P(S_t = S_i, \boldsymbol{X}|\lambda)}{P(\boldsymbol{X}|\lambda)} = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} \tag{3.12}$$

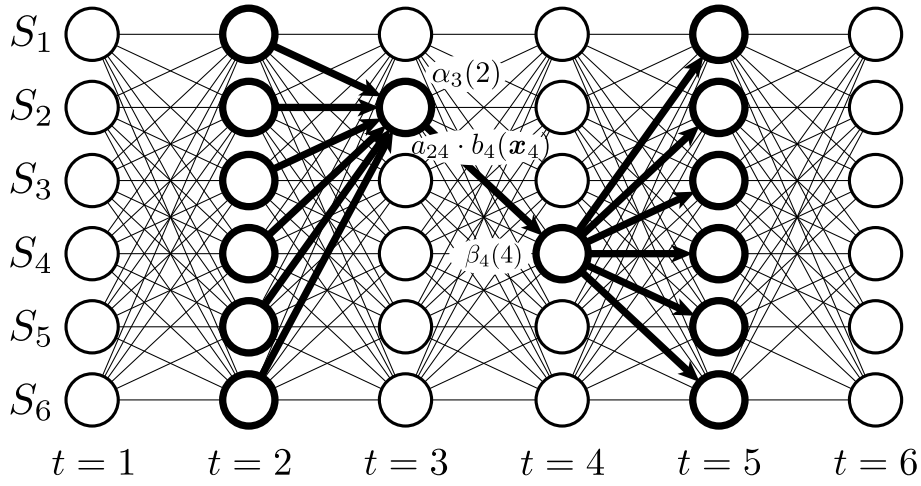
$\sum_{t=1}^{T} \gamma_t(i)$ is the "expected (over time) number of times that the state S_i is visited", given the model λ and the observation sequence $\boldsymbol{X}$ and when the upper index in the summation is $T-1$, this quantity is the "expected number of transitions from state S_i" (Rabiner 1989).

The forward-backward algorithm can determine the HMM parameters $\widehat{\lambda} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ from the old λ by maximizing the likelihood $P(\boldsymbol{X}|\lambda)$ following the principle of the EM algorithm (Huang et al. 2001, Sec. 8.2.4, p. 388).

Forward-backward algorithm

Step 1: *Initialization*
Description based on Rabiner 1989; Bishop 2006, (Sec. 13.2.2, p. 623); Theodoridis and Koutroumbas 2003, (Sec. 9.6, p. 368)

Selection of a suitable initial model $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$. As the algorithm normally leads to local maxima, the initial conditions are very important in order to find an appropriate local maximum for $P(\boldsymbol{X}|\lambda)$. $\boldsymbol{A}$ and $\boldsymbol{\pi}$ parameters are generally initialized in a randomly or uniformly way. In the case that the set $\boldsymbol{B}$ is modeled by Gaussians, the parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ can be initialized through the K-means algorithm (see Sec. 5.1.1) to the data. More bootstrapping techniques will be discussed in Chapter 5.



$$\xi_3(2,4) = \frac{\alpha_3(2) \cdot a_{24} \cdot b_4(\boldsymbol{x}_4) \cdot \beta_4(4)}{\sum_{i=1}^{6} \sum_{j=1}^{6} \alpha_3(i) \cdot a_{ij} \cdot b_j(\boldsymbol{x}_4) \cdot \beta_4(j)}$$

Figure 3.8: *The relationship of the adjacent α and β is displayed. α is calculated recursively from left to right, and β from right to left. So, we can compute $P(\boldsymbol{X}|\lambda) = \sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)$ and $\xi_t(i,j) = \alpha_t(i) \cdot a_{ij} \cdot b_j(\boldsymbol{x}_{t+1}) \cdot \beta_{t+1}(j) / \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(\boldsymbol{x}_{t+1}) \cdot \beta_{t+1}(j)$, (adapted from Huang et al. 2001, Sec. 8.2.4, p. 388; Rabiner 1989).*

Step 2: *E-step*

Description based on Huang et al. 2001, (Sec. 8.2.4, p. 391)

Transformation of the objective function $P(\boldsymbol{X}|\lambda)$ into an auxiliary function $Q(\widehat{\lambda}, \lambda)$ as a measure between the initial model λ and the updated one $\widehat{\lambda}$.

Step 3: *M-step*

Description based on Huang et al. 2001, (Sec. 8.2.4-8.3.2, pp. 390-395); Rabiner 1989

Computation of $\widehat{\lambda}$ according to the following re-estimation equations that maximize the auxiliary Q-function:

$$\widehat{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

The initial probability π_i can be seen as a particular instance of the transition probability. π_i is usually defined as $\pi_1 = 1$ in speech recognition systems. For the emission probabilities in the discrete case, we have:

$$\widehat{b_j}(\boldsymbol{x}_t) = \frac{\sum\limits_{\substack{t=1 \\ s.t.\ \boldsymbol{x}=\boldsymbol{x}_t}}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)}$$

If we are working with continuous emission probabilities, Gaussian mixture models (GMMs) are normally employed:

$$b_j(\boldsymbol{x}) = \sum_{m=1}^{M} cm_{j,m} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}) = \sum_{m=1}^{M} cm_{j,m} b_{j,m}(\boldsymbol{x})$$

where $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m})$ denotes a single Gaussian density function, M the number of mixture components, $\boldsymbol{\mu}_{j,m}$ is the mean vector, $\boldsymbol{\Sigma}_{j,m}$ the covariance matrix and $cm_{j,m}$ is the weight for the m^{th} mixture component of state S_j satisfying:

$$cm_{j,m} \geq 0 \quad \sum_{m=1}^{M} cm_{j,m} = 1$$

Here, the corresponding re-estimation formulas are:

$$\widehat{cm}_{j,m} = \frac{\sum\limits_{t=1}^{T} \gamma_t(j,m)}{\sum\limits_{t=1}^{T} \sum\limits_{m=1}^{M} \gamma_t(j,m)}$$

$$\widehat{\boldsymbol{\mu}}_{j,m} = \frac{\sum\limits_{t=1}^{T} \gamma_t(j,m) \cdot \boldsymbol{x}_t}{\sum\limits_{t=1}^{T} \gamma_t(j,m)}$$

$$\widehat{\boldsymbol{\Sigma}}_{j,m} = \frac{\sum\limits_{t=1}^{T} \gamma_t(j,m) \cdot (\boldsymbol{x}_t - \boldsymbol{\mu}_{j,m})(\boldsymbol{x}_t - \boldsymbol{\mu}_{j,m})'}{\sum\limits_{t=1}^{T} \gamma_t(j,m)}$$

where $\gamma_t(j,m)$ is the probability of being in state S_j at time step t with the m^{th} mixture component accounting for $\boldsymbol{x}_t$:

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j) \cdot \beta_t(j)}{\sum\limits_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j)} \right] \left[\frac{cm_{j,m} \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m})}{\sum\limits_{m=1}^{M} cm_{j,m} \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m})} \right]$$

In the semi-continuous case, we estimate the corresponding weights cm to the codebook used (composed of a predefined set of GMMs, $b_m(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$).

$$b_j(\boldsymbol{x}) = \sum_{m=1}^{M} cm_{j,m} b_m(\boldsymbol{x})$$

Step 4: *Iteration*

Update the parameters of the model $\lambda = \widehat{\lambda}$ and go to step 2 until the algorithm converges.

Despite the fact that the above description of the forward-backward algorithm is formulated with only one training sequence, it can be extended to a set of training samples under the independence assumption of these samples, so that the training of a model from R sequences is analogous to estimate the HMM parameters λ optimizing the joint likelihood (Huang et al. 2001, Sec. 8.2.4, p. 391):

$$\prod_{r=1}^{R} P(\boldsymbol{X}_r|\lambda) \tag{3.13}$$

The forward-backward algorithm is applied on each training sample to compute the counts in the estimation equations, which are summed up for all the R samples in the numerator and denominator and finally, the parameters are normalized (Huang et al. 2001, Sec. 8.2.4, p. 391).

If we observe the recursion steps in detail, we note that at each step the new updated values (often significantly less than the unity and coming from multiplications with previous values of the same magnitudes), go exponentially quick to zero with each multiplication and thus, possibly exceeding the precision range of the computer (Theodoridis and Koutroumbas 2003, Sec. 9.6, p. 369). Generally we can make a work-around of this problem in the evaluation of likelihood functions by taking logarithms[11] (Huang et al. 2001, Sec. 8.4.6, p. 403).

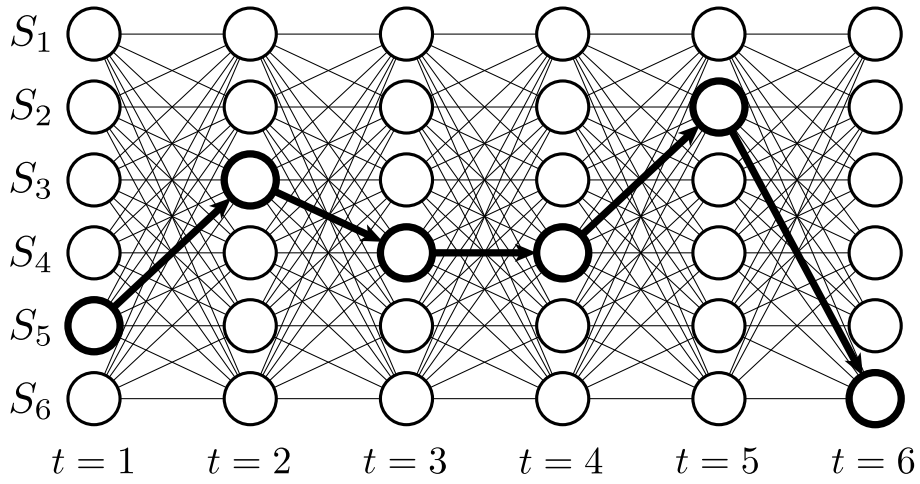
In addition to the Baum-Welch algorithm, many scientists have proposed other parameter estimation techniques, e.g. the so called Viterbi training (Fink 2008, Sec. 5.7.2, p. 86).

3.3.4 Decoding: Viterbi algorithm

The forward algorithm calculates the probability that the model λ can generate a set of observations by adding the probabilities of all possible successions of states, being the whole set of paths (including the best path) no longer enumerated (Huang et al. 2001, Sec. 8.2.3, p. 385). Nevertheless, it is necessary to estimate this best path in some applications, e.g. the search of the most probable sequence of speech unit models in continuous speech recognition (Bishop 2006, Sec. 13.2.5, p. 629; Huang et al. 2001, Sec. 8.2.3, p. 385).

In HMMs, the states are not observable (hidden), thus, the most extensively applied criterion is to select the succession of states $\mathcal{S}^*$ of model λ that reaches the highest probability $P(\mathcal{S}^*, \boldsymbol{X}|\lambda)$ of producing the observation $\boldsymbol{X}$ (Huang et al. 2001, Sec. 8.2.3, pp. 385-386).

[11] It is also possible to multiple the probabilities by a scaling constant, however it is not so beneficial as taking logarithms (see Huang et al. 2001, Sec. 8.4.6, p. 403).



$$\mathcal{S}^* = S_5, S_3, S_4, S_4, S_2, S_6$$

Figure 3.9: *Illustration of the Viterbi decoded path, where the best path is characterized by bold arrows, while the rest of the paths by thin lines (adapted from Fig. 9.1 in Theodoridis and Koutroumbas 2003).*

In HMM frameworks, the technique used is known as Viterbi algorithm, which is based on dynamic programming and is quite similar to the forward algorithm (Viterbi 1967: qtd. in Rabiner 1989). However, in contrast to the forward algorithm, there is a backtracking step and the probabilities of all possible paths that go through a state are not accumulated or added, but this addition is substituted by a maximization over all previous states in order to obtain the best succession of states[12] (Rabiner 1989).

Thanks to the optimization principle of Bellman, we only maintain the path to the maximum probability (Bellman 1957: qtd. in Theodoridis and Koutroumbas 2003, Sec. 9.4, p. 355). In this context, the computational complexity $O(N^2T)$ results from the N remaining sequences of states and the N computations at each step, which corresponds to N^2 for all the hidden states at each time step t (Theodoridis and Koutroumbas 2003, Sec. 9.4, p. 355).

Finally, the probability $P(\mathcal{S}^*, \boldsymbol{X}|\lambda)$ corresponding to the most likely path is computed and to recover the optimal sequence $\mathcal{S}^*$, we employ the backtracking procedure, which saves a register of the best succession of states (Bishop 2006, Sec. 13.2.5, p. 630).

[12]This technique is also called the best path method (Theodoridis and Koutroumbas 2003, Sec. 9.6, p. 365).

Viterbi algorithm

Description based on Huang et al. 2001 (Sec. 8.2.3, p. 386); Rabiner 1989.

Nomenclature:

$VIT_t(i)$ is the highest probability along a succession of states $\mathcal{S}$, at time step t, which has generated the observation sequence $\boldsymbol{x}_1^t$ (until t) and finishes in state S_i.

$$VIT_t(i) = \max_{S_1,\ldots,S_{t-1}} P(\boldsymbol{x}_1^t, S_1^{t-1}, S_t = S_i|\lambda) \tag{3.14}$$

$BTR_t(j)$ matches the most likely previous state S_i in the best path sequence $\mathcal{S}^*$ being currently in state S_j.

$$BTR_t(j) = \arg\max_{1\leq i\leq N} \quad [VIT_{t-1}(i) \cdot a_{ij}] \tag{3.15}$$

Step 1: *Initialization*

$VIT_1(i) = \pi_i \cdot b_i(\boldsymbol{x}_1) \quad 1 \leq i \leq N$

$BTR_1(i) = 0$

Step 2: *Recursion*

$VIT_t(j) = \max_{1\leq i\leq N}[VIT_{t-1}(i) \cdot a_{ij}] \cdot b_j(\boldsymbol{x}_t) \quad 2 \leq t \leq T; 1 \leq j \leq N$

$BTR_t(j) = \arg\max_{1\leq i\leq N} [VIT_{t-1}(i) \cdot a_{ij}] \qquad 2 \leq t \leq T; 1 \leq j \leq N$

Step 3: *Termination*

The best score $= \max_{1\leq i\leq N} VIT_T(i)$

$S_T^* = \arg\max_{1\leq i\leq N} \; VIT_T(i)$

Step 4: *Backtracking*

$S_t^* = BTR_{t+1}(S_{t+1}^*) \quad t = T-1, T-2, ..., 1$

$S^* = S_1^*, S_2^*, ..., S_T^* \qquad$ is the best sequence

More details of these algorithms explained in Sec. 3.3.2, 3.3.3, 3.3.4 can be found in Fink 2008.

3.4 Word learning and recognition in the system[13]

After the presentation of the main algorithms involved in HMMs, we explain how these algorithms are employed in our framework in combination with other techniques for the learning and recognition phase. During the learning phase, the parameters of the word-model are first defined, then initialized, estimated and finally adapted via several discriminative learning strategies as Fig. 3.10 illustrates.

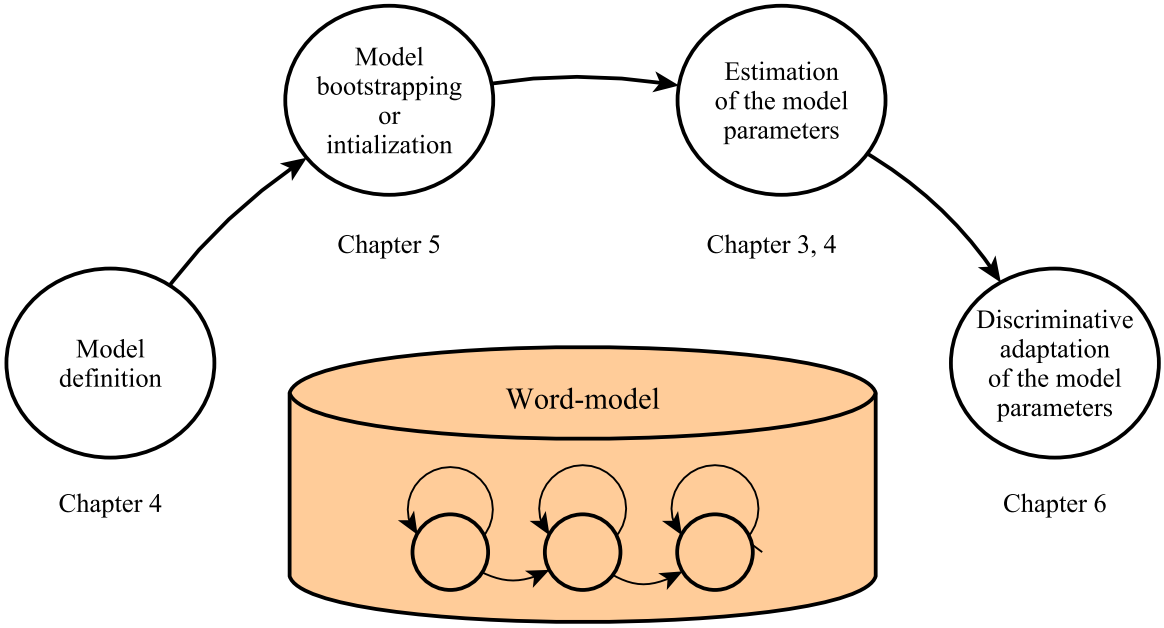


Figure 3.10: *Overview of the learning phases that a word-model experiments before being released as "trained" in our system.*

3.4.1 Model definition

In the previous sections, we described the principles under which HMMs operate and their basic elements, namely the hidden states, the observation probabilities and the relation between the states, i.e. initialization and transition probabilities. However, the configuration of these parameters is completely dependent on the desired application and often also on the number of training samples available in the system as further explained and referenced (both issues) in Chapter 4, where an extensive and specific discussion of the configuration of the model parameters of our framework is realized. This survey includes an analysis of the choice of the estimates regarding sparse training conditions as constraint for our system and the introduction of a threshold determined by the number of available training data presented in our previous work Ayllón Clemente et al. 2012.

[13]This section and its corresponding subsections take numerous paragraphs (also state-of-the-art reviews), some of them verbatim, from Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

In our efficient incremental learning framework, we aim at the construction of a user-friendly system in which the users do not have to tune the parameters, but the framework adapts to the changing conditions itself, i.e. different data structures and variable number of available training samples. We achieve this purpose through the dynamic configuration of the parameters, e.g. number of hidden states and an adaptive variance floor (see Sec. 4.2.3) dependent on the available number of samples in each case (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

3.4.2 Model bootstrapping

In our system, the HMM parameters are computed by a combination of different training stages of maximum likelihood (ML) and maximum a posteriori (MAP) estimation, see Sec. 4.1.4. In the estimation of these parameters, an initial estimation of the parameters is a crucial and critical stage to obtain good performance (Huang et al. 2001, Sec. 8.4.1, p. 396). Consequently, we propose and develop a novel bootstrapping or initialization technique, which can deal with few training samples in incremental word learning and provides suitable initial estimates of the word-models to learn according to our previous works in Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

Our initialization approach proposed previously in the last referred works consists of the combination of an unsupervised and a supervised learning process, i.e. it uses the labeled training samples of the words to learn and an additional set of unlabeled random speech segments pronounced by other speakers and containing different units than the ones to learn. The first stages of the mentioned initialization are based on Iwahashi 2007 and Brandl et al. 2008, where the use of unlabeled training data allows us to pre-train the system in an unsupervised way with generic speech models. In Fig. 3.1, the unlabeled random speech fragments are visualized by a group of four speakers, the labeled ones by a set of three speakers. The successful performance of the proposed method is due to a novel multiple sequence alignment (MSA) procedure (first introduced in Ayllón Clemente et al. 2010b as well as Ayllón Clemente and Heckmann 2009) that configures the final topology of the model. In Chapter 5, we present the referred model bootstrapping proposed and describe in more detail each of the phases of this initialization to obtain a good initial set of HMM parameters for the next processing stage "the estimation of the parameters", where the acoustic models are recalculated and the accumulators of the silence-models are updated (see Fig. 3.1; Ayllón Clemente et al. 2012).

3.4.3 Estimation of the parameters

Similar to our previous works and other standard approaches, once the parameters of the word-model have been initialized, they are re-estimated using the forward-backward algorithm. When working with data containing isolated spoken words, each training sample does not only contain the samples of the models to be estimated but usually

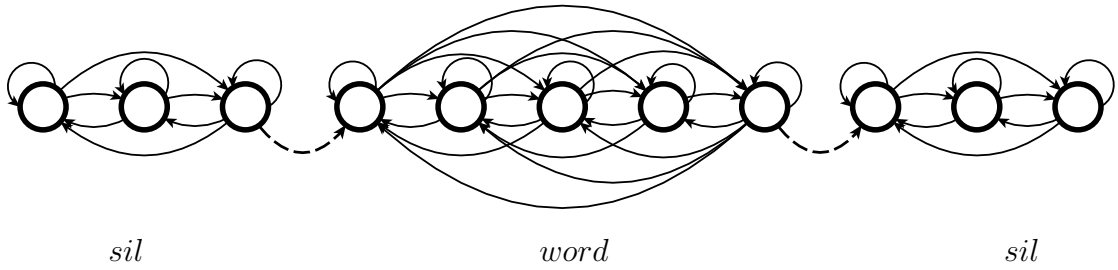


Figure 3.11: *In isolated word learning, the sentences are built as: sil - word - sil, as the language model shows in Fig. 3.2(a). For the training, the word-models and the silence-models are estimated together as a unique hidden Markov model (Huang et al. 2001, Sec. 9.5.2, pp. 439-441). A discussion about which model representations will be used can be found in Sec. 4.1, hence in this picture, the models are represented by a general HMM topology.*

silences at the beginning and end of each training segment, see Fig. 3.2(a) (also considered in Ayllón Clemente et al. 2012). The forward-backward algorithm presents the advantage of being able to automatically match each model to the portion of the sentence it belongs without needing any segmentation aid, so that the silence-model and the word-model are estimated jointly as one unique model and then fragmented after executing the forward-backward algorithm (Fig. 3.11; Huang et al. 2001, Sec. 9.5.2, pp. 439-441).

In a batch process, the whole training set is provided before the start of the estimation of the parameters, but not in incremental learning, where the system only receives part of the whole data set in each time step (see Bishop 2006, Sec. 3.1.3, pp. 143-144), i.e. each time that the samples of a new term enter the system, new samples for the silence-model are introduced too (considered also in Ayllón Clemente et al. 2012). Hence, similar to our previous publication, we work with silence accumulators of the estimates that are computed each time a new word-model is added. These accumulators are defined by the symbols sil', sil", sil"' in Fig. 3.1. During the re-estimation of the parameters of the new model, each update of the silence-model incorporates the accumulators of the previous silence data while reducing the computational cost, because it avoids the iterative estimation of the silence-model each time new training samples are introduced into the framework (see Ayllón Clemente et al. 2012).

3.4.4 Discriminative training

Another aspect to investigate in incremental learning is the execution of a discriminative training stage that refines the estimates of the parameters computed in the previous stages improving the generalization performance (see Ayllón Clemente et al. 2012). In the last decades, a trend for using discriminative methods in the estimation of the parameters of the HMMs in ASR systems has appeared (e.g. Kapadia 1998: qtd. in Bishop 2006

(Sec. 13.2.6, p. 632)). This interest is particularly strong in the case of large margin discriminative training (DT) methods, where large margin DT (LM DT) has emerged as a promising discriminative learning framework for building statistical classifiers in recent years (see Jiang et al. 2006; Sha and Saul 2006). This method makes use of the concept of the margin, which can be defined in this context for a training sample as: "distance between the discriminant score of the label w_i to the highest discriminant score of all the competing word sequences" according to Yu and Deng 2007. Large margin DT offers advantages of SVMs however without their drawbacks (see Sec. 2.3.5; Jiang et al. 2006). This makes large margin DT a good method to apply as final refinement stage in our system (see Ayllón Clemente et al. 2012).

Related to the discriminative training stage, two facts encourage us to investigate and analyze different strategies to execute large margin discriminative training in order to improve the recognition scores (Ayllón Clemente et al. 2012):

- The computation effort of the algorithms. A new cluster is added at each time step forcing the optimization of all previously learned parameters at each iteration as our system has to be improved and prepared to be used for further learning or recognition.
- Additionally, when the number of training samples is very small, all training segments are usually well classified.

Several strategies coming from previous works (see Appendix D), standard approaches used as basis and the optimization algorithms applied in our framework are described in Chapter 6.

3.4.5 Speech recognition

Up until now, we mainly focused on the learning phase. However, the real application of our framework should be the ability to recognize previously presented ("learned") terms by the user (see Chapter 1). This means that our system has to be endowed with techniques that enable the decoding of samples and their assignment to already learned words[14]. The recognition scheme of our framework is displayed in Fig. 3.12, where we employ isolated word (IWR) and continuous speech recognition (CSR) tools.

In isolated word recognition (IWR), word boundaries are not problematic (see Sec 3.4.3) as we can calculate the acoustic model probability $P(\boldsymbol{X}|\lambda)$ using the forward algorithm[15] and the word associated with the word-model λ^* with the highest probability is then identified as the sought word (Huang et al. 2001, Sec. 12.2.3, p. 604).

However, this procedure is not so simple in continuous speech recognition as we have to consider the fact that each word can start at any time step (Huang et al. 2001, Sec. 12.2.4,

[14]This is also part of standard approaches.
[15]Also possible using the Viterbi algorithm.

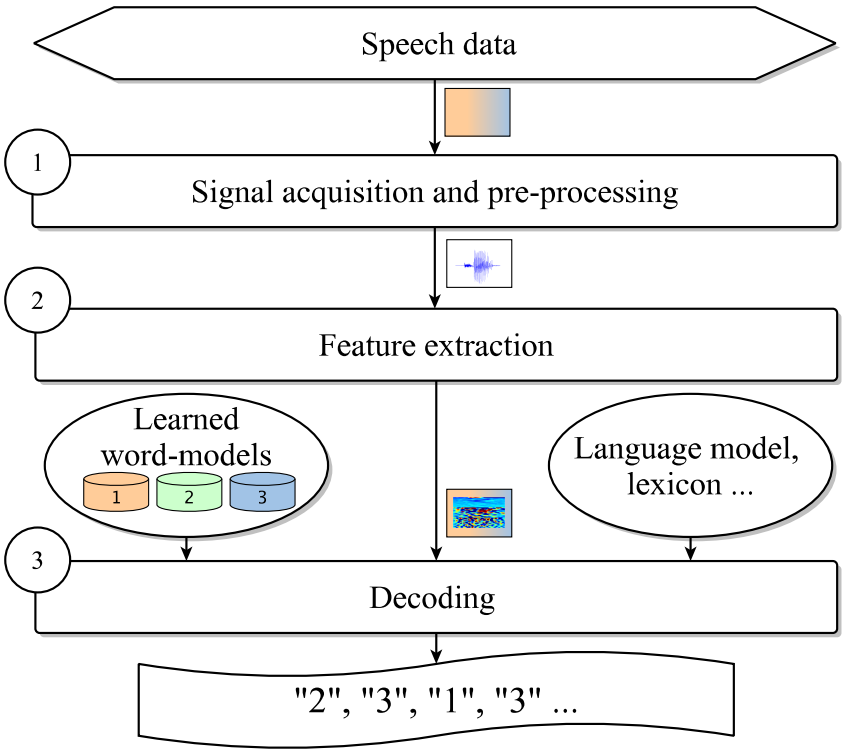


Figure 3.12: *Overview of our IWL framework in the recognition phase (adapted from Fig. 1.2 of Huang et al. 2001). In this snapshot, an unlabeled and unknown data sample has entered the system. The output is the set of labels of the decoded sample that the system believes to understand.*

p. 604). Some of the first ASR systems employed a combined method consisting in hypothesizing likely word boundaries and then applying pattern matching approaches for identifying each portion (Huang et al. 2001, Sec. 12.2.4, p. 604).

According to Jelinek 1997 (Sec. 1.3.4, p. 9), a simple hypothesis search is necessary, which ignores the huge number of potential candidates examining only those successions of words previously recommended by the acoustics. One of these pruning strategies is the beam search, where the purge of the less likely sequences reduces a lot the search (Jelinek 1997, Sec. 5.3, pp. 81-84). In our framework, we use an alternative formulation of the Viterbi algorithm called Token Passing Algorithm (Young et al. 1989) to generate the recognition results (see Ayllón Clemente et al. 2012).

3.5 Implementation details

As experimental platform[16], the system is implemented in Mathworks Matlab© using as baselines the hidden Markov models toolkit HTK (Young et al. 2009, see Appendix B), Netlab (Nabney 2002) and Voicebox (Brookes 2005) toolboxes.

[16]Similar to the one presented in Ayllón Clemente et al. 2012.

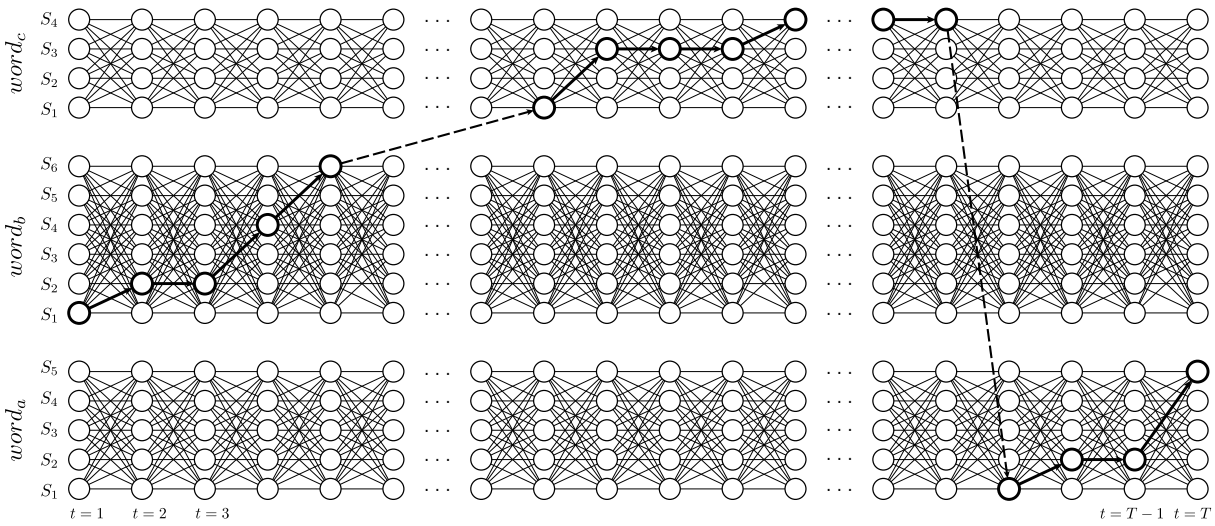


Figure 3.13: *Decoding in continuous speech recognition. In this picture, one can appreciate the huge amount of hypotheses needed by conventional decoding methods in a system containing three word-models (adapted from Fig. 12.11 in Huang et al. 2001 (Sec. 12.2.4, p. 605)). In the displayed example, the decoded sentence is* $word_b, word_c, word_a$ *as the marked path shows.*

In HTK (Young et al. 2009), we set all parameters to the default values, except the minimum number of training samples, which is normally set to 3; the word insertion penalty $-p$ and the grammar scale factor $-s$ are set to zero and additionally, the pruning option -t is set to the following incremental thresholds: 250.0 150.0 1000.0 (see Ayllón Clemente et al. 2012).

In order to integrate our proposed algorithms in the HTK framework (Young et al. 2009), an interface between HTK and Matlab© (in both directions) was built using several Matlab© and C libraries.

3.5.1 Evaluation metrics

Similar to our previous works (see Appendix D), we measure the performance of our proposed methods and the state-of-the-art techniques using the word error rate (WER, see Sec. 2.3.7) and we realize 25-fold cross-validations of each experiment in order to obtain more reliable results for our framework. The configuration of the cross-validation sets is explained in the following section.

3.5.2 Benchmarks

For the evaluation of the system, the TIDigits database (see Leonard and Doddington 1993) described in Appendix A.2 was used as target vocabulary[17]. In the utterances of

[17]The one also used for other previous works, see Appendix D.

the database, digits from “0” to “9”, “zero” and “oh” are present, which sentences consist of isolated digits or digits uttered in continuous speech in clean conditions pronounced by adults and children (see Leonard and Doddington 1993). For our evaluation, we only employ adult speakers and we filter the continuous digits utterances and the sentences pronounced by women from the training set, so that in the training phase we only find isolated digits and male speakers (Ayllón Clemente et al. 2012).

Otherwise, the test set comprises isolated words and continuous speech sentences uttered by adult speakers, where we use the female speakers in the test set (only for the one including isolated words) in order to prove the generalization power of our algorithms in very different unseen data (Ayllón Clemente et al. 2010a). In order to obtain the 25-fold cross-validations mentioned above, we divide randomly the resulting training set in different segments from 1 up to 10 training samples per subset being the number of different segments provided to the system 250, 25 different subsets for each combination of a determined number of training data.

3.6 Recapitulation

The analysis and references of conventional ASR systems in the last chapter pointed that standard batch processes are not suitable for interactive incremental systems (see Bishop 2006, Sec. 3.1.3, pp. 143-144). In spite of the introduction of incremental approaches by different authors to overcome these limitations, to our knowledge, the frameworks presented until now do not directly intend to address the problem of efficient learning in these scenarios and are often restricted to speaker-dependent applications, what motivated us to propose a framework able to operate with a reduced number of training samples in a speaker-independent incremental learning scenario (see Ayllón Clemente et al. 2012).

In this chapter, we presented the architecture of our efficient incremental word learning system including a brief overview of each of the stages and modules that compose our framework (Ayllón Clemente et al. 2012). In our platform, words are the speech units to model as they are the main vehicle of meaning (see Huang et al. 2001, Sec. 9.4.1, p. 427). Inspired in the first stages in children development, the simplest way of presenting new words is used, i.e. in isolation and clean conditions, as referred in Sec. 3.2. However, in order to build a more robust system, we decided to employ RASTA-PLP features to cope with a possible unquiet environment (see argumentation and references also in Sec. 3.2). Furthermore, although the learning phase only deals with isolated words, the system is prepared to recognize the learned terms in continuous speech utterances in the recognition stage (similar to our previous works Ayllón Clemente et al. 2012).

In our system, words are represented via generative models, in particular HMMs[18]. Despite their limitations, these statistical methods have emerged as a very robust tool for

[18]All our previous works, see Appendix D, are based on HMMs.

modeling sequential patterns, i.e. being a well suited technique for modeling speech (Huang et al. 2001, Ch. 8, Sec. 8.5, p. 375-409). At this point, we explained the most relevant algorithms used in HMMs for the evaluation, decoding and training of the models. Apart from the word-models, a uniform distribution is used as language model, so that each model has the same probability to appear (Huang et al. 2001, Sec. 12.2.4, p. 604). This fact matches with our goal of using little prior knowledge.

Each time a new word enters the system according to Fig. 3.1, the parameters of the new word-model are defined. This definition is crucial in order to protect the word-models from overspecialization, i.e. not recognizing unseen data, and to decrease the need of parameter tuning in changing conditions; thus we introduce a parameter to control this specialization during the learning phase in our framework (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). A detailed description of the configuration of the word-models (including the mentioned parameter) dependent on the training conditions will be presented in Chapter 4. Next, the parameters of the model are initialized in the model bootstrapping stage, being this initialization critical for the success of the HMM recognizer as mentioned by Huang et al. 2001 (Sec. 8.4.1, p. 396). In Chapter 5, we will present a bootstrapping method built over supervised and unsupervised algorithms including a novel multiple sequence alignment technique (Ayllón Clemente et al. 2010b; Ayllón Clemente et al. 2012). In the learning phase, the parameters are estimated with the help of the Baum-Welch algorithm (Huang et al. 2001, Sec. 8.2.4, p. 387). Finally, our generative word-models are adapted by means of several large margin DT strategies in order to improve the generalization performance, which will be explained in detail in Chapter 6 (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). All these functional modules work together to achieve the target of reducing the tutoring time, i.e. limiting the number of samples employed to learn a new word while maintaining a competent performance and enabling the use of different speakers in the learning and recognition phase of our incremental word learning framework (Ayllón Clemente et al. 2012). The contribution of each functional module to the whole performance of our system is analyzed in the following chapters.

4

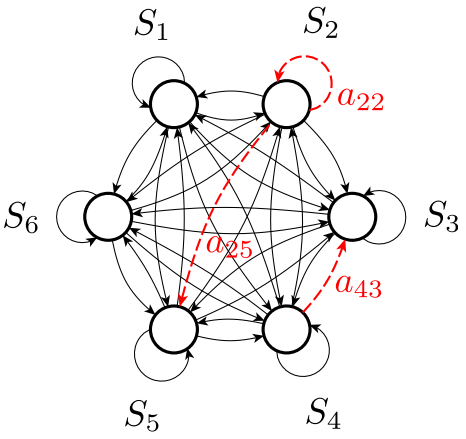
Speech modeling in sparse learning conditions

In real-world applications, the variability of the input data is so huge that the training samples are only able to contain a small portion of all possible input data resulting in a challenge in pattern recognition applications known as generalization or generalization performance (Bishop 2006, Ch. 1, p. 2). In the same context, conventional ASR modeling techniques require a large amount of labeled training data in order to estimate an optimal and considerable set of parameters[1] as in general, the more complex or flexible (with respect to the number of parameters) models are defined, the larger the number of training data required, although, it is not very satisfying to define the number of parameters in a model according to the number of training data (Bishop 2006, Sec. 1.1, p. 9). However, if the number of training samples is very small, the estimated models can be overspecialized to the data employed, where one alternative is to tie them together to decrease the number of parameters to estimate according to Huang et al. 2001 (Sec. 8.4.5, p. 401). Nevertheless, the mentioned approaches by Huang and colleagues suggest the need of some expertise about the relations among the parameters and the units to learn by the system that it is not compatible with our goal of learning without/with very little prior knowledge.

Additionally, labeled data are not easy to procure in interactive learning[2]; thus, some authors employ automatic labeling methods in order to avoid manual transcription from the user or partly avoiding transcriptions (e.g. Alshawi 2003 and Gorin et al. 1999), others use unsupervised learning or train the system with a smaller number of training data samples (e.g. Iwahashi 2007) as stated by Ayllón Clemente et al. 2012[1]. Due to the above mentioned disadvantages when few training samples are employed, it is very interesting to investigate how the behavior of a given model or parameter changes as the number of training samples varies too.

[1] Mentioned in our previous works enumerated in Appendix D.

[2] One way to get labels is to make the user to read predefined sentences, similar to the manner to create a speech database (e.g. Garofolo et al. 1993). However, this activity is tedious and not very user-friendly.



$$\boldsymbol{A} = a_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}$$

$$\boldsymbol{\pi} = (\pi_i)^t = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \end{pmatrix}$$

Figure 4.1: *HMM ergodic topology is the most permissive HMM representation as all possible transitions from one state to another are allowed (Rabiner 1989). The self-transition in state S_2 is plotted in red (dashed line) and defined as a_{22} in the transition probability matrix $\boldsymbol{A}$. Likewise, the transitions between states S_2 and S_5 as well as S_4 and S_3 are displayed respectively as a_{25} and a_{43}. In this scheme, the emission probabilities have not been plotted to simplify the representation.*

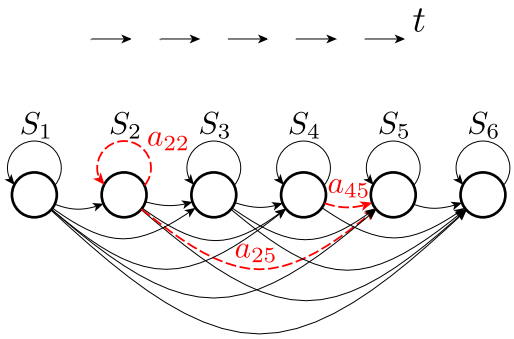
Outline of the chapter

In the next sections, we first analyze how the generalization performance changes according to the parameters used, then we discuss a critical issue in machine learning[3] algorithms that can appear with limited training data, and the way this can be avoided introducing an adaptive parameter dependent on the size of our training set. Additionally, we compare our proposed methods with several state-of-the-art approaches. One of the goals of this analysis is to predict how the parameters behave in different conditions and data structures in order to adapt them automatically without the help of the user. The fundamentals, techniques and state-of-the-art methods explained, the approaches proposed as well as the experiments and results presented in this chapter are based on Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012, where some paragraphs are taken verbatim from these sources as explained in Sec. 1.1.1.

4.1 Selection of the model structure for our efficient IWL framework

In our HMM framework, learning the parameters of a word-model is comparable to learn the sound of a word (Chaudhuri et al. 2011). HMMs are considered as a double-embedded stochastic process (Huang et al. 2001, Sec. 8.2, p. 378):

[3]Although our focus is on the speech recognition field, the approaches that we proposed could be also applied/extended to the modeling of other sequential data different from speech.



$$\boldsymbol{A} = a_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & a_{33} & a_{34} & a_{35} & a_{36} \\ 0 & 0 & 0 & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix}$$

$$\boldsymbol{\pi} = (\pi_i)^t = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 4.2: *HMM Bakis or left-to-right topology, where we do not need an initialization probability because the models are forced to start in the first state as the transitions between states are always from left to right reducing the number of parameters to compute as the not allowed transitions are permanently set to 0, e.g.* a_{43} *from Fig. 4.1 (adapted from Bishop 2006, Sec. 13.2, p. 614). In this scheme, the emission probabilities have not been plotted to simplify the representation.*

- In the first stochastic process, speech is represented as a succession of state transitions. This succession of states is not "directly observable".

- Secondly, the sequence of states is probabilistically related to an additional stochastic process, on this occasion observable, which produces the speech features, e.g. the emission/observation probabilities.

In the next sections, we analyze how the HMM parameters have to be selected and estimated in order to be appropriate for the efficient incremental framework proposed in the last chapter.

4.1.1 Model topology: connectivity of the network

In order to decide the most appropriate HMM representation, the purposed application might be taken into account, where the most used topologies are ergodic (Fig. 4.1), in which all kind of transitions are permitted, and Bakis (Bakis 1976) or left-to-right (Fig. 4.2) topology, in which only transitions from left to right are allowed (Bishop 2006, Sec. 13.2, p. 614; see also Rabiner 1989).

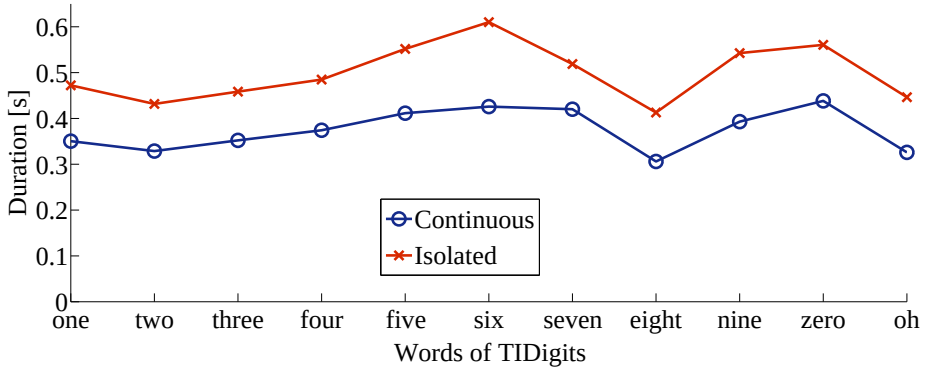


Figure 4.3: *Examples of the duration of the words when they are uttered in isolation ("x") and in continuous speech ("o"), where the pronunciation of isolated words is slower than the continuous speech according to the realized experiments. The duration of each word of the TIDigits database (Leonard and Doddington 1993, see Appendix A.2) have been obtained for this experiment using the forced alignment function of HTK (Young et al. 2009, see Appendix B).*

Analysis of the model topology[4]

In stationary cases compared to the non-stationary ones, only the data varies with time, but not the distribution producing the samples, which does not change (Bishop 2006, Ch. 13, p. 606). According to Huang et al. 2001 (Sec. 8.4.2, p. 397), "speech is a time-evolving non-stationary signal". In this context, the left-to-right (the direction of the time, see Fig. 4.2), topology owns characteristics to be the most convenient HMM topology for modeling speech units thanks to the "self-transition to each state" that enables the modeling of neighboring speech features generated by the same state and the transition to the right that ensures a consistent progression when the quasi-stationary speech fragments evolves (Huang et al. 2001, Sec. 8.4.2, p. 397).

Analysis of the skip transitions in the model[5]

In left-to-right topology, we can also introduce more constraints by defining admissible jumps between states or skip transitions, e.g. adding one skip transition or jump means that being located in state S_i in the next time step the location could be S_i, S_{i+1} or S_{i+2} and in case there are no skips, the possible transitions are going to state S_{i+1} or to stay in the current state S_i (Rabiner 1989; Young et al. 2009, Sec. 17.8, p. 273). A left-to-right topology and a small number of skip transitions ϵ have the advantage of reducing the number of parameters, as the elements a_{ij} are permanently set to zero if $j < i$ or $j > i + \epsilon + 1$ (Bishop 2006, Sec. 13.2, p. 614; Rabiner 1989).

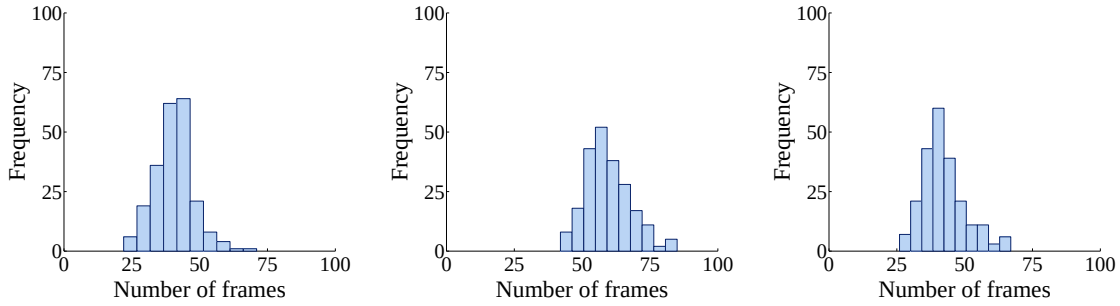
[4]See the analysis realized also in Ayllón Clemente et al. 2012.
[5]Extended from Ayllón Clemente et al. 2012.

Table 4.1: *WER (%) of the evaluation of the number of skip transitions ϵ in isolated word learning and isolated words (I) recognition as well as continuous speech (C) recognition. The left column is the number of training samples R (No.* ***X****) used for the evaluation of each configuration of the number of skip transitions (0, 1, 2 and 3 skips). The lowest WERs for I and C are marked in bold.*

No. ***X***	Nr. skips ϵ							
	0		1		2		3	
(R)	I	C	I	C	I	C	I	C
1	73.5	89.9	**44.5**	**80.8**	48.9	82.4	54.4	83.8
2	36.1	83.0	**33.7**	**77.7**	38.5	79.2	38.5	79.8
3	**15.2**	76.7	20.2	75.1	25.2	75.8	21.9	**74.4**
4	**9.9**	**68.2**	15.4	70.6	15.0	71.6	14.7	72.4
5	**7.7**	**54.7**	7.9	58.4	8.3	62.3	10.8	66.1
6	**4.1**	**43.4**	4.2	47.5	4.6	48.5	9.3	53.7
7	**2.2**	**32.2**	2.3	33.6	3.3	38.8	5.4	44.3
8	**2.0**	**23.4**	2.1	26.2	2.1	27.9	2.9	32.7
9	**1.6**	**18.9**	1.9	20.0	1.9	23.2	1.9	24.7
10	**1.3**	**14.6**	1.7	16.4	**1.3**	17.2	1.5	20.4
12	1.1	10.8	1.2	10.7	**1.0**	**10.0**	1.1	11.2
15	**0.9**	7.3	**0.9**	7.0	**0.9**	**6.9**	1.0	8.2
17	**0.7**	6.4	0.9	**5.6**	0.9	5.9	1.0	6.7
20	**0.6**	5.9	0.9	**5.0**	1.0	**5.0**	1.3	5.7
25	**0.5**	4.9	0.8	**3.8**	1.1	4.1	1.5	4.9
30	**0.4**	4.2	0.9	**3.3**	1.4	4.0	1.7	4.6
35	**0.4**	3.9	1.0	**3.0**	1.6	4.0	1.9	4.8
40	**0.4**	3.5	1.0	**2.8**	1.6	3.9	2.1	5.0
45	**0.4**	3.4	1.1	**2.5**	1.7	3.8	2.1	4.7
50	**0.5**	3.2	1.0	**2.4**	1.4	3.5	2.2	4.3
55	**0.4**	3.1	0.9	**2.2**	1.5	3.7	2.4	4.3

In Fig. 4.3, the experimentally extracted duration of the words of the TIDigits database (Leonard and Doddington 1993, see Appendix A.2) is depicted. The red line with the marker "x" represents the duration of the digits uttered in isolation and the blue line with the marker "o" the words pronounced in continuous natural speech. One can observe that the duration of the digits in isolation is longer than in continuous speech[6]. According to this, it seems beneficial to model changing duration when working with different recognition conditions. From the above descriptions of Bishop and Rabiner, one can directly infer that skip transitions incorporate flexibility in the topology of the model enabling jumps between the hidden states and then allowing different word duration from one ex-

[6]In Sec. 4.1.2, we show through experiments that also looking to a particular word uttered in a determined style, we will find differences in the duration.



(a) Digit: Two (b) Digit: Six (c) Digit: Eight

Figure 4.4: *Experimentally obtained distribution of the duration of words measured by their number of frames. The frequencies are exemplary estimated on three digits taken from the training set of the TIDigits database (Leonard and Doddington 1993, see Appendix A.2) for isolated digit recognition (only men were considered). The data were obtained off-line via the forced alignment function of HTK (Young et al. 2009, see Appendix B).*

emplar to another with the corresponding increase of the number of estimates as referred above. When using few training samples, we want to reduce the number of parameters, thus we investigate here if the use of a small number of skips is more suitable despite the loss of flexibility.

Experimental procedure

Similar to our experiments realized in Ayllón Clemente et al. 2012, we analyze the recognition scores for different configurations of the number of skip transitions (from 0 to 3) in order to evaluate which configuration is the most suitable one for our efficient learning framework.

We use different sizes of training sets in order to determine the behavior of this parameter when the amount of training samples increases. As extension to our previous experiments realized in Ayllón Clemente et al. 2012, these learning sets start with 1 sample per word, increment the number of samples stepwise and finishes with 55 samples per word. Here, the training sets also only contain isolated words. On the other hand, the evaluation test is subdivided into two groups: one comprising only isolated words (I) and other with continuous speech (C). So, one can see how relevant the flexibility in the duration of the exemplars is when employing a small number of training samples in both test cases. All training and test samples are extracted from the TIDigits database (Leonard and Doddington 1993, see Appendix A.2) and uttered by male speakers. In Table 4.1, the results obtained are shown.

As also stated in Ayllón Clemente et al. 2012, the achieved scores reveal that a topology with one skip transition (enabling different speaking rates as observed in Fig. 4.3) is superior to other constructions in continuous speech recognition when enough train-

ing data[7] are used, however, when we train with few samples (3 to 10), there are too many parameters for the small number of data samples provided and thus adding skip transitions only increases the number of parameters without improving the recognition results.

4.1.2 Number of hidden states per speech unit

According to the formulation of the HMMs provided by Rabiner 1989 (see Sec. 3.3.1), it is straightforward to notice that the number of hidden states is directly involved in the amount of parameters required to define the model[8]. Therefore, an appropriate choice of the number of states of the model could particularly reduce the required set of parameters as we stated in Ayllón Clemente et al. 2012. As reported by Rabiner 1989, there are two main approaches to set the number of states in each word-model:

- The number of states is approximated to the number of sounds (phonemes) that normally form a word, where models being composed of 2 to 10 states would fit. In this context, it seems plausible to use phoneme transcriptions to decompose the word-models and make an equivalence between the number of phonemes and the number of hidden states of the word-model. However, we do not have any phoneme information *a priori*, so the use of a pre-trained phoneme decoder would not fit into our scenario. On the other hand, we could also set all word-models with a fixed length. Nevertheless, a configuration with a fixed number of states is not advantageous when the terms to be learned present different duration (Fig. 4.3 and 4.4).

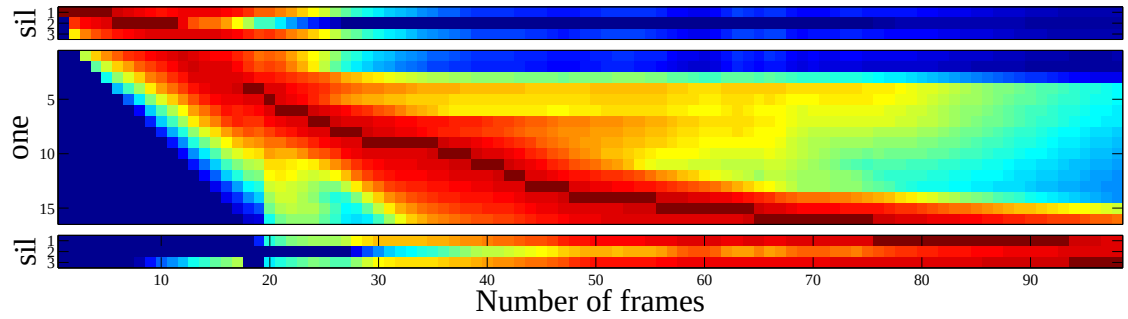
- Therefore, another technique is the estimation of the number of hidden states through the duration of the word, the concept known as the Bakis (length modeling) method/model (Bakis 1976), which consists in making the number of states proportional to the number of observations (frames) in the training samples of a word (qtd. in Geiger et al. 2010). In this way, there is a correspondence between each state and a portion of the frames associated with a word in each training sample.

Some researchers use so many hidden states as observations contained in the samples (e.g. Fink 2008[9]), Huang et al. 2001 (Sec. 8.4.2, p. 398) encourage to employ from 15 to 25 states for each second of the signal and others, as Geiger et al. 2010 combine different methods: fixed length (see Zimmermann and Bunke 2001) and Bakis length modeling (see Bakis 1976).

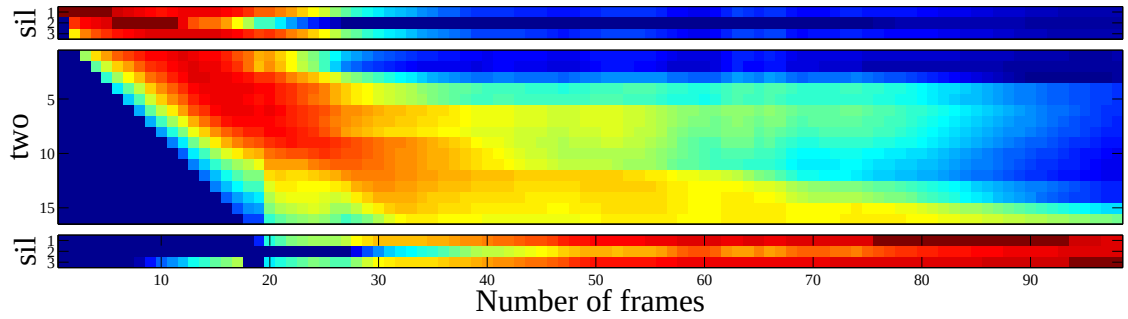
[7] Exceptions are 1, 2 and 3 samples.

[8] More information about the relevance of the number of states can be found in Bilmes 2006.

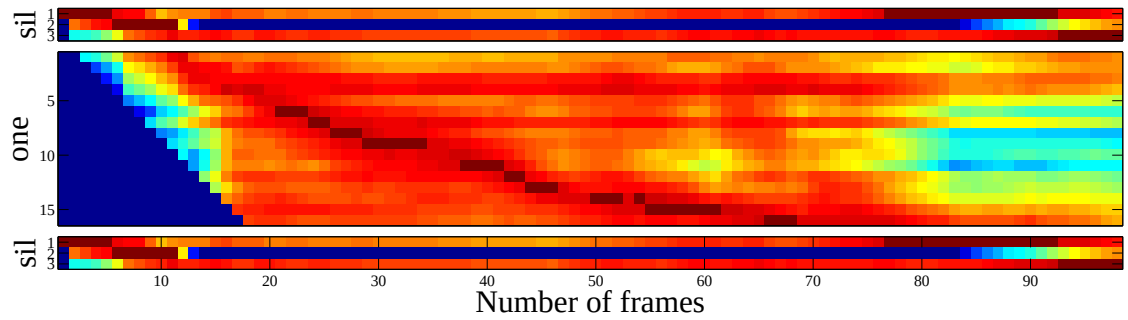
[9] In Esmeralda toolbox (Fink 2008, Sec. 13.3, p. 212), the number of states for each model is computed by the length of the shortest sample. For example, if the length of the shortest sample is 20 frames, then the number of states for the model would be 20.



(a) Viterbi decoding of the word "one" by the word-model "one"



(b) Viterbi decoding of the word "one" by the word-model "two"



(c) Emission probabilities of the Viterbi decoding of the word "one" by the word-model "one"

Figure 4.5: *Plot of the Viterbi decoding process of an isolated digit recognizer for a test sequence (using HTK (Young et al. 2009) and the interfaces named in Sec. 3.5), in which the word "one" is uttered. In (a) and (b), the color in each square represents the normalized partial probability $VIT_t(i)$ of being in a distinct state S_i of a word-model (y-axis) at a particular frame (x-axis), see Eq. 3.14. This is similar in (c) for the emission probabilities $b_i(\boldsymbol{x}_t)$. If the probability is low, the square is colored blue in contrast to high probability in dark red. The recognizer applies the grammar explained in Sec. 3.2: silence (sil), digit, silence (sil). In this example, the silence-model is an ergodic HMM with 3 emitting states and the word-models are Bakis HMMs with a fixed (16) number of hidden states. In (a), the proper HMM model for the decoding is employed in contrast to (b), where a wrong representation was used in order to appreciate the differences between the clear decoding path of the right model and the blurred path of the wrong one. In (c), the relevance that the observation probabilities have in the Viterbi decoding can be observed.*

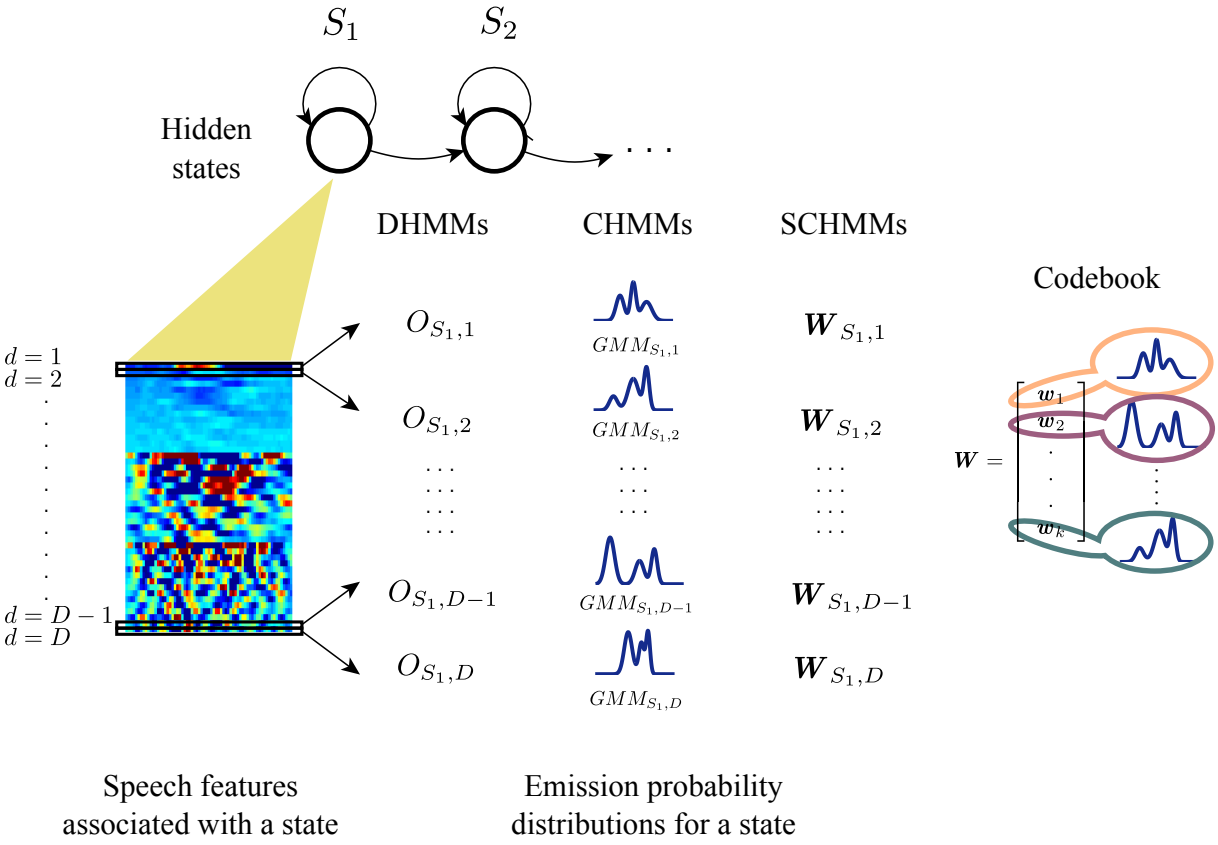


Figure 4.6: *Illustration of the different observation modeling methods for a hidden state: DHMMs, CHMMs, SCHMMs (see references in the body of the text for each type). Independent of the method, each of the feature dimensions d is represented by representative values $O_{S,d}$ in DHMMs, a distribution as GMMs in CHMMs or a vector of weights $\boldsymbol{W}_{S,d}$ to the elements of a codebook in SCHMMs. This means that the model possesses d values, d GMMs or d weight vectors per state.*

Analogous to our previous work in Ayllón Clemente et al. 2012, after considering the analysis of Rabiner 1989 and different configurations for this parameter, the number of hidden states for each new word-model is set proportionally to the number of observations (frames) of its training samples as the words to learn have normally different frame lengths in our experiments. Hence, we use a dynamic number of states (similar to the one used in Geiger et al. 2010) for each model making a relation between the average of the number of frames in the samples and the states (for example: 2 frames/state), where the number of frames of a training sample is calculated using forced alignment (HTK, Young et al. 2009, Sec. 2.3.3, p. 19) after the feature extraction process[10]. This method is also very sensitive when a significantly reduced number of training samples are provided, e.g. in

[10]In interactive isolated word learning scenarios, the use of voice activity detection (VAD) can substitute the forced alignment (for VAD, see e.g. Lokhande et al. 2012). On the other hand, in continuous speech other methods capable of segmenting words, e.g. based on energy (or other cues), can be used (see Prasad et al. 2004; Chowdhury et al. 2009).

the case of only one sample, the duration of this exemplar is going to be applied for the selection of the number of states of the HMM, hence a pre-processing is incorporated in order to limit the number of outliers (see extremes in Fig. 4.4). In the final configuration, we employ an initialization algorithm (to be presented in Chapter 5) to provide a suitable topology with a different number of states for each model.

4.1.3 Emission probabilities

In order to analyze the amount of information contained in these parameters, the emission probabilities, and their relevance[11] in the successful construction of an HMM model, we realized a visualization of the Viterbi decoding process in Fig. 4.5. Here, we compare the Viterbi decoding process for the utterance "one" using the word-model for "one" (Fig. 4.5(a)) and the word-model for "two" (Fig. 4.5(b)). The well-defined decoding progress in Fig. 4.5(a) is a useful reference for the reader to distinguish the contribution of these model parameters in Fig. 4.5(c) for an accurate decoding, i.e. the construction of a suitable model. By means of this analysis, we can conclude as expected that the emission probabilities have a determining role in an HMM model and that a non-appropriate or poor choice of these estimates will lead to deficient word-models.

There are three possibilities to model the emission probabilities in HMMs: discrete (DHMMs), semi-continuous (SCHMMs) and continuous (CHMMs or CDHMMs where D stands for density) HMMs (Huang et al. 2001, Sec. 9.5.1, p. 437). All of them are plotted in Fig. 4.6. According to Theodoridis and Koutroumbas 2003 (Sec. 9.6, p. 370), the modeling of continuous variables by discrete observations presents a main drawback: the possible loss of information during the vector quantization (VQ) of the signal, which can highly reduce the system quality.

The opposite approach is the modeling in continuous probability density functions, which can be generally modeled with Gaussian mixtures models, GMMs (see Fig. 4.7), (Huang et al. 2001, Sec. 8.3.1, p. 392). The use of GMMs as observation probabilities makes the number of parameters to be estimated quite large and without sufficient training data, the robustness of the system decreases (Theodoridis and Koutroumbas 2003, Sec. 9.6, p. 372). For CDHMMs using GMMs, there are a number of techniques to use in case of desiring less compact covariance matrices, namely tying them across various mixture components/hidden states or the use of diagonal covariance matrices (Huang et al. 2001, Sec. 8.4.5, p. 401). For the former case, Gales 1999 proposed to use a special kind of sharing covariance matrices between the different distributions although this kind of sharing normally needs to know the relations between some parameters previously. Other types of shared covariance matrices are the semi-continuous hidden Markov models, in which a set of mixture densities (codebook) is shared among all the states (Huang et al. 2001, Sec. 8.3.2, p. 394, Sec. 8.4.5, p. 401). However, a good set of codebooks or a pool of

[11] Although it is straightforward that these parameters are very important for the HMMs.

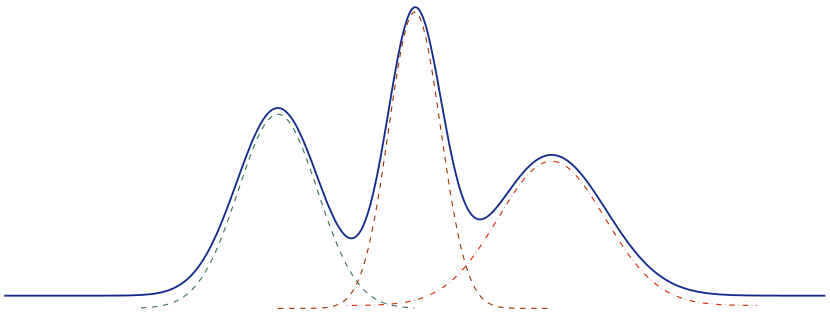


Figure 4.7: *In general, continuous distributions can be usually approximated by a set of Gaussian mixture models (GMMs), (Huang et al. 2001, Sec. 8.3.1, p. 392). In the picture, the combination of three "Gaussians" (dotted and dashed lines) models the continuous distribution (solid line).*

covariance matrices suggest the necessity of having a previously training and knowledge of the system. In the case of diagonal covariance matrices (variance vectors), although their use should be limited for the cases where the correlation among feature coefficients is not strong (Huang et al. 2001, Sec. 8.4.5, p. 402), we employ this simplification, similar to most ASR systems, despite the present correlation in RASTA-PLP features[12]. As we use diagonal covariance matrices, in the following we will employ indistinguishably the term covariance matrix $\boldsymbol{\Sigma}$ or variances $\boldsymbol{\sigma}^2$.

If we apply the principal component analysis (PCA) transformation, the feature vectors obtained are uncorrelated (Theodoridis and Koutroumbas 2003, Sec. 6.5, p. 219). In previous works in Ayllón Clemente et al. 2010a, PCA is applied to the training and test samples using a speech segment for initialization completely independent of the training and test samples and not containing the words to learn. Nevertheless, good scores can be also obtained without transformations as we show through this work.

4.1.4 Configuration of our acoustic models

As in our previous works (see Appendix D), we employ Gaussian mixture models (GMM) with diagonal covariances to represent the feature distribution in the hidden states (see Fig. 4.6), i.e. CHMMs and related to the number of components in each hidden state, we use 3 components[13] for the word-models and 6 components for silence- and pause-models. This number is derived from several preceding works of fellow research team members (see Domont 2009, Sec. 4.2.2, p. 49).

[12] An interested reader can visualize this correlation through Matlab©.
[13] The use of 3 components for the word-models is also employed by Murthy et al. 2004.

Additionally, our word-models are represented by left-to-right HMMs of varying length, the silence (auxiliary) model by an ergodic 3-emitting states HMM and the short pause-model by the middle state of the silence-model[14] (see Fig. 4.8, similar to HTK, Young et al. 2009, Sec. 3.2.2, p. 34). The latter are ergodic and simpler, because the segment to model is stationary and one to three states are enough (Huang et al. 2001, Sec. 8.4.2, p. 398).

Estimation of the AMs

As realized in our previous work in Ayllón Clemente et al. 2012, the initial parameters of the models are first computed for a single GMM and posteriorly split in different stages to increment the mixture components as advised by Sankar 1998 and Young et al. 2009 (Sec. 10.6, p. 174).

Following both sources, the parameters of the mixture component to split are first copied. Then, the weights of both copies are halved, and finally the means of the mixtures are slightly varied using the standard deviations of the component. After each splitting, several iterations are typically needed to refine the GMMs, where we can use in each iteration various learning criteria to estimate the model parameters as the ML and MAP (Sec. 2.3.4) analogous to our previous work mentioned above. In our system, we use a set of 20 iterations, the first 19 with ML, particularly Baum-Welch algorithm[15], and the last iteration comprises a MAP refinement. The Gaussian splitting procedure occurs during the first 19 iterations.

4.2 Overfitting problem[16]

One of the main drawbacks of using a small amount of training data is the overfitting problem (Bishop 2006, Sec. 1.1, p. 6), where the training algorithm may fit the parameters to some specific aspects of the data and then poorly generalize to unseen samples according to Ghahramani 2001. As mentioned by the last author, a method to avoid overfitting is to apply standard cross-validation learning. However, we cannot ensure enough training data on interactive systems to realize this technique. Alternative approaches for Ghahramani are to enable learning with few training samples through regularization methods with a penalty term (see e.g. Neukirchen and Rigoll 1998; Bishop 2006, Sec. 1.1, p. 10). In other cases, they directly constrain the number of independent parameters (e.g. Siu et al. 1999). One way mentioned before of reducing this number is by applying "parameter tying" (Huang et al. 2001, Sec. 8.4.5, p. 401; e.g. Yuan et al. 1998). The problem as mentioned before is to define these relations *a priori*, what is difficult in

[14] See also Ayllón Clemente et al. 2012.

[15] A complete description of Baum-Welch algorithm can be found in Sec. 3.3.3.

[16] This section and its corresponding subsections take numerous paragraphs (also state-of-the-art reviews), some of them verbatim, from Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012.

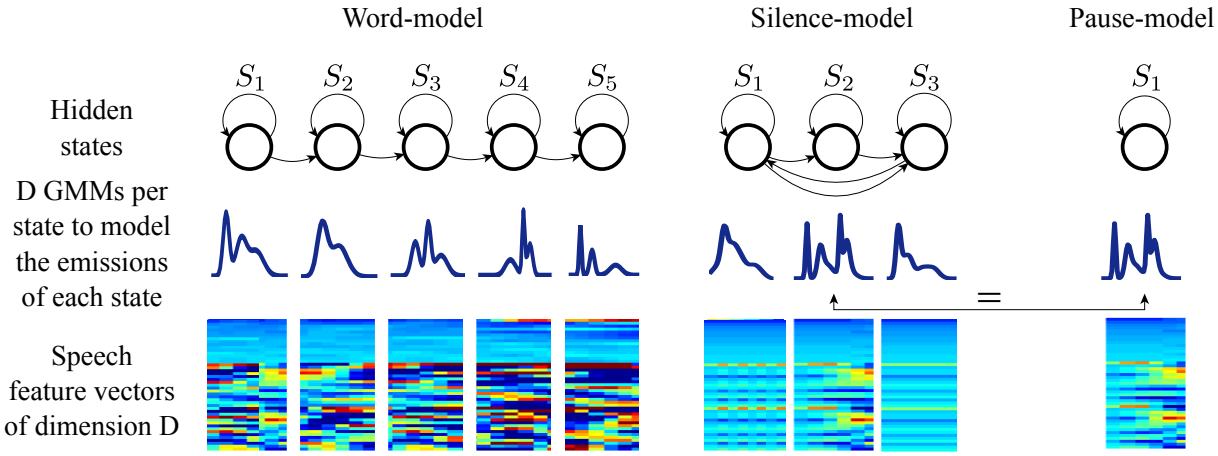


Figure 4.8: *Complete scheme of the word-, silence- and pause-models used in our framework. We use a left-to-right HMM topology without skip transitions, where each hidden state is modeled by a GMM with 3 components and the silence-model has an ergodic topology of 3 emitting states, where the emission probability of the middle state is shared with the pause-model as referred in the text.*

an incremental learning scenario without prior knowledge.

In Sec. 4.1.3, we analyzed that the emission probabilities, in our case GMMs, are decisive parameters for obtaining good performance in HMM frameworks. In GMMs, the magnitude of the variances can be crucial to recognize an unseen sample. According to Melin et al. 1998, when this magnitude is very low, the variance might not properly represent the distribution and samples belonging to the same class may not be identified as such if they differ to some extent to the training data. Thus, we analyze the behavior of the variances, especially their decrease in each developmental iteration (see Fig. 4.10). This decrease of the variances is the reason why the most extended technique to avoid overfitting is the integration of further constraints like the so called variance floor mentioned in Melin 1998. In cases of sparse training conditions, as the last author explained, the variance floor allows the system not to critically specialize to the training samples presented and guarantees that if other samples belonging to the model are shown, they have the chance to be recognized by the system (Fig. 4.9). We extend the current variance floor state-of-the-art methods going one step further by incorporating an adaptive factor dependent on the number of training data to prevent overfitting without applying a constant value that could be too large/small for the employed specific conditions (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

4.2.1 The variance floor

In hidden Markov models, expectation-maximization (EM) algorithms are often deployed to adapt the model parameters including the means and variances of their Gaussian

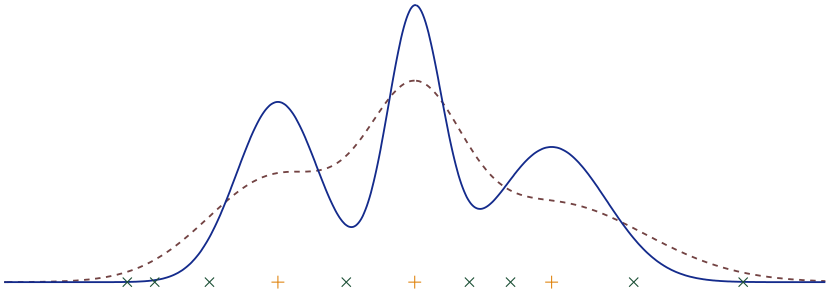


Figure 4.9: *When the estimation of the parameters occurs with a small number of training samples (indicated by some exemplary "+"), the model is overspecialized (solid line) and not able to recognize all unseen data ("x") belonging to the same class. In these situations, other models provide higher probabilities for the unseen samples. If the models are relaxed (dotted line), i.e. increasing the variance in the case of GMMs, the not modeled data can be detected as referred in the text.*

mixtures models (GMM), however a variance estimated from few training samples might not model properly the targeted distribution (Melin et al. 1998). In general, these estimated variances also decrease in each training iteration as Fig. 4.10 shows. If the variances drop too much the distributions become narrow with the risk of the occurrence of overfitting as mentioned by Melin and his colleagues. More precisely, there is a tendency of underestimating the variances in such cases, besides further numerical problems like division near zero (Ghoshal et al. 2005) resulting in the need of using a floor as threshold in order to avoid the decrease of the variances in each training iteration.

Generally speaking, any kind of parameter flooring can be viewed as a particular instance of interpolation with the uniform distribution (Huang et al. 2001, Sec. 8.4.5, p. 401). In practice, the training algorithms can be modified by including this lower threshold in the variance parameters, which is known as variance floor (Melin et al. 1998; see also Sim and Gales 2006; Melin and Lindberg 1999).

4.2.2 Computation of the variance floor

One simple way of computing the variance floor is to estimate the global variance of the speech segments and then scale it by a predefined factor (e.g. Stuttle 2003, Sec. B.1, p. 138; Young et al. 2009, Sec. 17.3.2, p. 257). In contrast, other researchers, as Yu et al. 2008, use a method to adapt the variance floor to the average of the variances over all GMMs in each dimension of the features or to floor the variances using a percentile variable (Young et al. 2009, Sec. 17.7.1, p. 268). In addition to the variance floor, the

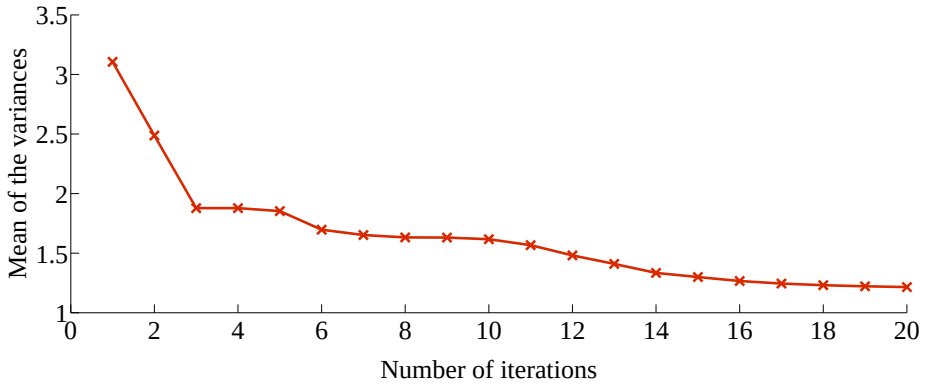


Figure 4.10: *The decrease of the estimated variances after each maximum likelihood (ML) iteration (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). The last iteration (20^{th}) of our system is however with MAP. The plotted example corresponds to the average of the variances of all components of the GMMs of all the word-models in TIDigits (Leonard and Doddington 1993, see Appendix A.2).*

minimum allowed variance is another threshold applied to the floor (Young et al. 2009, Sec. 17.7.2, p. 270), which can also replace it in cases where no variance floor is computed. Other authors suggest that the variances of the GMMs should be fixed at the beginning and not updated in each iteration of the re-estimation of the GMMs (Young et al. 2009, Sec. 8.3, p. 136; see Melin et al. 1998 for further sources).

Independently of the above approaches, Melin et al. 1998 identified three different manners to calculate and apply the variance floor depending on the HMM structure level considered for the computation: model-dependent, state-dependent and mixture component-dependent floor.

4.2.3 An efficient variance floor estimation

We apply in our incremental word learning framework a combination of the first approach referred to as model-dependent floor and one similar to the mixture-dependent method (from Melin et al. 1998) jointly with the adaptation that we propose in this section (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). In our system, instead of applying the same values for all the word-models, we calculate the variance floors considering each model independently as we cannot predict the differences between the models that are still unknown[17] (see the end of this section).

The next step is to decide which floor constant is optimal for our task (see Sec. 3.5.2 and previous works in Appendix D) evaluating how the variances behave when the number of samples R is reduced.

[17]The exemplars to build these models have not been presented to the system yet.

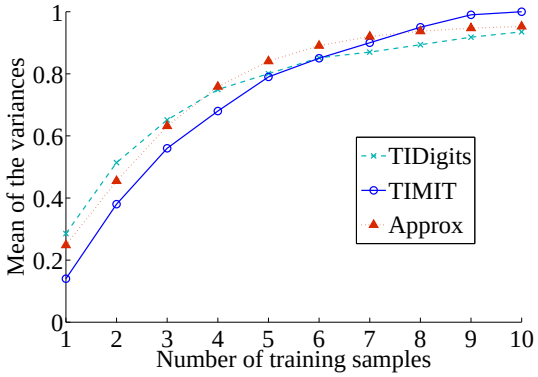


Figure 4.11: *The average $\overline{Var}(R)$ of the variances of the GMMs (3 per hidden state) over all the models of the TIDigits (Leonard and Doddington 1993) and TIMIT (Garofolo et al. 1993) databases for different numbers of samples R (x-axis) and its approximation by the Gompertz function (dotted line) are depicted here (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). For the estimation of the average, two sets of training data were randomly selected from TIDigits and TIMIT (see Appendix A). For the illustration, a training sample is a speech segment containing one word (silences are ignored for this experiment). We can take the behavior of the average as representative model because the variances of each feature dimension behave similarly according to our experiments plotted in Fig. 4.12.*

Analysis of the variance floor

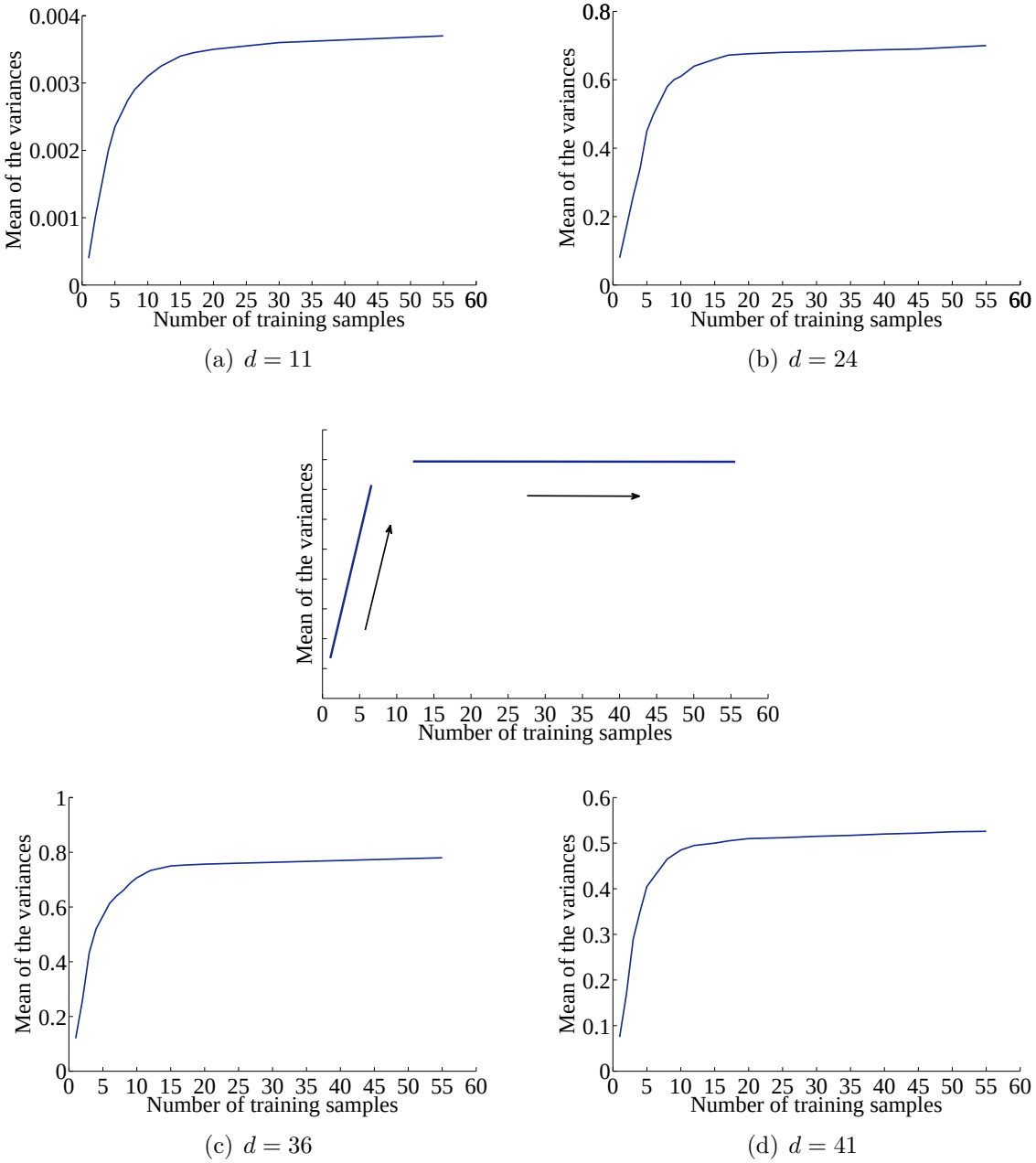
In Fig. 4.11, the average $\overline{Var}(R)$ over the variances of all Gaussian mixture models for all feature dimensions and models after one iteration of the ML and MAP estimation are displayed. One can observe that the variances rapidly grow at the beginning and then they increase slightly until they saturate at $\overline{Var}(\infty)$. Consequently, they seem highly correlated to the number of training samples when this number is relatively small[18].

In interactive and dynamic environments, it is not possible or advisable as referred in Chapter 1 to adapt continuously and manually the parameters of the system during operation. Hence, it would be very beneficial to have self-adaptive parameters as our extended variance floor that scales automatically according to the amount of training samples independent of the terms to be learned (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

Scaling the variance floor

According to the Fig. 4.11, the behavior of the variance floor can be easily modeled using a section of a sigmoid-like function, called the Gompertz function[19]:

[18] As observed by other authors, e.g. Melin and colleagues, see beginning of this section.
[19] After Benjamin Gompertz.



(a) $d = 11$ (b) $d = 24$

(c) $d = 36$ (d) $d = 41$

Figure 4.12: *Behavior of the variances of the GMMs in several feature dimensions (d) for sets with different amount of training samples R from the experiments we realized. There are so many dimensions as the speech features used possess (in our case 45 = 15 RASTA-PLP +15 first derivatives + 15 second derivatives, see definition of the data source in Sec. 3.2). The variances behave in a similar way (see central picture) independently of the speech feature dimension considered as demonstrated through the experiments.*

$$G(R; a, b, c) = a \cdot e^{b \cdot e^{c \cdot R}} \tag{4.1}$$

with the parameters a, b and c adapted to the data. In Fig. 4.11, the dotted red line depicts the estimated Gompertz function $var(R) = f(G(R; a, b, c))$, which best approximates the variances in our experiments. Once the behavior of the variances is estimated, $var(R)$ is normalized such that $var_{f1}(R)$ converges to 1 for $R \to \infty$, resulting in $var_{f1}(R) = \overline{Var}(\infty)/var(R)$. However, when the number of training samples drops, a larger variance floor is required (see Melin et al. 1998). The reinforcement factor $r_f(R) = 1 + e^{-R}$ is introduced to compensate this effect, by increasing the value of the function $var_{f1}(R)$ for a very small number of training samples. Finally, by multiplying the reinforcement factor $r_f(R)$ and the normalized variance function $var_{f1}(R)$ estimated above, one obtains a scaling function for the variance floor depending on the number of samples used:

$$var_f(R) = \frac{\overline{Var}(\infty) \cdot (1 + e^{-R})}{f(G(R; a, b, c))} \tag{4.2}$$

We use a generalized expression for the variance floor estimation used in the state-of-the-art, based on a larger number of training samples, through the Eq. 4.3. Here, K is the scaling constant and $\overline{V}_d$ stands for the three state-of-the-art types of variance floor estimations described in Sec. 4.2.2 and evaluated at the end of the chapter. The first case is the average variance over all mixture components in each dimension d (Yu et al. 2008), the second type is the global variance of the speech features data in each feature dimension d (e.g. Stuttle 2003, Sec. B.1, p. 138; Young et al. 2009, Sec. 17.3.2, p. 257) and the third one is the variance of the dimension d to find in the x^{th} percentile (in this case K=1), (Young et al. 2009, Sec. 17.7.1, p. 268).

$$V_F(d) = K \cdot \overline{V}_d \tag{4.3}$$

The proposed variance floor estimation is depicted in Eq. 4.4, where the variance floor function $var_f(R)$ and the scaling factor K are multiplied by $\overline{V}_d$ yielding a variance floor value depending not only on the dimension of the feature d in contrast to Eq. 4.3 but on the number of training samples R (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

$$V_F^*(d, R) = K \cdot var_f(R) \cdot \overline{V}_d \tag{4.4}$$

Computation of the floor in incremental learning

In our framework[20], in order to save the re-estimation of previously learned models when a new model enters the system, we only compute the parameters of the current model

[20] Also in the previous works mentioned in Appendix D.

(except in the discriminative phase where other competing models are adapted). Hence, the variance floor estimated only takes into account the parameters of the current model and its training data and is employed in the supervised phase of the initialization (see Chapter 5) and re-estimation steps.

4.2.4 Evaluation of the proposed scaled floor[21]

Algorithms to evaluate

We evaluate and compare state-of-the-art variance floor approaches enumerated in Sec. 4.2.2 to our method V^{*}[22], which incorporates a scaling factor adapted to the number of training samples (Fig. 4.13) in order to analyze if our technique can be applied to all cases. We use as baseline a system that relies on a flexible number of states and uses a uniform parameter initialization (see Chapter 5 for details). The state-of-the-art variance floor methods that we investigate are:

- The variance floor V_G (Yu et al. 2008): it computes the average variance over all mixture components in each feature dimension at each iteration of the learning algorithm. Then, this value is scaled by some constant and used as variance floor. As mentioned before, we compute the variance floors taking only into account the current model. Hence, the variance floor method only considers the variances of its own model. This variance floor method is used as reference to measure the improvement of our approaches.

- The variance floor method $V_{\bigcirc}$ (Stuttle 2003, Sec. B.1, p. 138; Young et al. 2009, Sec. 17.3.2, p. 257): it is calculated via the global variance. The use of a global variance as the variance floor $V_{\bigcirc}$ is almost independent of the number of training samples, as if we take the mean of the global variance of all possible combinations of data for each number of training samples, it usually tends to a constant value[23]. Following the model-dependent assumption, the global variance used for the calculus of the variance floor is estimated taking only into account the own training samples of the model. This floor does not change during the training process. Otherwise, the method referred to as variance floor V_G is very sensitive to the estimates realized by the expectation-maximization algorithm as it takes the average of the variances of the GMMs for each feature dimension and such average decreases in each iteration as shown in Fig. 4.10.

[21] As mentioned at the beginning of the section, several paragraphs (some verbatim) are taken from previous works as Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012, however some experiments have been extended in contrast to the previously published ones.

[22] First published under Ayllón Clemente et al. 2010a.

[23] According to several experiments realized by the author in the context of speech features. Some exceptions appeared at 1 and 2 samples.

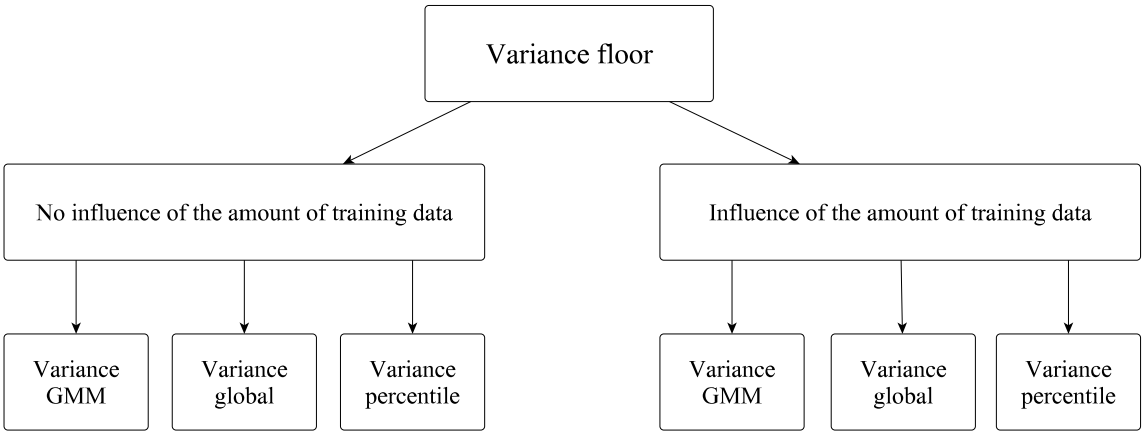


Figure 4.13: *Overview of the experiments realized to compare the different variance floor methods with our technique (Ayllón Clemente et al. 2012).*

- The variance floor $V_{\%}$ (Young et al. 2009, Sec. 17.7.1, p. 268): the threshold to be applied as variance floor for the variances of each dimension is the x^{th} percentile of the distribution of the variances for such feature dimension (see Sec. 4.2.2), i.e. it indicates the number of Gaussian components (as a percentage of the total number of Gaussian components) to update their variances to the variance floor. For the computation, only the variances of its own model are taken into consideration (i.e. model-dependent).

Databases

For the evaluation of our approach against other techniques, we use the TIDigits database containing numbers from "1" to "9", "zero" and "oh" (Leonard and Doddington 1993, see Appendix A.2). We employ 250 subsets from 1 to 10 training samples to obtain 25 cross-validation vectors for each configuration of a determined amount of training samples. The training samples are composed of adult male speakers uttering isolated numbers. On the other hand, the test set contains adult male and female speakers and the sentences comprise both isolated word and continuous speech utterances[24], in the last case only adult male speakers are employed. The test set containing adult female speakers is used to measure how the system behaves when quite different unseen data are presented[25].

[24]The continuous speech utterances analysis is extended here to all the methods to compare and not the best approach as in Ayllón Clemente et al. 2012.

[25]In Ayllón Clemente et al. 2010a, a test set containing female speakers was also used, however the parameterization and final configuration of the system was not the same as here.

Parameterization

The fact that each method uses different estimation techniques makes difficult to apply the same variance floor multiplier in all approaches. In the V_G method, applying a small scaling factor (without considering our additional factor dependent on the number of training samples) would not be enough in order to avoid overfitting. However, a large multiplier would be inefficient in the case of the global variance, as it would allow too many samples to be misclassified. Therefore, we select the corresponding best suited parameters for each technique[26].

Different experiments have been realized with the baseline system in order to select these suitable values. The values used for V_G and V_O are between [1/15, 2/3] and [1/20,1/2] respectively. The percentile method is a compromise between previous approaches as it selects the amount of variances that are supposed to be underestimated. In this case, the values used are between [10,60].

After the experiments realized, the mean constant values selected are $\sim$1/2, $\sim$2/5 and $\sim$50% for each approach respectively. All these values are prior to correction.

Results

As referred in Chapter 3, we evaluate the contribution of each approach that we propose using WER (%)[27]. The results of the methods using the scaling factor approach, which depends on the number of training samples, are shown in Table 4.2(b) and without the scaling factor in Table 4.2(a). In Table 4.3[28], the average of the relative improvements of all the proposed techniques and standard approaches for isolated word recognition (IWR) in men and women as well as for continuous speech recognition (CSR) with male speakers are presented. Similarly, we display in Fig. 4.14 the absolute and relative improvements, in male speakers for isolated word and continuous recognition, of our proposed method jointly with the percentile variance floor technique $V^*_{\%}$ against the baseline system V_G.

Compared to the variance floor estimation methods (Eq. 4.3) in state-of-the-art, our method V^* (Eq. 4.4) improves the results when a very small number of training samples are used. The technique of the variance floor percentile demonstrates that it does not only provide the best results in standard approaches, but also with our proposed scaling factor (Table 4.3). Although some overlap (see bars representing the maximum and minimum values of the cross-validations) exists between the variance floor percentile approach computed with our scaling factor $V^*_{\%}$ and the variance floor method computing the average of the variances of the GMMs V_G (Fig. 4.14(c)), the former achieves almost

[26]The parameterization is taken from our previous work in Ayllón Clemente et al. 2012.

[27]The results presented here are an extension (female speakers and more continuous speech experiments) of the ones explained in Ayllón Clemente et al. 2012.

[28]The values in Table 4.2 are rounded. The values calculated in Table 4.3, although also rounded, were calculated with the values of Table 4.2 without rounding.

Table 4.2: *Word error rates (WER %) of the state-of-the-art methods and our proposed approach. For each method, the WER values represent the mean of a 25-fold cross-validation on the training set evaluated on separated male (M) and female (W) speakers test sets with isolated words and another test set (C) with continuous speech utterances produced by male speakers (extended from Ayllón Clemente et al. 2012.). "No.* ***X****" stands for the number of training samples (R). The variance floor methods are the variance floor employing the average of the variances of the GMMs V_G, the variance floor using the global variance $V_{\bigcirc}$ and the variance floor according to the percentile $V_{\%}$. In the case of V_{-}^{*}, the * represents the incorporation of the scaling factor depending on the number of training samples in the computation of the variance floor. The best WERs are marked in bold.*

(a) Variance floor estimation without the influence of the No. ***X***

No.	V_G			$V_{\bigcirc}$			$V_{\%}$		
X	M	W	C	M	W	C	M	W	C
1	67.8	84.9	85.8	**64.1**	**77.4**	83.1	69.1	82.3	**82.1**
2	34.3	78.1	57.3	32.6	**54.1**	56.1	**32.3**	65.4	**56.0**
3	11.1	54.3	35.5	11.7	48.6	33.5	**10.9**	**39.3**	**31.0**
4	8.9	48.2	29.7	7.8	32.3	27.8	**7.4**	**29.3**	**25.8**
5	7.4	42.8	24.9	6.2	29.7	21.9	**3.2**	**25.2**	**20.8**
6	2.7	41.5	19.0	2.6	**24.3**	16.1	**2.3**	25.4	**15.9**
7	2.0	38.3	12.5	2.0	**22.1**	11.2	**1.8**	24.2	**11.1**
8	**1.7**	36.4	10.4	1.9	25.6	9.5	**1.7**	**23.8**	**9.4**
9	1.5	35.6	10.1	1.4	**22.3**	8.9	**1.3**	23.6	**8.4**
10	1.2	32.8	9.3	1.2	**22.1**	8.3	**1.1**	22.4	**7.8**

(b) Variance floor estimation with the influence of the No. ***X***

No.	V_G^*			$V_{\bigcirc}^*$			$V_{\%}^*$		
X	M	W	C	M	W	C	M	W	C
1	**48.9**	81.2	74.4	57.8	81.4	73.5	61.1	**75.4**	**71.2**
2	31.7	69.8	53.0	30.6	57.6	45.3	**29.8**	**52.3**	**44.3**
3	7.8	47.3	34.9	9.4	**35.3**	31.7	**7.2**	36.3	**30.5**
4	7.5	35.4	25.9	7.1	32.1	24.2	**6.4**	**27.5**	**22.2**
5	5.9	40.3	18.5	5.3	27.3	18.7	**3.7**	**23.1**	**17.8**
6	2.4	37.2	15.5	2.1	23.3	15.2	**2.0**	**22.4**	**14.2**
7	1.9	36.1	10.3	2.0	**22.4**	9.4	**1.5**	22.6	**9.2**
8	1.5	33.4	9.9	1.5	23.4	9.3	**1.4**	**21.7**	**9.2**
9	1.4	32.3	9.4	**1.1**	22.1	8.8	**1.1**	**21.9**	**8.2**
10	**1.0**	31.2	7.1	**1.0**	20.6	7.1	**1.0**	**20.4**	**6.4**

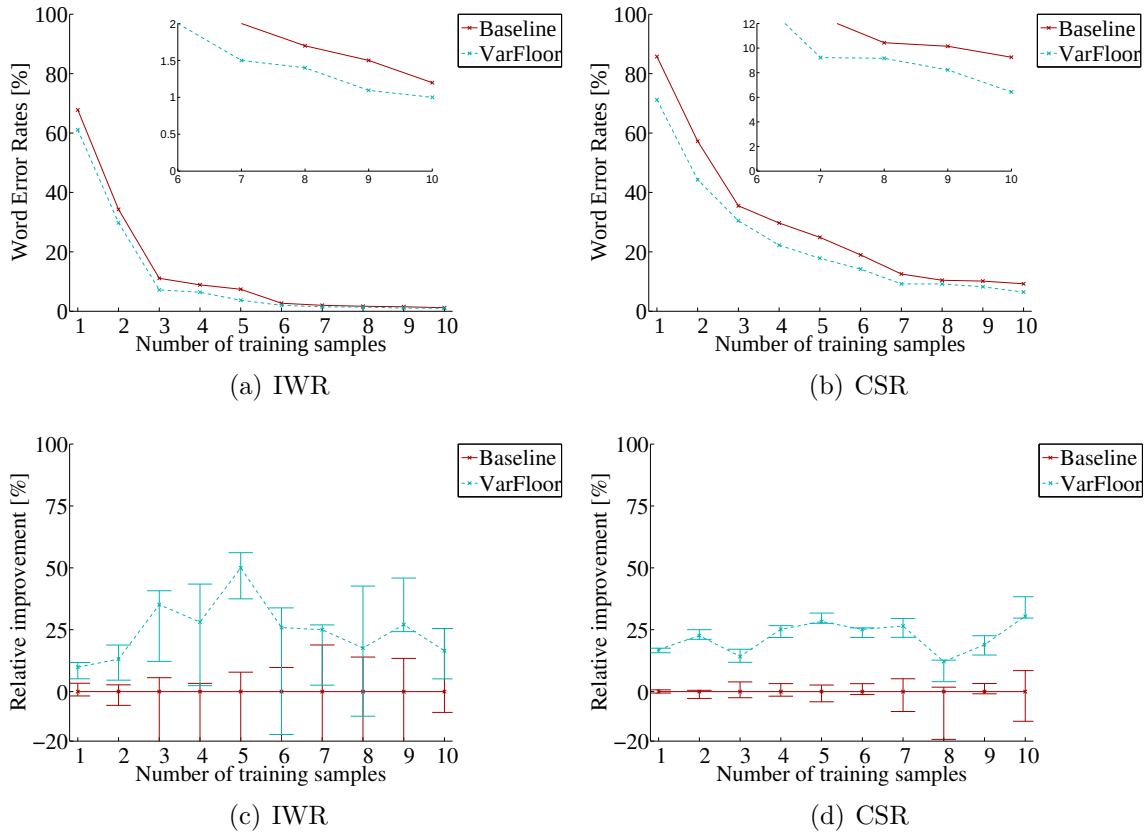


Figure 4.14: *Word error rates (a,b) and relative improvement (c,d), in test sets using male speakers, of the proposed method with the variance floor percentile $V^*_{\%}$ (VarFloor) against the technique of the average of the variances of the GMMs (Baseline), see Ayllón Clemente et al. 2012. In Fig. (a) and (b), a zoom view is provided in order to distinguish the values of both methods when the resolution of the scale axis is too coarse. The bars in Fig. (c) and (d) indicate the minimum and maximum values of the improvements of the 25 cross-validations. They target to measure a possible overlap of the improvements.*

25% (in average) relative improvement. In the continuous speech case, we observe that, although the results are not as good as in isolated digit recognition, our method V^* notably improves the recognition scores.

In the female test set, a larger improvement is obtained in most cases. However, no samples are used for the learning stages containing female speakers. For this reason, even though the recognition scores are quite improved for female speakers, the system performs better for male speakers than for female speakers. If the distributions are widened, the unseen test data such as samples containing female speakers can be more easily recognized. However, we cannot increase $\boldsymbol{\sigma}^2$ too much, because that would include data not belonging to the class.

In conclusion, the introduction of a variance floor dependent on the number of training samples clearly outperforms baseline systems. The most successful combination is

Table 4.3: *Relative improvements of the word error rates (WER %) comparing all the variance floor techniques explained in this chapter (extended from Ayllón Clemente et al. 2012 (female speakers scores)). For abbreviations, see Table 4.2.*

		Not influenced by the number of training data			Adapted taking into account the amount of training data		
		V_G	$V_{\bigcirc}$	$V_{\%}$	V_G^*	$V_{\bigcirc}^*$	$V_{\%}^*$
ISOLATED	M	–	3.2	12.7	15.2	16.7	24.8
	W	–	29.7	30.3	9.9	33.1	37.0
CONTINUOUS	M	–	8.6	12.0	13.3	18.2	22.0

achieved via the percentile technique and our proposed adaptive floor $V_{\%}^*$.

4.3 Synopsis

In the last chapter, we presented the architecture and main features of our efficient incremental word learning framework[29]. The efficiency of the system depends on the number of training samples that are required to obtain a good generalization performance[30]. Hence, we studied how to learn words with limited training data, i.e. how to define the word-models under these conditions, through some experiments and the existing literature. In particular, two issues were analyzed: an appropriate selection of the parameters of the HMMs and the avoidance of overfitting through the introduction of an adaptive variance floor dependent on the number of available training samples (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

First, we analyzed the most suitable configuration of HMMs for our word-models to learn. According to the consulted literature in this chapter, the Bakis or left-to-right topology ensures the appropriate modeling of the temporal progression of speech and needs less parameters than the ergodic one. The topology for the word-model, which we selected, does not possess any skip transition. This configuration has demonstrated in the realized experiments in Sec. 4.1.1 and previous works to be more beneficial when the number of training samples is very limited. In our system, words are learned in isolation but the system can recognize them in isolation and in continuous speech, which is characterized through a faster speech rate (see references and figures in Sec. 4.1.1). When the number of training samples is more generous, the best configuration to recognize continuous speech is with one skip transition according to the experiments realized. As argued in Sec. 4.1.1 and

[29]Based on our previous works, see Appendix D.
[30]The argumentation and corresponding references can be found at the beginning of the chapter.

previous works, the addition of one skip transition increases the flexibility of the models and enhances the performance when dealing with different word duration. Related to the duration of the speech units is also the number of hidden states that each model should have (see Rabiner 1989). Following the argumentation of some authors enumerated in Sec. 4.1.2, the most profitable way of selecting the number of hidden states is according to the duration in frames of the words to learn. Finally, the emission probabilities of the hidden states are represented by GMMs, a suitable and widely extended approach to model a large set of different distributions (Huang et al. 2001, Sec. 8.3.1, p. 392).

The second part of the chapter was dedicated to the overfitting problem or the overspecialization of the model parameters when a reduced number of training samples are used for the learning phase (see Bishop 2006, Sec. 1.1, p. 6). In this context, we discussed the importance of the variance floor as a threshold to stop the decreasing value of the variances of the GMMs in each training iteration (see Melin et al. 1998 and Fig. 4.10). Similarly, the relation between the number of training samples and the narrowness of the distributions (overspecialization - detecting less unseen samples belonging to their class) was analyzed in Fig. 4.9. Therefore, we proposed a dynamic variance floor threshold, which depends on the training data available in each case (see Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). We evaluated our approach together with three different state-of-the-art methods, namely the variance floor calculated using the average of the variances (Yu et al. 2008), the global variance of the data (Stuttle 2003, Sec. B.1, p. 138; Young et al. 2009, Sec. 17.3.2, p. 257) or depending on a percentile (Young et al. 2009, Sec. 17.7.1, p. 268). The combination of our technique with the percentile approach provides the system with an approximate relative improvement of 25% over standard methods.

By means of the mentioned above successfully adaptive configuration of the framework, our system is able to set up the required parameters in a suitable manner for changing conditions, i.e. variable data structures and different availability of training samples. This constitutes a considerable advance towards a fully automated user-friendly system.

Once the model definition has been realized considering sparse learning conditions, the next step in our framework is the initialization of the estimates in order to find a suitable local optimum in the computation of the model parameters, see Fig. 3.10 and Chapter 5. Additionally, the generalization performance for the appropriate incremental classification of new terms will be enhanced in our system through different discriminative learning strategies, which explanation is subjected to Chapter 6.

5
Model bootstrapping

In the model definition phase explained in the last chapter, the structure of the word-model was configured. Following the related literature, we arranged the disposition of the different parameters to represent the sequential nature of speech and considered efficiency aspects by including an adaptive floor dependent on the number of training samples[1].

After the definition of the structure of the word-model, the parameters of this model have to be determined according to the standard ASR systems (see Fig. 2.5). The expectation-maximization algorithms used for the estimation of the HMM parameters get easily stuck in local optima, what makes the initialization of the parameters a critical phase to obtain suitable results (Huang et al. 2001, Sec. 8.4.1, p. 396). This matter is even more relevant if we consider that this estimation in our system takes place in limited training data conditions. For that purpose, we investigate an approach to obtain a good set of initial estimates in order to achieve a suitable optimum[2]. This initialization or model bootstrapping phase takes place previous to the estimation of the parameters, see Fig. 3.1.

Outline of the chapter

We start the chapter reviewing the most used state-of-the-art initialization methods in the literature. After that, we present a model bootstrapping method, which introduces a novel manner of initializing the HMM parameters in order to obtain good optima when few training samples are used, being the highlight of this initialization the proposal of a multiple sequence alignment (MSA) technique[2] that provides an appropriate topology and initial estimates of the word-model to train. Finally, we evaluate our approach and compare it with other well-known techniques introduced at the beginning of the chapter. The state-of-the-art methods described, the algorithms proposed, the experiments and their resulting scores are founded[3] on Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

[1] First presented in Ayllón Clemente et al. 2010a.
[2] First published in Ayllón Clemente and Heckmann 2009 as well as Ayllón Clemente et al. 2010b.
[3] Some paragraphs are also taken verbatim from these previous works.

5.1 Initialization methods[4]

Since the training process in HMMs is based on expectation-maximization algorithms, the efficiency of the procedure strongly depends on the initialization of the parameters as mentioned before (Rabiner 1989, Huang et al. 2001, Sec. 8.4.1, p. 396). The standard initialization approach explained by Rabiner 1989 segments the training data frames into states and mixture components by means of the well-known K-means clustering (explained in Sec. 5.1.1) and Viterbi decoding. This joint initialization technique is executed in a supervised manner (it needs labeled training data samples), but this bootstrapping method gives the possibility to iteratively estimate new models, as they are determined independently from each other (see Young et al. 2009, Sec. 8.2-8.3, pp. 132-136).

This standard initialization proposal applies the concept of HMMs as generative models, able to produce speech samples (Young et al. 2009, Sec. 8.2, p. 132). Therefore, the means and variances of the GMMs related to a hidden state could be computed if the identity of the state generating the corresponding features of the samples was known and the transition matrix estimated by summing and normalizing the number of frames that are related to each state (Young et al. 2009, Sec. 8.2, p. 132; Rabiner 1989). The components of the GMMs can be estimated assigning the feature vector to the component with the highest probability and applying K-means clustering (Young et al. 2009, Sec. 8.2, p. 133). The Viterbi algorithm decodes each training sample providing the most likely succession of states and computes the likelihood of the data so that, the estimation is iteratively repeated until a local maximum of the likelihood over all training samples is achieved (Young et al. 2009, Sec. 8.2, p. 133).

As we cannot decode without HMM parameters, whether the first iteration includes a uniform segmentation of the data with the following association of the segments to the different states (left-to-right topology in speech recognition) or the parameters are initialized randomly (Young et al. 2009, Sec. 8.2, p. 133; Rabiner 1989). An illustration of the whole standard process can be found in Fig. 5.1.

Another state-of-the-art initialization method is “flat start”, where all Gaussian mixtures models (GMMs) of the classes are initialized with identical parameters equal to the global mean and variance of the training data (Itaya et al. 2005; Young et al. 2009, Sec. 8.3, pp. 135-136). The advantage of this method is that the models can be easily initialized without the need of labels, i.e. unsupervised (Young et al. 2009, Sec. 8.3, p. 135) and its disadvantage is that either the global mean and variance are exclusively computed from the training data of the new cluster, or all the computations have to be repeated each time a new class is introduced into the system. Some instances of similar initialization

[4]The review of the state-of-the-art methods presented by Rabiner 1989, Young et al. 2009 (Sec. 8.2-8.3, pp. 132-136), Itaya et al. 2005, Brandl et al. 2008, Iwahashi 2007 are based on Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012, where some paragraphs of this review are taken verbatim from both previous works.

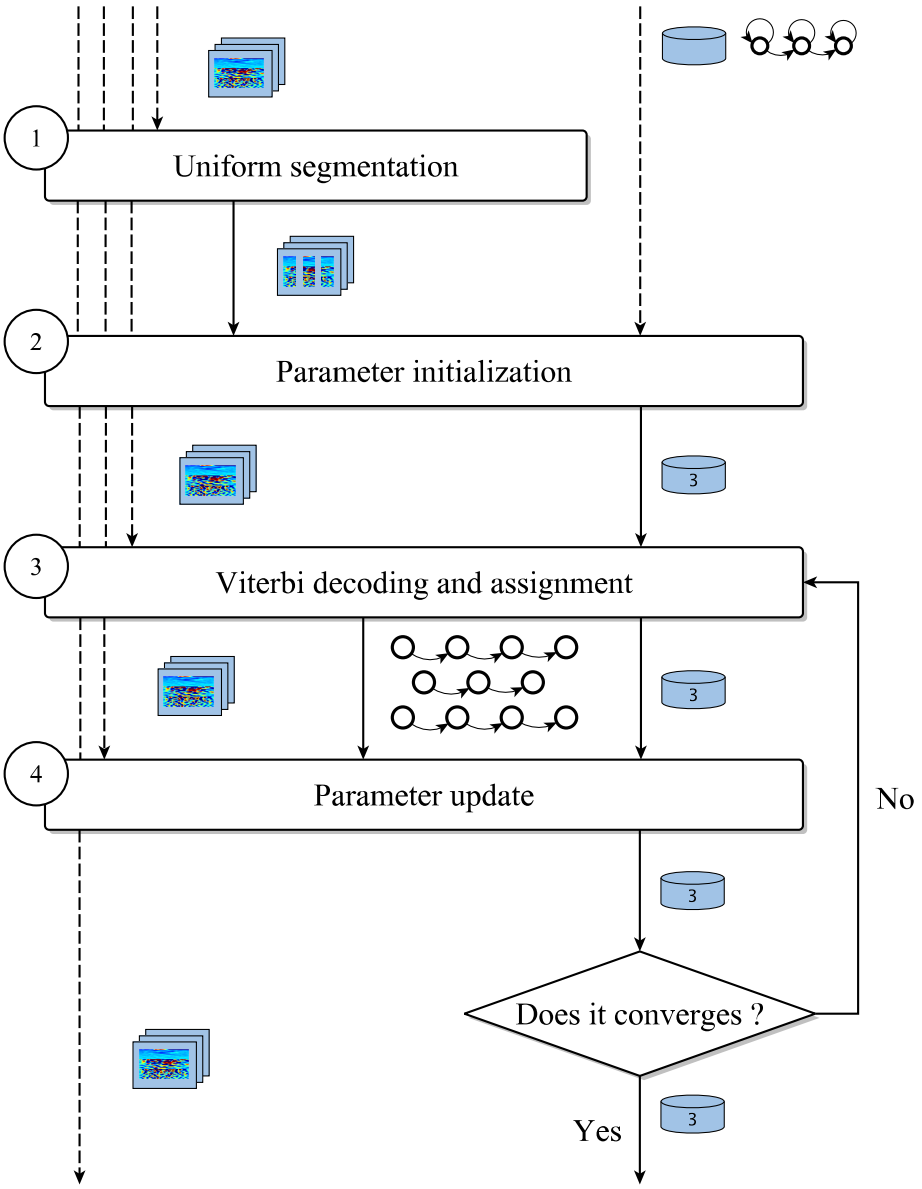


Figure 5.1: *Standard bootstrapping method for HMM parameters based on the following steps (Young et al. 2009, Sec. 8.2, pp.132-133; Rabiner 1989). First, the feature vectors of the training samples belonging to the word-model are segmented uniformly (1^{th} block). Next, the parameters of the empty models are initialized according to the segmentation (2^{th} block). It is also possible to initialize the parameters randomly. Afterwards, the training samples are decoded by the Viterbi algorithm and the feature vectors are associated with the corresponding states (3^{th} block). To determine which feature vector is assigned to each component of the GMMs in a particular state, K-means clustering can be used. Finally, the parameters are updated (4^{th} block) and the previous step (3^{th} block) is repeated until the process converges. This plot represents a piece of an ASR system, more precisely the steps between model definition and estimation of the parameters.*

techniques to the above mentioned ones can be found in Nathan et al. 1996.

Moving towards an unsupervised initialization technique for incremental learning in interactive environments, some authors like Brandl et al. 2008 and Iwahashi 2007 initialize the system by means of an unsupervised acquisition procedure for building sub-word models, which are subsequently used to construct word-models. In Brandl 2009 (Sec. 6.3.1-6.4.3, pp. 75-82), the initialization technique is partly based on K-means clustering and Viterbi decoding, where a small time slot of stored input speech is segmented in K clusters to train the output probabilities of K HMMs. The probabilities to go from one HMM to another are estimated by frame level counts (Brandl 2009, Sec. 6.3.1, p. 75). Due to Monte Carlo sampling, the most frequent succession of HMMs are chained yielding an initial set of sub-word models in Brandl's system. Subsequently, the Akaike information criterion (AIC) is employed by Brandl to optimize the size of the set of sub-word models (Akaike 1974). Finally, the sub-word models are decoded and later concatenated to build the targeted HMM initialization in which the states, each one being modeled by a GMM, that contribute least to the concatenated succession of sub-word models are pruned via Viterbi alignment (Brandl 2009, Sec. 6.4.3, p. 82). The idea of using a decoder with sub-word units to provide a bootstrapping to the system was also employed by Roy 2003.

Other authors use previously trained models by another language as their initialization (e.g. Lööf et al. 2009 and Vu et al. 2011) or automatic transcription algorithms (e.g. Gollan and Ney 2008). Although some researchers stated to bootstrap the system with unsupervised techniques, they also used some previous trained models (e.g. Kemp and Waibel 1999, Wessel and Ney 2005 and Lamel et al. 2002b) to decode data without transcriptions. In this way, additional training data are obtained for initializing other models or themselves (see also Lamel et al. 2001; Lamel et al. 2002a). So, some kind of supervision or previous knowledge has been employed to initiate the system until now.

5.1.1 K-means algorithm for clustering

This is one of the most popular and well-known unsupervised[5] clustering algorithms, also known as the generalized Lloyd algorithm (Lloyd 1982: qtd. in Huang et al. 2001, Sec. 4.4.1.2, p. 169). It is the principal component of the first step of the initialization approach presented in the next sections.

As its name indicates, the algorithm separates different observation data $\boldsymbol{x}_1, ..., \boldsymbol{x}_L$ so to build K clusters where each of the L observations is a D dimensional variable $\boldsymbol{x}$ (Bishop 2006, Sec. 9.1, p. 424). For this purpose, a set formed by K vectors $\boldsymbol{\mu}_k$ of dimension D are introduced, in which $\boldsymbol{\mu}_k$ is a prototype or representative related to the k^{th} cluster and is considered a representation of the centers of them (Bishop 2006, Sec. 9.1, p. 424). Hence, the goal is to determine an assignment $r_{l,k}$ of each vector to the clusters and to compute the set of vectors $\boldsymbol{\mu}_k$, also called codebook (Huang et al. 2001, Sec. 4.4.1.2, p. 166; Bishop

[5]See Vector Quantization, Huang et al. 2001, Sec. 4.4.1, p. 164.

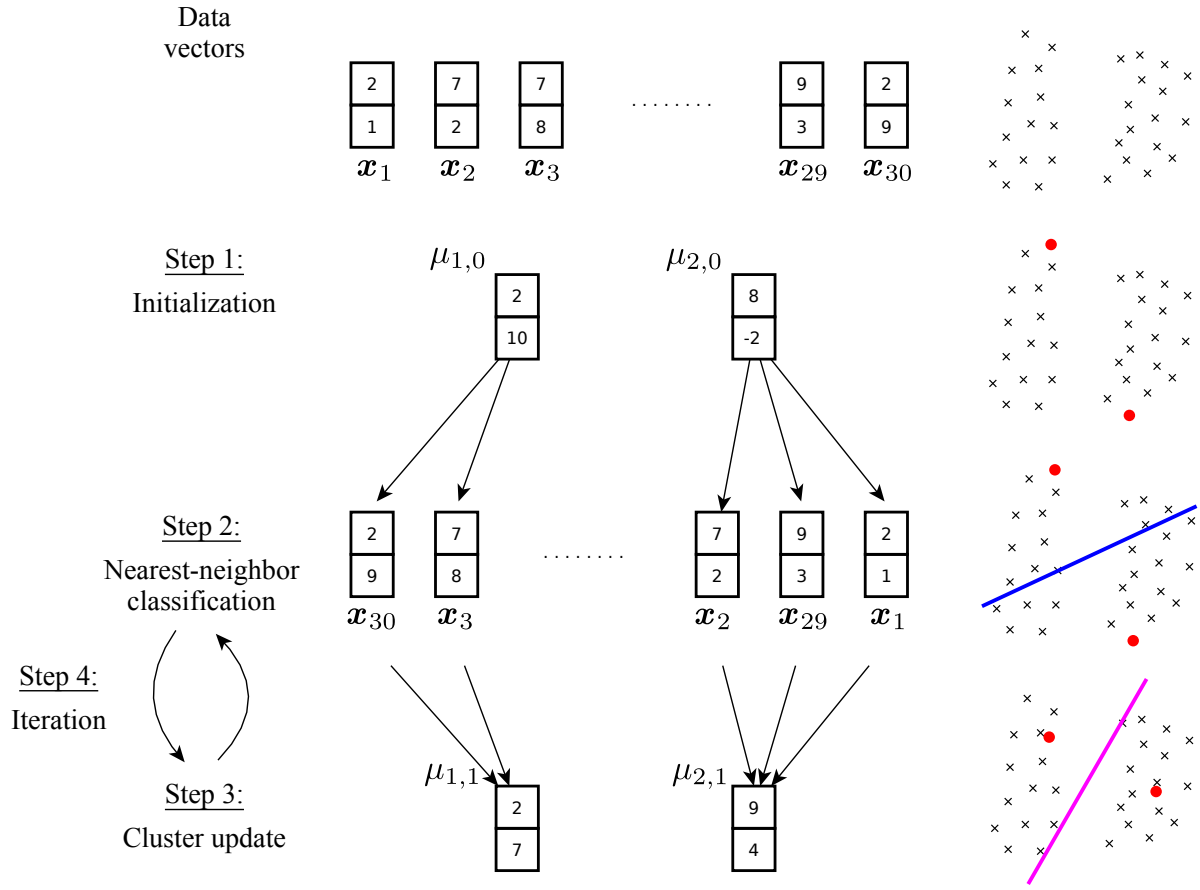


Figure 5.2: *Overview of the K-means algorithm (adapted from Bishop 2006, Sec. 9.1, pp. 424-428). This learning method classifies L data vectors of dimension D, in the example $L = 30$ and $D = 2$, in K clusters, in the example $K = 2$; and calculates the representative $\boldsymbol{\mu}_{k,it}$, where "it" is the iteration number (in the example only one iteration is shown), of each cluster k as the mean of the vectors assigned to them.*

2006, Sec. 9.1, p. 424). An objective function for the minimum distance optimization, sometimes called distortion or dissimilarity measure J, is given by (Huang et al. 2001, Sec. 4.4.1.2, p. 169; Bishop 2006, Sec. 9.1, p. 424; Theodoridis and Koutroumbas 2003, Sec. 14.5.1, p. 531):

$$J = \sum_{l=1}^{L} \sum_{k=1}^{K} r_{l,k} \cdot \|\boldsymbol{x}_l - \boldsymbol{\mu}_k\|^2 \tag{5.1}$$

where $r_{l,k}$ is defined by:

$$r_{l,k} = \begin{cases} 1 & if \quad k = \arg\min_j \|\boldsymbol{x}_l - \boldsymbol{\mu}_j\|^2 \\ 0 & otherwise \end{cases} \tag{5.2}$$

The iterative optimization procedure corresponding to this objective function is the fol-

lowing:

K-means algorithm

Description merged from Huang et al. 2001, Sec. 4.4.1.2, pp.166-169; Theodoridis and Koutroumbas 2003, Sec. 14.5.1, p. 532; Bishop 2006, Sec. 9.1, pp. 424-428.

Step 1: *Initialization*

Choose some initial $\boldsymbol{\mu}_k$ vectors. These can be also randomly selected.

Step 2: *Nearest-neighbor classification*

Assign each data vector ($\boldsymbol{x}_l$) to the closest prototype $\boldsymbol{\mu}_k$ so that $\|\boldsymbol{x}_l - \boldsymbol{\mu}_k\|^2 \leq \|\boldsymbol{x}_l - \boldsymbol{\mu}_j\|^2$ for all $j \neq k$. By doing this, J is minimized and $\boldsymbol{\mu}_k$ fixed.

Step 3: *Cluster updating*

Recalculate the prototypes $\boldsymbol{\mu}_k$ according to:

$$\boldsymbol{\mu}_k = \frac{\sum_l r_{l,k} \cdot \boldsymbol{x}_l}{\sum_l r_{l,k}} \tag{5.3}$$

$\boldsymbol{\mu}_k$ is then the mean vector of all the observations $\boldsymbol{x}_l$ related to cluster k. Consequently, this method is called the K-means algorithm. Compared to the last step, J is minimized and $r_{l,k}$ fixed.

Step 4: *Iteration*

Go to step 2 until the new distortion J exceeds a predetermined value relative to the distortion at the previous iteration, the assignments do not vary in the next iteration or the maximum number of iterations is achieved.

Although the value of the objective function J is reduced in each iteration, the algorithm is not forced to converge to a global minimum of J according to Bishop 2006 (Sec. 9.1, p. 425). Bishop also stated that the two stages 2 and 3 of updating $r_{l,k}$ and $\boldsymbol{\mu}_k$ can be associated with the E (expectation) and M (maximization) steps of the EM algorithm respectively.

A major benefit of applying the K-means algorithm is its simple nature (Theodoridis and Koutroumbas 2003, Sec. 14.5.1, p. 532). The computational cost of the algorithm is of the order $O(KL)$, (Bishop 2006, Sec. 9.1, p. 428). However, the principal disadvantage of this method is that the number of clusters must be previously determined, what has motivated the presentation of different approaches in the context of incremental vector based algorithms to cope with this challenge (see Fritzke 1998; Jokusch and Ritter 1999), although it is still an unsolved problem.

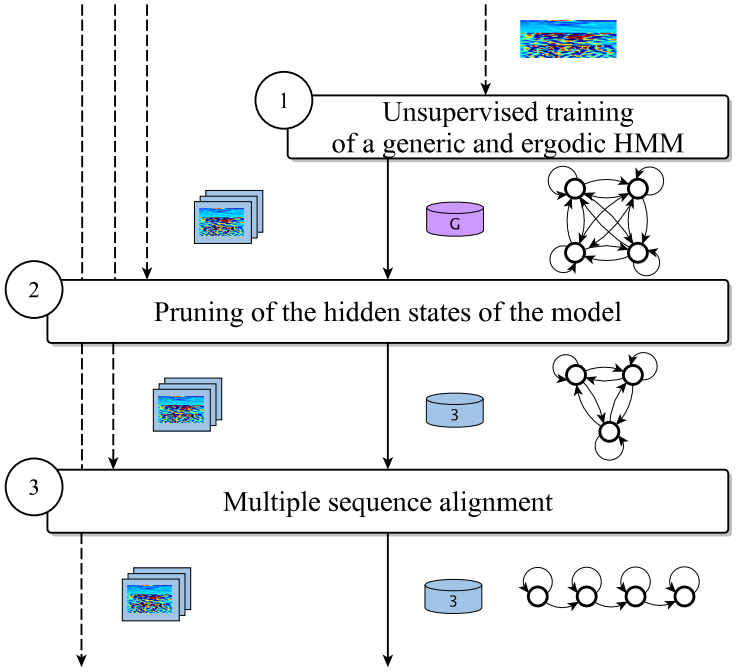


Figure 5.3: *Overview of the proposed model bootstrapping technique to initialize the word-model (see Ayllón Clemente et al. 2010b, Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2012). The bootstrapping method consists of three main stages. First, an ergodic HMM representing a generic speech model is computed using only unsegmented training data and K-means clustering similar to the proposals of Brandl 2008 and Iwahashi 2007, where the states of the HMM allow all the transitions among the states representing general speech segments. Afterwards, this ergodic model is pruned into a specific HMM for the word-model to be learned removing the unnecessary states analogous to Brandl 2009 (Sec. 6.4.3, p. 82). Finally, the ergodic specific HMM is converted into a left-to-right model using the proposed multiple sequence alignment (first published in Ayllón Clemente et al. 2010b as well as Ayllón Clemente and Heckmann 2009). The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a). This plot is an extract of the Fig. 3.1, more precisely the steps between model definition and estimation of the parameters.*

5.2 Multiple sequence model bootstrapping[6]

The proposed model initialization method is shown in Fig. 5.3. This algorithm comprises three main steps. First, an ergodic and full generic HMM is trained in an unsupervised way similar to Brandl et al. 2008 and Iwahashi 2007, where a shared HMM initialization model is built introducing only unlabeled training data. Next, the training of the previously obtained HMM employing ML estimation (Baum-Welch algorithm) and la-

[6]This section and its corresponding subsections take numerous paragraphs (also state-of-the-art reviews), some of them verbatim from the previous works enumerated in Appendix D. The approach proposed was first published in Ayllón Clemente et al. 2010b and Ayllón Clemente and Heckmann 2009.

beled training data is performed. Then, the pruning of the least occupied hidden states provides an ergodic word-level HMM analogous to Brandl 2009 (Sec. 6.4.3, p. 82). The ergodic HMM obtained is transformed into a Bakis, i.e. left-to-right, configured HMM by means of a novel multiple sequence alignment (MSA) technique proposed (Ayllón Clemente et al. 2010b). These steps constitute the basis for the construction of a new word-model.

5.2.1 Unsupervised training of an ergodic and generic HMM

This first step summarized in this subsection is as mentioned before similar to the one employed by Brandl et al. 2008 and Iwahashi 2007, where some minutes of arbitrary unlabeled input speech are recorded. The employed input speech in our framework do not contain any of the words to be learned afterwards and the speech segments are also pronounced by different (male and female) speakers. Next, the acoustic features of the segments are extracted and clustered by the K-means algorithm. The feature assignment obtained allows the estimation of the emission probabilities of an HMM composed of K states. The training is performed in a completely unsupervised manner and it is only executed once. The resulting HMM is stored to be used as pre-initialization for each new word-model to be created.

The transition probabilities between the states are uniformly distributed, so that the word-model is called ergodic[7]. Additionally, it is also named generic because it does not represent a specific word.

5.2.2 Pruning of the hidden states of the word-model

Next, the ergodic and still generic HMM is pruned to eliminate the hidden states that contribute least to the model to train (alike Brandl 2009, Sec. 6.4.3, p. 82) and also to build a more compact word-model. Before pruning is applied, the HMM can be retrained by means of the Baum-Welch algorithm using the labeled training data associated with the word to learn.

Similar to Brandl 2009 (Sec. 6.4.3, p. 82), a Viterbi decoding of the labeled training segments is performed and then, the least occupied states are pruned. We evaluate the occupation of the states using two criteria[8]:

- The maximum value of the emission probabilities considering the scores obtained in each state for each decoded frame of the speech sequences. The emission probability b of a state S_i for a frame $\boldsymbol{x}_t$ of a speech segment $\boldsymbol{X}_r$ is defined in Sec. 3.3.1 for our framework. After the computation of all emission probabilities b of the speech

[7] All possible transitions between the states are allowed (see Fig. 5.4, more details of model topologies and literature references in Sec. 4.1.1).

[8] Included in Ayllón Clemente et al. 2012 as mentioned at the beginning of the section.

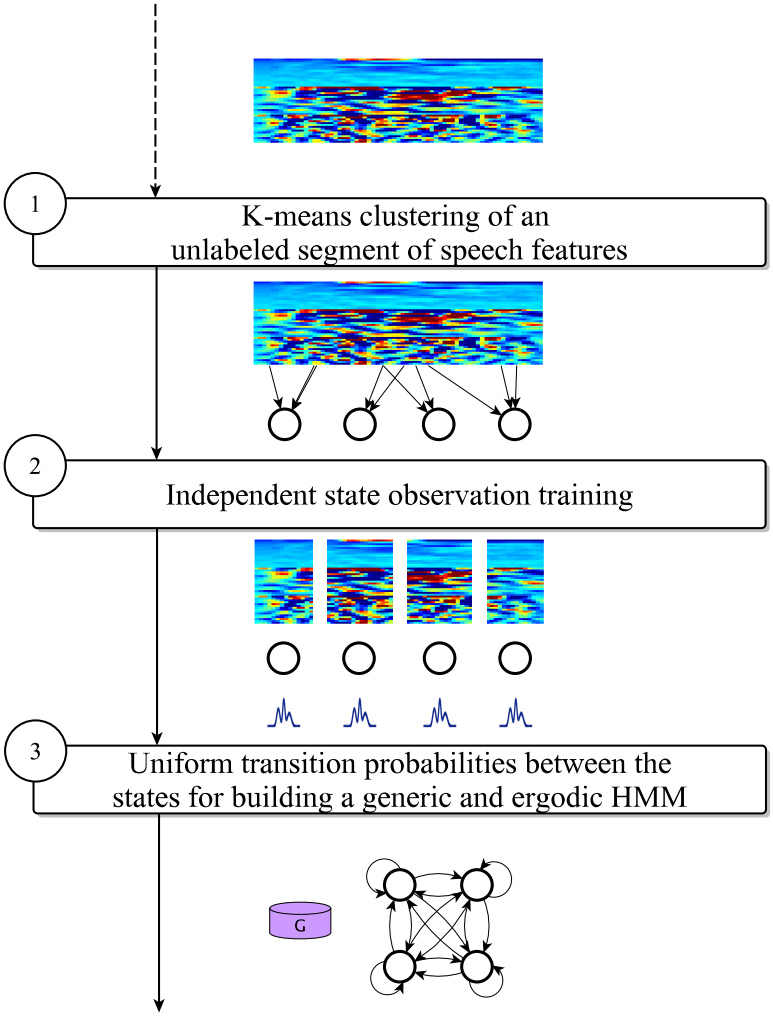


Figure 5.4: *Unsupervised training of a generic and ergodic HMM (analogous to the proposed method of Iwahashi 2007 and Brandl 2008). A short sequence of speech, containing different terms as the ones to learn and uttered by several male and female speakers, is segmented with the help of the K-means algorithm. The clusters obtained serve to estimate the emission probabilities of the states that are then joined in a generic and ergodic HMM. This plot is an extract of the Fig. 5.3.*

frames for each state S_i, we obtain a vector of emission probabilities b for that state S_i. Next, we store the maximum of this vector in $b_{S_i}^{max}$ and remove the state S_i, if $b_{S_i}^{max}$ is less than a predefined threshold.

$$b_{S_i}^{max} = \max\left[b_{\boldsymbol{x}_1,\boldsymbol{X}_1}, b_{\boldsymbol{x}_2,\boldsymbol{X}_1}, ..., b_{\boldsymbol{x}_t,\boldsymbol{X}_r}, ..., b_{\boldsymbol{x}_T,\boldsymbol{X}_R}\right] \tag{5.4}$$

- The frequency of appearance of each state in the decoded sequences. Once the speech segments have been decoded, we can calculate the frequency at which a determined state S_i appears. If the presence in the decoded sentences of a state S_i

is less than a previously defined minimum frequency, we remove the state.

After pruning, the observation estimates of the remaining states are not changed and the transition matrix is constructed by eliminating the rows and columns of the matrix belonging to the pruned states. A further run of the Baum-Welch algorithm refines the parameters. Afterwards, the HMM is no longer generic but model specific.

5.2.3 Multiple sequence alignment (MSA)

As mentioned in Sec. 4.1.4, we use a Bakis topology, i.e. left-to-right, for our models. Thus, the configuration of the states has to be changed from an ergodic topology to a Bakis one. Hidden Markov models are generative models, so we can perform Viterbi decoding of the training segments to obtain the most likely (temporal) succession of states generating the data[9]. The result of the decoding procedure is R optimal path sequences $\mathcal{S}^*$, one for each of the R training samples associated with the word to learn, containing each succession of states information about a potentially underlying configuration of these hidden states in a left-to-right topology for its corresponding training segment[10]. Thus, all decoded sequences have to be merged into one sequence that encodes the information contained in all the Viterbi decoding sequences, which is realized by means of a novel "multiple sequence alignment" algorithm (Ayllón Clemente and Heckmann 2009; Ayllón Clemente et al. 2012).

In Bioinformatics, the term sequence alignment is employed to define a procedure to organize and compare different biological sequences (e.g. protein and gene sequences) with the goal of recognizing or searching similar fractions (patterns) that could indicate some kind of correlation between them (Mount 2004, Ch. 3, p. 70; Sharma 2009, Ch. 2, p. 41). These algorithms aim to find the sequence alignment, which is most suitable based on a scoring function or matrix, and in most cases is delivered with a similarity matrix, also called substitution matrix with scores for all possible replacements of one element by a different one (see Henikoff and Henikoff 1992; Kuznetsov and McDuffie 2015).

Our purpose is to merge all the decoded sequences in a succession of states representing the best alignment between the state sequences through the multiple sequence alignment method proposed, which is depicted in Fig. 5.6 and comprises six building blocks (Ayllón Clemente et al. 2010b; Ayllón Clemente and Heckmann 2009[11]):

- The first block is the computation of a cost matrix $\boldsymbol{D}$ assigning different costs to the permutation of state transitions in the observed succession of states in contrast to the substitution or similarity matrix from the above mentioned Bioinformatics literature, where the cost matrix represents the probability of a character to be

[9] Argumentation also used by the standard initialization described in Sec. 5.1, Young et al. 2009, Sec. 8.2, p. 132.
[10] See description of the Viterbi decoding algorithm in Sec. 3.3.4 according to the known literature.
[11] The reader should visit this publication for examples.

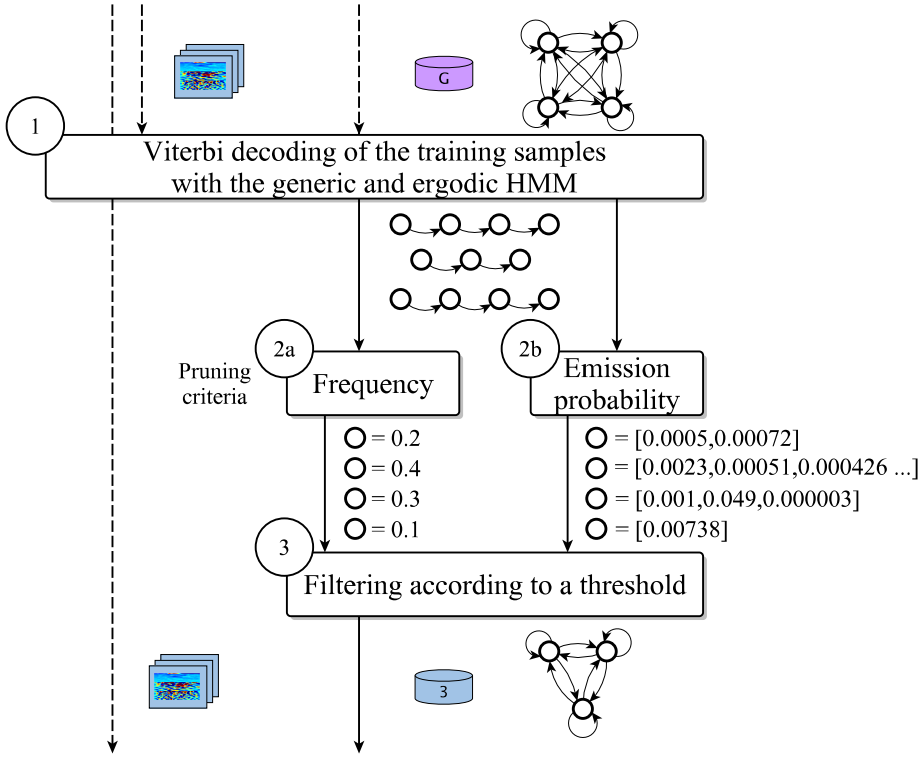


Figure 5.5: *Pruning of the hidden states (analogous to Brandl 2009, Sec. 6.4.3, p. 82). In this phase, the generic ergodic HMM is specialized in the word-model to learn at the current learning iteration. For that purpose, the least occupied states in the decoded training sequences are pruned using two criteria: frequency and emission probability. This plot is an extract of the Fig. 5.3.*

replaced or respectively aligned by another one. Each element of our cost matrix $\boldsymbol{D}(i,j)$ corresponds to the probability[12] of a hidden state S_j to be followed by another, different, state S_i and is computed based on the frequency of the sequence $S_j \rightarrow S_i$ in the succession of states found in the Viterbi decoding (see Eq. 5.5), where gap penalties are ignored.

$$\boldsymbol{D}(i,j) = K_\delta \frac{\sum_{s_1^*}^{s_R^*} \delta_{j \rightarrow i}}{\sum_{j} \sum_{s_1^*}^{s_R^*} \delta j \rightarrow i}; \quad K_\delta \leq 1 \tag{5.5}$$

- The second main building block is similar to the Bioinformatics approaches and it computes the pairwise similarity distances between all sequences, which are captured by the comparison matrix $\boldsymbol{C}$. In our framework, these distances are calculated by means of a special weighted edit distance using dynamic programming. In par-

[12] The values in $\boldsymbol{D}$ can be considered probabilities if $K_\delta = 1$.

ticular, we construct a 2-D grid $\boldsymbol{S}$, which can be considered as a varied fusion of the H-Matrix of the algorithm proposed in Smith and Waterman 1981 and the matrix (array) proposed in Needleman and Wunsch 1970:

Computation of the elements of the grid $\boldsymbol{S}$

Description based on Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

Step 1: *Pre-processing of the sequences*

Let $\boldsymbol{v}$ and $\boldsymbol{t}$ be the corresponding decoded successions of states for two training data. Modification of the original state sequences $\boldsymbol{v}$ and $\boldsymbol{t}$ by the addition of tokens. Two tokens are introduced marking the start and end of the sequences. These tokens are incorporated in order to strengthen local alignment especially at the start of the sequences. Previous to the addition of the tokens, the sequences must not contain repetitions of adjacent numbers.

Step 2: *Initialization*

Construction of a grid with size $(w+1)x(y+1)$, where w and y are the respective lengths of the modified sequences $\boldsymbol{v}'$ and $\boldsymbol{t}'$. The states of both sequences are represented on each axis (see the right picture under the 2^{th} block in Fig. 5.6). To simplify the description of the algorithm, we position the sequence $\boldsymbol{v}'$ in the abscissa i-axis and $\boldsymbol{t}'$ in the ordinate j-axis as example. In order to measure the other way around, these positions have to be exchanged.

As in Smith and Waterman 1981, the nodes of the first column and row of the grid $\boldsymbol{S}$ are set to 0. In the second row and column of the grid, the alignment of the sequences $\boldsymbol{v}'$ and $\boldsymbol{t}'$ starts. This is the reason why the indexing of each sequence is related to $i-1$ and $j-1$ respectively.

Step 3: *Computation of the elements*

The value of each node (i,j) in the grid is computed by the similarity measure function $\boldsymbol{S}(i,j)$, which determines the correspondence between the individual states in the sequences $\boldsymbol{v}$ and $\boldsymbol{t}$:

$$\begin{gathered} \boldsymbol{S}(1,j) = 0 \;\; ; \;\; \boldsymbol{S}(i,1) = 0 \\ \boldsymbol{S}(i,j) = \max\left[\boldsymbol{S}(i,j-1), \boldsymbol{S}(i-1,j-1)\right] + m(i,j) \end{gathered} \tag{5.6}$$

The cost m is 1 if $\boldsymbol{v}'(i-1)$ and $\boldsymbol{t}'(j-1)$ are aligned, more precisely, $m(i,j)$ receives the value 1 if the element to analyze of both sequences is the same. Otherwise,

the cost is the value coded in the cost matrix $\boldsymbol{D}$. For the computation of $\boldsymbol{D}$, the modified sequences are to be used, i.e. the start and end tokens are also considered in the computation of $\boldsymbol{D}$[13]. The value is read from the cost matrix by taking as column index the element $i-1$ of the sequence $\boldsymbol{v}'$ and as row index the element $j-1$ of the sequence $\boldsymbol{t}'$.

$$m(i,j) = \begin{cases} 1 & if \quad a = b \\ \boldsymbol{D}(b,a) & if \quad a \neq b \end{cases}$$
$$with \tag{5.7}$$
$$a = \boldsymbol{v}'(i-1)$$
$$b = \boldsymbol{t}'(j-1)$$

Step 4: *Termination*

When all elements of $\boldsymbol{S}$ have been calculated, the similarity distance measure[14] between the two sequences is defined as the value of $\boldsymbol{S}(w+1, y+1)$.

The comparison matrix $\boldsymbol{C}$ is filled with the value of the similarity measure computed above in step 4 for each pair of sequences. For the comparison of the similarity distance measures, the distance of the sequence $\boldsymbol{v}$ over the sequence $\boldsymbol{t}$ as well as the distance of $\boldsymbol{t}$ over $\boldsymbol{v}$ are computed.

- In the third block, the proposed method enables two ways of merging. One is starting the merge with the least similar sequences (smallest $\boldsymbol{S}$) and the other one with the most similar ones. Both methods are valid. In our approach, we start merging with the least similar sequences.

- In the 4^{th} block of Fig. 5.6, the sequences are pairwise merged based on their similarity measure. The maximum similarity path decodes the most suitable alignment between the two sequences to merge by effectively warping the states of the second sequence to the states of the first one following the similarity of both sequences, which is also known as the variational similarity concept, i.e. realizing some changes we can convert one sequence in the other (Theodoridis and Koutroumbas 2003, Sec. 8.2.2, p. 326).

[13]The step 1 is realized for all the sequences jointly with the computation of $\boldsymbol{D}$ once and then, it is not repeated in further iterations.
[14]This should not be understood in a strictly mathematical sense.

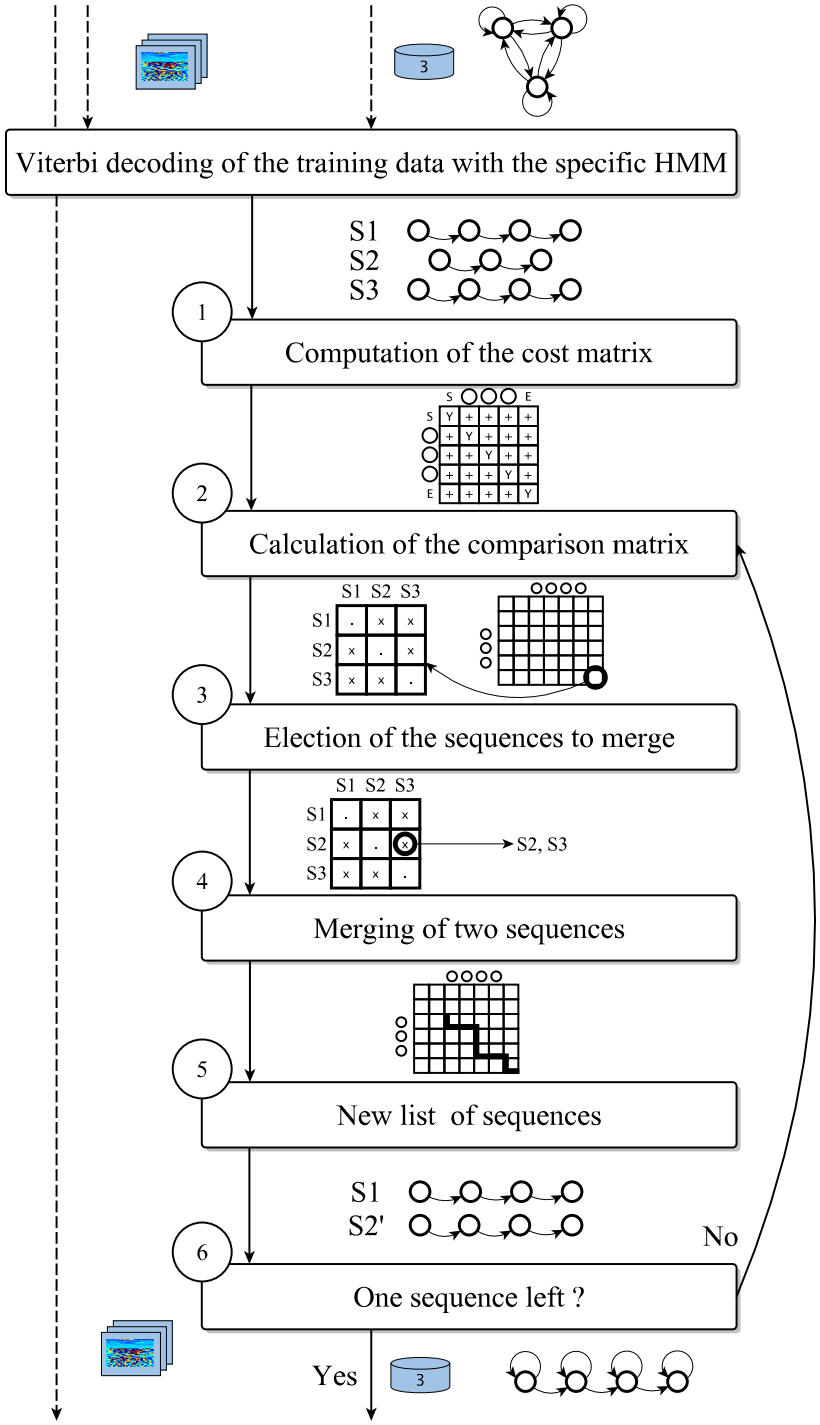


Figure 5.6: *Multiple sequence merging procedure inspired by the progressive methods of MSA in Bioinformatics (see Sharma 2009, Sec. 4.7, p. 121), where the R decoded succession of states are merged into one well suited sequence (first published in Ayllón Clemente et al. 2010b as well as Ayllón Clemente and Heckmann 2009). The values of the elements of the cost matrix* $\boldsymbol{D}$ *are the probabilities of a state to follow a different state. These probabilities are used to build a grid* $\boldsymbol{S}$*, which provides the similarity values between the different sequences to evaluate in the comparison matrix* $\boldsymbol{C}$*. After the selection of the sequences to align and the merging of these sequences, the process is iteratively repeated until only one sequence is left. This plot is an extract of the Fig. 5.3.*

Decoding of the grid

Description based on Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

Step 1: *Initialization*

Starting point: Row $i = 3$ and column $j = 3$ of the grid $\boldsymbol{S}$.

Calculation of the index of the maximum value of the column j. If the maximum value is less than 2, jump to the next column incrementing i in one unit and calculate the maximum there. This indicates that the initial element of the sequences (not considering the start token) is not the best choice to start the new merged sequence. Otherwise, take the index of the row obtained as the value of *ind*.

Step 2: *Sequence decoding*

a) If the (absolute) difference between *ind* and i is bigger than one unit, go to step 3 without copying any element[15].

b) If i is bigger than *ind*, copy the element $\boldsymbol{t}'(j-1)$.

c) If i is equal to *ind*:

- If i is equal to $w+1$, copy the element $\boldsymbol{t}'(j-1)$.
- If i is smaller than $w+1$, copy the element $\boldsymbol{v}'(i-1)$. If $\boldsymbol{v}'(i-1)$ and $\boldsymbol{t}'(j-1)$ are not aligned, also copy the element $\boldsymbol{t}'(j-1)$. $i = i+1$.

d) If i is smaller than *ind*, copy the element $\boldsymbol{v}'(i-1)$ and increment i in one unit. Repeat the above process until i and *ind* are equal, afterwards copy $\boldsymbol{v}'(i-1)$ and $\boldsymbol{t}'(j-1)$. $i = i+1$, if i is smaller than $w+1$.

Step 3: *Iteration and termination*

$j = j+1$, until $j = y+1$. In case j is not bigger than $y+1$, calculate the index of the maximum value of the column j, take the index *ind* of the row obtained and go to step 2.

Step 4: *Post-processing*

[15]It is optional. If it is not used, the resulting sequence is generally longer, in the case that the sequence in the i-axis is shorter than the sequence in the j-axis. In the case that the shortest sequence is placed in the j-axis, it is necessary to disable this option. If this option is still operating, the end of the sequence in the i-axis will be often disregarded.

Clean all repetitions of adjacent numbers in the resulting sequence and place the tokens properly again if there are still sequences to merge.

The merging algorithm proposed above is similar to the walk back proposed by Smith-Waterman. However, our walk is realized forwards while picking the elements of the new sequence starting at the upper left corner of the matrix and descending, it ends at the maximum value on the lower right corner. This method offers the possibility to correct errors in the sequences, so that if an element (a state) of one sequence is placed wrongly, the algorithm uses the cost matrix[16], and the comparison of the whole sequence with another sequence, to prune it. In the same way, new correct elements would be integrated.

- In the 5^{th} block, once two sequences are merged, a new list of sequences is established including the new sequence and deleting the two old ones.

- In the 6^{th} block, all the sequences are merged pairwise until only one is left. After merging all sequences, we obtain a well suited succession of hidden states for the word-model.

The final step of the bootstrapping method is to construct the new hidden Markov model with Bakis topology using the succession of states computed in the last step. The Gaussian mixture parameters are conserved from the previous steps of the bootstrapping; however, here we propose that the transition matrix $\boldsymbol{A}$ of this new HMM with Bakis topology is initialized with the values calculated by the transpose of the cost matrix $\boldsymbol{D}$.

The model bootstrapping presented in this section sets a suitable succession of states and establishes a dynamic number of hidden states combining the information contained in the speech and the signals instead of considering only their duration. Our method also allows us to eliminate states that are very similar or only repetitions of themselves and then contributes to reduce the number of states. The merging of the sequences is a dynamic procedure that is not fixed to a determined number of elements, so that if the system requires a predefined number of states, we remove the last states to shorten the sequence or add similar states at the end if the sequence is too short[17]. An analysis and literature references of the influence of the number of hidden states of the word-models is performed in Sec. 4.1.2.

[16] As mentioned before, it represents the probability that an element is followed by another one in contrast to the Bioinformatics literature.

[17] Other possibility is to include repetitions of states along the sequence. However, the experiments realized in this direction were not successful.

5.3 Evaluation of our algorithms[18]

Approaches to compare

Once the parameters of the system are fixed (Chapter 4), our bootstrapping method (see last section) initializes the estimates and we compare it with some of the most well-known state-of-the-art methods (see Sec. 5.1) such as:

- "Flat start" - FS method (Itaya et al. 2005; Young et al. 2009, Sec. 8.3, pp.135-136): in this technique, the means and variances of the GMMs are initialized to the global mean and variance of the features, i.e. all means and variances are the same for all the states of the word-model.

- The combination of uniform segmentation and Viterbi decoding - US method (Rabiner 1989; Young et al. 2009, Sec. 8.2, pp. 132-133): in this method, it is assumed that each training sample has been generated by the HMM to estimate. Hence, by knowing the state that produces each frame of the training data, the emission probabilities of that state can be computed. For the association of each feature vector with a state, Viterbi decoding is employed. Similarly, the further assignment of a feature vector to a component of a GMM inside a state can be realized by K-means clustering and taking the component with the highest probability. In the first iteration of the algorithm, a uniform segmentation can take place in order to initialize the parameters or these can be randomly initialized.

We make use of the Netlab and Voicebox Matlab© toolboxes to implement our algorithms. Additionally, all methods integrate our proposed variance floor estimation (variance floor percentile together with the scaling factor dependent on the number of training samples) of the last chapter and we do not use the number of states provided by our algorithm, but the number of states based on the average of the frames of the samples. In this way, all algorithms are set with the same conditions. The baseline system is the same as in Sec. 4.2.4.

Afterwards, we compare our multiple sequence alignment (MSA) based bootstrapping with simpler initialization methods and other types of alignments in order to distinguish if the alignment proposed is essential or other alignments can provide the same performance:

- The approach, which we refer to as best Viterbi alignment (BVA) (Brandl 2009, Sec. 6.3.1-6.4.3, pp.75-82): it is based on a very similar initialization (unsupervised learning phase and pruning of the states) as our approach including also a Viterbi decoding for all training data. The difference with our method is that in BVA instead of aligning and merging all sequences of states, the sequence of concatenated

[18] Several paragraphs (some verbatim) are taken from previous works as Ayllón Clemente et al. 2010b, Ayllón Clemente and Heckmann 2009, Ayllón Clemente et al. 2012, however some experiments have been extended in contrast to the previously published ones.

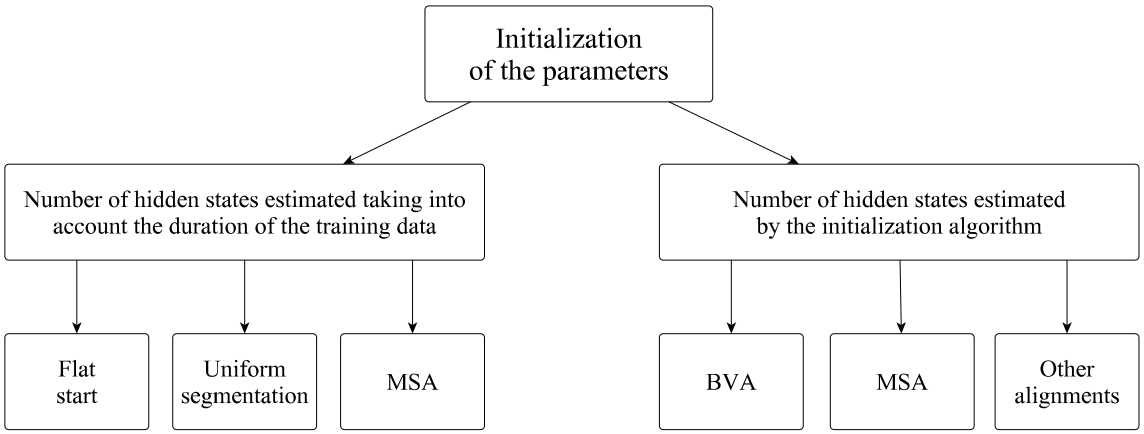


Figure 5.7: *Overview of the experiments realized for the evaluation of the different initialization methods (Ayllón Clemente et al. 2012). MSA stands for multiple sequence alignment and BVA for best Viterbi alignment.*

sub-word models with the highest probability for all training data is chosen for the construction of the new model.

- Two variations of the MSA:
  - $\text{MSA}_{\cup}$: we combine all the different states that appear in the decoded Viterbi sequences.
  - $\text{MSA}_{\cap}$: only the common hidden states that appear in all the sequences are combined.

Related to the step where the states are pruned after the unsupervised training of the model[19], we analyze all the above algorithms applying to the Viterbi decoded sequences both pruning criteria mentioned in Sec. 5.2.2: the frequency each state appears in these sequences and the maximum emission probability of the states during the decoding process.

We evaluate the initialization methods together with the estimation or learning phase. As mentioned at the beginning of the chapter, the goal of applying model bootstrapping is to provide the estimation phase with a good set of initialized parameters (Huang et al. 2001, Sec. 8.4.1, p. 396); hence we measure the bootstrapping approaches jointly with the learning stage.

Data sets

The TIDigits database (Leonard and Doddington 1993, see Appendix A.2) containing single numbers ("1" to "9" and "zero", "oh") is here used as it was applied for the

[19] Analogous step is also realized in Brandl 2009 (Sec. 6.4.3, p. 82).

evaluation of the variance floor in the preceding chapter. Here, 250 different subsets of training data are employed each of them containing up to 10 samples, where each subset is evaluated by a test set divided in isolated words uttered by men and women as well as a continuous speech test set[20], where only adult male speakers were included. As mentioned before, the interest of presenting samples coming from a different speaker gender is to analyze if the improvements cannot only appear by different speakers but also by another gender[21] (female and male voices are quite different).

In particular, the TIMIT database (Garofolo et al. 1993, Appendix A.1) was chosen for the unsupervised learning step in order to initialize the models with different data as the training samples of the words to learn. TIMIT does not contain explicitly digits. In this case, adult male and female speakers mixed in a continuous speech segment[22] are used for the training of the generic speech model.

Parameters employed

In the first phase of the initialization, we employ a number of states for the generic and ergodic speech model similar to the number of phonemes in American English language as we use American English databases. In the pruning step, we apply a threshold of 0.01 to 0.1 in the case of the frequency criterion and 1e-10 to 1e-5 for the emission probability criterion. Additionally, a post-processing in the pruning phase is applied in order to ensure a minimum number of 3-4 states per word. Finally, the multiple sequence alignment stage is executed. The resulting number of states is not allowed to be over 25 and not lower than 4-5 states. This threshold can be determined using the duration of the training samples as fall-back method.

Results

The average of the recognition scores of all the initialization approaches depicted in Fig. 5.7 are shown in Table 5.1(a) and 5.1(b) for isolated word recognition in men and women as well as for continuous speech recognition using male speakers[23]. In Table 5.1(b), the algorithms dynamically set the number of states in contrast to the other approaches, where the number of states is fixed by the duration of the training data (Table 5.1(a)). Moreover, we display in Fig. 5.8(a) and 5.8(b) the absolute recognition scores of the proposed method for the initialization phase (including the percentile scaled

[20] The continuous speech utterances analysis is extended here to all the methods to compare and not the best approach as in Ayllón Clemente et al. 2012.

[21] In Ayllón Clemente et al. 2010a, also a test set containing female speakers was used, however the parameterization and final configuration of the system was not the same as here.

[22] In the database, each sentence is uttered by one speaker. Hence, we concatenated several continuous speech sentences pronounced by different speakers (also different genders) in order to have a long utterance of few minutes.

[23] The results presented here are an extension (female speakers and more continuous speech experiments) of the ones presented in Ayllón Clemente et al. 2012.

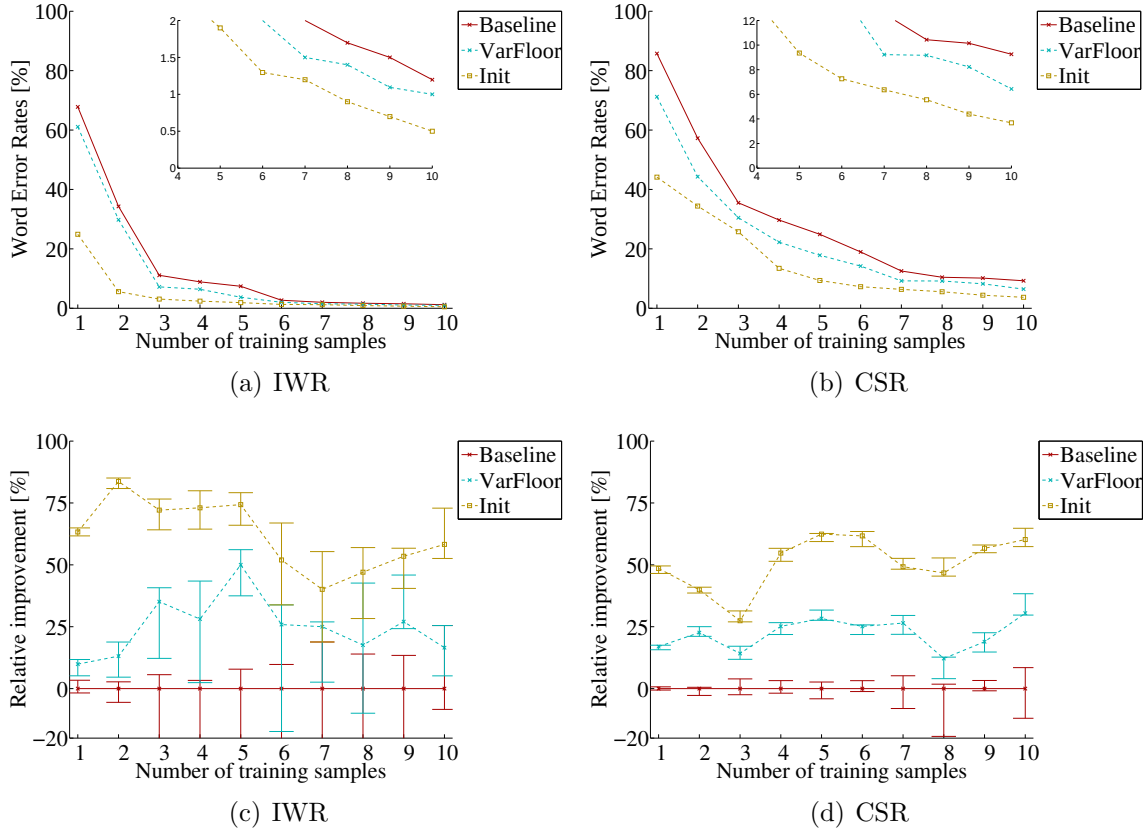


Figure 5.8: *Word error rates (a,b) and relative improvement (c,d), in test sets with male speakers, using the percentile variance floor scaled by the number of samples (VarFloor) and adding the incorporation of the MSA method where the algorithm sets the number of states (Init) (see Ayllón Clemente et al. 2012). The method used as baseline (Baseline) is explained in Sec. 4.2.4. In Fig. (a) and (b), a zoom view is presented in order to distinguish the values of the different methods when the resolution of the scale axis is too rough. The bars in Fig. (c) and (d) indicate the minimum and maximum values of the improvements of the 25 cross-validations. They target to measure a possible overlap of the scores relative to each stage of our system.*

variance floor dependent on the amount of training data) of the incremental word learning system. In addition to the recognition scores, the relative improvements are shown in Fig. 5.8(c), 5.8(d) and Table 5.2[24]. In spite of the relevance of the improvements, we also depict the minimum and maximum values of the cross-validations in Fig. 5.8(c) and 5.8(d) in order to measure a possible overlap (bars) of the improvements for each stage of our system.

In our first analysis, the number of hidden states of each word-model depends on the duration in frames of the training samples. We can observe that MSA outperforms state-

[24]The values in Table 5.1 are rounded. The values calculated in Table 5.2, although also rounded, were calculated with the values of Table 5.1 without rounding.

Table 5.1: *Word error rates (WER %) of the state-of-the-art methods and our proposed algorithms. For each method, the WER values represent the mean of a 25-fold cross-validation on the training set evaluated on separated male (M) and female (W) speakers test sets with isolated words and another test set (C) with continuous speech utterances produced by male speakers (extended from Ayllón Clemente et al. 2012). "No. **X**" stands for the number of training samples (R). FS stands for "flat start", US for uniform segmentation, MSA for multiple sequence alignment, BVA for best Viterbi alignment, $MSA_{\cup}$ for the initialization method using all the different states present in the decoded successions of states and $MSA_{\cap}$ employing only the common states. The best WERs are marked in bold.*

(a) Initialization with No. of states fixed by the data

No.	FS			US			MSA		
X	M	W	C	M	W	C	M	W	C
1	61.1	75.4	71.2	58.4	57.9	74.9	**23.5**	**51.2**	**45.4**
2	29.8	52.3	44.3	28.4	46.5	37.6	**6.3**	**44.2**	33.5
3	7.2	36.3	30.5	6.9	34.2	30.1	**4.7**	**32.1**	**25.4**
4	6.4	**27.5**	22.2	6.7	33.1	19.6	**3.8**	28.1	**14.4**
5	3.7	**23.1**	17.8	**2.8**	24.6	12.6	2.9	25.7	**9.2**
6	**2.0**	22.4	14.2	2.2	**21.2**	8.9	2.2	22.6	**8.3**
7	**1.5**	22.6	9.2	1.9	25.6	8.7	**1.5**	**19.3**	**6.6**
8	1.4	21.7	9.2	**1.1**	19.3	6.7	1.3	**16.2**	**6.3**
9	1.1	21.9	8.2	**0.9**	**14.8**	**5.1**	1.2	15.7	6.0
10	1.0	20.4	6.4	**0.8**	**14.1**	**4.3**	**0.8**	14.4	**4.3**

(b) Initialization with No. of states fixed by the algorithm

No.	BVA			MSA			$MSA_{\cup}$			$MSA_{\cap}$		
X	M	W	C	M	W	C	M	W	C	M	W	C
1	**24.9**	**51.6**	**44.1**	**24.9**	**51.6**	**44.1**	**24.9**	**51.6**	**44.1**	**24.9**	**51.6**	**44.1**
2	10.9	42.1	**34.3**	**5.6**	43.4	34.4	6.9	**40.3**	35.3	6.4	45.9	36.6
3	7.0	32.0	**24.2**	**3.1**	**31.1**	25.8	4.2	33.2	27.2	3.9	41.8	28.7
4	4.9	29.1	17.3	**2.4**	**28.7**	**13.4**	2.9	29.6	14.4	2.7	39.6	14.1
5	3.0	27.6	12.9	**1.9**	**24.3**	**9.4**	2.3	25.3	10.6	2.4	34.5	11.3
6	2.1	22.6	8.8	**1.3**	**20.5**	**7.3**	1.9	22.1	9.8	1.9	32.7	8.9
7	1.5	21.9	6.5	**1.2**	**18.3**	**6.4**	1.7	20.6	8.2	1.8	29.8	8.0
8	**0.9**	18.7	5.9	**0.9**	**15.4**	**5.6**	1.5	19.8	7.0	1.6	25.4	6.1
9	**0.7**	13.9	5.0	**0.7**	**12.6**	**4.4**	1.1	17.6	5.7	1.3	19.6	5.9
10	0.7	13.6	4.3	**0.5**	**12.1**	**3.7**	0.9	13.4	4.8	1.0	18.7	4.6

Table 5.2: *Relative improvement of the word error rates (WER %) of all methods explained in this chapter compared to the baseline system (Sec. 4.2.4) (extended from Ayllón Clemente et al. 2012 (female speakers scores)). For abbreviations, see Table 5.1.*

		Number of hidden states estimated via the duration of the samples			Number of hidden states estimated via the initialization algorithm			
		FS	US	MSA	BVA	MSA	$MSA_{\cup}$	$MSA_{\cap}$
ISOLATED	M	24.8	28.8	44.0	46.3	61.7	45.0	42.3
	W	37.0	42.8	46.8	45.8	49.8	45.4	30.2
CONTINUOUS	M	22.0	36.9	46.9	45.9	50.8	42.7	43.5

of-the-art methods in most of the cases (Table 5.1(a)), also in quite different unseen data such as the test set containing female speakers. In Table 5.2, it is shown that our method achieves a relative improvement of 40% against the baseline system and an average of 15% improvement compared to the best state-of-the-art method, uniform segmentation (US), for the test containing isolated words uttered by male speakers.

Furthermore, our method provides a different number of states for each word-model independent of the information of the duration of the training data. In these conditions, our model bootstrapping technique MSA is superior to BVA in most situations according to Table 5.1(b). MSA also outperforms $MSA_{\cup}$ and $MSA_{\cap}$ in all cases except for two samples with the test set of female speakers. When only one training segment is used, the recognition results obtained in our approach and in the other alignments are the same. In this case, the multiple sequence alignment algorithm described in Sec. 5.2.3 is not executed, because only one training segment is sampled by the Viterbi decoding. In this situation, our technique and the other alignments have the same behavior. In reference to the step where the states are pruned after the unsupervised training of the models (similar to Brandl 2009, Sec. 6.4.3, p. 82), the criterion using the maximum values of the emission probabilities does not provide better results than the frequency one, so we evaluate all the above algorithms using the pruning criterion according to the frequency, i.e. how often a determined state can be found in the Viterbi decoded sequences.

From Table 5.2, one can see that BVA also obtains a large improvement in comparison with the baseline system, but our system is still superior. $MSA_{\cup}$ and $MSA_{\cap}$ methods almost always show better scores than the baseline system, nevertheless they do not outperform our proposed multiple sequence alignment.

The proposed method, setting the number of hidden states autonomously, is the best one for female speakers, although the results obtained are very similar in almost all ex-

periments as shown in Table 5.2. The average score obtained in $MSA_{\cap}$ on isolated word recognition (IWR) with female speakers is worse than the baseline system, because we only consider the common information contained in the samples. This causes overspecialized models that do not generalize to unseen data as is the case of the female speakers.

5.4 Discussion[25]

In addition to the model definition and overfitting issue discussed in the previous chapter, we also addressed the initialization problem in order to improve the starting conditions of the ML estimation. We presented in this chapter a multiple sequence bootstrapping technique composed of three phases (Ayllón Clemente et al. 2010b; Ayllón Clemente et al. 2012[26]): unsupervised learning of a generic and ergodic HMM similar to the methods proposed in Iwahashi 2007 and Brandl et al. 2008, pruning of the least occupied states in the generic model to build a word-specific HMM analogous to the one realized in Brandl 2009 (Sec. 6.4.3, p. 82) and transformation of the ergodic HMM into a left-to-right word-model applying a novel multiple sequence alignment method.

Related to the first phase inspired in Iwahashi 2007 and Brandl et al. 2008, the unsupervised learning of a generic speech model previous to the proper learning of the word can be seen as unusual. However, these approaches are especially suitable for the natural and user-friendly learning scenario that we aim at. As referred at the beginning of the Chapter 4, the usage of unsupervised algorithms is beneficial as it helps us in cases of limited training data. Furthermore, from the point of view of the user less tutoring time might be desired, what increases the acceptance among the future users.

We showed that our bootstrapping approach is superior to state-of-the-art methods, e.g. "flat start" (Itaya et al. 2005; Young et al. 2009, Sec. 8.3, pp. 135-136) and uniform segmentation (Rabiner 1989; Young et al. 2009, Sec. 8.2, pp. 132-133). When recognizing isolated words uttered by male speakers, our proposed initialization achieves an improvement of 44% relative to the baseline system and approximately 15% over the best state-of-the-art methods mentioned in this chapter (see Table 5.2). This improvement of our system is not uniform as we can observe in Table 5.1(a). In this experiment, our system was not allowed to establish dynamically the number of states, they were externally imposed. So, the sequence of states provided by the MSA algorithm is truncated to a fixed predefined number of hidden states (in this case dependent on the average duration in frames of the samples). This kind of fixing the length of the HMM is not optimal for our bootstrapping and hence, it is only superior when the number of training samples is very small. In a second experiment, the number of hidden states is not fixed by the duration of the training data and the algorithm sets the number of states via the multiple

[25] Several paragraphs (some verbatim) are taken from previous works as Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

[26] Also published for a patent in Ayllón Clemente and Heckmann 2009.

sequence alignment (MSA) method. Here, our approach (see Table 5.1(b)) outperforms all the other alignments presented.

Hence, we were able to demonstrate that although the first phases of the unsupervised initialization is common for the initialization algorithms BVA, $MSA_{\cup}$ and $MSA_{\cap}$, taking a sequence obtained via the proposed multiple sequence alignment is better than taking the sequence with the highest probability to represent the class (BVA - Brandl 2009, (Sec. 6.4.3, p. 82)), merging only the states that are present in all the sequences ($MSA_{\cap}$) or taking all the different states contained in the sequences in order to keep the maximum information ($MSA_{\cup}$).

One of the principal deviations between BVA and our approach is that in Brandl 2009 (Sec. 6.4.3, p. 82) a "concatenated" sequence of HMMs is built to construct a syllable model, while in our approach different sequences of states are built and later "merged" into a new sequence to build a word-model. This concept of merging (condensing and recombining) several state sequences into one sequence to build a word-model is our novel contribution here (multiple sequence alignment for the transformation of an ergodic HMM into a Bakis one - Ayllón Clemente and Heckmann 2009; Ayllón Clemente et al. 2010b). Moreover, our system is able to select the appropriate number of hidden states for each case.

In conclusion, the improvement obtained with our bootstrapping approach (jointly with the use of the variance floor dependent on the number of training samples) is more than 50% in continuous speech and even better for isolated word recognition (IWR) using male speakers. Regarding the consistency of the results, we can observe in Fig. 5.8(d) that our initialization method provides substantial and regular improvements in continuous speech and almost so good results in isolated word recognition (IWR) where the number of overlaps is small. The advantage of our model bootstrapping phase is not only to find a suitable set of initial estimates but also to improve substantially the recognition scores of the whole system. This enables a reduction of the training samples (less tutoring time, thus less effort). Furthermore, our initialization also sets the necessary number of hidden states for each word-model. After this initialization phase is realized and the topology of the new HMM is fixed, ML estimation for the learning of the parameters is employed alternatively with the routine of splitting the mixture components (see the corresponding references in Sec. 4.1.4). Once the final number of mixture components is reached, the parameters are tuned using MAP estimation. Finally, a large margin discriminative training refinement is realized (see Fig. 3.1) to optimize the estimates coming from the EM (ML and MAP) training procedure.

6
Discriminative training

Finally, we arrive to the last building block of our incremental architecture, the discriminative training (DT) stage. Up to this point, we have learned all the words in isolation, i.e. without knowing if other word-models exist in the system and how their parameters are configured. In this stage, we aim to refine the parameters of the word-models by enhancing their generalization performance via the usage of discriminative training, which has been widely investigated in ASR system (see e.g. Juang et al. 1997 and McDermott 1997). According to Huang et al. 2001 (Sec. 4.3, p. 150), ML and MAP techniques target to maximize the probability (likelihood) that the training data $\boldsymbol{X}_i$ correspond to the model λ_{w_i}. Unfortunately, these training criteria cannot ensure that the observation $\boldsymbol{X}_i$ belonging to class w_i achieves a higher likelihood $P(\boldsymbol{X}_i|\lambda_{w_i})$ than $P(\boldsymbol{X}_i|\lambda_{w_j})$, where λ_{w_j} is different to λ_{w_i}, since neither ML nor MAP estimation aim to discriminate between classes (Huang et al. 2001, Sec. 4.3, p. 150).

In the last decade, several authors have investigated the advantage of combining generative and discriminative methods in the machine learning field (e.g. Jebara 2004 and Lasserre et al. 2006: qtd. in Bishop 2006, Sec. 1.5.4, p. 44). Recently, one can observe this combination in the form of new speech recognition techniques, called large margin discriminative training (LM DT) approaches[1], which improve the generalization ability through the optimization of the margin between the classifiers (e.g. Yu et al. 2008, Li et al. 2005, Chang et al. 2008, Sha and Saul 2007, Jiang et al. 2006 and Yu and Deng 2007).

Outline of the chapter

First, we provide the reader with an overview of the most relevant discriminative training techniques. Next, we introduce large margin discriminative training methods explaining the importance of the margin to separate classes. In this context, two relevant decision rules to compute the LM DT are presented and several optimization techniques are discussed. Then, we propose different strategies for incremental discriminative learning

[1] Also employed in our previous works such as Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012.

with limited training samples. After outlining our proposals, we evaluate the strategies in different conditions. The basic principles of the state-of-the-art methods explained, the techniques and strategies proposed, the experiments realized as well as their corresponding results are founded on the ones[2] introduced in Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012.

6.1 Optimization of the AMs: hybrid approaches and DT algorithms[3]

One of the most commonly used discriminative training methods in ASR is the minimum classification error (MCE) training, which aims to minimize the classification errors as its name indicates (e.g. Juang et al. 1997, McDermott 1997, Juang and Katagiri 1992, Macherey et al. 2005 and He et al. 2008). However, when a small number of training samples are used as in our scenario, the training data are already well classified and hence, it is difficult to optimize based on the misclassified training samples[4].

Another important discriminative training method is maximum mutual information (MMI) estimation, which aims to find a new parameter set that maximizes the mutual information between the training exemplars and the words associated with them (He et al. 2008). These methods do not have the simplicity and nearly rapid convergence of the EM algorithm, but if they are suitably developed, they can provide satisfactory recognition scores (Woodland and Povey 2002: qtd. in Sha and Saul 2007).

On the other hand, the predominant behavior of the observation probabilities that we analyzed in Sec. 4.1.3, has encouraged several authors to improve the representation of the HMMs through the investigation of new formulations of these parameters, e.g. by means of some hybrid approaches combining HMMs with other models as ANNs (e.g. Morgan et al. 1993; Hinton et al. 2012[5]) and SVMs (e.g. Stadermann and Rigoll 2004; Ganapathiraju et al. 2000) to replace the standard GMMs as observation probabilities. In the last case, in order to compute the probability of the model, the unprocessed outputs of the SVMs are transformed into a magnitude similar to a probability measure by a sigmoid function (Jiang et al. 2006). In the combination with ANNs, these can be employed to substitute the emission densities directly (e.g. Morgan and Bourlard 1995, Hennebert et al. 1997, Zavaliagkos et al. 1994 and Huang et al. 2001, Sec. 9.8.1, p. 455).

Although these hybrid approaches have achieved a certain success among the speech recognition community, standard ANNs and SVMs, as stated in Sec. 2.3.5, cannot cap-

[2] Some paragraphs are also taken verbatim from these previous works.

[3] The review of the state-of-the-art methods of this section (included subsections) are based on Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2010b and Ayllón Clemente et al. 2012. Some paragraphs are taken verbatim.

[4] Argumentation extracted from the experiments realized in Ayllón Clemente et al. 2010b.

[5] Deep neural networks (DNN) have become an important approach during the redaction of this thesis, however they need the estimation of a considerable set of parameters compared to the large margin approaches explained below.

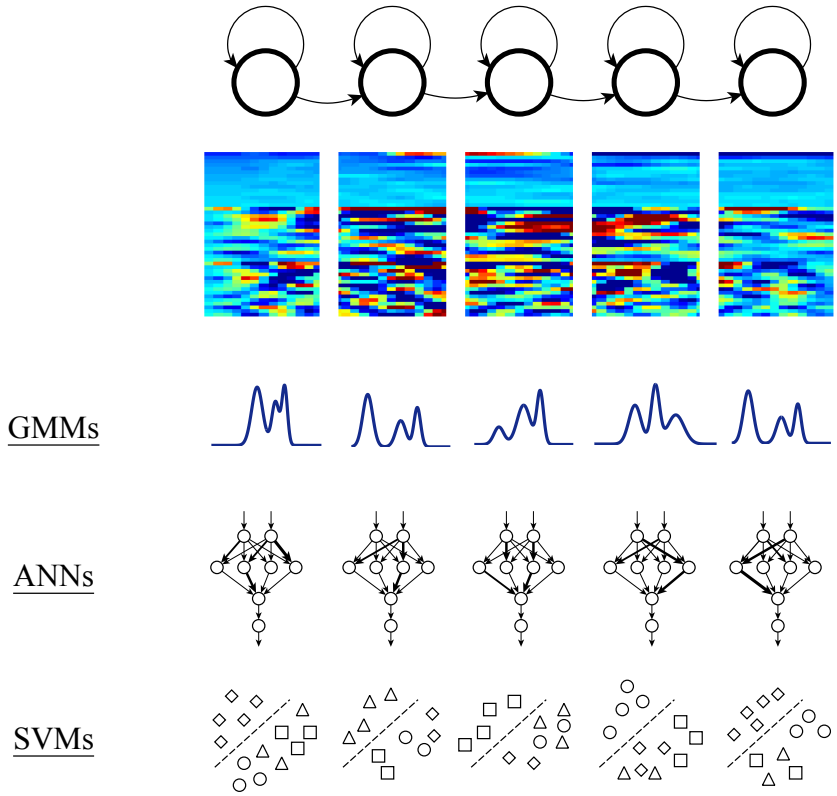


Figure 6.1: *Hybrid models. Instead of GMMs, other approaches as ANNs (e.g. Morgan et al. 1993) or SVMs (e.g. Stadermann and Rigoll 2004) can be used for modeling the observations or emissions.*

ture the nature of speech in so an efficient manner as HMMs do[6]. Consequently, other techniques for speech recognition have appeared that, instead of substituting the HMMs with ANNs, SVMs or building hybrid systems of the latter with HMMs, examine how to directly improve continuous density HMMs (keeping Gaussian mixture models) founded on the DT principle, meaning the case of large margin DT as investigated by authors such as Li et al. 2005, Sha and Saul 2007 and Jiang et al. 2006.

Regarding the above mentioned issues, we adhere to the large margin DT techniques and use them as basis for the last refinement step in our framework (Fig. 3.1). In the next sections, we explain the fundamentals of large margin discriminative training (LM DT) and introduce some novel strategies for its application in efficient learning situations.

6.1.1 The margin

Another criterion to estimate the HMM parameters λ different from maximizing the likelihood is, for instance, to minimize the total number of misclassified training samples

[6]See the Sec. 2.3.5 for detailed references and argumentation, Jiang et al. 2006.

following MCE optimization (Juang et al. 1997; Li et al. 2005). Nevertheless, having a low error rate in the training data does not necessarily imply to have good generalization in the classifier (suitability to recognize unseen samples), which can be after all measured through its margin (Vapnik 1998: qtd. in Jiang et al. 2006). So, a classifier that aims to maximize the margin usually achieves better performance in not previously observed data, being then the optimization of the margin in the construction of HMMs (also when the training samples are already well classified) a very promising principle (Jiang et al. 2006).

The best known classifiers based on the concept of the margin principle are the SVMs (e.g. Ganapathiraju et al. 2004b, Bazzi and Katabi 2000 and Wan and Campbell 2000), where hyperplanes are used to separate labeled data and are designed to maximize the minimum distance or margin from any sample to the decision boundary (see Fig. 6.2, Sha and Saul 2006). Although several authors have realized extensive efforts to introduce SVMs in ASR systems, for instance in speaker or speech recognition, SVMs present some disadvantages, e.g. the need of kernels for speech recognition and their unsuitability to deal with sequence modeling (see Sec. 2.3.5; Jiang et al. 2006).

In order to maximize the distances between the word-models, a group of researchers directly extend well-known discriminative training methods used in ASR systems as MCE to the large margin principle (e.g. Yu et al. 2007, Yu et al. 2008 and Ratnagiri et al. 2011), while others concentrate on improving the GMMs of the HMMs updating their means (e.g. Li et al. 2005; Jourani et al. 2010) as well as their variances (e.g. Chang et al. 2008; Sha and Saul 2007). In our presented system in Ayllón Clemente et al. 2010b, the minimum classification error estimation using the extended Baum-Welch (EBW) algorithm proposed in He et al. 2008 was performed. However, the application of incremental MCE training in our previous work was not beneficial because of the very small number of training samples used (training samples already well classified). Therefore, we only apply the techniques that refine the GMMs in our current framework that we are going to denote them as $\mathrm{LM}_{\boldsymbol{\mu}}$ (only means are updated, Li et al. 2005) and $\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ (means and variances are recalculated, Sha and Saul 2007).

Next, we give some definitions about decision rules and optimization algorithms employed in large margin HMM classifiers and explain that the estimation of a large margin HMM can be defined as a minimax optimization problem (Li et al. 2005).

6.1.2 Decision rules for updating the GMMs in LM DT

In the following part, the estimation of HMMs for speech recognition based on the large margin principle will be explained. We start identifying a subset of support samples or tokens $\overline{\boldsymbol{X}}_S$ from all utterances in the training set $\overline{\boldsymbol{X}}$, which are characterized to be very close to the classifier boundary but still has a positive margin (correct classified) (Li et al. 2005):

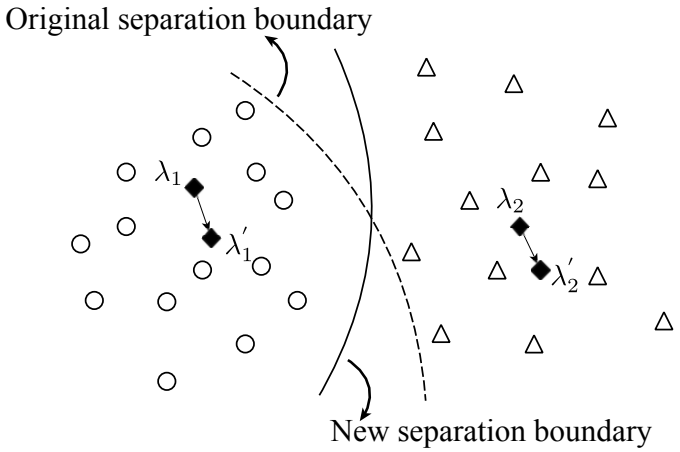


Figure 6.2: *Illustration of decision boundaries in case of ML estimation (original) and when using as criterion the maximization of the margin between clusters (new), Jiang et al. 2006.*

$$\overline{\boldsymbol{X}}_S = \{\boldsymbol{X}_i |\ \boldsymbol{X}_i \in \overline{\boldsymbol{X}} \text{ and } 0 \leq d(\boldsymbol{X}_i) \leq \nu\} \tag{6.1}$$

where $\nu > 0$ is a previous selected positive threshold and d is a distance measure or margin to the separation boundary.

To improve the generalization ability, the HMM parameters λ must be estimated with the goal that all support tokens are separated the maximum possible distance from the decision boundaries (Li et al. 2005). In other words, the HMM acoustic models λ must be calculated according to the rule of maximizing the minimum margin of all tokens, being this kind of computation called large margin estimation (LME) and the respective models $\widehat{\lambda}$ obtained consequently large margin HMMs (Li et al. 2005):

$$\widehat{\lambda} = \arg \max_{\lambda} \min_{\boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S} d(\boldsymbol{X}_i) \tag{6.2}$$

Here, we describe two distance measures[7] based on the Euclidean and Mahalanobis distances.

Euclidean distance

In 1-D, the Euclidean distance[8] between two points (y_i, y_j) is defined as:

[7] To our best knowledge, the distance measures (similar to the definition of margins) presented in the following paragraphs are not exclusive. Thus, novel distance measures can emerge without violating the large margin principle.

[8] A standard in Mathematics.

$$d_{i,j} = |y_i - y_j| \tag{6.3}$$

In the context of our problem, one can see in y an equivalence to the discriminant function F[9] and hence, the separation margin against other classes for a word sample $\boldsymbol{X}_i$ assuming that it belongs to class w_i can be formulated as (Jiang et al. 2006; see also Chang et al. 2008):

$$\begin{aligned} d(\boldsymbol{X}_i) &= F(\boldsymbol{X}_i|\lambda_{w_i}) - \max_{\substack{w_i,w_j \in \boldsymbol{w} \\ w_i \neq w_j}} F(\boldsymbol{X}_i|\lambda_{w_j}) \\ &= \min_{\substack{w_i,w_j \in \boldsymbol{w} \\ w_i \neq w_j}} [F(\boldsymbol{X}_i|\lambda_{w_i}) - F(\boldsymbol{X}_i|\lambda_{w_j})] \end{aligned} \tag{6.4}$$

where $\boldsymbol{w}$ is the set of the previously learned words, F the discriminant function and λ_{w_j} the word-model of the class w_j. Indeed, if $d(\boldsymbol{X}_i) \leq 0$, $\boldsymbol{X}_i$ will be misclassified and if $d(\boldsymbol{X}_i) > 0$, $\boldsymbol{X}_i$ will be correctly categorized to the class w_i (Li et al. 2005).

Mahalanobis distance

Another approach to estimate the GMMs is proposed by Sha and Saul 2007, whose technique makes use of the Mahalanobis distance M to optimize the GMMs (assumed as ellipsoids) of the HMM λ_{w_i}:

$$M(\boldsymbol{x}_t) = (\boldsymbol{x}_t - \boldsymbol{\mu}_{m_{S_t,\lambda_{w_i}}})^T \boldsymbol{\Sigma}^{-1}_{m_{S_t,\lambda_{w_i}}} (\boldsymbol{x}_t - \boldsymbol{\mu}_{m_{S_t,\lambda_{w_i}}}) + \theta; \quad \boldsymbol{x}_t \in \boldsymbol{X}_i \tag{6.5}$$

where the additional parameter θ is a non-negative scalar offset as well as $\boldsymbol{\mu}_{m_{S_t,\lambda_{w_i}}}$ and $\boldsymbol{\Sigma}_{m_{S_t,\lambda_{w_i}}}$ are the means and covariance matrices for the mixture component m in a GMM of the state S_t of model λ_{w_i}. According to Sha and Saul 2007, one way of obtaining a label for the vector $\boldsymbol{x}_t$ of the sample $\boldsymbol{X}_i$ is to consider the component of the state S_t with the highest emission probability according to the ML estimation. In the original description of Sha and Saul 2007, each class w_i is modeled by one state. In our HMM framework, the initial concept has been extended to cope with multiple states with the help of the Viterbi decoding algorithm.

6.1.3 Optimization algorithms applied in LM DT

In this section, different optimization algorithms for large margin DT are discussed. One is based on gradient descent, iterative localized optimization (Li et al. 2005), and another aims at convex optimization[10] (Sha and Saul 2006).

[9] $F(\boldsymbol{X}_i|\lambda_w) = P(\lambda_w) \cdot P(\boldsymbol{X}_i|\lambda_w)$, see Sec. 2.3.3 for references.

[10] Other authors (e.g. Li and Jiang 2007 and Chang et al. 2008) also employed semidefinite programming or convex relaxation, but we concentrate on Sha and Saul 2006.

Iterative localized optimization[11]

As stated in Li et al. 2005, the maximin problem from Eq. 6.2 integrating Eq. 6.4 can be transformed into the following minimax optimization problem:

$$\widehat{\lambda} = \arg\min_{\lambda} \max_{\substack{\boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S \\ w_i, w_j \in \boldsymbol{w} \\ w_j \neq w_i}} [F(\boldsymbol{X}_i|\lambda_{w_j}) - F(\boldsymbol{X}_i|\lambda_{w_i})] \tag{6.6}$$

subject to:

$$\begin{gathered} F(\boldsymbol{X}_i|\lambda_{w_j}) - F(\boldsymbol{X}_i|\lambda_{w_i}) < 0 \\ \forall \boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S; \quad w_i, w_j \in \boldsymbol{w}; \quad w_j \neq w_i \end{gathered} \tag{6.7}$$

Unfortunately, the constraints of Eq. 6.7 do not ensure that a minimax point exists, being necessary more constraints in order to find a solution to the optimization problem. According to Li et al. 2005, this issue can be solved by only updating one model in each optimization stage (for RLM the new learned word-model, see Sec. 6.2.2) until the minimum margin is achieved. This localized optimization strategy gives the name of *iterative localized optimization* to the method and the Eq. 6.6 can be then redefined as (Li et al. 2005):

$$\widehat{\lambda}_{w_m} = \arg\min_{\lambda_{w_m}} \max_{\substack{\boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S \\ w_i, w_j \in \boldsymbol{w} \\ w_j \neq w_i \\ w_j = w_m \| w_i = w_m}} [F(\boldsymbol{X}_i|\lambda_{w_j}) - F(\boldsymbol{X}_i|\lambda_{w_i})] \tag{6.8}$$

subject to:

$$\begin{gathered} F(\boldsymbol{X}_i|\lambda_{w_j}) - F(\boldsymbol{X}_i|\lambda_{w_i}) < 0 \\ \forall \boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S; \quad w_i, w_j \in \boldsymbol{w}; \quad w_j \neq w_i; \quad w_j = w_m \parallel w_i = w_m \end{gathered} \tag{6.9}$$

This minimax problem can be reduced to a minimization task if the "max" of the Eq. 6.8 is substituted by a differentiable function, where the gradient descent algorithm (Appendix C) can be employed to update the estimates minimizing the simplified objective function $Q(\lambda_{w_m})$, Li et al. 2005:

$$Q(\lambda_{w_m}) = \frac{1}{\eta} log[\sum_{\substack{\boldsymbol{X}_i \in \overline{\boldsymbol{X}}_S \\ w_i, w_j \in \boldsymbol{w} \\ w_j \neq w_i \\ w_j = w_m \| w_i = w_m}} exp[\eta \cdot F(\boldsymbol{X}_i|\lambda_{w_j}) - \eta \cdot F(\boldsymbol{X}_i|\lambda_{w_i})]] \tag{6.10}$$

when $\eta \to \infty$, this function approaches the maximization in Eq. 6.8, being the constant η then larger than 1.

[11] Description based on Li et al. 2005.

Iterative localized optimization
Description based on Li et al. 2005.

Step 1: *Initialization*

- Classification of all training samples $\overline{\boldsymbol{X}}$ based on the current estimates λ.
- Identification of the set of support tokens $\overline{\boldsymbol{X}}_S$ according to Eq. 6.1.

Step 2: *Iteration*

- Selection of the support token $\boldsymbol{X}_m$ from $\overline{\boldsymbol{X}}_S$, which currently achieves the minimum margin.
- Selection of the true model of $\boldsymbol{X}_m$, i.e. λ_{w_m} for optimization in this step.
- Minimization of the objective function by only updating the model $\lambda_{w_m} : \lambda_{w_m} \Rightarrow \widehat{\lambda_{w_m}}$ using a gradient descent algorithm.

Step 3: *Termination*

Repeat until some threshold is reached or the maximum number of iterations is exceeded.

Towards convex optimization[12]

In order to avoid local minima, a convex optimization should be used (Sha and Saul 2007). With the goal of simplifying the Eq. 6.5, an expanded matrix $\boldsymbol{\phi}_{m_{S_t,\lambda_{w_i}}}$ can be formulated as (Sha and Saul 2006):

$$\boldsymbol{\phi}_{m_{S_t,\lambda_{w_i}}} = \begin{pmatrix} \boldsymbol{\Sigma}^{-1}_{m_{S_t,\lambda_{w_i}}} & -\boldsymbol{\Sigma}^{-1}_{m_{S_t,\lambda_{w_i}}} \boldsymbol{\mu}_{m_{S_t,\lambda_{w_i}}} \\ -\boldsymbol{\mu}^T_{m_{S_t,\lambda_{w_i}}} \boldsymbol{\Sigma}^{-1}_{m_{S_t,\lambda_{w_i}}} & \boldsymbol{\mu}^T_{m_{S_t,\lambda_{w_i}}} \boldsymbol{\Sigma}^{-1}_{m_{S_t,\lambda_{w_i}}} \boldsymbol{\mu}_{m_{S_t,\lambda_{w_i}}} + \theta \end{pmatrix} \tag{6.11}$$

The matrix $\boldsymbol{\phi}_{m_{S_t,\lambda_{w_i}}}$ is positive semidefinite, so that the Mahalanobis distance M can be formulated as (Sha and Saul 2006):

$$M(\boldsymbol{x}_t) = \boldsymbol{z}^T \cdot \boldsymbol{\phi}_{m_{S_t,\lambda_{w_i}}} \cdot \boldsymbol{z} \; ; \quad \text{where} \quad \boldsymbol{z} = \begin{pmatrix} \boldsymbol{x}_t \\ 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{x}_t \in \boldsymbol{X}_i \tag{6.12}$$

Next, Eq. 6.12 is integrated in a new formulation for an objective function (Sha and Saul 2007), in which numerous semidefinite matrices are incorporated to take into account all the mixture components of the GMMs and states inside each word-model.

Reviewing Sha and Saul 2007, one can observe that the maximization of these expressions can lead to convex optimization after some necessary transformations. Although these problems can be normally solved by interior-point algorithms, Sha and Saul 2007 advised (due to the magnitude of the problem) to use an efficient special-purpose solver applying

[12]Description based on Sha and Saul 2006 as well as Sha and Saul 2007.

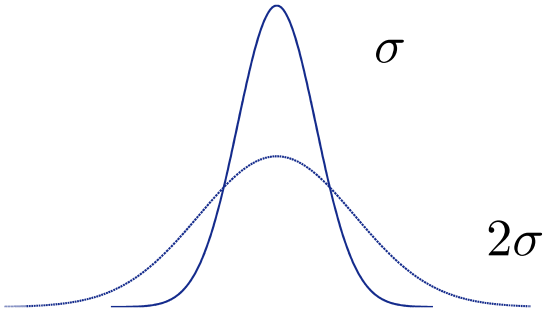


Figure 6.3: *Visualization of how appointed Gaussians are widened when the variances are multiplied by a positive constant larger than 1.*

projected subgradient methods[13] or more aggressive optimizers combining the latter with conjugate gradients (see Sha and Saul 2007 for more details).

6.2 Large margin computational strategies for limited data[14]

Following the same reasoning as in Chapter 4[15], when a modest number of samples are employed, the HMM models may be overtrained, i.e. the word-models already seem well separated in the learning phase and the components of the GMMs very elongated (characterized by large "distances" between the classes). Therefore, optimizing the model in each iteration with the maximization of distances as referred in the state-of-the-art methods explained in the last section does not improve recognition results significantly[16]. Here, we present three strategies to cope with this challenge (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012): multiplying artificially the variances by a scaling factor, retraining the last model and using a confidence based selection method, where the two last strategies are combined with the first one.

6.2.1 Scaling factor (SF)

We propose one heuristic method that slightly increases the variances with the goal to obtain overlapping models (Fig. 6.3). We multiply the variances of the GMMs by a scaling factor SF in order to intentionally detriment the estimations and to force the algorithms to recalculate the means ($\mathrm{LM}_{\boldsymbol{\mu}}$, Li et al. 2005) and in some cases also the variances of the GMMs ($\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$, Sha and Saul 2007).

[13] For projected subgradient methods see Bertsekas 1999: qtd. in Sha and Saul 2007.

[14] This section and its corresponding subsections take numerous paragraphs, some of them verbatim, from the previous works Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012.

[15] See references directly in Chapter 4.

[16] Experiments have been realized additional to the ones effectuated in Ayllón Clemente et al. 2010b to support this argumentation.

In addition to the scaling factor SF, we propose two complementary strategies to the distance criteria, which are used in combination with the algorithms proposed by Li et al. 2005 and Sha and Saul 2007. These strategies select a set of classes in each iteration step and the models of these classes are the ones to be updated during that step. This selection reduces the computational cost, i.e. it saves the retraining of some models, and improves the recognition results for a reduced number of training samples when used jointly with the scaling factor SF.

6.2.2 Retraining the last introduced word-model (RLM)

In Chapter 3, we explained how and why the HMMs are estimated incrementally in our word learning system. We can take advantage of the "incremental" nature of our system and only optimize the last computed model. For example, if word-models 1 and 2 have already been estimated and optimized, word-model 3 may only affect the relation between 1 and 3, and 2 and 3, but it may not influence the relation between word-models 1 and 2. Hence, this approach assumes that the only word-model to adapt in each step is the last model added to the system. This strategy reduces the computational cost significantly, because only one model is modified in each time step (Fig. 6.4). However, this simplification neglects the non-pairwise interactions among future unseen classes. This method is referred to as retraining the last model (RLM).

It is also straightforward to think that if incremental learning is not applied, the RLM criterion is no longer suitable.

Retraining the last model (RLM) – LM strategy
Based on the proposed approach in Ayllón Clemente et al. 2010a integrated in the technique of Li et al. 2005/Sha and Saul 2007.

Precondition

More than one model has been already learned (not taking into account the silence-model).

Step 1: *Initialization - Scaling factors*

Application of the scaling factor SF to the variances of the new learned model λ_{w_n} (see Sec. 6.3 for examples of scaling factor values).

Step 2: *Iteration*

a) *Search of support tokens*

Evaluation of the training samples to obtain the support tokens $\overline{\boldsymbol{X}}_S$ of the learned models λ_{w_1}, λ_{w_2}, ... , λ_{w_n}.

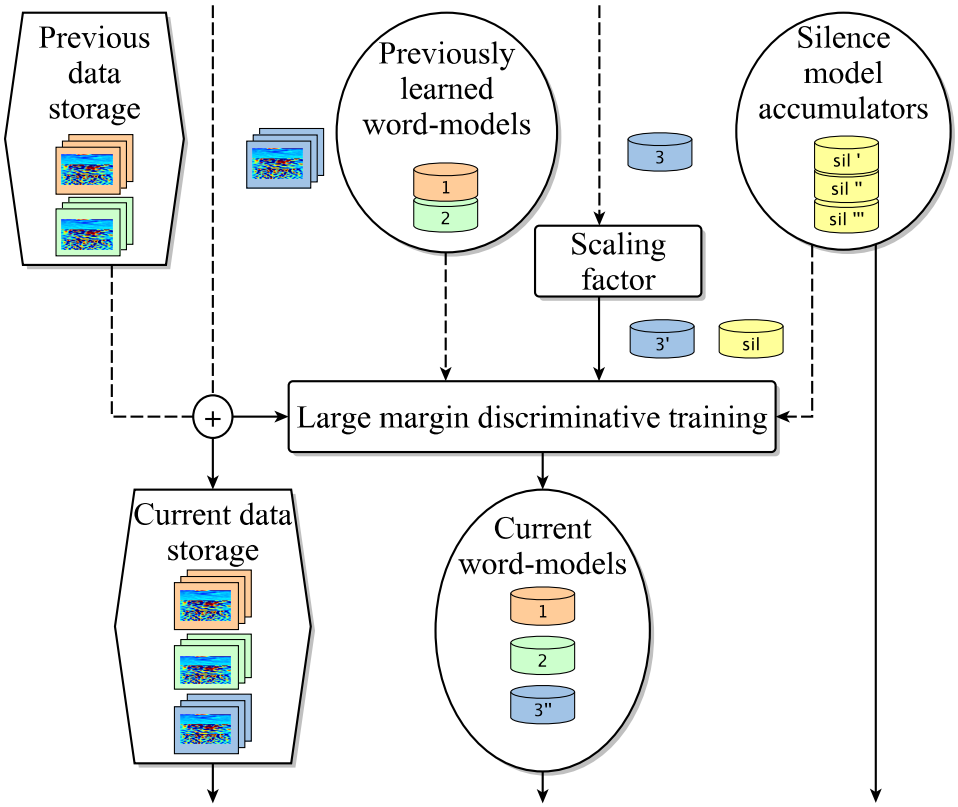


Figure 6.4: *Scheme of the strategy called retraining the last model (RLM), where only the last learned model is updated (Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2012). The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a). This plot is an extract of the Fig. 3.1.*

b) *Update of the new word-model*

Leaving the parameters of the word-models λ_{w_1}, λ_{w_2}, ... , $\lambda_{w_{n-1}}$ fixed, the margin is maximized by only updating:

- the means of the word-model λ_{w_n} ($\boldsymbol{\mu}_{\lambda_{w_n}} : \boldsymbol{\mu}_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_n}}$). The optimization is realized by means of a gradient descent algorithm, which minimizes the objective function $Q(\lambda_{w_n})$ from Eq. 6.10 (Li et al. 2005).
- the means ($\boldsymbol{\mu}_{\lambda_{w_n}} : \boldsymbol{\mu}_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_n}}$) and the variances ($\boldsymbol{\sigma}^2_{\lambda_{w_n}} : \boldsymbol{\sigma}^2_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_n}}$) of the word-model λ_{w_n}. The optimization is realized using the special-solver proposed by Sha and Saul 2007.

Step 3: *Termination*
The step 2 will be repeated until convergence or the maximum number of iterations is reached. In each iteration, it is important to check that the margin criteria are still valid for the current support tokens $\overline{\boldsymbol{X}}_S$.

6.2.3 Selecting word-models via confidence intervals (CBS)

As mentioned in previous sections, it is not feasible to obtain a reliable and uniform estimation of the word-models when few data samples are used. In this context, our criterion to select which word-models are the most appropriate ones to update is based on the use of a confidence-like interval over the probability distribution of the discriminant functions F of each model. Thus, this strategy is named confidence based selection (CBS).

In order to identify which word-models have to be adapted, we implement the following procedure (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012):

- Firstly, we take all samples $\boldsymbol{X}_j$ that are assigned to a particular class w_j and compute the score of the discriminant function $F(\boldsymbol{X}_j|\lambda_{w_j})$ for each sample $\boldsymbol{X}_j$.
- Next, a distribution of the scores obtained for each class w_j is constructed (e.g. with the help of histograms) and a confidence-like interval is defined.
- Afterwards, we take all samples $\boldsymbol{X}_i$ that are not assigned to that class and evaluate through $F(\boldsymbol{X}_i|\lambda_{w_j})$ how many of them fall into this confidence-like interval, i.e. how many of them have a large confidence/probability for a wrong class assignment (Fig. 6.6).
- Finally, if the distribution of a class w_j shows a strong overlap with other classes, then the word-model of this class has to be adapted (Fig. 6.5).

If no incremental learning is used, the CBS criterion can be also employed to reduce the computational cost (not updating all the models), on the contrary the algorithm will behave like any other state-of-the-art discriminative system when all samples and word-models are considered.

Confidence based selection (CBS) – LM strategy
Based on the proposed approach in Ayllón Clemente et al. 2010a integrated in the technique of Li et al. 2005/Sha and Saul 2007.

Precondition

More than one model has been already learned (not taking into account the silence-model).

Step 1: *Confidence based selection*

a) Construction of the distribution of the discriminant functions $F(\boldsymbol{X}_j|\lambda_{w_j})$ of each word-model λ_{w_j}.

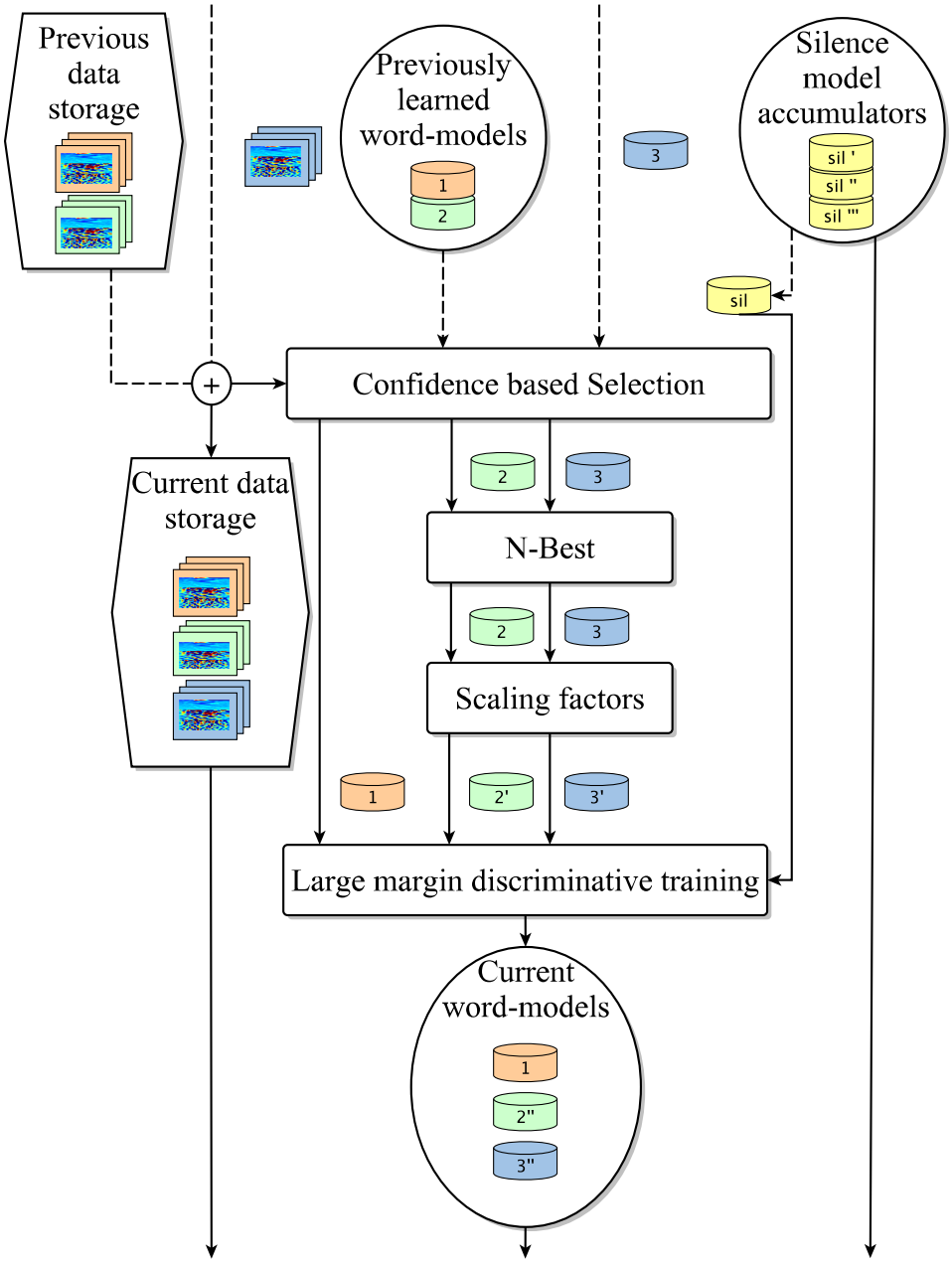


Figure 6.5: *Scheme of the strategy called confidence based selection (CBS), (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). In this example, the word-models 2 and 3 are updated in contrast to RLM where only model 3 is updated. N-Best can be computed to save computation, however re-ranking is needed when an updated model is obtained. The language model (LM) was not displayed to simplify the picture, see Fig. 2.8(a). This plot is an extract of the Fig. 3.1.*

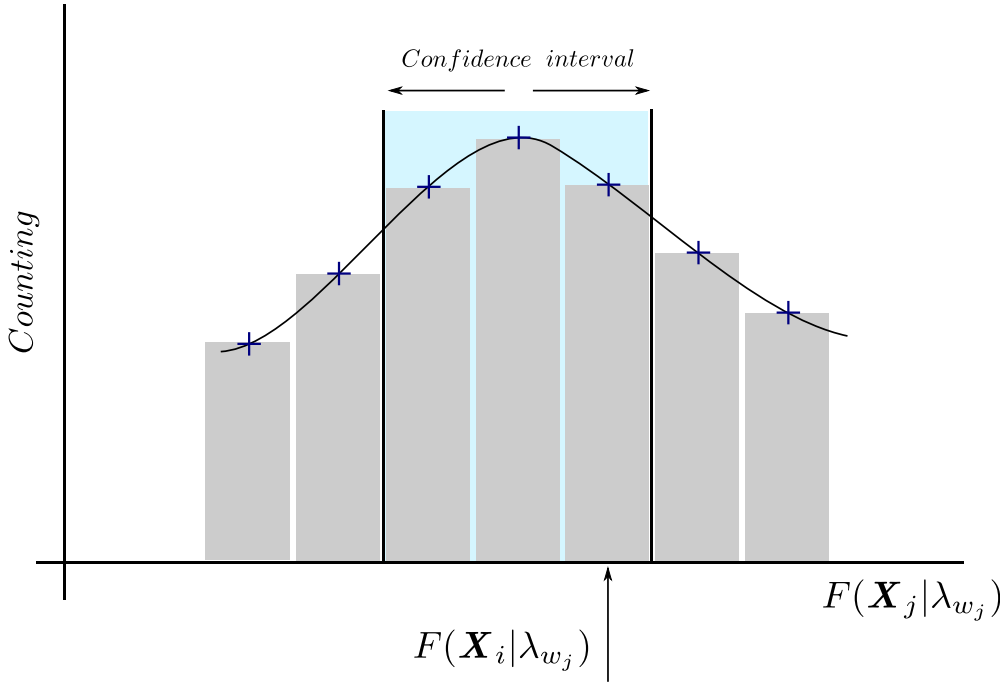


Figure 6.6: *Showcase of the distribution of the discriminant function $F(\boldsymbol{X}_j|\lambda_{w_j})$, Ayllón Clemente et al. 2012. This distribution helps us to identify if a sample $\boldsymbol{X}_i$ could be interpreted to belong to the class encoded by the discriminant function $F(\boldsymbol{X}_j|\lambda_{w_j})$. This latter is the case illustrated in the plot.*

b) Evaluation of each training sample $\boldsymbol{X}_i$ with all other competing word-models λ_{w_j} through the computation of $F(\boldsymbol{X}_i|\lambda_{w_j})$.

c) Analysis of the score obtained for $\boldsymbol{X}_i$ using the word-model λ_{w_j} to determine if it is distributed in the same way as the samples $\boldsymbol{X}_j$ of λ_{w_j} using the distribution for each word-model λ_{w_j} of a), see Fig. 6.6.

d) Computation of the number of word-models "λ_{w_i}" that the word-model λ_{w_j} overlaps applying the condition in c).

Step 2: *N-Best (optional)*

Then, all the word-models λ_{w_j} that overlap other word-models λ_{w_i} and the new learned word-model λ_{w_n} are selected to be adapted in the next steps. In order to save computation time, we strongly recommend the application of the N-Best method[17], if the number of such models exceeds a predefined threshold, e.g. 6 word-models. The input of the filter employed in these cases is the ranking obtained through step 1, d).

Step 3: *Scaling factors*

Application of the scaling factor SF to the variances of the word-models selected in step 2

[17]Related to Chow and Schwartz 1989.

(see Sec. 6.3 for examples of scaling factor values). It is also possible to apply the scaling factors before the CBS if the set of word-models λ_{w_j} that overlap other word-models λ_{w_i} is empty.

Step 4: *Iteration*

a) *Search of support tokens*

Evaluation of the training samples to obtain the support tokens $\overline{\boldsymbol{X}}_S$ of the learned word-models λ_{w_1}, λ_{w_2}, ... , λ_{w_n}.

b) *Update of the overlapping models*

Leaving the parameters of the word-models not selected in step 2 fixed (e.g. λ_{w_2}, λ_{w_4}, $\lambda_{w_{n-1}}$), the margin is maximized by only updating:

- the means of the word-models λ_{w_1}, λ_{w_3}, λ_{w_5}, ... , $\lambda_{w_{n-2}}$, λ_{w_n}:

$$\begin{aligned}
&\boldsymbol{\mu}_{\lambda_{w_1}} : \boldsymbol{\mu}_{\lambda_{w_1}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_1}} \\
&\boldsymbol{\mu}_{\lambda_{w_3}} : \boldsymbol{\mu}_{\lambda_{w_3}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_3}} \\
&\boldsymbol{\mu}_{\lambda_{w_5}} : \boldsymbol{\mu}_{\lambda_{w_5}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_5}} \\
&\quad . \quad . \quad . \\
&\boldsymbol{\mu}_{\lambda_{w_{n-2}}} : \boldsymbol{\mu}_{\lambda_{w_{n-2}}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_{n-2}}} \\
&\boldsymbol{\mu}_{\lambda_{w_n}} : \boldsymbol{\mu}_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_n}}
\end{aligned}$$

  The optimization is realized by means of a gradient descent algorithm, which minimizes the objective function $Q(\lambda_w)$ and updates only one word-model in each step according to the iterative localized optimization of Li et al. 2005.

- the means and the variances of the word-models λ_{w_1}, λ_{w_3}, λ_{w_5}, ... ,$\lambda_{w_{n-2}}$, λ_{w_n}:

$$\begin{aligned}
&\boldsymbol{\mu}_{\lambda_{w_1}} : \boldsymbol{\mu}_{\lambda_{w_1}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_1}} && \boldsymbol{\sigma}^2_{\lambda_{w_1}} : \boldsymbol{\sigma}^2_{\lambda_{w_1}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_1}} \\
&\boldsymbol{\mu}_{\lambda_{w_3}} : \boldsymbol{\mu}_{\lambda_{w_3}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_3}} && \boldsymbol{\sigma}^2_{\lambda_{w_3}} : \boldsymbol{\sigma}^2_{\lambda_{w_3}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_3}} \\
&\boldsymbol{\mu}_{\lambda_{w_5}} : \boldsymbol{\mu}_{\lambda_{w_5}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_5}} && \boldsymbol{\sigma}^2_{\lambda_{w_5}} : \boldsymbol{\sigma}^2_{\lambda_{w_5}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_5}} \\
&\quad . \quad . \quad . && \quad . \quad . \quad . \\
&\boldsymbol{\mu}_{\lambda_{w_{n-2}}} : \boldsymbol{\mu}_{\lambda_{w_{n-2}}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_{n-2}}} && \boldsymbol{\sigma}^2_{\lambda_{w_{n-2}}} : \boldsymbol{\sigma}^2_{\lambda_{w_{n-2}}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_{n-2}}} \\
&\boldsymbol{\mu}_{\lambda_{w_n}} : \boldsymbol{\mu}_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\mu}}_{\lambda_{w_n}} && \boldsymbol{\sigma}^2_{\lambda_{w_n}} : \boldsymbol{\sigma}^2_{\lambda_{w_n}} \Rightarrow \widehat{\boldsymbol{\sigma}}^2_{\lambda_{w_n}}
\end{aligned}$$

  The optimization is realized using the special-solver proposed by Sha and Saul 2007.

Step 5: *Termination*

The step 4 will be repeated until convergence or the maximum number of iterations is reached. In each iteration, it is important to check that the margin criteria are still valid for the current support tokens $\overline{\boldsymbol{X}}_S$.

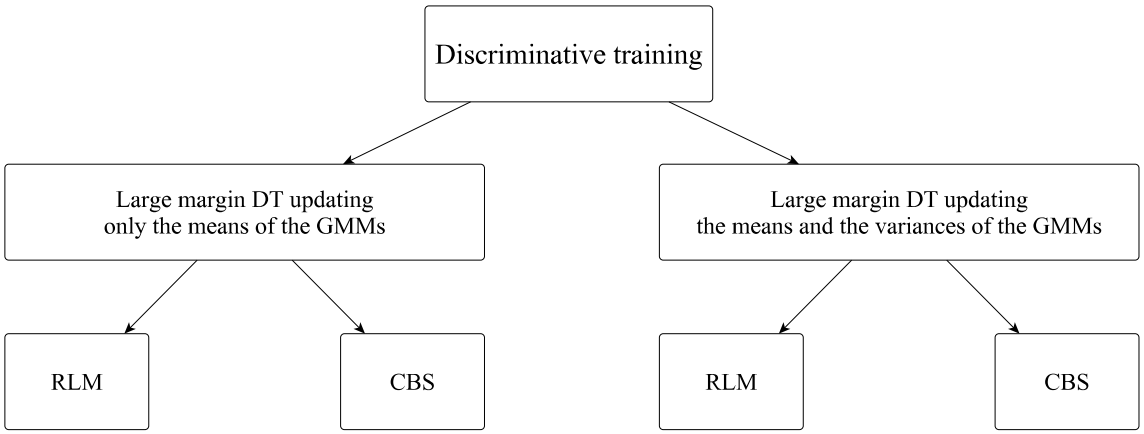


Figure 6.7: *Overview of the experiments realized in discriminative training (Ayllón Clemente et al. 2012). RLM stands for retraining the last model and CBS for confidence based selection. All these methods are combined with the scaling factor SF.*

6.3 Evaluation[18]

Techniques evaluated

In this section, we evaluate the advantage of the incorporation of a large margin discriminative training stage analyzing two state-of-the-art algorithms previously explained (see Fig. 6.7):

- $\mathrm{LM}_{\boldsymbol{\mu}}$ (Li et al. 2005): only the means of the GMMs of the word-models are updated. The adaptation of these parameters is realized according to the iterative localized optimization proposed by Li et al. 2005. In interactive learning, it is crucial to reduce the computation time, thus we use the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (see Nocedal and Wright 1999, Sec. 7.2, p. 177) for the iterative localized optimization.

- $\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ (Sha and Saul 2007): the means and the variances of the GMMs of the word-models are recalculated. The method applied is here based on the Mahalanobis distance, the semidefinite matrices and the special-solver introduced in Sha and Saul 2007.

The goal of the comparison of the algorithms $\mathrm{LM}_{\boldsymbol{\mu}}$ and $\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ is to determine the benefits of the additional computation of the variances in $\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$, despite the increased computational cost. The baseline system used is the same as in previous chapters.

[18]Several paragraphs (some verbatim) are taken from previous works as Ayllón Clemente et al. 2010a, Ayllón Clemente et al. 2012, however some experiments have been extended in contrast to the previously published ones.

In our incremental word learning framework, the parameters of the word-models are estimated from few training samples. This leads to a possible overspecialization of the learned word-models that already seem well separated from the perspective of the available training data (see previous sections). Unfortunately, standard LM DT algorithms are not able to further optimize the models efficiently in these conditions[19]. Therefore, we proposed three strategies to cope with the efficient learning challenge and the high computational cost involved in DT algorithms (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012): the inclusion of a scaling factor (SF), the retraining of the last learned word-model (RLM) and a selection strategy based on confidence-like intervals (CBS). All these strategies are explained in Sec. 6.2 and will be evaluated jointly with the $\mathrm{LM}_{\boldsymbol{\mu}}$ and $\mathrm{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ algorithms as the Fig. 6.7 illustrates.

Parameters employed[20]

The scaling factor is set to $SF = 1.1$ for RLM. In CBS, SF is set to 1 if the word-model λ_{w_j} of the class w_j overlaps more than 2 word-models λ_{w_i} of different classes w_i, to 1.1 if it overlaps 1 or 2 models and to 1.2 if it does not overlap any word-model. When the number of word-models λ_{w_j} that overlaps other models is very reduced or there is no word-model that overlaps another, it is recommended to apply the scaling factor before CBS.

Setting the length of the confidence-like interval to σ, with σ being the standard deviation of the distribution of the discriminant functions, saves computation time and does not impair the recognition results. The rest of the parameters are set to default values, more information can be found in Li et al. 2005 and Sha and Saul 2007.

Databases

As in the previous experiments, 250 different subsets of TIDigits DB (Leonard and Doddington 1993, see Appendix A.2) are used to obtain the 25 cross-validations of each configuration of the number of training samples (from 1 to 10). In each of these sets employed for the learning stage, single numbers ("1" to "9", "zero" and "oh") are uttered by different adult male speakers in isolation. In the evaluation phase, three test sets are used: two of them contain isolated words pronounced by different men and women[21] respectively and in the third one, continuous speech sentences[22] are produced by men.

[19] According to the experiments realized.

[20] This parameterization is similar to the one used in our previus works Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012.

[21] In Ayllón Clemente et al. 2010a, a test set containing female speakers was also used, however the parameterization and final configuration of the system was not the same as here.

[22] The continuous speech utterances analysis is extended here to all the methods to compare and not the best approach as in Ayllón Clemente et al. 2012.

Table 6.1: *Word error rates (WER %) of the state-of-the-art methods and our proposed algorithms. For each method, the WER values represent the mean of a 25-fold cross-validation on the training set evaluated on separated male (M) and female (W) speakers test sets with isolated words and another test set (C) with continuous speech utterances produced by male speakers (extended from Ayllón Clemente et al. 2012). "No.* $\boldsymbol{X}$*" stands for the number of training samples (R). In the discriminative training stage, the variance floor approach applied is* $V^*_{\%}$ *and the initialization used MSA. Here, we differentiate between the* $LM_{\boldsymbol{\mu}}$ *and the* $LM_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ *approaches. RLM is the strategy called retraining the last model and CBS the strategy named confidence based selection. All the scores are computed including the strategy denominated scaling factor SF. The best WERs are marked in bold.*

	$LM_{\boldsymbol{\mu}}$						$LM_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$					
No.	RLM			CBS			RLM			CBS		
$\boldsymbol{X}$	M	W	C	M	W	C	M	W	C	M	W	C
1	22.3	50.5	43.8	21.3	49.8	42.4	20.3	48.5	**40.5**	**19.2**	**47.4**	42.2
2	5.1	41.2	36.0	5.0	41.0	29.7	3.9	40.4	27.8	**3.7**	**39.1**	**25.1**
3	2.7	29.6	22.4	2.6	28.2	19.0	2.5	27.3	15.8	**2.2**	**24.1**	**13.7**
4	2.1	27.3	14.8	2.0	25.1	11.0	2.0	24.2	9.9	**1.3**	**23.0**	**9.3**
5	1.7	21.1	9.5	1.5	20.0	7.1	1.4	19.8	7.1	**1.1**	**19.0**	**6.2**
6	1.2	19.8	7.8	1.1	19.4	4.1	**1.0**	**17.4**	3.9	**1.0**	17.5	**3.2**
7	1.1	18.1	4.7	0.7	19.1	3.7	0.7	16.5	3.6	**0.6**	**15.4**	**2.8**
8	0.7	14.4	3.5	**0.5**	15.8	3.4	**0.5**	**13.4**	3.1	**0.5**	13.6	**2.1**
9	0.6	13.9	3.2	0.5	13.8	2.9	0.4	13.2	2.3	**0.2**	**13.1**	**2.0**
10	0.4	**12.0**	2.9	0.3	12.1	2.4	0.3	12.9	2.1	**0.2**	12.3	**1.7**

Results

The average values of the scores obtained, when recognizing isolated words uttered by men and women as well as the recognition scores of continuous speech sentences pronounced by male speakers, using the proposed strategies are detailed in Table 6.1. As in previous chapters, we display in Fig. 6.8(a) and 6.8(b) the average of the absolute recognition scores of the proposed methods for each phase of the incremental word learning system. In addition to the recognition scores, the average of the relative improvements are shown in Fig. 6.8(c), 6.8(d) and Table 6.2[23]. Despite the relevance of the average of the relative improvements, we also depict the minimum and maximum values of the cross-validations in Fig. 6.8(c) and 6.8(d) in order to identify a possible overlap (bars) of the improvements in each phase of our system.

Tables 6.1 and 6.2 demonstrate that the introduction of large margin discriminative

[23]The values in Table 6.1 are rounded. The values calculated in Table 6.2, although also rounded, were calculated with the values of Table 6.1 without rounding.

Table 6.2: *Relative improvement of the word error rates (WER %) of all methods presented in this chapter compared to the baseline system described in Sec. 4.2.4 (extended from Ayllón Clemente et al. 2012 (female speakers scores)). For abbreviations, see Table 6.1.*

		MSA	$LM_{\boldsymbol{\mu}}$		$LM_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$	
			RLM	CBS	RLM	CBS
ISOLATED	M	61.7	66.8	72.5	74.2	78.9
	W	49.8	51.7	52.2	54.6	56.4
CONTINUOUS	M	50.8	56.0	64.1	67.3	71.4

training strategies (LM combined with SF, RLM and CBS) in the last phase of our incremental word learning framework generally improves the recognition results obtained in the previous phases of the system. $\text{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ (Sha and Saul 2007) with CBS provides the best recognition scores. When our system uses the $\text{LM}_{\boldsymbol{\mu}}$ (Li et al. 2005) algorithm as last step, which only updates the means of the GMMs, the complete framework obtains an improvement of about 70% in the test set containing isolated words uttered by male speakers. The joint update of the means and variances increases the relative improvement by about 5%. If we consider the whole system, this increment signifies almost 80% improvement for isolated digits uttered by men, more than 55% by women and over 70% in CSR with male speakers. Furthermore, the improvement of the large margin strategies is quite significant. In Fig. 6.8(c), the minimum and maximum values of the cross-validations, for the independent initialization and re-estimation phase on one hand and the subsequent LM DT refinement stage applying $\text{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ (Sha and Saul 2007) with CBS and SF on the other hand, do not almost overlap.

6.4 Summary and discussion[24]

In the previous stage of our incremental learning system, the parameters are computed using ML and MAP estimation approaches, which, although standard for this kind of task, do not directly maximize the separation between clusters (see Huang et al. 2001, Sec. 4.3, p. 150). The latter can be achieved using discriminative training techniques being then the last stage of our architecture a DT refinement stage to improve the classification scores.

In the last decades, numerous discriminative training methods have been proposed in the speech recognition community (see e.g. Juang et al. 1997 and McDermott 1997). Large

[24]Several paragraphs (some verbatim) are taken from previous works as Ayllón Clemente et al. 2010a and Ayllón Clemente et al. 2012.

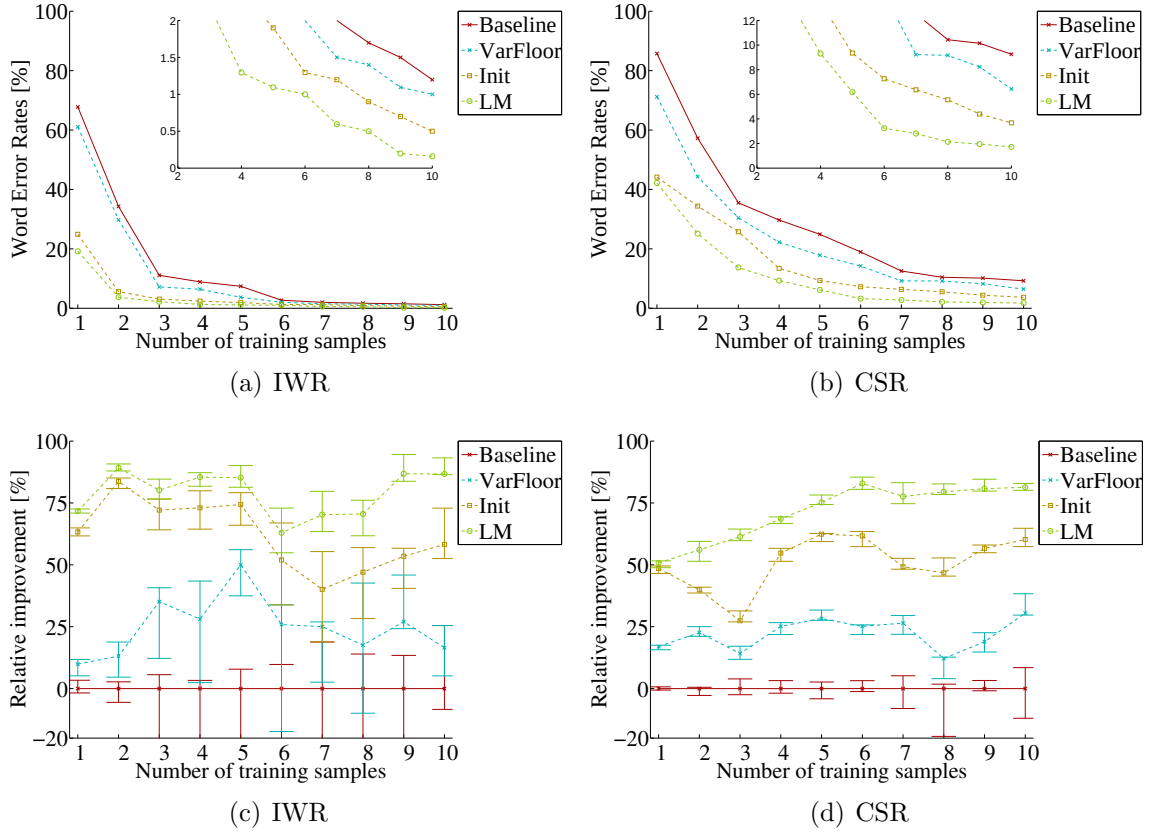


Figure 6.8: *Word error rates (a,b) and relative improvement (c,d) using the proposed methods in male speakers (see Ayllón Clemente et al. 2012). In Fig. (a) and (b), a zoom view is presented in order to distinguish the values of the different methods when the resolution of the scale axis is too coarse. The bars in Fig. (c) and (d) indicate the minimum and maximum values of the improvements of the 25 cross-validations. They target to measure a possible overlap of the improvements in each stage of our system. "VarFloor" stands for the $V_{\%}^{*}$, "Init" for the MSA jointly with the $V_{\%}^{*}$ and "LM" for the $LM_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ together with $V_{\%}^{*}$, MSA, SF and CBS. In the same way, "Baseline" stands for the baseline system described in Sec. 4.2.4.*

margin discriminative training is one of the most promising approaches at the moment, where large margin classifiers are characterized by good generalization power obtaining much lower recognition errors in unseen test data (Jiang et al. 2006). In the case of the SVMs, these are not an optimal approach for ASR systems (see Sec. 2.3.5, see Jiang et al. 2006). Thus, some researchers such as Li et al. 2005 and Sha and Saul 2007 studied different algorithms suitable for ASR systems based on the principle of the large margin. The common idea of these algorithms is to obtain decision boundaries, defined by the optimized HMM parameters, with the maximum classification margin (Jiang et al. 2006).

As we discussed in the chapter, in the context of efficient learning, the general state-of-the-art LM DT algorithms are not straightforward to apply. Hence, we investigated

several strategies to combine with the general approaches. The first strategy is the use of a scaling factor SF that forces the models to overlap. The other two approaches to apply jointly with SF are retraining the last model (RLM) and confidence based selection (CBS). In RLM, only the last model is updated in contrast to CBS where several models are re-estimated according to a confidence-like interval. Additionally, the state-of-the-art methods employed are the approach proposed by Li et al. 2005, where only the means of the GMMs are updated ($\text{LM}_{\boldsymbol{\mu}}$) and the one introduced by Sha and Saul 2007 that adapts the means and the variances of the GMMs ($\text{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$).

Although the system perfectly classifies the training data and the number of samples is relatively small, we were able to improve the recognition results obtained by ML/MAP estimation using our strategies. The combination of $\text{LM}_{\boldsymbol{\mu},\boldsymbol{\sigma}^2}$ (Sha and Saul 2007) and the CBS/SF strategy provided the best results in spite of the high computational cost (in contrast to $\text{LM}_{\boldsymbol{\mu}}$ (Li et al. 2005) and RLM[25]).

The improvements obtained in the large margin discriminative training stage are large and reliable for isolated word and continuous speech recognition as Fig. 6.8(c) and 6.8(d) show. Here, there are almost no overlaps between the improvements obtained in this stage and the results achieved only taking into account the new MSA initialization step. Furthermore, we demonstrated that the whole chain of processing steps and algorithms (generative and discriminative) provide more than 70% improvement (in male speakers) against the baseline system in speaker-independent and sparse learning conditions. This enables a significant reduction (approx. 50%, see Sec. 7.1) of the number of training samples without impairing the performance.

[25]The combination of the simplest methods can be used with 10% loss of performance.

7
Summary

As mentioned in Chapter 1, in cognitive robotics and artificial intelligence researchers aim to understand and model cognitive processes with the long-term goal of endowing robotic systems with more complex behavior and the ability of autonomous learning[1] (Romanelli 2010). Robotic systems have to acquire knowledge in uncontrolled conditions without the help of experts and transfer this knowledge to new situations in order to realize daily activities as well as to interact with different users (Romanelli 2010; Seabra Lopes 2002), e.g. to assist people demanding care (see Roger et al. 2012). A basic ability to achieve this goal is enabling the communication (written or verbal) between humans and machines (Bailey 1992). Due to the use of different dialects and vocabularies, it is crucial that artificial agents are able to learn new words of our own lexicon and terms coming from the shared experiences between the users and the agents (Iwahashi 2007). Language acquisition is a complex mental ability of humans, where such skills could be computationally modeled by means of ASR systems (ten Bosch et al. 2009). Despite the huge advances of ASR in the last decade, these systems are still far away of achieving human performance (Lippmann 1997; Furui 2009). In addition, no computational model can precisely reproduce language acquisition entirely based on data extracted from sensors until now (Boves et al. 2007).

In this thesis, we proposed an incremental word learning framework that can be applied in an interactive learning scenario with little prior knowledge and a small number of training samples based in our previous works (see Appendix D), where the usage of limited training data reduces the tutoring time required to learn a novel word, hence making the system more user-friendly. In our HMM framework with four main stages, several issues are addressed to cope with the limited number of training samples while maintaining a good generalization performance and assuring further attributes such as the ability to efficiently operate in a speaker-independent context (Ayllón Clemente et al. 2012):

- First, the parameters of the new word-model that enters the system are defined.

[1] Some paragraphs of this summary (including the following sections) are taken, sometimes verbatim, from Ayllón Clemente et al. 2012.

Each word is modeled by a Bakis or left-to-right HMM (without skip transitions) with continuous emission probabilities and dynamic number of hidden states. Although the amount of states can be determined according to the duration of the signals (see Sec. 4.1.2 for references), this number is established through the bootstrapping phase of our system. Related to the emission probabilities, Gaussian mixture models (GMMs) are employed. In this context and in order to avoid overfitting, a new manner of setting the variance floor scaled according to the number of training samples is presented and a very detailed comparison of our variance floor estimation with numerous techniques is realized (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012).

- The second phase is the initialization of the estimates, where we extended a supervised and unsupervised bootstrapping technique and introduced a novel multiple sequence alignment method, which configures the topology of the model (number of states and order of these) and initializes its parameters (Ayllón Clemente et al. 2010b; Ayllón Clemente et al. 2012). As in our previous works (see Appendix D), we compared our initialization algorithm to other state-of-the-art methods such as "flat start" and uniform segmentation (US) as well as to other type of alignments, namely BVA, $\text{MSA}_{\cup}$ and $\text{MSA}_{\cap}$.

- The next stage performs the re-estimation of the HMM parameters. Once the parameters of the system are initialized with our algorithm, these are recomputed using ML and MAP estimation with our new above mentioned variance floor estimation.

- Finally, we investigated large margin discriminative training to improve the generalization performance of the system in the last phase. Here, the GMMs of the word-models are optimized through different techniques, where the jointly adaptation of the means and variances using the method proposed by Sha and Saul 2007 and our SF and CBS strategies reported a very notable improvement (more than 70%[2]) over the whole system on the TIDigits database (Ayllón Clemente et al. 2010a; Ayllón Clemente et al. 2012). Additionally, all the above mentioned algorithms were not only tested using isolated digits, but also continuous speech (Ayllón Clemente et al. 2012).

In the next sections, we conclude this thesis with an outline of the highlights of the presented work enhancing the principal introduced approaches and their improvements, as well as an outlook giving some starting points for future research and the limitations of the framework pending to be overcome.

[2]The maximum improvement is when the test samples have been uttered by male speakers, which have also produced the samples that were employed during the training phase.

7.1 Achievements

The proposed system works under a determined supervision and operates in an efficient and incremental manner. The results of our thesis can be summarized as follows (including the works of Appendix D):

- Firstly, we defined an incremental learning framework, which owns capabilities required for the fast acquisition of new words with as little a priori knowledge as possible inspired by the children language acquisition process[3] (Ayllón Clemente et al. 2012). Additionally, our speaker-independent system is able to generalize to different speakers, also when they are not included in the training set. Moreover, although all the words were learned in isolation, they were recognized in continuous speech thereby yielding very promising results.

- Secondly, we proposed how to avoid overfitting under sparse learning conditions with the help of an adjustable variance floor threshold, which value depends on the training data available in each situation allowing that more exemplars of the same category could be properly recognized (first published in Ayllón Clemente et al. 2010a). Here, we realized a comparison between this dynamic threshold and several standard approaches. Our variance floor approach showed an average improvement over 10% against baseline and an improvement of almost 25% relative to the reference system when applied jointly with the variance floor percentile method.

- Next, we presented a novel multiple sequence alignment (MSA) method integrated in a successful bootstrapping technique, which accommodates supervised and unsupervised learning algorithms and merges the information contained in the different samples of the learning set (previously decoded) into a succession of HMM states determining the topology of the word-model to be learned and its initialization (first published in Ayllón Clemente et al. 2010b; Ayllón Clemente and Heckmann 2009). This approach was shown to be superior to other state-of-the-art methods on the investigated conditions. The improvement of the MSA method, in its full configuration, was over 30% against standard initializations and over 15% when competing with more sophisticated bootstrapping algorithms in a test set containing isolated words uttered by male speakers. In the evaluation of the recognition of continuous speech utterances, the MSA method outperforms all other approaches, although the improvements were more modest.

- In the last stage, the word-models were optimized using large margin discriminative training (LM DT) approaches. As a consequence of the very reduced number of training samples available and the high computational cost of state-of-the-art LM

[3]However, without aiming to model this process, see argumentation in Sec. 3.1, our goal is the construction of a technical application/system.

DT algorithms, we introduced three techniques or strategies to employ in these conditions: the introduction of a scaling factor (SF), a strategy called retraining the last model (RLM) and a third method based on confidence-like intervals (CBS) applied on discriminant functions (first published in Ayllón Clemente et al. 2010a). We employed these efficient strategies with two established LM DT algorithms: the one proposed by Li et al. 2005, where only the means of the GMMs are updated and the one that recalculates both means and variances via the technique from Sha and Saul 2007. The combination of updating the means and the variances with the help of standard LM DT approaches, the use of the scaling factor (SF) to force further optimizations and the approach based on the confidence-like intervals (CBS) achieved the best results. This final refinement phase, i.e. LM DT and our integration of SF and CBS, provided our system with an additional improvement of more than 15% relative to the previous stage when recognizing isolated words uttered by male speakers and over 20% in the case of recognizing continuous speech utterances.

- Finally, the introduction of the proposed approaches (an adaptable and dynamic variable floor, an optimized initialization of the models via the integration of a multiple sequence alignment method and the strategic use of several techniques to efficiently transform the GMMs of the HMMs through large margin discriminative training) provides our framework with a global improvement of 70-80% against a baseline system (both evaluated[4] on the TIDigits database[5]) allowing a considerable reduction of the number of training samples while maintaining a good performance (Ayllón Clemente et al. 2012):

  - In the case of isolated word recognition, the reduction of the number of training samples can go from 10 (1.2% WER) to 5 (1.1% WER) exemplars.
  - The results related to the recognition of continuous speech utterances were also quite promising. We reported 1.7% WER, when only 10 training samples were employed in contrast to the 9.3% WER achieved by the baseline system and the 0.5% WER obtained when all training data available (isolated words and continuous speech utterances) are used.

Therefore, we conclude that the presented approaches and algorithms make possible to decrease the tutoring time notably and consequently, to achieve a more user-friendly

[4]The discussion about the evaluation and improvements against other competing state-of-the-art methods and other similar formulations in each phase in order to highlight the advantages and disadvantages of our work have been handled respectively in Chapter 4, 5 and 6.

[5]It is quite difficult to compare our framework against commercial systems without access to the source code or the parameterization of these systems for an equitable evaluation. Additionally, as we do not aim at modeling the process of language acquisition in children and only take inspiration from this process to build our technical application, we do not compare ourselves with systems modeling it.

system without impairing the generalization performance and representing these achievements a substantial advance towards efficient language acquisition, although humans still overcome all existing technological approaches.

7.2 Outlook and limitations of the system

The main purpose of our work was to investigate, design, develop and evaluate a system that is endowed with the necessary features for efficient incremental word learning when aiming at an interactive scenario. Due to the very extended nature of the referred problem and despite the successful and promising results presented through this thesis, there are related scientific topics and system limitations that are worthy for additional investigation and experimentation. Thus, these topics should be also addressed to reach our long-term goal: the natural speech acquisition in artificial agents to achieve a user-friendly interaction between humans and machines.

Firstly, as stated in our previous work Ayllón Clemente et al. 2012, it is necessary to integrate the system in a natural environment, where the terms are acquired directly from continuous speech utterances. One major limitation of our system is the restriction of learning only new terms in isolation. To be able to learn from continuous speech utterances, the system must deal with the segmentation problem (see Huang et al. 2001, Sec. 2.3.2, p. 53) mentioned in Chapters 1 and 2[6]:

- The most comfortable way of word segmentation for the users or tutors is unsupervised (without effort) in spite of being still an open question that authors aim to answer by investigating different techniques to circumvent this weakness (e.g. Aimetti et al. 2010).
- One possibility is to use the already learned words in isolation to further segment the speech stream similar to infants (see Ambridge and Lieven 2011, Sec. 2.4.1.1.1, p. 32). It is normally simpler if the system knows the position of the word to segment as in the case of the so called presentational constructions in CDS, e.g. "It's a X", "There's a X" (in the case of nouns in English, Tomasello 2003, Sec. 3.1.1, p. 49).
- Some experiments have shown that CDS prosodic characteristics ease infants the detection of word boundaries (see e.g. Thiessen et al. 2005).
- One more method for the segmentation of the important words to learn is using the synchrony between word uttering and movement of the tutors (see e.g. Schillingmann et al. 2011; Rolf et al. 2009).

Secondly, our system is not able to determine if a word has already entered the system when the same one is presented but with a different label. In particular, this deficiency

[6]Enumerated as not in the scope for our system in Sec. 1.1.

would provoke problems in the large margin discriminative training stage, as this stage would try to separate the similar models. The drawback would be solved if our algorithms could differentiate (independent of the label presented) already known words from unknown ones during the training phase. Otherwise, the detection of out-of-vocabulary (OOV) words can increase the learning and the performance rate (see Huang et al. 2001, Sec. 11.6.1, p. 572). A review about out-of-vocabulary words, keyword spotters and confidence measures can be found in Jiang 2005.

Additionally, our incremental word learning system is not integrated in an interactive learning scenario yet, which is the long-term goal. The main focus of this thesis was the investigation and implementation of an efficient framework to reduce the needed tutoring time to learn new words. Hence, the next step is to incorporate our proposed system in an artificial agent able to interact with a human tutor. This integration would also experimentally[7] demonstrate the advantage of having such efficient incremental word learning framework. Moreover, one should not forget that language is not only passive or exclusively based on the statistical information that can be extracted from speech (Kuhl 2007; e.g. Huckvale et al. 2009). In this context, the user can help the artificial agent to acquire words by supplying non-verbal information such as visual cues, which can be learned at the same time, e.g. Steels and Kaplan 2001, Roy and Pentland 2002 as well as Iwahashi 2007. The latter employed additionally touch sensors (slapping the robot hand) as corrective feedback. However, not only the tutors are able to give feedback, e.g. some special robots can provide us with face gestures as feedback or confidence signals (see Lang et al. 2010). An artificial agent, which is able to produce sentences, could also provide the user or tutor with a hint of the previously learned words, e.g. Iwahashi 2007.

Furthermore, our system presents the drawback of requiring all the samples for the discriminative training phase, hence it is needed to select the most representative and rich acoustic data in order to reduce the necessary storage space for the samples as we argued in Ayllón Clemente et al. 2012. In this direction, some researchers such as Wu et al. 2007, Lin and Bilmes 2011 or Isogai and Mizuno 2010 have started to investigate this kind of task without still solution when only few training samples are available.

Finally, it would be very useful to introduce and decode context information in our system (see Huang et al. 2001, Ch. 17, p. 836). People understand speech even in noisy environments due to the help of context information (Reddy 2001; Ligorio et al. 2010). Related to the noisy environments[8], a system being part of a moving artificial agent will be often working under ego-noisy conditions (see Ince et al. 2010). Hence, the incorporation of different techniques for building a robust system against noises is very advisable as these agents are very sensitive to noise compared to humans as investigated by several authors such as Lippmann 1997, Allen 1994 and Sroka and Braida 2005.

[7] In future evaluations, the benchmarks should be also extended to databases with natural/spontaneous speech, although they are very difficult to obtain.

[8] Dealing with noise was not in the scope of the thesis, see Sec. 1.1.

In conclusion, all the above mentioned limitations/themes for future research must be taken into account for the fulfillment of our long-term goal, where the promising results of our efficient incremental learning framework motivates the further investigation in this field. During the redaction of this thesis, several publications, e.g. Sun 2012 and Ons et al. 2014, have appeared with new approaches related to the efficient incremental framework of our previous works. The author expects that this community will grow in the next years bringing us closer to our shared goal.

A
Speech databases

There are several annotated corpora/databases available. Two of them that have been employed during several decades as standard benchmarks for the evaluation of speech recognition algorithms, are TIMIT (Garofolo et al. 1993) and TIDigits (Leonard and Doddington 1993) databases. We describe both in the following sections.

A.1 TIMIT

The TIMIT (Texas Instruments (TI) and Massachusetts Institute of Technology (MIT)) database was designed by TI, MIT and Stanford Research Institute (SRI) and developed to supply speech utterances in order to build English acoustic-phonetic models as well as to develop and evaluate the ASR systems (Garofolo et al. 1993). All the sentences were taped in clean environments using a Sennheiser close-talking microphone digitized at 20 kHz and then digitally filtered and downsampled to 16 kHz (Lamel et al. 1989).

Table A.1: *TIMIT speech material (Garofolo et al. 1993).*

Sentence Type	Nr. Sentences	Nr. Speakers per sentence	Total	Nr. Sentences per speaker
Dialect (SA)	2	630	1260	2
Compact (SX)	450	7	3150	5
Diverse (SI)	1890	1	1890	3
Total:	2342		6300	10

The corpus TIMIT is composed of a total of 6300 phrases (dialect sentences, phonetically compact phrases and phonetically diverse utterances) comprising 10 sentences uttered by 630 speakers (male and female speakers) from 8 dialect regions of USA (Garofolo et al. 1993). It is recommended to use 70-80% of the corpus for training and the rest (20-30%) for the evaluation of the system (Garofolo et al. 1993).

A.2 TIDigits

The TIDigits database was recorded to be applied in the design and evaluation of approaches for speaker-independent recognition of successions of numbers (Leonard 1984). The corpus vocabulary consists on the digits: “1” to “9”, plus “oh” and “zero”, so a total of 11 words (Leonard and Doddington 1993). The database contains a total of 326 speakers and more than 25000 utterances, which comprise between one and seven digits (Leonard and Doddington 1993).

Table A.2: *TIDigits description of speakers: number and age ranges (Leonard and Doddington 1993).*

Category	Number	Age range (years)
Man	111	21-70
Woman	114	17-59
Boy	50	6-14
Girl	51	8-15

The adult speakers were chosen so that there were minimum 5 male and 5 female speakers using different American English dialects (precisely 21) as well as 5 black male and 6 black female speakers were included (Leonard and Doddington 1993). Each speaker uttered 77 sequences, which comprise the following types: 22 isolated digits and more than 50 continuous digits sequences, so that from each speaker we got 253 numbers as well as 176 digit transitions (Leonard and Doddington 1993).

The recording was performed in a quiet environment and several human listening experiments were realized to yield certification of the transcription of the succession of the digits as well as to get knowledge about the speech recognition performance in humans and the implicit recognizability of the samples (Leonard and Doddington 1993).

B

Hidden Markov model toolkit

Currently, there is a wide range of HMM toolboxes employed by the speech recognition community such as Julius (e.g. Lee and Kawahara 2009), Esmeralda (e.g. Fink 1999) or Sphinx (e.g. Seymore et al. 1998); however one of the most used toolboxes (also employed in this work) is HTK (Young et al. 2009).

The hidden Markov model toolkit (HTK) is a compact toolkit for constructing and handling HMMs especially designed for speech recognition research but its application can be extended to many other uses such as symbol recognition, speech synthesis or any kind of sequential modeling (Woodland 2009). HTK comprises a kit of libraries and tools programmed in C language, which offer advanced commands for digital processing, training, testing, analysis as well as building DHMMs and CDHMMs to construct sophisticated frameworks (Woodland 2009).

C
Gradient descent algorithm

The gradient descent is one of the most extensively applied algorithms for solving optimization problems, where the target is to iteratively minimize a differentiable objective function $f(\boldsymbol{x})$ (Theodoridis 2015, Sec. 5.2, p. 163).

The algorithm begins with a first guess of the minimum point $\boldsymbol{x}_0$ and the next iterations are like (Theodoridis and Koutroumbas 2003, Sec. C.1, p. 659):

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} - \kappa \bigtriangledown f(\boldsymbol{x}) \tag{C.1}$$

where $\kappa > 0$ and $\bigtriangledown f(\boldsymbol{x})$ is the gradient of the objective function $f(\boldsymbol{x})$. If we are looking for a maximum, the approach is called gradient ascent and the minus sign in the equation turns to be a plus sign (Theodoridis and Koutroumbas 2003, Sec. C.1, p. 659). According to the Eq. C.1, the new value $\boldsymbol{x}_{new}$ is estimated so that the value of the objective function $f(\boldsymbol{x})$ decreases, whose convergence depends on (following issues are described in Theodoridis and Koutroumbas 2003, Sec. C.1, p. 659):

- Initialization

  The approach usually converges to a local minimum, a global minimum or a saddle point of $f(\boldsymbol{x})$, i.e. to a point where $\bigtriangledown f(\boldsymbol{x}) = 0$. The initial conditions $\boldsymbol{x}_0$ determine to which point the algorithm arrives.

- Step size κ

  Depending on this value, the corrections are small or large, influencing the convergence of the procedure.

A solution to find a step size that diminishes the objective function as much as possible is to employ "line search" (see Nocedal and Wright 1999, Sec. 2.2, p. 19). In this context, one of the most efficient techniques is the quasi-Newton method, which only needs that the gradient of the objective function $f(\boldsymbol{x})$ is provided for each iteration similar to the gradient descent method and in contrast to Newton's methods, they do not need second derivatives (Nocedal and Wright 1999, Ch. 8, pp. 193-194).

The most well-known quasi-Newton algorithm is the BFGS method, which receives its name from its finders Broyden, Fletcher, Goldfarb and Shanno (Nocedal and Wright 1999, Sec. 8.1, p. 194). For optimization problems with a large number of variables (parameters), an extension of the BFGS, the limited-memory method L-BFGS, is well-suited due to its modest memory/storage requirements (Nocedal and Wright 1999, Sec. 9.1, p. 224).

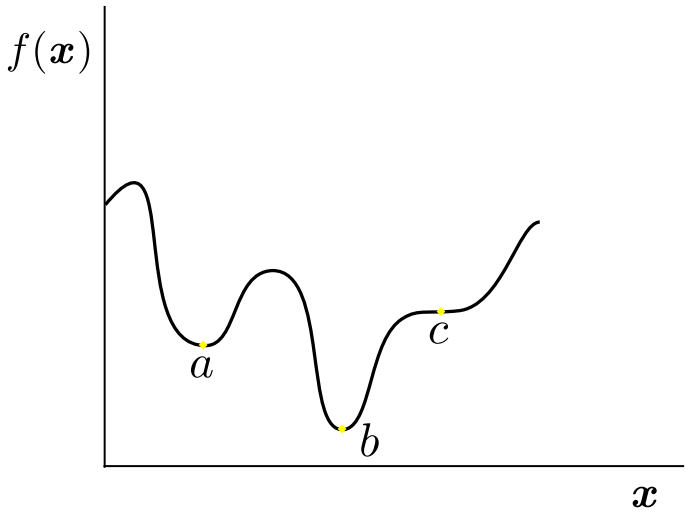


Figure C.1: *Illustration of examples of a local minimum (a), a global minimum (b) and a saddle point (c) of the objective function $f(\boldsymbol{x})$, Theodoritis and Koutroumbas, C. 1, p. 660.*

D

List of relevant publications by the author

Ayllón Clemente, I. and M. Heckmann (2009). Automatic speech recognition system integrating multiple sequence alignment for model bootstrapping; App. No.: EP 09171957.5; Date of publication: 13.04.2011; Bulletin 2011/15; EP 2309487 A1; Status: Published.

Ayllón Clemente, I., M. Heckmann, A. Denecke, B. Wrede, and C. Görick (2010a). Incremental word learning using large-margin discriminative training and variance floor estimation. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 889-892. Paper ID: 79597.

Ayllón Clemente, I., M. Heckmann, G. Sagerer, and F. Joublin (2010b). Multiple sequence alignment based bootstrapping for improved incremental word learning. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*. Doi: 10.1109/ICASSP.2010.5494990.

Ayllón Clemente, I., M. Heckmann, and B. Wrede (2012). Incremental word learning: efficient HMM initialization and large margin discriminative adaptation. *Speech Communication 54* (9), 1029-1048. Doi: 10.1016/j.specom.2012.04.005.

Bibliography

ACORNS Project (2011). Documents: Acquisition of Communication and Recognition Skills. `lands.let.ru.nl/acorns/documents/index.html`. Accessed: 18.08.2015.

Aimetti, G., R. K. Moore, and L. ten Bosch (2010). Discovering an optimal set of minimally contrasting acoustic speech units: a point of focus for whole-word pattern matching. In *Proceedings of the International Conference on Spoken Language Processing*.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE© Transactions on Automatic Control 19*(6), 716–723.

Allen, J. B. (1994). How do humans process and recognize speech? *IEEE© Transactions on Speech and Audio Processing 2*(4), 567–577.

Alshawi, H. (2003). Effective utterance classification with unsupervised phonotactic models. In *Proceedings of the Conference of the North American Association for Computational Linguistics on Human Language Technology*.

Ambridge, B. and E. V. M. Lieven (2011). *Child Language Acquisition. Contrasting Theoretical Approaches*. Cambridge University Press.

Arenas-Garcia, J. and F. Perez-Cruz (2003). Multiclass support vector machines: a new approach. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Arora, S. and B. Barak (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press.

ASIMO (2015). The Honda Humanoid Robot ASIMO. `http://world.honda.com/ASIMO/new/`. Accessed: 12.04.2015.

ASIMO Gallery (2015). ASIMO official website - Gallery of pictures of ASIMO. `http://www.asimo.honda.com/gallery/`. Accessed: 20.04.2015.

Aslin, R. N., J. Z. Woodward, N. P. LaMendola, and T. G. Bever (1996). Models of word segmentation in fluent maternal speech to infants. In *Signal to Syntax: Bootstrapping from speech to grammar in early acquisition*.

Asthana, A. N. and S. Khorana (2013). Unlearning machine learning: the challenge of integrating research in business applications. *Middle-East Journal of Scientific Research 15*(2), 266–271.

Austermann, A., S. Yamada, K. Funakoshi, and M. Nakano (2010). Learning naturally spoken commands for a robot. In *Proceedings of the International Conference on Spoken Language Processing*.

Ayllón Clemente, I. and M. Heckmann (2009). Automatic speech recognition system integrating multiple sequence alignment for model bootstrapping; App. No.: EP 09171957.5; Date of publication: 13.04.11; Bulletin 2011/15; EP 2309487 A1; Status: Published.

Ayllón Clemente, I., M. Heckmann, A. Denecke, B. Wrede, and C. Görick (2010a). Incremental word learning using large-margin discriminative training and variance floor estimation. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 889–892. Paper ID: 79597.

Ayllón Clemente, I., M. Heckmann, G. Sagerer, and F. Joublin (2010b). Multiple sequence alignment based bootstrapping for improved incremental word learning. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*. Doi: 10.1109/ICASSP.2010.5494990.

Ayllón Clemente, I., M. Heckmann, and B. Wrede (2012). Incremental word learning: efficient HMM initialization and large margin discriminative adaptation. *Speech Communication 54*(9), 1029–1048. Doi: 10.1016/j.specom.2012.04.005.

Bailey, C. W. (1992). Truly intelligent computers. In *Thinking Robots. An Aware Internet, and Cyberpunk Librarians. Library and Information Technology Association, 99–104*.

Bakis, R. (1976). Continuous speech word recognition via centisecond acoustic states. In *Proceedings of the Acoustical Society of America Meeting*.

Baldi, P. and Y. Chauvin (1994). Smooth on-line learning algorithms for hidden Markov models. *Neural Computation 6*(2), 307–318.

Bardideh, M., F. Razzazi, and H. Ghassemian (2007). An SVM based confidence measure for continuous speech recognition. In *Proceedings of the IEEE© International Conference on Signal Processing and Communications*.

Bates, E., D. Thal, B. L. Finlay, and B. Clancy (2002). Early language development and its neural correlates. In *Handbook of Neuropsychology, Vol. 6, Child Neurology*.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics 41*(1), 164–171.

Bazzi, I. and D. Katabi (2000). Using support vector machines for spoken digit recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Benesty, J., M. M. Sondhi, and Y. Huang (2008). *Springer Handbook of Speech Processing*. Springer.

Benzeghiba, M., R. DeMori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens (2007). Automatic speech recognition and speech variability: a review. *Speech Communication 49*(10,11), 763–786.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Biadsy, F., P. J. Moreno, and M. Jansche (2012). Google's cross-dialect Arabic voice search. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Bilmes, J. A. (2006). What hmms can do. *IEICE - Transactions on Information and Systems E89-D*(3), 869–891.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bortfeld, H., J. L. Morgan, R. M. Golinkoff, and K. Rathbun (2005). Mommy and me: familiar names help launch babies into speech stream segmentation. *Psychological Science 16*(4), 298–304.

Boves, L., L. ten Bosch, and R. K. Moore (2007). ACORNS - Towards computational modeling of communication and recognition skills. In *Proceedings of the IEEE© International Conference on Cognitive Informatics*.

Brandl, H. (2009). *A computational model for unsupervised childlike speech acquisition*. Ph. D. thesis, Bielefeld University.

Brandl, H., B. Wrede, F. Joublin, and C. Görick (2008). A self-referential childlike model to acquire phones, syllables and words from acoustic speech. In *Proceedings of the IEEE© International Conference on Development and Learning*.

Brent, M. R. and J. M. Siskind (2001). The role of exposure to isolated words in early vocabulary development. *Cognition 81*(2), 33 – 44.

Broen, P. A. (1972). *The verbal environment of the language-learning child*. American Speech and Hearing Association Monographs, No. 17.

Bronzaft, A. L. (1997). Beware: noise is hazardous to our children's development. *Hearing Rehabilitation Quarterly 22*(1).

Brookes, M. (2005). *Voicebox: Speech processing toolbox for Matlab*.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, 121–167.

Burrows, T. L. and M. Niranjan (1994). The use of recurrent neural networks for classification. In *Proceedings of the IEEE© Workshop on Neural Networks for Signal Processing*.

Carey, S. and E. Bartlett (1978). Acquiring a single new word. *Papers and reports on Child Language Development 15*, 17–29.

Chang, T.-H., Z.-Q. Luo, L. Deng, and C.-Y. Chi (2008). A convex optimization method for joint mean and variance parameter estimation of large-margin CDHMM. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Chaudhuri, S., M. Harvilla, and B. Raj (2011). Unsupervised learning of acoustic unit descriptors for audio content representation and classification. In *Proceedings of the International Conference on Spoken Language Processing*.

Cheshire, J. (2005). Age and generation-specific use of language. In *Sociolinguistics: An Introductory Handbook of the Science of Language and Society, Mouton de Gruyter, 1552–1563*.

Childers, J. B. and M. Tomasello (2002). Two-year-olds learn novel nouns, verbs, and conventional actions from massed or distributed exposures. *Developmental Psychology 38*(6), 967–968.

Chow, Y.-L. and R. Schwartz (1989). The N-Best algorithm: an efficient procedure for finding top N sentence hypotheses. In *Proceedings of the DARPA Speech and Natural Language Workshop*.

Chowdhury, N., M. A. Sattar, and A. K. Bishwas (2009). An efficient algorithm Bangla speech separation. In *Proceedings of the IEEE© International Conference on Computer Sciences and Convergence Information Technology*.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297.

Crammer, K. and Y. Singer (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research 2*, 265–292.

Cristianini, N. and J. Shawe-Taylor (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human-robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences 362*(1480), 679–704.

Dautenhahn, K. and B. Robins (2015). The Aurora Project. `http://www.aurora-project.com`. Accessed: 18.04.2015.

Dautenhahn, K., S. Woods, C. Kaouri, M. Walters, K. Koay, and I. Werry (2005). What is a robot companion - friend, assistant or butler? In *Proceedings of the IEEE© International Conference on Intelligent Robots and Systems*.

Davis, S. and P. Mermelstein (1980). Comparison of parametric representations for

monosyllabic word recognition in continuously spoken sentences. *IEEE© Transactions on Acoustics, Speech and Signal Processing 28*(4), 357–366.

de Boysson-Bardies, B. (1999). *How Language Comes to Children: From Birth to Two Years*. The MIT Press.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithms. *Journal of the Royal Statistical Society, Series B 39*(1), 1–38.

Deng, L., M. Aksmanovic, X. Sun, and C. F. J. Wu (1994). Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. *IEEE© Transactions on Speech and Audio Processing 2*(4), 507–520.

Dominey, P. F. and C. Dodane (2004). Indeterminacy in language acquisition: the role of child directed speech and joint attention. *Journal of Neurolinguistics 17*(2,3), 121–145.

Domont, X. (2009). *Hierarchical spectro-temporal features for robust speech recognition*. Ph. D. thesis, Darmstadt University of Technology.

Dragon speech recognition software (2015). Speech recognition technologies in Nuance. `http://www.nuance.co.uk/dragon/index.htm`. Accessed: 16.06.2015.

Eimas, P. D. (1975). Auditory and phonetics coding of the cues for speech: discrimination of the /r/-/l/ distinction by young infants. *Perception and Psychophysics 18*(5), 341–347.

Fernald, A. (1985). Four-month-old infants prefer to listen to motherese. *Infant Behavior and Development 8*(2), 181–195.

Fernald, A. and C. Mazzie (1991). Prosody and focus in speech to infants and adults. *Developmental Psychology 27*(2), 209–221.

Fernald, A. and T. Simon (1984). Expanded intonation contours in mothers' speech to newborns. *Developmental Psychology 20*(1), 104–113.

Fernández, J. L., D. P. Losada, and R. Sanz (2008). Enhancing building security systems with autonomous robots. In *Proceedings of the IEEE© International Conference on Technologies for Practical Robot Applications*.

Fink, G. A. (1999). Developing HMM-based recognizers with Esmeralda. *Text, Speech and Dialogue; Lectures Notes in Artificial Intelligence 1692*, 229–234.

Fink, G. A. (2008). *Markov models for pattern recognition: from theory to applications*. Springer.

Friederici, A. D. and J. M. Wessels (1993). Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception and Psychophysics 54*(3), 287–295.

Fritzke, B. (1998). *Vektorbasierte Neuronale Netze*. Habilitation Treatise. Friedrich-Alexander-Universität Erlangen-Nürnberg.

Fujimura, H., T. Masuko, and M. Tachimori (2010). A duration modeling technique with incremental speech rate normalization. In *Proceedings of the International Conference on Spoken Language Processing*.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE© Transactions on Acoustics, Speech, and Signal Processing 29*(2), 254–272.

Furui, S. (2009). Selected topics from 40 years of research on speech and speaker recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Gales, M. J. F. (1999). Semi-tied covariance matrices for hidden Markov models. *IEEE© Transactions on Speech and Audio Processing 7*(3), 272–281.

Ganapathiraju, A., J. Hamaker, and J. Picone (2000). Hybrid SVM/HMM architectures for speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Ganapathiraju, A., J. Hamaker, and J. Picone (2004a). Applications of support vector machines to speech recognition. *IEEE© Transactions on Signal Processing 52*(8), 2348–2355.

Ganapathiraju, A., J. Hamaker, and J. Picone (2004b). Applications of support vector machines to speech recognition. *IEEE© Transactions on Signal Processing 52*(8), 2348–2355.

Garofolo, J. S., L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren (1993). *The DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM, Readme file*. Linguistic data consortium. `catalog.ldc.upenn.edu/docs/LDC93S1/timit.readme.html`. Accessed: 25.08.2015.

Gauvain, J.-L. and C.-H. Lee (1991). Bayesian learning of Gaussian mixture densities for hidden Markov models. In *Proceedings of the DARPA Speech and Natural Language Workshop*.

Geiger, J., J. Schenk, F. Wallhoff, and G. Rigoll (2010). Optimizing the number of states for HMM-based on-line handwritten whiteboard recognition. In *Proceedings of the International Conference on Frontiers in Handwriting Recognition*.

Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence 15*(1), 9–42.

Ghoshal, A., P. Ircing, and S. Khudanpur (2005). Hidden Markov models for automatic annotation and content-based retrieval of images and video. In *Proceedings of the International Conference on Research and Development in Information Retrieval*.

Gish, H. and K. Ng (1993). A segmental speech model with applications to word spotting. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Goldin-Meadow, S., M. E. P. Seligman, and R. Gelman (1976). Language in the two-year old: receptive and productive stages. *Cognition 4*(2), 189–202.

Gollan, C. and H. Ney (2008). Towards automatic learning in LVCSR: rapid development of a Persian broadcast transcription system. In *Proceedings of the International Conference on Spoken Language Processing*.

Goodsitt, J. V., J. L. Morgan, and P. K. Kuhl (1993). Perceptual strategies in prelingual speech segmentation. *Journal of Child Language 20*(2), 229–252.

Google Glass (2015). Voice actions in Google Glass - Speech recognition technologies in Google. `https://support.google.com/glass/answer/3079305?hl=en`. Accessed: 17.06.2015.

Gorin, A. L., D. Petrovksa-Delacretaz, G. Riccardi, and J. H. Wright (1999). Learning spoken language without transcriptions. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*.

Grieser, D. L. and P. K. Kuhl (1988). Maternal speech to infants in a tonal language: support for universal prosodic features in motherese. *Developmental Psychology 24*(1), 14–20.

Gupta, A. K. and S. K. Arora (2007). *Industrial Automation and Robotics*. Laxmi Publications (P) Ltd.

Hanson, B. and T. H. Applebaum (1990). Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with Lombard and noisy speech. In *Proceedings of the IEEE© International Conference on Acoustics, Speech, and Signal Processing*.

He, X., L. Deng, and W. Chou (2008). Discriminative learning in sequential pattern recognition - a unifying review for optimization-oriented speech recognition. *IEEE© Signal Processing Magazine 25*(5), 14–36.

Henikoff, S. and J. G. Henikoff (1992). Amino acid substitution matrices from protein blocks. *National Academy of Sciences of the United States of America 89*(22), 10915–10919.

Hennebert, J., C. Ris, H. Bourlard, S. Renals, and N. Morgan (1997). Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. In *Proceedings of the European Conference on Speech Communication and Technology*.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America 87*(4), 1738–1752.

Hermansky, H. (2011). Speech recognition from spectral dynamics. *Sadhana 36*(5), 729–744.

Hermansky, H. and N. Morgan (1994). RASTA processing of speech. *IEEE© Transactions on Speech and Audio Processing 2*(4), 578–589.

Hinton, G., L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury (2012). Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine 29*, 82–97.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence 40*(1,3), 185–234.

Ho, C.-C., K. F. MacDorman, and Z. A. D. Pramono (2008). Human emotion and the uncanny valley: a glm, mds, and isomap analysis of robot video ratings. In *Proceedings of the ACM/IEEE© International Conference on Human Robot Interaction*.

Hörnstein, J. and J. Santos-Victor (2010). Learning words and speech units through natural interactions. In *Proceedings of the International Conference on Spoken Language Processing*.

Houston, D. M., P. W. Jusczyk, and J. Tager (1998). Talker-specificity and persistence of infants' word representations. In *Proceedings of the Annual Boston University Conference on Language Development*.

Huang, S., X. Xie, and P. Fung (2008). Using output probability distribution for OOV word rejection. In *Proceedings of the IEEE© Spoken Language Technology Workshop*.

Huang, X., A. Acero, and H.-W. Hon (2001). *Spoken language processing: a guide to theory, algorithm and system development*. Prentice Hall.

Huckvale, M., I. S. Howard, and S. Fagel (2009). KLAIR: a virtual infant for spoken language acquisition research. In *Proceedings of the International Conference on Spoken Language Processing*.

iCub (2015). An open source cognitive humanoid robotic platform. `http://www.icub.org`. Accessed: 12.04.2015.

Ince, G., K. Nakadai, T. Rodemann, H. Tsujino, and J.-i. Imura (2010). A robust speech recognition system against the ego noise of a robot. In *Proceedings of the International Conference on Spoken Language Processing*.

Isogai, M. and H. Mizuno (2010). Speech database reduction method for corpus-based TTS system. In *Proceedings of the International Conference on Spoken Language Processing*.

Itaya, Y., H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura (2005). Deterministic annealing EM algorithm in acoustic modeling for speaker and speech recognition. *IEICE Transactions on Information and Systems E88-D*(3), 425–431.

Iwahashi, N. (2007). Robots that learn language: a developmental approach to situated human-robot conversations. In *Sankar, N. (ed.) Human-Robot Interaction, I-Tech Education and Publishing, 95-118*.

Jebara, T. (2004). *Machine learning: discriminative and generative*. Kluwer Academic Publishers.

Jelinek, F. (1997). Statistical methods for speech recognition. The MIT Press.

Jiang, H. (2005). Confidence measures for speech recognition: a survey. *Speech Communication 45*(4), 455–470.

Jiang, H., K. Hirose, and Q. Huo (1999). Robust speech recognition based on a Bayesian prediction approach. *IEEE© Transactions on Speech and Audio Processing 7*(4), 426–440.

Jiang, H., X. Li, , and C. Liu (2006). Large margin hidden Markov models for speech recognition. *IEEE© Transactions on Audio, Speech and Language Processing 14*(5), 1584–1595.

Jianjun, Z. and N. Xingfang (2007). A novel out-of-vocabulary rejection method for isolated word recognition. In *Proceedings of the Conference on Wireless, Mobile and Sensor Networks*.

Jokusch, J. and H. Ritter (1999). An instantaneous topological mapping model for correlated stimuli. In *Proceedings of International Joint Conference on Neural Networks*.

Jourani, R., K. Daoudi, R. André-Obrecht, and D. Aboutajdine (2010). Large margin Gaussian mixture models for speaker identification. In *Proceedings of the International Conference on Spoken Language Processing*.

Juang, B. H. (1985). Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal 64*(6), 1235–1249.

Juang, B. H. and T. Chen (1998). The past, present, and future of speech processing. *IEEE© Signal Processing Magazine 15*(3), 24–48.

Juang, B. H., W. Hou, and C. H. Lee (1997). Minimum classification error rate methods for speech recognition. *IEEE© Transactions on Speech and Audio Processing 5*(3), 257–265.

Juang, B. H. and S. Katagiri (1992). Discriminative learning for minimum error classification (pattern recognition). *IEEE© Transactions on Signal Processing 40*(12), 3043–3054.

Juang, B. H., S. E. Levinson, and M. M. Sondhi (1986). Maximum-likelihood estimation for multivariate mixture observations of Markov chains. *IEEE© Transactions on Information Theory 32*(2), 307–309.

Jusczyk, P. W. and R. N. Aslin (1995). Infants' detection of the sound patterns of words in fluent speech. *Cognitive Psychology 29*(1), 1–23.

Jusczyk, P. W., E. A. Hohne, and A. Bauman (1999a). Infants' sensitivity to allophonic cues for word segmentation. *Perception and Psychophysics 61*(8), 1465–1476.

Jusczyk, P. W., D. M. Houston, and M. Newsome (1999b). The beginnings of word segmentation in English-learning infants. *Cognitive Psychology 39*(3), 159–207.

Kagan, J. and M. Lewis (1965). Studies of attention in the human infant. *Merill-Palmer Quaterly 11*(2), 95–127.

Kapadia, S. (1998). *Discriminative training of hidden Markov models*. Ph. D. thesis, University of Cambridge.

Kemp, T. and A. Waibel (1999). Unsupervised training of a speech recognizer: recent experiments. In *Proceedings of the European Conference on Speech Communication and Technology*.

Kit, C. (2003). How does lexical acquisition begin? A cognitive perspective. *Cognitive Science 1*(1), 1–50.

Kubota, N. and T. Mori (2009). Conversation system based on Boltzmann selection and Bayesian networks for a partner robot. In *Proceedings of the IEEE© International Symposium on Robots and Human Interactive Communications*.

Kuhl, P. K. (2007). Cracking the speech code: how infants learn language. *Acoustical Science and Technology 28*(2), 71–83.

Kupferberg, A., S. Glasauer, M. Huber, M. Rickert, A. Knoll, and T. Brandt (2011). Biological movement increases acceptance of humanoid robots as human partners in motor interaction. *AI and Society 26*(4), 339–345.

Kurzweil, R. (2005). *The Singularity is near*. Viking Press.

Kuznetsov, I. and M. McDuffie (2015). PR2ALIGN: a stand-alone software program and a web-server for protein sequence alignment using weighted biochemical properties of amino acids. *BMC Research Notes 8:187*.

Lacerda, F., E. Marklund, L. Lagerkvist, L. Gustavsson, E. Klintfors, and U. Sundberg (2004). On the linguistic implications of context-bound adult-infant interactions. In *Proceedings of the International Workshop on Epigenetic Robotics*.

Lamel, L., J.-L. Gauvain, and G. Adda (2001). Investigating lightly supervised acoustic model training. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Lamel, L., J.-L. Gauvain, and G. Adda (2002a). Lightly supervised and unsupervised acoustic model training. *Computer Speech and Language 16*(1), 115–129.

Lamel, L., J.-L. Gauvain, and G. Adda (2002b). Unsupervised acoustic model training. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Lamel, L. F., R. Kassel, and S. Seneff (1989). Speech database development: design and analysis of the acoustic-phonetic corpus. In *ISCA Tutorial and Research Workshop on Speech Input/Output Assessment and Speech Databases*.

Lang, C., S. Wachsmuth, H. Wersing, and M. Hanheide (2010). Facial expressions as feedback cue in human-robot interaction - a comparison between human and automatic recognition performances. In *Proceedings of the Workshop on Human Communicative Behavior Analysis*.

Lasserre, J. A., C. M. Bishop, and T. Minka (2006). Principled hybrids of generative and discriminative models. In *Proceedings of the IEEE© Computer Society Conference on Computer Vision and Pattern Recognition*.

Lee, A. and T. Kawahara (2009). Recent development of open-source speech recognition engine Julius. In *Proceedings of the Annual Summit and Conference of the Asia-Pacific Signal and Information Processing Association*.

Leonard, R. (1984). A database for speaker-independent digit recognition. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Leonard, R. G. and G. R. Doddington (1993). *A Speaker-Independent Connected-Digit Database, Readme file*. Linguistic data consortium. `catalog.ldc.upenn.edu/docs/LDC93S10/tidigits.readme.html`. Accessed: 26.08.2015.

Levinson, S. E. (1986). Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language 1*(1), 29–45.

Li, X. and H. Jiang (2007). Solving large-margin hidden Markov model estimation via semidefinite programming. *IEEE© Transactions on Audio, Speech, and Language Processing 15*(8), 2383–2392.

Li, X., H. Jiang, and C. Liu (2005). Large margin HMMs for speech recognition. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Ligorio, T., S. L. Epstein, R. J. Passonneau, and J. B. Gordon (2010). What you did and didn't mean: Noise, context, and human skill. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Lin, H. and J. Bilmes (2011). Optimal selection of limited vocabulary speech corpora. In *Proceedings of International Conference on Spoken Language Processing*.

Lin, P., K. Abney, and G. A. Bekey (2011). Robot ethics: the ethical and social implications of robotics. Series in Intelligent Robotics and Autonomous Agents, The MIT Press.

Liporace, L. (1982). Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE© Transactions on Information Theory 28*(5), 729–734.

Lippmann, R. (1997). Speech recognition by machines and humans. *Speech Communication 22*(1), 1–15.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE© Transactions on Information Theory 28*(2), 129–137.

Lokhande, N. N., N. S. Nehe, and P. S. Vikhe (2012). Voice activity detection algorithm for speech recognition applications. In *Proceedings of the International Conference in Computational Intelligence*.

Lööf, J., C. Gollan, and H. Ney (2009). Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system. In *Proceedings of the International Conference on Spoken Language Processing*.

Macherey, W., L. Haferkamp, R. Schlüter, and H. Ney (2005). Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Maddieson, I. (1984). *Patterns of sounds*. Cambridge University Press.

Mari, J. F., J. P. Haton, and A. Kriouile (1997). Automatic word recognition based on second-order hidden Markov models. *IEEE© Transactions on Speech and Audio Processing 5*(1), 22–25.

Markov, K. and S. Nakamura (2007). Never-ending learning with dynamic hidden Markov networks. In *Proceedings of the International Conference on Spoken Language Processing*.

Mattys, S. L. and P. W. Jusczyk (2001). Phonotactic cues for segmentation of fluent speech by infants. *Cognition 78*(2), 91–121.

Mattys, S. L., P. W. Jusczyk, P. A. Luce, and J. L. Morgan (1999). Phonotactic and prosodic effects on word segmentation in infants. *Cognitive Psychology 38*(4), 465–494.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology 5*, 115–133.

McDermott, E. (1997). *Discriminative training for speech recognition*. Ph. D. thesis, Waseda University.

Melin, H. (1998). Optimizing variance flooring in HMM-based speaker verification. In *Proceedings of the COST 250 Workshop on Speaker Recognition by Man and by Machine: Directions for Forensic Applications*.

Melin, H., J. W. Koolwaaij, J. Lindberg, and F. Bimbot (1998). A comparative evaluation of variance flooring techniques in HMM-based speaker verification. In *Proceedings of the International Conference on Spoken Language Processing*.

Melin, H. and J. Lindberg (1999). Variance flooring, scaling and tying for text-dependent speaker verification. In *Proceedings of the European Conference on Speech Communication and Technology*.

Menyuk, P. (1977). *Language and maturation*. The MIT Press.

Mikhailova, I., M. Heracles, B. Bolder, H. Janssen, H. Brandl, J. Schmüdderich, and C. Görick (2008). Coupling of mental concepts to a reactive system: incremental approach in system design. In *Proceedings of the International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*.

Minematsu, N., S. Asakawa, Y. Qiao, D. Saito, T. Nishimura, and K. Hirose (2009). Implementation of robust speech recognition by simulating infants' speech perception based on the invariant sound shape embedded in utterances. In *Proceedings of the International Conference on Speech and Computer*.

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2012). *Foundations of Machine Learning*. The MIT Press.

Moreno, P. J. and P. P. Ho (2003). A new SVM approach to speaker identification and verification using probabilistic distance kernels. In *Proceedings of the European Conference on Speech Communication and Technology*.

Morgan, N., H. Boulard, S. Renals, M. Cohen, and H. Franco (1993). Hybrid neural network/hidden Markov model systems for continuous speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence 7*(4), 899–916.

Morgan, N. and H. Bourlard (1995). Continuous speech recognition: an introduction to the hybrid HMM/connectionist approach. *IEEE© Signal Processing Magazine 12*(3), 24–42.

Mount, D. W. (2004). *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press.

Murthy, H. A., T. Nagarajan, and N. Hemalatha (2004). Automatic segmentation and labeling of continuous speech without bootstrapping. In *Proceedings of the European signal processing conference*.

Nabney, I. T. (2002). *NETLAB: algorithms for pattern recognition*. Springer, Series in Advances in Pattern Recognition.

NAO (2015). Who is NAO? `http://www.aldebaran-robotics.com/en/humanoid-robot/nao-robot`. Accessed: 15.09.2015.

Nathan, K., A. Senior, and J. Subrahmonia (1996). Initialization of hidden Markov models for unconstrained on-line handwriting recognition. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular*

Biology 48(3), 443–453.

Neukirchen, C. and G. Rigoll (1998). Controlling the complexity of HMM systems by regularization. In *Advances in Neural Information Processing Systems*.

Nguyen, S. M., A. Baranes, and P.-Y. Oudeyer (2011). Bootstrapping intrinsically motivated learning with human demonstration. In *Proceedings of the IEEE© International Conference on Development and Learning*.

Ninio, A. (1993). On the fringes of the system: Children's acquisition of syntactically isolated forms at the onset of speech. *First Language 13*(39), 291–313.

Nocedal, J. and S. J. Wright (1999). *Numerical optimization*. Springer.

Nozza, R. J., R. N. Rossman, L. C. Bond, and S. L. Miller (1990). Infant speech-sound discrimination in noise. *Journal of the Acoustical Society of America 87*(1), 339–350.

O'Dea, T. and P. Mukherji (2000). *Understanding Children's Language and Literacy*. Stanley Thornes.

Ons, B., J. F. Gemmeke, and H. Van hamme (2014). Fast vocabulary acquisition in an NMF-based self-learning vocal user interface. *Computer Speech and Language 28*(4), 997 – 1017.

Pine, J. M. and E. V. Lieven (1993). Reanalysing rote-learned phrases: individual differences in the transition to multi-word speech. *Journal of Child Language 20*(3), 551–571.

Prasad, V. K., T. Nagarajan, and H. A. Murthy (2004). Automatic segmentation of continuous speech using minimum phase group delay functions. *Speech Communication 42*(3,4), 429 – 446.

Quinlan, P. and B. Dyson (2008). *Cognitive Psychology*. Pearson Education Limited (Prentice Hall).

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE© 77*(2), 257–286.

Rabiner, L. R. and R. W. Schafer (2007). *Introduction to Digital Speech Processing*. Foundations and Trends in Technology. Now Publishers.

Ratnagiri, M. V., B.-H. Juang, and L. R. Rabiner (2011). Large margin minimum classification error using sum of shifted sigmoids as the loss function. In *Proceedings of the International Conference on Spoken Language Processing*.

Reddy, R. (2001). *Foreword. In Huang, X., Acero, A. and Hon, H.-W. Spoken language processing: a guide to theory, algorithm and system development*. Prentice Hall.

Roger, K., L. Guse, E. Mordoch, and A. Osterreicher (2012). Social commitment robots and dementia. *Canadian Journal on Aging / La Revue canadienne du vieillissement 31*(1), 87–94.

Rolf, M., M. Hanheide, and K. J. Rohlfing (2009). Attention via synchrony: making use of multimodal cues in social learning. *IEEE© Transactions on Autonomous Mental Development 1*(1), 55–67.

Romanelli, F. (2010). Hybrid control techniques for static and dynamic environments: a step towards robot-environment interaction. *Robot Manipulators: New Achievements, Chapter 29*, 551–576.

Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan.

Roy, D. (2003). Grounded spoken language acquisition: experiments in word learning. *IEEE© Transactions on Multimedia 5*(2), 197–209.

Roy, D. (2009). New horizons in the study of child language acquisition. In *Proceedings of the International Conference on Spoken Language Processing*.

Roy, D. and A. Pentland (2002). Learning words from sights and sounds: a computational model. *Cognitive Science 26*(1), 113–146.

Rumelhart, D. E. and J. L. McClelland (1986). *Parallel distributed processing: explorations in the microstructure of cognition, Vol. 1: Foundations*. The MIT Press.

Russell, M. and R. Moore (1985). Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Saffran, J. R., R. N. Aslin, and E. L. Newport (1996). Statistical learning by 8 month-old infants. *Science 274*(5294), 1926–1928.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development 3*(3), 210–229.

Sankar, A. (1998). Experiments with a Gaussian merging-splitting algorithm for HMM training for speech recognition. In *Proceedings of the DARPA Speech Recognition Workshop*.

Sathya, R. and A. Abraham (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence 2*(2), 34–38.

Saunders, J., H. Lehman, Y. Sato, and C. L. Nehaniv (2011). Towards using prosody to scaffold lexical meaning in robots. In *Proceedings of the IEEE© International Conference on Development and Learning*.

Saxton, M. (2009). The inevitability of child directed speech. In *Advances in Language acquisition. Palgrave Macmillan, 62–86*.

Schillingmann, L., P. Wagner, C. Munier, B. Wrede, and K. Rohlfing (2011). Using prominence detection to generate acoustic feedback in tutoring scenarios. In *Proceedings of the International Conference on Spoken Language Processing*.

Schuster, M. and K. Nakajima (2012). Japanese and Korean voice search. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Schwartz, R. G. and B. Y. Terrell (1983). The role of input frequency in lexical acquisition. *Journal of Child Language 10*(1), 57–64.

Seabra Lopes, L. (2002). CARL: from situated activity to language level interaction and learning. In *Proceedings of the International Conference of Intelligent Robots and Systems*.

Seabra Lopes, L. (2007). How many words can my robot learn? An approach and experiments with one-class learning. *Interaction Studies 8*, 53–81.

Seymore, K., S. Chen, S. Doh, M. Eskenazi, E. Gouvea, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern, and E. Thayer (1998). The 1997 CMU Sphinx-3 English broadcast news transcription system. In *Proceedings of the DARPA Speech Recognition Workshop*.

Sha, F. and L. K. Saul (2006). Large margin Gaussian mixture modeling for phonetic classification and recognition. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Sha, F. and L. K. Saul (2007). Large margin hidden Markov models for automatic speech recognition. In *Advances in Neural Information Processing Systems*.

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers 37*(1), 10–21.

Sharma, K. R. (2009). *Bioinformatics: sequence alignment and Markov models*. McGraw-Hill.

Shute, B. and K. Wheldall (1999). Fundamental frequency and temporal modifications in the speech of British fathers to their children. *Educational Psychology 19*(2), 221–233.

Sim, K. C. and M. J. F. Gales (2006). Minimum phone error training of precision matrix models. *IEEE© Transactions on Audio, Speech and Language Processing 14*(3), 882–889.

Siri (2015). Speech recognition technologies in Apple. `http://www.apple.com/uk/ios/siri/`. Accessed: 16.06.2015.

Siu, M., M. Jonas, and H. Gish (1999). Using a large vocabulary continuous speech recognizer for a constrained domain with limited training. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Smith, N. and M. Gales (2001). Speech recognition using SVMs. In *Advances in Neural Information Processing Systems*.

Smith, T. F. and M. S. Waterman (1981). Identification of common molecular subsequences. *Journal of Molecular Biology 147*(1), 195–197.

Sroka, J. J. and L. D. Braida (2005). Human and machine consonant recognition. *Speech Communication 45*(4), 401–423.

Stadermann, J. and G. Rigoll (2004). A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Steels, L. (1995). When are robots intelligent autonomous agents? *Robotics and Autonomous Systems 15*(1), 3–9.

Steels, L. and F. Kaplan (2001). Aibo's first words: the social learning of language and meaning. *Evolution of Communication 4*(1), 3–32.

Stuttle, M. N. (2003). *A Gaussian mixture model spectral representation for speech recognition*. Ph. D. thesis, Hughes Hall and Cambridge University Engineering Department.

Sun, M. (2012). *Constrained non-negative matrix factorization for vocabulary acquisition from continuous speech*. Ph. D. thesis, Katholieke Universiteit Leuven.

Sung, J., H. I. Christensen, and R. E. Grinter (2009). Sketching the future: assessing user needs for domestic robots. In *Proceedings of the IEEE© International Symposium on Robot and Human Interactive Communication*.

Sutton, R. S. and A. G. Barto (1998). Reinforcement learning: an introduction. The MIT Press.

ten Bosch, L., L. Boves, H. Van hamme, and R. K. Moore (2009). A computational model of language acquisition: the emergence of words. *Fundamenta Informaticae 90*(3), 229–249.

Theodoridis, S. (2015). *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press.

Theodoridis, S. and K. Koutroumbas (2003). *Pattern recognition*. Elsevier Academic Press.

Thiessen, E. D., E. A. Hill, and J. R. Saffran (2005). Infant-directed speech facilitates word segmentation. *Infancy 7*(1), 53–71.

Tomasello, M. (2003). *Constructing a language: a usage-based theory of language acquisition*. Harvard University Press.

Vaisipour, S., B. Babaali, and H. Sameti (2007). Using and evaluating new confidence measures in word-based isolated word recognizers. In *Proceedings of the International Symposium on Signal Processing and Its Applications*.

Van hamme, H. (2008). HAC-models: a novel approach to continuous speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*.

Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley and Sons.

Versteegh, M., L. ten Bosch, and L. Boves (2011). Modelling novelty preference in word learning. In *Proceedings of the International Conference on Spoken Language Processing*.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE© Transactions on Information Theory 13*(2), 260–269.

Vu, N. T., F. Kraus, and T. Schultz (2011). Rapid building of an ASR system for under-resourced languages based on multilingual unsupervised training. In *Proceedings of the International Conference on Spoken Language Processing*.

Waibel, A. and K. F. Lee (1990). *Readings in speech recognition*. Morgan Kaufmann Publishers.

Wan, V. and W. M. Campbell (2000). Support vector machines for speaker verification and identification. In *Proceedings of the IEEE© Workshop on Neural Networks for Signal Processing*.

Wang, X. and L. C. W. Pols (1997). A preliminary study about robust speech recognition for a robotics application. In *Proceedings of the Institute of Phonetic Sciences*.

Werker, J. F. and R. N. Desjardins (1995). Listening to speech in the first year of life: experiential influences on phoneme perception. *Current Directions in Psychological Sciences 4*(3), 76–81.

Wessel, F. and H. Ney (2005). Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE© Transactions on Speech and Audio Processing 13*(1), 23–31.

Widrow, B. and M. E. Hoff (1960). Adaptive switching circuits. In *Proceedings of the IRE WESCON Convention Record*.

Woodland, P. (2009). What is HTK? `http://htk.eng.cam.ac.uk/`. Accessed: 20.08.2015.

Woodland, P. C. and D. Povey (2002). Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language 16*(1), 25–47.

Woodward, A. L., E. M. Markman, and C. M. Fitzsimmons (1994). Rapid word learning in 13- and 18-month-olds. *Developmental Psychology 30*(4), 553–566.

Woog, A. (2010). *Scratchbot, a great idea*. Norwood House Press.

Wu, Y., R. Zhang, and A. Rudnicky (2007). Data selection for speech recognition. In *Proceedings of the IEEE© Workshop on Automatic Speech Recognition and Understanding*.

Young, S. J., G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland (2009). *The HTK Book, Version 3.4*. Cambridge University Engineering Department.

Young, S. J., N. H. Russell, and J. H. S. Thornton (1989). Token passing: a simple conceptual model for connected speech recognition systems. Technical report, Cambridge University Engineering Department.

Yu, D. and L. Deng (2007). Large-margin discriminative training of hidden Markov models for speech recognition. In *Proceedings of the IEEE© International Conference on Semantic Computing*.

Yu, D., L. Deng, X. He, and A. Acero (2007). Large-margin minimum classification error training for large-scale speech recognition tasks. In *Proceedings of the IEEE© International Conference on Acoustics, Speech and Signal Processing*.

Yu, D., L. Deng, X. He, and A. Acero (2008). Large-margin minimum classification error training: a theoretical risk minimization perspective. *Computer Speech and Language 22*(4), 415–429.

Yuan, B., C. Guan, G. Loudon, and H. Li (1998). Optimization of parameter-tying for Chinese acoustic modeling. In *Proceedings of the International Symposium on Chinese Spoken Language Processing*.

Zavaliagkos, G., Y. Zhao, R. M. Schwartz, and J. Makhoul (1994). A hybrid segmental neural net/hidden Markov model system for continuous speech recognition. *IEEE© Transactions on Speech and Audio Processing 2*(1), 151–160.

Zimmermann, M. and H. Bunke (2001). Hidden Markov model length optimization for handwriting recognition systems. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*.