Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Incremental word learning: Efficient HMM initialization and large margin discriminative adaptation

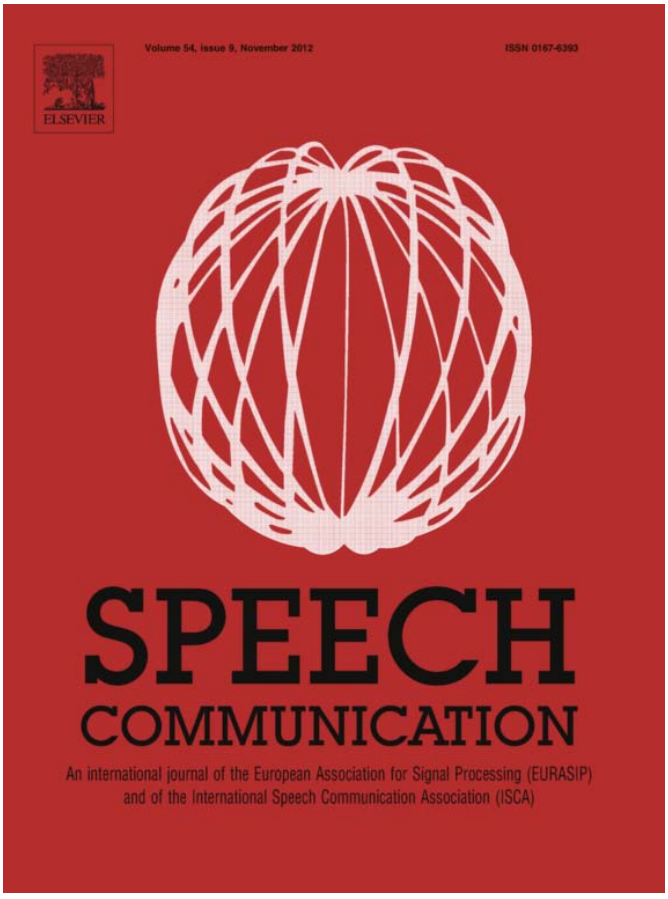
Irene Ayllón Clemente, Martin Ernst Heckmann, Britta Wrede

2012

Preprint:

This is an accepted article published in Speech Communication. The final authenticated version is available online at: https://doi.org/[DOI not available]

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and educational use, including for instruction at the author's institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's Personal Copy

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Speech Communication 54 (2012) 1029–1048

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Incremental word learning: Efficient HMM initialization and large margin discriminative adaptation

Irene Ayllón Clemente [a,b,*], Martin Heckmann [b], Britta Wrede [a]

[a] Bielefeld University, Research Institute for Cognition and Robotics – CoR-Lab, D-33615 Bielefeld, Germany
[b] Honda Research Institute Europe GmbH, D-63073 Offenbach am Main, Germany

Received 22 September 2011; received in revised form 19 April 2012; accepted 19 April 2012
Available online 28 April 2012

Abstract

In this paper we present an incremental word learning system that is able to cope with few training data samples to enable speech acquisition in on-line human robot interaction. As with most automatic speech recognition systems (ASR), our architecture relies on a Hidden Markov Model (HMM) framework where the different word models are sequentially trained and the system has little prior knowledge. To achieve good performance, HMMs depends on the amount of training data, the initialization procedure and the efficiency of the discriminative training algorithms. Thus, we propose different approaches to improve the system. One major problem of using a small amount of training data is over-fitting. Hence we present a novel estimation of the variance floor dependent on the number of available training samples. Next, we propose a bootstrapping approach in order to get a good initialization of the HMM parameters. This method is based on unsupervised training of the parameters and subsequent construction of a new HMM by aligning and merging Viterbi decoded sequences. Finally, we investigate large margin discriminative training techniques to enlarge the generalization performance of the models using several strategies suitable for limited training data. In the evaluation of the results, we examine the contribution of the different stages proposed to the overall system performance. This includes the comparison of different state-of-the-art methods with our presented techniques and the investigation of the possible reduction of the number of training data samples. We compare our algorithms on isolated and continuous digit recognition tasks. To sum up, we show that the proposed algorithms yield significant improvements and are a step towards efficient learning with few examples.
© 2012 Elsevier B.V. All rights reserved.

Keywords: Incremental word learning; Learning from few examples; Speech recognition; Multiple sequence alignment; Discriminative training; Bootstrapping

1. Introduction

In cognitive robotics researchers target for the understanding and modeling of cognitive processes in order to endow robotic systems with more complex behavior and the ability of autonomous learning. Robotic systems have to acquire knowledge during unconstrained interaction and transfer this knowledge to new situations in order to realize daily activities as well as to interact with humans, e.g. to assist people requiring care. This requires the development of methods to directly interact with human tutors rather than hard-coded programs or predefined behaviors to fulfill only specific predefined tasks.

A basic ability to achieve this goal is a communication between human and machine, i.e. a personal robot has to understand written or verbal communication (Charles and Bailey, 1992). It would enable a natural interaction with the user and the possibility to get a direct feedback during interaction as well as to transfer knowledge without the requirement of programming skills. Since we communicate using different dialects and vocabulary, it is crucial that robots not only know our native language, but also

* Corresponding author at: Bielefeld University, Research Institute for Cognition and Robotics – CoR-Lab, D-33615 Bielefeld, Germany.
E-mail addresses: iayllon@cor-lab.uni-bielefeld.de (I. Ayllón Clemente), martin.heckmann@honda-ri.de (M. Heckmann), bwrede@cor-lab.uni-bielefeld.de (B. Wrede).

0167-6393/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.specom.2012.04.005

Author's Personal Copy

are able to learn new words of our own lexicon, e.g. dialects.

Language acquisition is a complex mental ability of humans (Bosch et al., 2009). Nowadays automatic speech recognition (ASR) systems are considered as potential computational models of these corresponding cognitive skills. Despite the huge advances of ASR in the last decade, all conventional ASR systems still perform substantially worse than humans (Bosch et al., 2008). In addition, until now no computational model can precisely explain the acquisition of language and communication skills (Boves et al., 2007). Thus, our target is the efficient construction of speech models in order to enable the acquisition of knowledge in robots. In this way, an interactive learning system where a human tutor teaches a robot is a very attractive scenario for our long-term goal.

The framework we propose is inspired by the process of speech acquisition in children. Compared to current systems children learn through constant communication and interaction with their parents and other children. Hence for auditory learning in small children, we assume that a closed loop of speech perception and production plays an important role. While some authors concentrate on jointly solving both aspects (Moore, 2007; Minematsu, 2010), others (Van Segbroeck and Van hamme, 2009; Bosch et al., 2009) constrain their work on the robust perception, i.e. the recognition of words, as it is in itself still a widely unsolved problem in ASR systems. Moreover, our learning framework should constitute a user-friendly system adequate to acquire language in an unsupervised way and to require a low tutoring time, i.e. few training samples.

In ASR systems, the standard statistical model to represent the structure of speech is the continuous density Hidden Markov Model (CDHMM), where Gaussian Mixture Models (GMM) are usually employed to model the feature distribution in the hidden states. In HMMs, the amount of available training data to obtain the estimates of the model parameters, the initialization of the models and the discriminative training algorithms used are critical factors. These aspects are explained in the following paragraphs.

With conventional off-line training techniques, a large amount of labeled training data is required to estimate an optimal set of parameters. Unfortunately, it is very difficult to obtain this data in interactive learning; hence researchers aim to train the system in an unsupervised manner and/or to train it with a smaller number of training data samples (Iwahashi, 2006). One of the main drawbacks of using a small amount of training data is the over-fitting problem (Bishop, 2006). In this case the learning algorithms of the HMMs may fit the model parameters to some specific features of the dataset but do not generalize to unseen examples (Ghahramani, 2001). There are different methods to avoid over-fitting. A common method is to apply standard cross-validation learning (Devijver and Kittler, 1982). However, such an approach relies on off-line training and we cannot ensure enough training data on on-line systems to realize this approach. Alternative approaches to enable learning with few training samples apply regularization methods with a penalty term (Neukirchen and Rigoll, 1998) or in other cases, they reduce the number of free parameters (Siu et al., 1999). Following this idea, Neukirchen et al. (1998) and Kadous (2002) proposed to use 'parameter tying'. The idea is to set up an equivalence relation between HMM parameters in different states. In this way the number of independent parameters in the model is reduced. The problem here is to define these relations *a priori*, which is also difficult in an incremental learning scenario without prior knowledge. Nevertheless, the widely used technique to avoid over-fitting is the integration of further constraints like the so called variance floor. Here the training algorithms can be modified by including a lower threshold on the variance parameters, a variance floor (Melin, 1998; Sim and Gales, 2006; Melin and Lindberg, 1999). 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 (Stuttle, 2004). In contrast, other researchers as Dong et al., 2008 use a method to adapt the variance floor to each dimension of the features or to floor the variances using a percentile variable (Lee et al., 1992). Additionally to the variance floor, the minimum allowed variance (Young et al., 2006) is another threshold applied to the floor, 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 (Liu, 1994; Rosenberg et al., 1998; Mokbel and Collin, 1999).

Since the training process in HMMs is based on Expectation-Maximization, the efficiency of the procedure strongly depends on the initialization of the parameters. The standard initialization approach was proposed by Rabiner (1989). It segments the training data frames into states by means of K-means clustering and Viterbi decoding. This initialization technique is executed in a supervised manner requiring the use of many labeled training data samples. However this bootstrapping method gives the possibility to iteratively estimate new models. Another state-of-the-art initialization method is 'flat start'. In this case all Gaussians Mixtures Models (GMMs) of the classes are initialized with identical parameters equal to the global speech mean and variance of the training data (Itaya et al., 2005; Young et al., 2006). The advantage of this method is that the models can be easily initialized without the need of labels, i.e. unsupervised. Nevertheless, this technique does not allow an efficient iterative construction of new models. 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 techniques can be found in Nathan et al. (1996) and Smith (2002). Moving towards an unsupervised initialization technique for incremental learning in interactive environments, some authors like Brandl et al. (2008) and Iwahashi (2006) initialize the system by means of an unsu-

Author's Personal Copy

pervised phone model acquisition procedure. This technique is also based on K-means clustering and Viterbi decoding and it enables the initialization of syllable models in an on-line word learning system (Brandl, 2010) using previously estimated phone models.

Finally, discriminative training (DT) techniques have been widely studied in ASR systems in order to improve the generalization performance of the HMMs (Juang et al., 1997; McDermott, 1997). In general, DT methods outperform conventional HMMs training techniques when the amount of training data is limited and does not allow reliable parameter estimation. Some of these approaches directly minimize classification errors on the training data as the model estimation criterion. Minimum classification error (MCE) is currently one of these prominent DT approaches. One efficient MCE technique has been proposed by He et al. (2008). Recently, the generalization ability of HMMs has been further improved by taking the margin of the classifier into account, these techniques are called large-margin discriminative training (Dong et al., 2008; Li et al., 2005; Chang et al., 2008; Sha and Saul, 2007). Dong et al., 2008 targets combining the large margin principle with MCE, while other authors concentrate on improving the GMMs by updating their means (Li et al., 2005) as well as their variances (Chang et al., 2008; Sha and Saul, 2007).

We present an efficient incremental word learning framework using few training data and being suitable to operate in an interactive scenario where a human tutor teaches a robot. The use of only limited data is motivated by the reduction of the tutoring time in order to make the interaction and learning process more efficient. In these conditions, the selection of a suitable set of parameters for our scenario is essential in order to avoid over-fitting when few data samples are provided. Hence, we analyze how to select the appropriate topology of the HMMs and investigate approaches to estimate the variance floor that best suits to the system. We extend the state-of-the-art variance floor methods going one step further by incorporating the standard approaches a scaling factor dependent on the number of training data. The proposed framework consists of three main steps displayed in Fig. 1. The first stage of the framework is to optimally initialize the parameters of the Hidden Markov Models (HMMs) in order to get better performance. Our work is influenced by the approaches of Brandl et al. (2008) and Iwahashi (2006). In this way, we propose a bootstrapping method (Ayllon Clemente et al., 2010b) that can deal with few training samples in incremental word learning and provide initial estimates of the word-models to learn. Our approach consists of the combination of unsupervised and supervised training, where a transformation of an ergodic HMM into a left-to-right HMM takes place. This transformation is performed by means of a novel multiple sequence alignment procedure. This first step allows the estimation of a good initial set of HMM parameters that are trained by Maximum Likelihood (ML) (Rabiner, 1989) and Maximum A Posteriori (MAP) estimation in the next phase. After retraining the HMMs, the last refining stage uses discriminative training. In cases where all training segments are well classified as in our framework and the number of training samples is very low, we may still want to maximize the minimum margin to build an HMM-based large margin classifier. Therefore, we also investigate and analyze different strategies to execute large margin discriminative training to improve the recognition scores when the number of training samples is very limited.

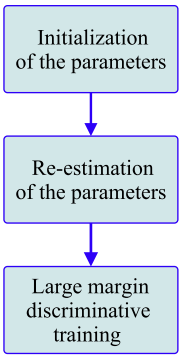


Fig. 1. Overview of the presented framework. The blocks display the different phases of the system. After the parameters of the HMMs are initialized, the estimates are refined in the re-estimation phase. Finally a discriminative training step improves the generalization performance of the system.

The remainder of the paper is organized as follows. In Section 2 the problem description is presented. Hidden Markov Models (HMMs) as ASR systems are introduced, the differences between batch training and incremental learning are described and several state-of-the-art methods for on-line learning are discussed. Next, we introduce the architecture of our system and subsequently all the different stages of our framework in Section 3. First, the estimation of the variance floor incorporating the number of training data samples is described. Afterwards, we present the model bootstrapping, describing in more detail the different phases of the initialization in consecutive subsections. Here an unsupervised initialization of the parameters of an HMM is performed, followed by the retraining and construction of a new HMM using multiple sequence alignment (MSA). Then the different large-margin discriminative training strategies proposed are described. In Section 4 we describe experiments in which we compare and evaluate our approach with different standard techniques and report the recognition scores obtained on an isolated and continuous digit recognition task. In Section 5 we discuss the results we obtain, possible improvements and give an outlook on future work. Finally we summarize the work presented in Section 6.

2. Problem description

A learning procedure to model speech units can be described by the process of constructing statistical models

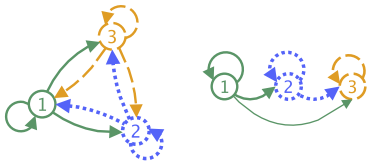
Author's Personal Copy

from the training data in order to map the continuous speech signals on a discrete representation of a fixed and limited number of predefined patterns (Bosch et al., 2009). Hidden Markov Models (HMMs) are the standard statistical approach for this purpose in ASR systems. They encode the information contained in a speech unit as a sequence of states and a set of parameters. HMMs have the advantage of being simple and not computationally demanding to use in ASR systems. However, as Huang et al. (2001) pointed out they have several limitations like the sensitivity to the initialization of their estimates, the need for a great amount of training data, the modeling of the duration of the speech units, the assumption of the process being a first-order Markov Chain and the assumption of the observations being conditional independent. Hence researchers aim for new speech recognition methods (Boves et al., 2007) or possibilities to construct hybrid systems (Morgan et al., 1993; Ganapathiraju et al., 2000). On the other hand, in principle the latter limitations can be overcome by incorporating more parameters (Bilmes, 2002). Hence the remaining drawbacks are the strong necessity for a large amount of training data to estimate the huge number of parameters and the initialization of them. Despite the known problems, HMMs are still the most employed ASR systems and part of many successful approaches (Young et al., 2006).

In this Section, we give an overview on HMMs focusing on their parameters and learning algorithms. Additionally, we introduce the differences between batch and on-line (incremental) learning algorithms for word-model HMMs and discuss some state-of-the-art methods for incremental word learning.

2.1. HMMs for ASR systems

Similar to other techniques in machine learning the choice of the model complexity requires *a priori* knowledge and experience with the learning problem (Rabiner, 1989). There are different representations to build an HMM, the most used topologies are ergodic and Bakis (see Fig. 2). Speech is a time-series of non-stationary signals. In HMMs, each state aims to encode a quasi-stationary speech segment of the non-stationary signal. The Bakis, i.e. left-to-right, topology has a self-transition to each state modeling adjacent speech features that belong to the same state. When the stationary segment evolves, then the transition to the right ensures a natural progression of this evolution (Huang et al., 2001). A Bakis topology and a low number of skip transitions (Fig. 2) have the advantage of reducing the number of parameters. The more restrictions we impose on the model the less parameters have to be estimated. In the case that speech features are independent, the full covariance matrices of the GMMs in an HMM can be simplified into diagonal matrices (variance vectors) reducing the number of parameters to estimate (Gales, 1999). Most state-of-the-art approaches rely on this assumption. However, despite the reduction of the parameter space,



(a) Ergodic (b) Bakis

Fig. 2. Different HMM topologies. In the ergodic topology (a) transitions between all states are allowed in contrast to Bakis (b) where only transitions from left to right are possible. In the latter we can also define how fast the last state of the sequence is achieved by using skip transitions (thin continuous line).

there are still too many parameters to estimate and they require a lot of training data samples (Huang et al., 2001). In order to improve the recognition results if limited training samples are provided, some authors use a variance threshold or also called variance floor (Melin, 1998). This parameter will be introduced in the next Sections. From the definition of the HMMs (Rabiner, 1989), we observe that the number of hidden states directly influences the size of the parameter set. Hence an efficient selection of the necessary number of hidden states of each word-model can notably reduce the required number of parameters. Some authors relate the number of hidden states of a model to the duration of the speech units (Rabiner, 1989). As we want to model words, one idea is to approximate the number of states of all the models to the number of sounds (phonemes) normally presented within a word. Hence models containing between 2 and 10 states would be appropriate. In this way the word-models have a fixed length, which is the same for all the word-models. Some exceptions are silences and pauses, which are stationary signals. However a fixed number of states is not optimal in cases where the different speech units to model do not have similar duration. Thus, another concept known as the Bakis length modeling method (Bakis, 1976) is to let the number of states correspond roughly to the average number of observations (frames) in a spoken version of the word. In this manner each state corresponds to an observation interval. Several authors, as Fink (2008) indicates, model one state per time unit (frame) of the duration of the speech signal, others set this relation heuristically (Geiger et al., 2011).

In the training of the HMMs, the initial parameters of the models are usually computed for a single GMM. If we want to model the emission probabilities with more that one component of GMMs, we iteratively split the Gaussian density function in different stages to increment the mixture components as advised by Young et al. (2006). After each splitting, several iterations are typically needed to refine the GMMs. In each iteration we can use different criteria to estimate the model parameters. The most used estimation methods are Maximum Likelihood (ML) and Maximum A Priori (MAP) estimation. These approaches are both supervised learning techniques (Huang et al., 2001). ML estimation aims to find the set of model parameters that maximizes the likelihood of generating the observed data

Author's Personal Copy

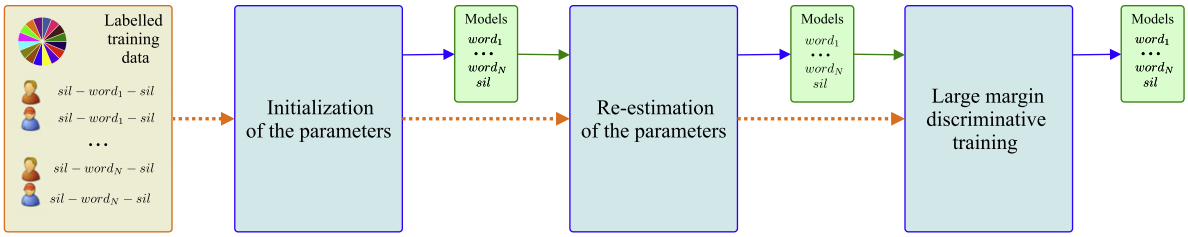


Fig. 3. Batch ASR system. Here all the labeled training data are introduced into the system. Each training sample used is represented by a colored sector in a circle. In the example presented, two training data per word are displayed. Each sample is uttered from a different speaker. All the N word-models including the silence model are estimated after each training step of the system.

without any prior knowledge as mentioned by Theodoridis and Koutroumbas (2008). Because of the lack of information contained in the prior, a sufficient number of training samples is often a prerequisite for ML estimation. On the other hand, MAP estimation can effectively deal with an insufficient number of data profiting from the prior information about existing models (Gauvain and Lee, 1994). The algorithm adjusts the estimates of pre-trained models in such a way that limited training samples would adapt the model parameters conducted by the prior knowledge in order to prevent large deviations of the parameters unless the training samples provide strong evidence (Huang et al., 2001). When the amount of training data increases, MAP estimation converges to ML estimation as the training samples overwrite any important prior that may influence the final estimate. In our system a combination of both methods is used (Section 3.3.4).

2.2. Incremental word learning

In a state-of-the-art batch process the whole training set is provided to the system and the models are jointly estimated in each iteration of the training phase, namely the bootstrapping (or initialization) and the re-estimation of the model parameters. Afterwards, the models are usually optimized by means of a discriminative training algorithm. This procedure is illustrated in Fig. 3.

However in interactive learning the system does not receive all the data at the same time. In this case, a tutor provides the data samples incrementally and often jointly with some kind of visual stimulus or pseudolabel for the word to learn. This is why authors as Roy (2003); Bosch et al. (2009); Van hamme (2008); Markov and Nakamura (2007); Iwahashi (2006) or Brandl et al. (2008) have proposed different approaches to deal with these challenges.

Roy (2003) introduced CELL (Cross-Channel Early Lexical Learning), a computational model of sensor-grounded word learning. He addressed the segmentation of continuous spontaneous speech without a pre-existing lexicon. However a recurrent neural network (RNN) trained off-line for phoneme recognition with the TIMIT database (Garofolo et al., 1993) was used to facilitate the segmentation process. In CELL, a speech synthesizer provided feedback to the user. The speech corpus used to train the system contained approximately 7600 utterances comprising 37000 words uttered by six different speakers. These speech samples were recorded through experiments with caregivers playing with prelinguistic infants.

The ACORNS project (Bosch et al., 2009) aimed to develop computational models that demonstrate the capability to acquire language and communication skills on the basis of sensory input. The challenge of the project was to model an infant who has no prior knowledge (in contrast to the above approach) about grammar, words or speech sounds when it starts to listen to its caregivers. The learning agent (the robot) was passive and did not explore the environment, however it received a feedback (auditory) from the (simulated) caregivers. Several methods, Non-Negative Matrix Factorization (NMF) or DP-Ngram, were modified in order to make them suitable for incremental learning and to extract patterns (word discovery) from continuous utterances. The speech corpus contained references to observable physical objects and events in the environment and it was recorded in three different languages: Dutch, Finnish and Swedish. For each language the database consisted of utterances from 2 male and 2 female speakers. Each speaker uttered 1000 sentences in adult-directed and infant-directed modes for a total of 2000 utterances per speaker. Per speaker, the set of 1000 sentences contained 10 repetitions of combinations of 10 target words and 10 carrier sentences.

Van hamme (2008) described a bottom-up recognition system able to operate in continuous speech. The framework is based on non-negative matrix factorization and on histograms of acoustic co-occurrence (HAC-models). These models are able to detect, order and locate words in sentences. The system is tested on the TIDigits database. Although the framework presents very promising results, HMMs still outperform it.

Markov and Nakamura (2007) developed an unsupervised on-line adaptive learning system. The network is composed of hidden Markov states and can detect previously unseen speech patterns. If a new pattern is found, new states are added to the network while preserving the previously acquired knowledge. To avoid an uncontrolled growth of the system, states corresponding to spurious events or noises (rarely visited) models are gradually removed. In their experiments they recognized utterances

Author's Personal Copy

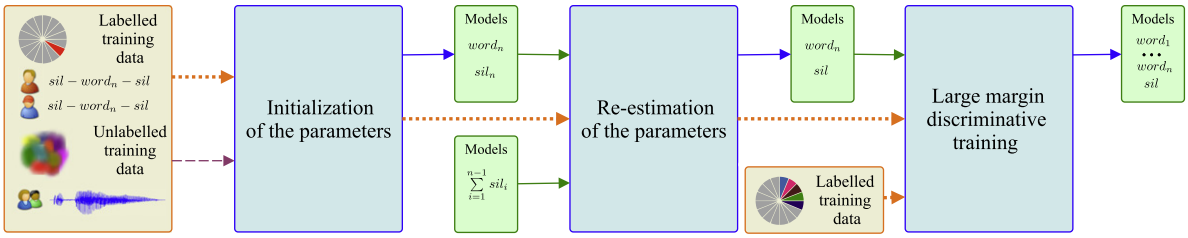


Fig. 4. Incremental word learning framework. In each iteration of the system, only the new word-model n is computed in the first and second estimation steps using the labeled training samples (colored sector) corresponding to the word-model. In the initialization stage, not only the labeled training samples of the word-model n are used, but unlabeled training data pronounced by other speakers and containing different words than the ones to learn. In the re-estimation of the parameters also the accumulators $\sum_{i=1}^{n-1} sil_i$ of the silence models are updated. Finally the large margin discriminative training phase refines the estimates of the model n using the previous learned models with their respective training samples.

consisting of single samples of 22 English letters uttered by Japanese speakers. The speech data set was composed of 440 speech sentences.

Iwahashi (2006) presented a developmental approach for language processing to be used in human-robot conversations with little prior knowledge. The robot was able to learn words referring to objects names and motions of moving objects using utterances containing single words and images of the objects or movements. Next, the system iteratively acquired some simple grammar maintaining 'conversations' with the tutor. Novelty detection was also integrated into the system. The robot could detect known and unknown words with the help of visual stimuli that were learned jointly with the speech units. If the robot could not decide if the word was already learned it asked the user for feedback. To obtain good performance, the robot was trained using a lot of interaction episodes. In each of this learning episode not only the user but the robot could ask the tutor to realize an action. In cases of incorrect behavior of the robot, the user slapped the hand of the robot (a touch sensor) to provide feedback to the system.

Brandl et al. (2008) proposed a model for unsupervised acoustic speech acquisition inspired by principles which are believed to play a role in early infant speech acquisition. The key concepts of their approach included a regularization scheme that might ensure asymptotic homeostasis and the combination of unsupervised and supervised speech processing in isolated word learning. When only 6 training samples were used, the system performed quite well in a very limited speaker-dependent scenario. In contrast to the above mentioned approaches, Brandl et al. (2008) did not provide any kind of feedback to the robot.

A common aspect between Iwahashi (2006) and Brandl et al. (2008) is the integration of a phone unsupervised bootstrapping phase in their systems in order to save training time. Both also implemented their approaches on an HMM framework.

All these methods aim to train a system, a robot, in interaction with a human tutor. However, one key aspect that is not directly addressed by these approaches is reducing the number of repetitions during the training while maintaining good performance and not over-specializing to one individual tutor. A system which achieves this is much more intuitive and user-friendly as it reduces the required tutoring time.

3. An efficient incremental word learning system

In this Section, our proposed framework and methods are explained, which allow using only limited training data and propose a novel manner to initialize an HMM system. First, we define the architecture of our incremental word learning system. Next, we explain our variance floor estimation that is computed in the training steps for the initialization and re-estimation of the parameters. Then, the different stages of the model bootstrapping algorithm are described in detail. Finally we introduce large margin discriminative training and emphasize different strategies for limited training data and the optimization algorithms used.

3.1. Architecture

The incremental word learning system proposed comprises three main processing stages and is illustrated in Fig. 4. In incremental learning we do not provide all the data at the same time, but only the data relative to the model to estimate in each time step. In Fig. 4, only the sector corresponding to the training samples of the current word to learn is colored.

First, the parameters of the Hidden Markov Models are initialized by means of a model bootstrapping method, also called initialization, that uses the labeled training samples of the words to learn and additionally a set of unlabeled random speech segments. In Fig. 4 the unlabeled random speech fragments are visualized by a group of (two) speakers. This initialization method is explained in more detail in Section 3.3. The initialization step allows the estimation of a good initial set of HMM parameters that are recalculated in the next stage using ML and MAP estimation algorithms. Instead of computing all the models again in each time step if desired, we only estimate the actual word-model. In this way we do not need to estimate all the previous models when data of a still unseen class is entered into the system.

Author's Personal Copy

The fact that we only consider the model to estimate and its respective training samples for the initialization and re-estimation of the parameters influences the computation of the variance floor. At the same time, the variance floor is part of the estimation of the variances of the Gaussian Mixture Models (GMMs) representing the observation probabilities. This constraint has been chosen in order to avoid the re-estimation of all the already learned models when a new model enters the system. If we only take into account to calculate the variance floor of the already learned models, the variance floor values of the models previously estimated will be lower. Thus, the models would not be equally estimated. Hence, the variance floor estimated here only takes into account its own model and training data. Unfortunately, it suggests that the estimation becomes highly sensitive to unseen data. The variance floor estimated in each time step is employed in the initialization and re-estimation steps.

When working with data containing isolated spoken words, each training sample does not only contain the samples of the models to be estimated but silences at the beginning and end of each training segment. In a batch process the whole training set is provided before the start of the estimation of the parameters, but not in incremental learning. Here the system only receives part of the whole data set of the silence model in each time step. Hence we work with silence accumulators of the estimates that are computed each time a word-model is added. These accumulators are defined by the symbol $\sum_{i=1}^{n-1} sil_i$ in Fig. 4, where n is the current word to learn. Each update of the silence model incorporates the accumulators of the previous silence data. This procedure reduces the computational cost because it avoids the iterative estimation of the silence model each time new training samples are introduced into the framework. During the re-estimation of the parameters of the new model, the accumulators of the silence model are fused and the silence model is updated.

Finally, 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 stage before. In a batch process all the models are introduced into the system before starting a discriminative training procedure. On the other hand, in incremental learning a new cluster is added at each time step forcing the optimization of the parameters at each iteration. As we assume that the system has to be improved and ready after each new cluster has entered, in interactive systems the computation effort of an algorithm is a significant issue. Therefore, some adequate discriminative training strategies are presented in Section 3.4.

3.2. Estimation of the variance floor

In Hidden Markov Models, Expectation-Maximization (EM) algorithms are often deployed to adapt the parameters of HMMs including the means and variances of their Gaussian Mixtures Models (GMM). However, a variance estimated from only a few training samples might not be representative for the underlying distribution. In general, these estimated variances also decrease in each training iteration as Fig. 5 shows. If the variances drop too much the distributions become narrow and over-fitting could occur. More precisely, there is a tendency of underestimating the variances in such cases (besides further numerical problems like division near zero (Ghoshal, 2005)). In these cases, it is necessary to use a variance floor as threshold in order to avoid the decrease of the variances in each training iteration (Ghoshal, 2005; Bimbot et al., 1998). Melin et al. (1998) identified three different approaches to calculate and apply the variance floor. The first method is called the model-dependent floor. In this technique the variance floor vector for a word-model is calculated from the variance of all vectors with speech segments exclusively identified to belong to this word. The second approach, state-dependent floor, creates a vector for a particular state in a word-model that is computed as a linear combination of variance parameters of the Gaussian mixtures in this specific state in the model. In contrast, the third technique named mixture component-dependent floor, constructs a variance vector for a specific mixture component in a determined state of a word-model as the variance of this mixture-component in a model calculated using one minute of speech. We apply in our incremental word learning framework the first approach referred to as model-dependent floor.

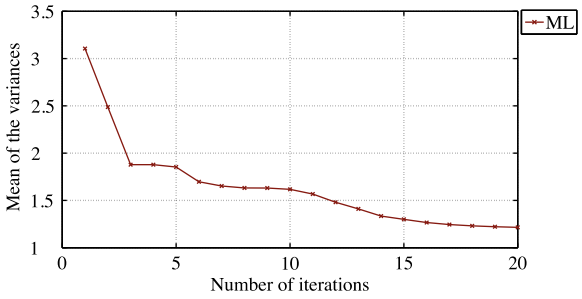


Fig. 5. The decrease of the estimated variances after each Maximum Likelihood (ML) iteration. The plotted example corresponds to the average of the variances of all components of the GMMs of all the word-models.

In our system, the first step is to decide which floor constant is optimal for our task (see Section 4.1). Hence it is necessary to evaluate how the variances decrease when the number of samples n is reduced. Although the magnitudes of the RASTA-PLP features of the individual dimensions are quite different, their behavior is very similar when varying the number of training samples. Therefore, taking as basis for setting the variance floor the average of the variances of the dimensions of the features seems a well-suited approximation. In Fig. 6 the average $\overline{X}_V(n)$ 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 in the beginning and then only slightly increase until they saturate at $\overline{X}_V(\infty)$ (Fig. 6).

Author's Personal Copy

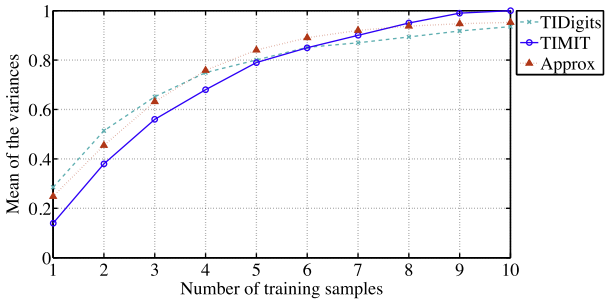


Fig. 6. The average $\overline{X}_V(n)$ of the variances of the GMMs (3 per hidden state) over all the models of the TIDigits and TIMIT databases for different numbers of samples n (x-axis) and its approximation by the Gompertz function (red dotted line) are depicted here. For the estimation of the average, two sets of training data were randomly selected from TIDigits and TIMIT. A training sample is a speech segment containing one word.

Therefore, it is crucial to scale the variance floor in cases with few training samples. This behavior can be easily modeled using a sigmoid function, called the Gompertz function:

$$G(n;a,b,c) = a \cdot e^{b \cdot e^{c \cdot n}} \tag{1}$$

with the parameters a, b and c adapted to the data. In Fig. 6 the dotted red line depicts the estimated Gompertz function $v(n) = f(G(n;a,b,c))$, that best approximates the variances. Once the behavior of the variances is estimated, $v(n)$ is normalized such that $v_{f1}(n)$ converges to 1 for $n \to \infty$, resulting in $v_{f1}(n) = \overline{X}_V(\infty)/v(n)$. However, as the number of training samples drops, a larger variance floor is required (Melin et al., 1998). The reinforcement factor $r_f(n) = 1 + e^{-n}$ compensates this effect, by increasing the value of the function $v_{f_1}(n)$ for a very small number of training samples. Finally, by multiplying the reinforcement factor $r_f(n)$ and the normalized variance function $v_{f1}(n)$ estimated above, one obtains a scaling function for the variance floor depending on the number of samples used:

$$v_f(n) = \frac{\overline{X}_V(\infty) \cdot (1 + e^{-n})}{f(G(n;a,b,c))} \tag{2}$$

The variance floor estimation used in the state-of-the-art, based on a larger number of training samples is shown in Eq. 3. Here K is the scaling constant and $\overline{X}_{V,d}$ stands for the three different types of variance floor estimations mentioned in Section 1. The first case is the average variance over all Gaussian components in each dimension d, the second type is the global variance of the speech features data in each feature dimension d and the third one is the variance of the dimension d to find in the xth percentile (in this case $K = 1$).

$$V_F(d) = K \cdot \overline{X}_{V,d} \tag{3}$$

The proposed variance floor estimation is depicted in Eq. 4. The variance floor function $v_f(n)$ and the scaling factor K are multiplied by $\overline{X}_{V,d}$. In contrast to Eq. 3, this yields a variance floor value depending not only on the dimension of the feature d but on the number of training samples n.

$$V_F^*(d,n) = K \cdot v_f(n) \cdot \overline{X}_{V,d} \tag{4}$$

This equation is used in each training iteration realized for the supervised phase of the initialization and the re-estimation of the parameters, namely the first and second steps of the system.

3.3. Model bootstrapping

The Expectation-Maximization algorithms used for the re-estimation of the HMM parameters are based on gradient descent methods. These algorithms usually get stuck in local minima. Thus, it is essential to have a model bootstrapping method which provides a suitable initialization for the HMM parameters in order to obtain good optima.

The proposed model initialization system is shown in Fig. 7. This algorithm comprises three main steps: the unsupervised training of an ergodic and full generic HMM (similar to Brandl et al. (2008) and Iwahashi (2006)), in which a shared HMM initialization model is built introducing only unlabeled training data. Next, pruning and training of the previously obtained HMM employing ML estimation is performed, using the Baum-Welch algorithm (Rabiner, 1989) and labeled training data. The pruning of the less occupied hidden states provides ergodic pruned word-level HMMs. The topology of the HMM obtained is transformed into a Bakis, i.e. left-to-right, configuration by means of a novel multiple sequence alignment proposed by Ayllon Clemente et al. (2010b). These steps constitute the basis for the construction of a new word-level HMM.

3.3.1. Unsupervised training of a generic HMM

In our initialization procedure, some minutes of arbitrary unlabeled input speech (Section 4.1) are recorded. These do not contain any of the words to be learned afterwards and the speech segments are also pronounced by different speakers. This is illustrated in Fig. 7. Next, the acoustic features of the segments are extracted and clustered by the K-Means algorithm. In Brandl (2010) the K clusters that result after applying K-Means are employed to train K three-state continuous HMMs, where each state is modeled by a GMM. However, in our approach an ergodic and generic Hidden Markov Model with K hidden states is constructed instead of the K three-state HMMs used by Brandl (2010). It is called ergodic because all possible transitions between the states are allowed (see Fig. 7 and Section 2.1) and generic because it does not represent a specific word. 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.

3.3.2. Pruning of the hidden states of the word model

Next, the ergodic and still generic HMM is retrained by means of the Baum-Welch algorithm using a labeled train-

Author's Personal Copy

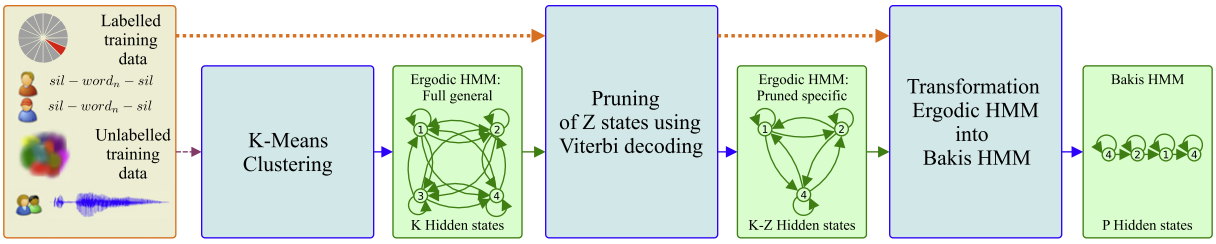


Fig. 7. Overview of the proposed model bootstrapping system to initialize the HMMs. The bootstrapping method consists of three stages. First an ergodic HMM representing a generic speech model is computed using only the unsegmented training data and K-means clustering. Here the states of the HMM allow all the transitions among the states representing general speech segments. Afterwards, this ergodic model is pruned into a specific HMM for the word-model n to be learned. In this case the unnecessary states are removed. Finally the ergodic specific HMM is converted into a left-to-right model using the proposed multiple sequence alignment.

ing data set. Afterwards, the HMM is no longer generic but model specific. In order to eliminate the hidden states that contribute less to the model and also to build a more compact word-model, a state pruning is performed (Brandl, 2010). A Viterbi decoding of the training segments is performed and then the least occupied states are pruned. We evaluate the occupation of the states using two criteria: the value of the emission probabilities o of each state for each frame of the speech sequences and the frequency of appearance of each state in the decoded sequences.

The emission probability o of a state i for a frame x of a speech segment S is defined in Rabiner (1989). After the computation of all emission probabilities of the speech frames, we obtain a vector of emission probabilities. Next we take the maximum O_i and remove the state i if O_i is less than a predefined threshold.

$$O_i = max[o_{1,1}, o_{2,1}, \ldots, o_{x,S}, \ldots, o_{m,T}] \tag{5}$$

The criterion of the frequency is simpler. After decoding the speech segments we count the frequency at which a determined state i appears. If the frequency occupation of a state 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.

3.3.3. Multiple sequence alignment

When modeling speech units we use a Bakis topology for our models. Thus, the configuration of the states has to be changed. The idea behind the Bakis-topology, i.e. left-to-right topology as explained in Section 2.1, is that transitions between states are only forwards, i.e. from left to the right. Hidden Markov Models are generative models, so we perform Viterbi decoding of the training segments to obtain the most likely succession of states generating the data. The result of the decoding procedure is N optimal path sequences, one for each of the N training samples for one individual word. Each succession of states contains information about a potential underlying configuration of these hidden states in a left-to-right topology for its corresponding training segment. Thus, all decoded sequences have to be merged into one sequence that encodes the information contained in the Viterbi decoding sequences. This is realized by means of an algorithm called 'multiple sequence alignment'.

In Bioinformatics the term sequence alignment is employed to define a way of arranging different biological sequences in order to identify similar regions that may be an indication of some kind of relationship between these sequences (Needleman and Wunsch, 1970; Smith et al., 1981; Jones et al., 1992). The goal of these algorithms is to find an alignment, which is optimal under a scoring function. In most cases, the scoring function is delivered with a similarity matrix. This matrix designates costs for the replacement of one element by a different one. Additionally the scoring function is designed in a way such that the best alignment is only to be expected if both sequences are the same.

Our purpose is to merge all the decoded sequences in a succession of states representing the best alignment between the state sequences. The multiple sequence alignment method we propose (Ayllon Clemente et al., 2010b) is depicted in Fig. 8 and comprises five main building blocks. The first block is the computation of a cost matrix D assigning different costs to the permutation of state transitions in the observed succession of states. In contrast to the approaches applied in Bioinformatics, the cost matrix does not represent the probability of a character to be replaced by another one. Each element of the cost matrix corresponds to the probability of a state to be followed by another, different, state. The element $D(i,j)$ of the cost matrix D is computed based on the frequency of the sequence $j \rightarrow i$ in the succession of states following the Viterbi decoding (see Eq. 6).

$$D(i,j) = K_\delta \frac{\sum_{seq} \delta_{j \rightarrow i}}{\sum_i \sum_{seq} \delta_{j \rightarrow i}} \quad K_\delta \leqslant 1 \tag{6}$$

The second main building block is the computation of the distances between all sequences, which are captured by the comparison matrix C. The distances between two sequences are calculated by means of a special weighted edit distance using dynamic programming. This is computed as

Author's Personal Copy

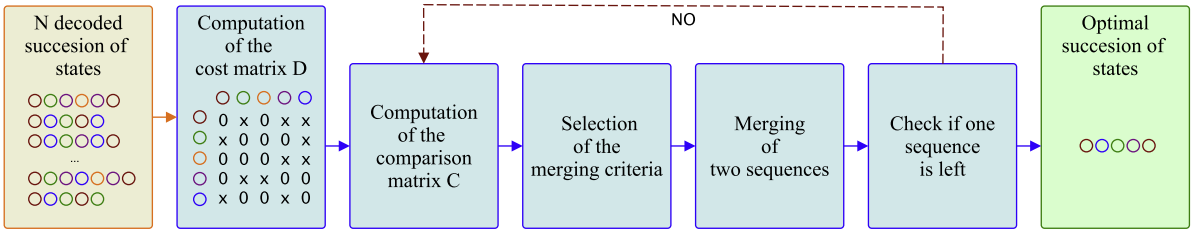


Fig. 8. Multiple sequence merging procedure. Then N decoded successions of states are merged into one optimal sequence. The values of the elements of the cost matrix D are the probabilities of an element to follow a different element. These probabilities are used to compute the similarity matrix, which provides the similarity values to compare the different sequences in the matrix 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.

a result of the construction of a distance matrix S (see Eq. 7). The distance matrix S can be considered as a modified fusion of the H-Matrix of the algorithm proposed in Smith et al. (1981) and the matrix proposed in Needleman and Wunsch (1970), often referred to as a F-Matrix. As in Smith et al. (1981), the first column and row of the matrix S is set to 0. After merging all sequences, we obtain a well-suited succession of hidden states for the word-model In Eq. 7, the variable $c(i,j)$ takes a value depending on the alignment of the states i and j of the sequences v and t to compare and the cost matrix D (see Eq. 8). More precisely, $c(i,j)$ receives the value 1 if the element to analyze of both sequences is the same. For the comparison of the distances, the distance of the sequence v over t as well as the distance t over v are computed.

$$\begin{aligned} &S(1,j) = 0; \quad S(i,1) = 0 \\ &S(i,j) = max[S(i,j-1), S(i-1,j-1)] + c(i,j) \end{aligned} \tag{7}$$

$$\begin{aligned} &c(i,j) = \begin{cases} 1 & if\ a = b \\ D(b,a) & if\ a \neq b \end{cases} \\ &with \\ &a = v(i-1) \\ &b = t(j-1) \end{aligned} \tag{8}$$

Finally, in the last blocks different sequences are merged based on their similarity measure until only one sequence is left. The proposed method enables two ways of merging. One is starting the merge with the least similar sequences (smallest 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. The merging procedure is a forward decoding of the matrix S, used to calculate the weighted edit distance explained before. This merging routine is similar to the backtracing proposed by Smith et al. (1981). However, in our method the walk is forwards. In a nutshell, the forward decoding starts at the left upper corner of the matrix and with descending progress ends at the maximum value on the right lower corner. The asymmetry in Eq. 7 enables a straightforward decoding of the new sequence as it can be realized forwards and backwards in each decoding step. After merging all sequences, we obtain a well-suited succession of hidden states for the word-model.

Once the succession of states is computed, the new HMM with Bakis topology is built. The parameters of the GMMs are preserved from the previous steps, however the transition matrix A of this new HMM with Bakis topology is filled with the values contained in the cost matrix D.

The model bootstrapping presented in this Section sets an optimal succession of states and establishes a dynamic number of hidden states combining the information contained in the speech considering the signals instead of only their duration. Our method allows us to eliminate states that are only repetitions of themselves and then contributes to shorten the number of states. The merging of the sequences is a dynamic procedure that is not fixed to one determined number of elements. 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.

3.3.4. Re-estimation of the parameters

After the initialization phase is realized and the topology of the new HMM is fixed, ML estimation is employed during the routine of splitting the mixture components (see Section 2.1). Once the final number of mixture components is reached, the parameters are refined using MAP estimation.

3.4. Large-margin discriminative training

As argued by Vapnik (1998) and based on theoretical results in machine learning, large margin classifiers are characterized by good generalization power. These classifiers generally yield much lower generalization errors in unseen test data. This was already demonstrated for support vector machines (SVM, Scholkopf and Smola (2001)). Despite the fact that several hybrid systems have been built using SVMs (Ganapathiraju et al., 2000), they are not an optimal classifier for ASR systems as Sha and Saul (2007) stated. Thus, some researchers such as Dong et al. (2008) and Li et al. (2005) study different algorithms suitable for ASR systems based on the principle of the large margin. The common idea of these algorithms is to estimate the HMM parameters in such a way that the decision

Author's Personal Copy

boundary determined by the estimated HMMs in the previous step achieves the maximum classification margin, i.e., increasing the generalization performance.

The multi-class separation margin for a word sample s_i, assuming that it belongs to class W_i, is defined by Li et al. (2005) and Chang et al. (2008) as:

$$d(s_i) = F(s_i|\lambda_{W_i}) - \max_{W_j \in \Omega; j \neq i} F(s_i|\lambda_{W_j}) = \min_{W_j \in \Omega; j \neq i} [F(s_i|\lambda_{W_i}) - F(s_i|\lambda_{W_j})] \quad (9)$$

where Ω denotes the set of all possible words, F is a discriminant function, in Li et al. (2005) $F(s_i|\lambda_W) = P(s_i)P(s_i|\lambda_W)$, and λ_{W_j} is the word-level HMM of the class W_j. The large margin method proposed by Li et al. (2005) will be referred as to LM_μ, because it updates the means μ of the GMMs.

However there is more than one definition for the margin between the different classes. As an example, Sha and Saul (2007) use the same kind of margin and as discriminant function F to measure the distance between a sample s and a class W, they apply the Mahalanobis distance M to the GMMs of the HMM λ_W defined as:

$$M(s|\lambda_W) = (s - \mu_{\lambda_W})^T \Sigma_{\lambda_W}^{-1} (s - \mu_{\lambda_W}) + \theta_{\lambda_W} \quad (10)$$

where θ_{λ_W} is a non-negative scalar offset of the GMMs of the model λ_W and $\mu_{\lambda_W}, \Sigma_{\lambda_W}$ are the means and covariance matrices of the GMMs of the model λ_W. In this approach, the means μ and the variances σ of the GMMs are adapted, so we called it $LM_{\mu,\sigma}$.

Other authors directly extend well-known discriminative training methods to the large margin principle. Dong et al. (2008) incorporate the concept of the margin to the minimum classification error (MCE) maximizing the distances between the word models. In our system presented in Ayllon 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 was not beneficial because of the very low number of training samples used. Thus, in our framework we only apply the remaining techniques explained above (LM_μ and $LM_{\mu,\sigma}$).

Following the same reasoning, as in Section 1, when a modest number of samples are employed, the HMM models may be over-trained. In this case, the word-models already seem well separated and the components of the GMMs very elongated. This can be characterized by large distances between them. Therefore, optimizing the model in each iteration with the maximization of distances as referred in the state-of-the-art methods does not improve recognition results significantly.

One heuristic method we propose is to slightly increase the variances with the goal to obtain overlapping models. We multiply the variances of the GMMs by a scaling factor S in order to intentionally detriment the estimations and to force the algorithms to recalculate the means (LM_μ) and in some cases also the variances of the GMMs ($LM_{\mu,\sigma}$). In addition to the scaling factor S, we propose two alternative strategies to the distance criteria used in combination with the algorithms proposed by Li et al. (2005) and Sha and Saul (2007). These strategies select a set of classes to optimize their models in each iteration, thereby reducing the computational cost, i.e. saving the retraining of some models, and improving the recognition results for a reduced number of training data samples. After the description of the strategies, different optimization algorithms for large-margin methods are presented.

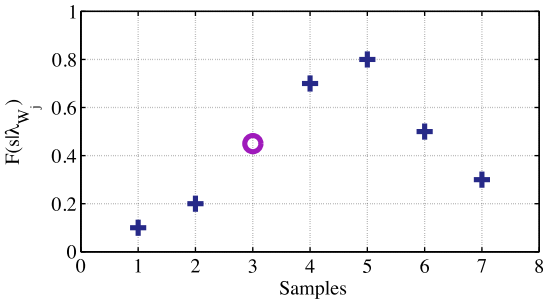


Fig. 9. Showcase of the distribution of the discriminant function. The marker "+" stands for $F(s_j|\lambda_{W_j})$, where s_j is a sample belonging to the model λ_{W_j}. Otherwise the marker "o" represents the value of $F(s_i|\lambda_{W_j})$ where s_i is a sample belonging to another model, e.g. λ_{W_i}.

3.4.1. Retraining the last introduced word-model (RLM)

In Section 3.1 we explained how and why the HMMs are estimated incrementally in our word learning system. We can take advantage of this fact optimizing only the last computed model. That means, if word-model 1 and 2 have already been estimated and optimized, word-model 3 can 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 into the system. This strategy reduces the computational cost significantly, because only one model is modified in each time step. However this simplification neglects the interactions between future unseen classes. This method is referred to as retraining last model (RLM).

3.4.2. Selecting word-models via confidence intervals (CBS)

As mentioned above, it is not feasible to obtain a reliable estimation of the word-models when using a very small number of data samples. Furthermore, not all models are uniformly estimated. Hence, instead of using the distance of the training samples to each already learned model (see Eq. 9), we propose the selection of the word-models by using the probability distribution of their discriminant functions F, i.e., confidence based selection (CBS). To construct each distribution, the discriminant function $F(s_j|\lambda_{W_j})$ is calculated for each sample s_j belonging to each class W_j, which is modeled by the word-level HMM λ_{W_j} (Fig. 9). Once the distributions of the discriminant functions are estimated, we evaluate the number of training samples s_i that may be wrongly generated by the class W_j using the

Author's Personal Copy

discriminant function $F(s_i|\lambda_{W_j})$ of the sample s_i (Fig. 9) and confidence intervals. Finally, the models of the classes W_j wrongly scoring a high confidence that sample s_i belongs to these misrecognized classes are retrained.

3.4.3. Optimization algorithms

Different optimization algorithms for large margin DT have been investigated. The most prominent algorithms are gradient descent (Li et al., 2005) and semidefinite programming (Li and Jiang, 2006; Chang et al., 2008; Sha and Saul, 2007). Semidefinite programming algorithms obtain the best results, however they have a high computational cost. In this direction, Sha and Saul (2007) mentioned that the Eq. 10 can be efficiently reparameterized to get a more elementary expression. For each word-model λ_W, a new formulation takes the means μ_{λ_W}, the covariance matrices Σ_{λ_W} and the offsets θ_{λ_W} into an expanded matrix ϕ_{λ_W}.

$$\phi_{\lambda_W} = \begin{pmatrix} \Sigma_{\lambda_W}^{-1} & -\Sigma_{\lambda_W}^{-1}\mu_{\lambda_W} \\ -\mu_{\lambda_W}^T\Sigma_{\lambda_W}^{-1} & \mu_{\lambda_W}^T\Sigma_{\lambda_W}^{-1}\mu_{\lambda_W} + \theta_{\lambda_W} \end{pmatrix} \tag{11}$$

The matrix ϕ_{λ_W} is positive semidefinite. Hence the Mahalanobis distance M can be formulated as:

$$M(s|\lambda) = z^T \cdot \phi_{\lambda_W} \cdot z \quad \text{where } z = [s, 1] \tag{12}$$

Eq. 12 becomes a linear objective function composed of semidefinite matrices and according to Sha and Saul (2007) the maximization of these margins results in convex optimization. These semidefinite problems can be solved by interior point algorithms. To guarantee polynomial time, a special-purpose solver was implemented using projected sub-gradient methods (Bertsekas, 1999; Sha and Saul, 2007).

4. Experiments

4.1. Speech databases

For the evaluation of the system, two different databases were employed. One database was applied for the unsupervised learning stage in Section 3.3.1 and the other one contains the labeled speech units to adapt the model in the following processing steps (Section 3.3.2 and 3.3.3).

In the unsupervised phase, the database used is a subset of TIMIT (Garofolo et al., 1993) with alternating utterances from men and women. TIMIT contains recordings of 630 speakers, each reading ten phonetically rich sentences.

To evaluate the following supervised phases of our incremental word learning system, a subset of the TIDIGITS corpus (Leonard, 1984) containing only male voices was used. This subset contains utterances from 112 men collected from 21 regions of the United States. There are a total of eleven words (digits) in the corpus vocabulary (digits of "1" to "9", plus "oh" and "zero"). From this subset, several datasets are generated. First, the subset is split into a test set and several training sets. The test set contains 4311 samples for continuous digits and 1232 samples including only isolated digits. Each training set contains up to 10 repetitions for each isolated digit. In contrast to Brandl et al. (2008), each digit is uttered by a different speaker. The TIMIT database was chosen for the unsupervised step in order to initialize the models with different data as the training samples of the keywords to learn because TIMIT database does not contain any digit.

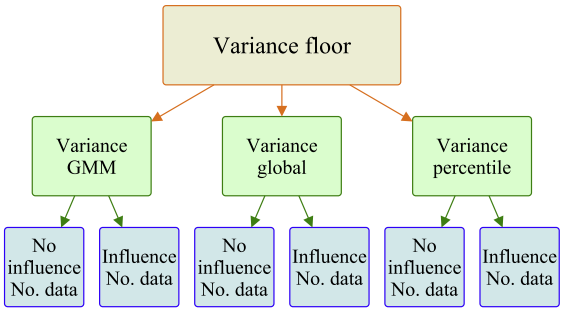


Fig. 10. Overview of the experiments realized to compare the different variance floor methods with our technique.

4.2. Settings of the incremental learning framework

We extract in our experiments 45-dimensional acoustic feature vectors comprising 15 RASTA-PLP coefficients (Hermansky and Morgan, 1994) and their first and second order time derivatives. The data is sampled at a rate of 16 kHz.

The models used in our system are continuous density HMMs (CDHMMs) with a different number of hidden states depending on the experiment, where the silence model is set to 3 emitting states in all cases. Further, each state is described by a Gaussian Mixture Model (GMM) with 3 components and a diagonal covariance matrix per state (see Section 2.1). As a baseline system, the Hidden Markov Models Toolkit (HTK) (Young et al., 2006) and a statistical Matlab Toolbox called NETLAB (Nabney, 2002) are used.

In HTK, there are some relevant parameters that influence the recognition scores. We set all parameters to the default values[1] except the minimum number of training samples, that is normally set to 3.

After evaluating two different configurations for the number of hidden states (fixed and flexible, Section 2.1), the number of hidden states for each model is set proportionally to the duration (in frames) of its samples.

[1] The word insertion penalty $-p$ and the grammar scale factor $-s$ are set zero. Especially, the pruning option -t is set to the following incremental thresholds: 250.0 150.0 1000.0.

Author's Personal Copy

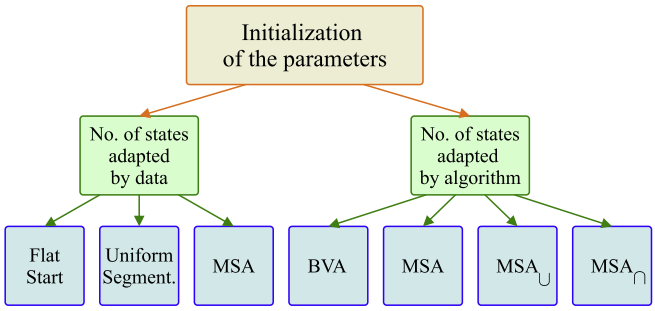


Fig. 11. Overview of the experiments realized for the evaluation of the different initialization methods. MSA stands for multiple sequence alignment and BVA for best Viterbi alignment.

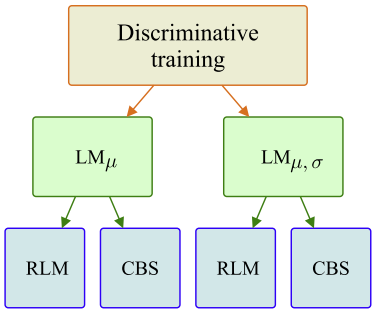


Fig. 12. Overview of the experiments realized in discriminative training. LM stands for large margin, RLM for retraining the last model and CBS for confidence based selection.

4.3. Experimental results

In this Section, we describe all the experiments performed and their evaluation. Three experimental blocks can be distinguished, namely the different variance floor approaches (Fig. 10), several initialization techniques (Fig. 11) and various discriminative training strategies (Fig. 12). All the approaches are evaluated on isolated (Section 4.3.1, 4.3.2, 4.3.3)continuous (Section 4.3.4) speech recognition

We measure the performance of our proposed methods and the state-of-the-art techniques using Word Error Rate (WER). Moreover, we realize 25-fold cross-validations of each experiment in order to obtain more reliable recognition scores.

4.3.1. Estimation of the variance floor

We compare state-of-the-art variance floor approaches to our method which incorporates a scaling factor adopted to the number of training samples (Fig. 10). The variance floor methods that we investigate are the ones computed by averaging the variances of the GMMs (V_G), the floor calculated via the global variance ($V_{\bigcirc}$) and the floor using a percentile on the complete variance set ($V_{\%}$). The variance floor method V_G computes the average variance over all Gaussian components in each feature-dimension. Then this value is scaled by some constant and used as variance floor. In the case of $V_{\bigcirc}$, the global variance of a determined number of samples is almost independent of the selection of the training samples. This makes the global variance a good measure for a variance floor formula and better than the variance of each iteration of the Baum-Welch algorithm as in V_G (Fig. 5). However it makes it difficult to apply the same variance floor multiplier in both methods. In the latter case applying a small scaling factor (independent of the additional factor dependent on the number of training samples) would not be enough in order to avoid over-fitting. However a large multiplier would be inefficient in the case of the global variance, allowing too many data to be misclassified. Therefore we selected the corresponding best-suited parameters for each technique. Different random experiments have been realized with the baseline system in order to select these suitable values. The values used for V_G and $V_{\bigcirc}$ are between [1/15, 2/3] and [1/20, 1/2] respectively. The percentile method is a compromise between previous approaches. It selects the amount of variances that are supposed to be underestimated. In other words, the variances of each dimension are floored to the xth percentile of the distribution of the variances for that dimension (Lee et al., 1992), i.e. it indicates the maximum number of Gaussian components (as a percentage of the total number of Gaussian components) to update their variances to the variance floor. In this case, the random values used are between [10, 60]. After the random experiments, 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.

We investigate our variance floor methods V^* using the system proposed by Rabiner (1989) as the baseline. The used system relies on a flexible number of states and uses as model initialization 'flat start' (Section 1). As a reference system to measure the improvement of our framework we employ the variance floor method V_G and for all the calculations the model-dependent estimation (Section 3.2). Hence the variance floor method that employs the concept of percentile or the average of the variances of the GMMs, only takes into account the variances of its own model. Similarly the global variance used for the calculus of a variance floor method is estimated taking only its own training samples of the model.

The results of the methods using the scaling factor approach that depends on the number of training data samples are shown in Table 1(b) and without the scaling factor in Table 1(a). Compared to the variance floor estimation methods (Eq. 3) in state-of-the-art, our method (Eq. 4) improves the results when a very small number of training samples are used. The technique of the variance floor percentile demonstrates that it not only provides the best results in standard approaches, but also with the proposed scaling factor where it achieves almost 25% relative improvement on isolated digit recognition against a variance floor computed using the average of the variances of the GMMs (Table 2). In Fig. 13(a) and (c) it is shown how the variance floor estimation method proposed provides consistent improvements when a very few number

Author's Personal Copy

Table 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 data set, evaluated on separated test data sets. 'No. X_i' stands for the number of training data samples. The variance floor methods are the variance floor GMM V_G, the variance floor global $V_{\bigcirc}$ and the variance floor 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. In the initialization methods, the variance floor method applied is $V^*_{\%}$. FS stands for 'flat start', US for uniform segmentation, MSA for multiple sequence alignment, BVA for best Viterbi alignment, $\cup$ for the initialization method using all the different states present in the decoded succession of states and $\cap$ employing only the common states. Finally in the discriminative training stage, the variance floor approach is $V^*_{\%}$ and the initialization used MSA. Here we differentiate the LM_{μ} method proposed by Li et al., 2005 and the $LM_{\mu,\sigma}$ from Sha and Saul, 2007. RLM is the retraining last model strategy and CBS the strategy confidence based selection. Green full boxes represent the situations where our methods improve the other techniques, otherwise the red boxes highlight where the other algorithms provide the best scores for each number of training samples. For neutral conditions, we employ grey oval boxes.

(a) V_{Floor} estimation not influenced by the No. of data

No. X_i	V_G	$V_{\bigcirc}$	$V_{\%}$
1	67.8	64.1	69.1
2	34.3	32.6	32.3
3	11.1	11.7	10.9
4	8.9	7.8	7.4
5	7.4	6.2	3.2
6	2.7	2.6	2.3
7	2.0	2.0	1.8
8	1.7	1.9	1.7
9	1.5	1.4	1.3
10	1.2	1.2	1.1

(b) V_{Floor} estimation influenced by the No. of data

No. X_i	V^*_G	$V^*_{\bigcirc}$	$V^*_{\%}$
1	48.9	57.8	61.1
2	31.7	30.6	29.8
3	7.8	9.4	7.2
4	7.5	7.1	6.4
5	5.9	5.3	3.7
6	2.4	2.1	2.0
7	1.9	2.0	1.5
8	1.5	1.5	1.4
9	1.4	1.1	1.1
10	1.0	1.0	1.0

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

No. X_i	FS	US	MSA
1	61.1	58.4	23.5
2	29.8	28.4	6.3
3	7.2	6.9	4.7
4	6.4	6.7	3.8
5	3.7	2.8	2.9
6	2.0	2.2	2.2
7	1.5	1.9	1.5
8	1.4	1.1	1.3
9	1.1	0.9	1.2
10	1.0	0.8	0.8

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

No. X_i	BVA	MSA	$MSA_{\cup}$	$MSA_{\cap}$
1	24.9	24.9	24.9	24.9
2	10.9	5.6	6.9	6.4
3	7.0	3.1	4.2	3.9
4	4.9	2.4	2.9	2.7
5	3.0	1.9	2.3	2.4
6	2.1	1.3	1.9	1.9
7	1.5	1.2	1.7	1.8
8	0.9	0.9	1.5	1.6
9	0.7	0.7	1.1	1.3
10	0.7	0.5	0.9	1.0

(e) Discriminative training

No. X_i	LM_{μ} RLM	LM_{μ} CBS	$LM_{\mu,\sigma}$ RLM	$LM_{\mu,\sigma}$ CBS
1	22.3	21.3	20.3	19.2
2	5.1	5.0	3.9	3.7
3	2.7	2.6	2.5	2.2
4	2.1	2.0	2.0	1.3
5	1.7	1.5	1.4	1.1
6	1.2	1.1	1.0	1.0
7	1.1	0.7	0.7	0.6
8	0.7	0.5	0.5	0.5
9	0.6	0.5	0.4	0.2
10	0.4	0.3	0.3	0.2

(f) Continuous recognition

No. X_i	V_G	$V^*_{\%}$	MSA	$LM_{\mu,\sigma}$ CBS
1	85.8	71.2	44.1	42.2
2	57.3	44.3	34.4	25.1
3	35.6	30.5	25.8	13.7
4	29.7	22.2	13.4	9.3
5	24.9	17.8	9.4	6.2
6	19.0	14.2	7.3	3.2
7	12.5	9.2	6.4	2.8
8	10.4	9.2	5.6	2.1
9	10.1	8.2	4.4	2.0
10	9.3	6.4	3.7	1.7

Author's Personal Copy

Table 2
Relative improvement of the Word Error Rates (WER %) of all methods. For each method, the relative WER values represent the mean of a 25-fold cross-validation on the training data set, evaluated on separated isolated and continuous test data sets. For abbreviations, see Table 1.

	VARIANCE FLOOR						INITIALIZATION							DISCRIMINATIVE TRAINING			
				Adapted to No. data			No. states via data			No. states via algorithm				LM_{μ}		$LM_{\mu,\sigma}$	
	V_G	$V_{\bigcirc}$	$V_{\%}$	V_G	$V_{\bigcirc}$	$V_{\%}$	FS	US	MSA	BVA	MSA	$MSA_{\cup}$	$MSA_{\cap}$	RLM	CBS	RLM	CBS
ISOLATED	–	3.2	12.7	15.2	16.7	24.8	24.8	28.8	44.0	46.3	61.7	45.0	42.3	66.8	72.5	74.2	78.9
CONTINUOUS	–	8.6	12.0	13.3	18.2	22.0	22.0	36.9	46.9	45.9	50.8	42.7	43.5	56.0	64.1	67.3	71.4

of training samples are used, from 1 to 5 training samples. However from 5 to 10 training samples the bars (minimum and maximum values from the cross-validation) overlap with those of the reference system (Section 4.3).

4.3.2. Model bootstrapping

Once the parameters of the system are fixed, the bootstrapping method initializes the estimates. In Section 1 we have explained that there are several initialization techniques where the most well-known are: 'flat start' and uniform segmentation. In uniform segmentation, each training segment is considered as the output of the HMM whose parameters are to be estimated. Hence, if the state that generated each frame or observation vector in the training segment was known, then the means μ, and variances σ of the GMMs of the model could be computed by averaging all the vectors associated with each state. The idea behind this so-called flat start training is similar to the uniform segmentation. By taking the same global mean and variance for all models and states, the first iteration of the training will effectively rely on a uniform segmentation of the data. Afterwards the models differentiate in the second and subsequent iterations and align with the training data as intended (Young et al. (2006)). Both methods are jointly evaluated with our model bootstrapping technique explained in Section 3.3 implementing our proposed variance floor estimation and a flexible number of hidden states. In order to compare the state-of-the-art methods with our proposed model, we do not use the number of states provided by the 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 model parameters. Afterwards, we want to compare our multiple sequence alignment (MSA) based bootstrapping with a simpler initialization method. In Brandl (2010) an approach we refer to as best Viterbi alignment (BVA) is proposed. It is based on a very similar initialization as our approach followed by a Viterbi decoding for all training data. The difference with our method is that in BVA instead of aligning and merging all sequences the state sequence with the highest probability for all training data is chosen. We compare our method with variations of our alignments in order to distinguish if the alignment proposed is essential or other alignments can provide the same performance. Here two variants of the MSA are investigated: $MSA_{\cup}$ and $MSA_{\cap}$. In both methods, instead of merging the sequences according to the algorithm described in Section 3.3.3, we combine all the different states that appear in the decoded Viterbi sequences in $MSA_{\cup}$ or only the hidden states that appear in all the sequences in $MSA_{\cap}$. In reference to the step where the states are pruned after the unsupervised training of the models, we evaluate all the above algorithms using both pruning criteria, according to the frequency each state can be found in the decoded sequences of the Viterbi decoding algorithm and the emission probabilities of the states in the decoded succession of states. The MSA model bootstrapping explained in Section 3.3 using our proposed variance floor estimation method is the final configuration used as a basis for the next step where discriminative training is employed (see Fig. 4).

The recognition scores of all the initialization approaches depicted in Fig. 11 are shown in Table 1(c) and (d). In the latter the algorithms dynamically set the number of states in contrast to the approaches (Table 1(c)) where the number of states is fixed by the duration of the training data. Hence, 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-of-the-art methods in most of the cases. We achieve more than 40% improvement relative to the baseline system and more than 15% in comparison with the best state-of-the-art method (Table 2).

Moreover, 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 achieves similar or better results compared to BVA. Especially, for a very small number of training samples MSA is superior to BVA. MSA also outperforms $MSA_{\cup}$ and $MSA_{\cap}$ in all cases. 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 Section 3.3.3 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 unsuper-

Author's Personal Copy

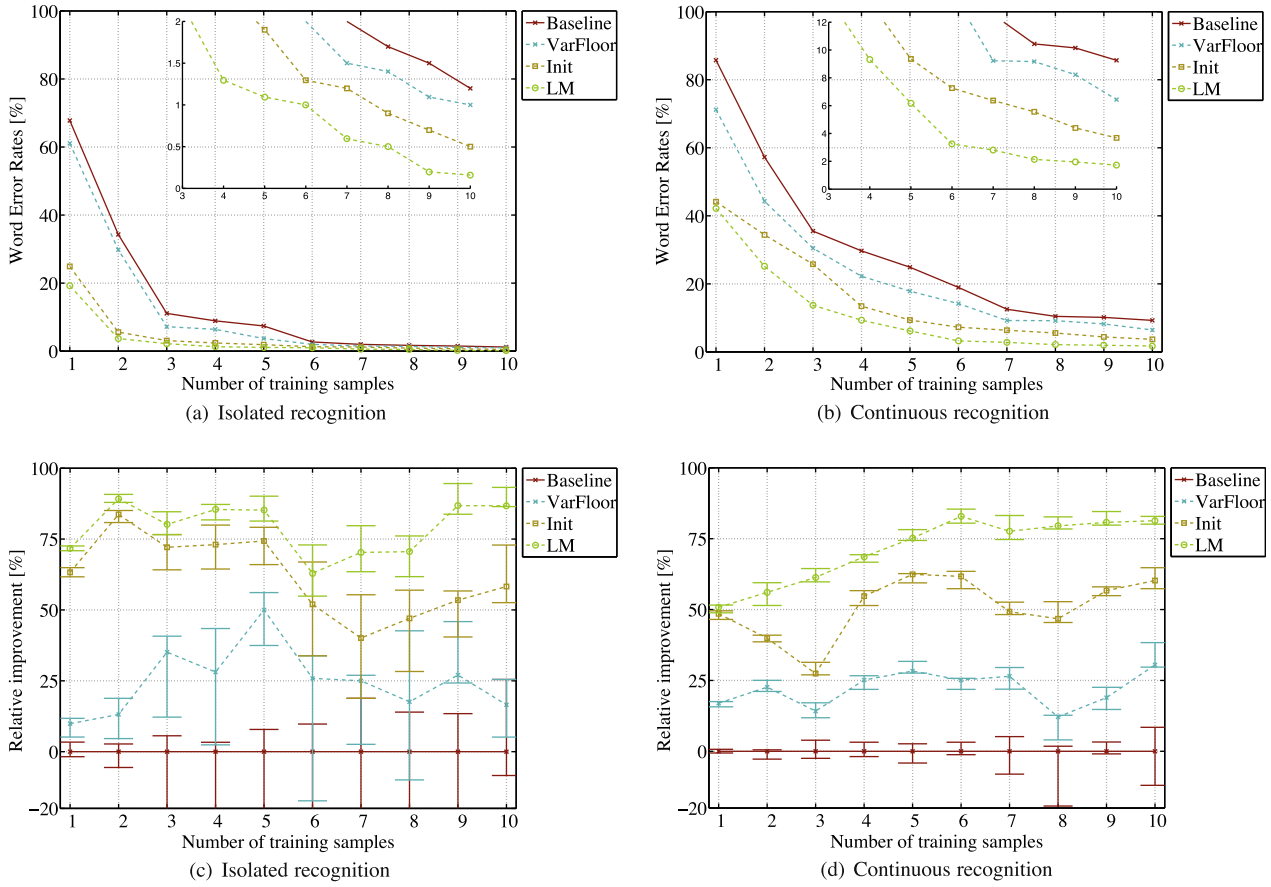


Fig. 13. Word error rates (a, b) and relative improvement (c, d) using the proposed methods in male voices. In Figs. (a, b) a zoom view is presented in order to distinguish the values of the different methods when the resolution of the scale axis is too coarse. The bars in Figs. (c, d) indicate the minimum and maximum of the improvements of the 25 cross-validations. They serve as an indication of possible overlap of the improvements in each stage of our system.

vised training of the models, the emission probabilities criterion does not improve the results much, so we evaluate all the above algorithms using the pruning criteria according to the frequency each state can be found in the decoded sequences of the Viterbi algorithm. Setting the number of states using the MSA algorithm improves the recognition scores further by more than 15%. From Table 2, one can see that BVA also obtains a large improvement in comparison with the baseline system, but our system is still superior. $\text{MSA}_{\cup}$ and $\text{MSA}_{\cap}$ methods show better scores than the baseline system, nevertheless they do not outperform our proposed multiple sequence alignment.

4.3.3. Discriminative training

We evaluate the advantage of adding the large margin discriminative training stage by analyzing the scaling factor S together with the two proposed large-margin discriminative training strategies (RLM and CBS) described in Section 3.4. Here we apply two of the state-of-the-art algorithms previously explained, LM_{μ} and $\text{LM}_{\mu,\sigma}$ (Fig. 12). In on-line learning it is crucial to reduce the computation time, hence we use for the methods referred by Li et al. (2005) the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (see Nocedal and Wright (1999)) and for the methods developed by Sha and Saul (2007) the projected subgradient method in order to ensure faster convergence as explained above. The L-BFGS method is employed with LM_{μ} and the projected subgradient method with $\text{LM}_{\mu,\sigma}$. Notice that LM_{μ} in contrast to $\text{LM}_{\mu,\sigma}$ only updates the means of the GMMs, so the values of the covariance matrices do not change.

The scaling factor is set to $S = 1.1$ for RLM. In CBS, S is set to 1 if the model of the class W_j overlaps more than 2 word-models 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 model. Setting the length of the confidence interval to $\pm\sigma$, with σ being the standard deviation of the distribution of the discriminant functions F by LM_{μ} or M by $\text{LM}_{\mu,\sigma}$, saves computation time and does not impair the recognition results.

Tables 1(e) and (f) demonstrate that large-margin discriminative training strategies in combination with MSA model bootstrapping outperform the baseline system, independent of the strategy used. Nevertheless, $\text{LM}_{\mu,\sigma}$ with CBS provides the best recognition scores. The LM_{μ} algorithm only updates the means of the GMMs obtaining an

Author's Personal Copy

improvement of about 70% in combination with the other stages. The update of the means jointly with the variances increases the improvement by about 5% in all cases, obtaining almost 80% improvement for the whole system for isolated digits. In contrast to the other proposed techniques the large margin strategies show significant improvements for all the cases. As can be seen from Fig. 13(c) the minimum and maximum values of the improvements of the cross-validations for the independent initialization and the subsequent large margin stage do not overlap.

4.3.4. Isolated learning, but continuous recognition

All the experiments explained above are realized in an isolated digit recognition scenario, where the system operates on isolated words. This is the easiest form of recognition to perform because the words are already segmented. In continuous speech, it is difficult to determine when a word starts and ends in order to get a proper label for the training phase. Another problem is 'coarticulation'. The synthesis of each phone is slightly modified depending on the pronunciation of surrounding sounds, and in the same way it also happens with the start and end of the words. Finally, another factor is the rate of the speech. In natural speech, speakers talk faster without emphasizing each word, this complicates the recognition process.

In continuous recognition the Viterbi algorithm is modified to include not only the hypothesis of different states, but also the optimal path of the recognized words. We use an alternative formulation to the Viterbi algorithm called the token passing algorithm, which is included in HTK Toolkit and described by Young et al. (2006).

The incorporation of skip transitions enable more flexible models not forcing the HMM to visit all states. Here we perform two experiments, one with all isolated training data from the TIDigits database and another with few examples applying different skip transition configurations. Afterwards we repeat all the experiments explained above but using the continuous TIDigits test set.

After the analysis of the different skip transition configurations we confirmed that one skip transition provides better scores when all the training data is used, because it allows shorter durations of the words that are learned in slow rate isolated learning. However when we train with few samples there are too many parameters for the low number of data samples provided. In these conditions adding skip transitions only increases the number of parameters without improving the recognition results.

In Table 1(f) the results obtained by the best approaches in each step are shown. We observe that although the results are not as good as in isolated digit recognition, our methods notably improve the recognition scores. The minimum and maximum values of the improvements of the recognition scores of our proposed algorithms are shown in Fig. 13(d) where these values (bars) do not overlap with the scores obtained in the baseline system. The introduction of our proposed variance floor method and the initialization step achieve an improvement of 20% and more than 50%, respectively against the baseline system. In the discriminative training phase, the subsequent large margin discriminative training strategies yield a final improvement of 70%.

5. Discussion

In this paper we proposed an incremental word learning framework that can cope with few training samples and different speakers. The problem we addressed is in principle similar to enrolling a telephone list in a voice-activated mobile phone. However in these applications usually a very well trained sub-unit model (e.g.: phones, triphones) is used as a basis to build word-models. On the other hand our inspiration is the way little children acquire language. We therefore focus on the learning of novel words with as little a priori knowledge as possible. The usage of limited training data reduces the time required to learn a novel word. In our framework, as in most ASR systems, we use continuous density Hidden Markov Models (CDHMMs) with observation probabilities modeled by GMMs. In this scenario, we addressed several issues to cope with a limited number of training samples and at the same time maintain a good generalization performance. These aspects are the estimation of variance floors to avoid over-fitting, the initialization of the estimates and the discriminative training.

First, we investigated how to avoid over-fitting under sparse training data conditions by applying a variance floor threshold depending on the training data available in each case. Section 4.3.1 showed that the proposed variance floor threshold is superior to the state-of-the-art methods (V_G, $V_{\bigcirc}$ and $V_{\%}$). Despite these improvements of the variance floor estimation in isolated recognition, it is difficult to evaluate the recognition scores because of the partial overlap of the bars (minimum and maximum relative improvements WER%) between our proposed model and the reference system V_G. In contrast, there is no overlap in continuous recognition.

In addition to over-fitting we also handle the initialization problem in order to improve the starting condition of the ML estimation. We presented a multiple sequence bootstrapping technique. We have shown that this approach is superior to state-of-the-art methods, e.g. 'flat start' and uniform segmentation. Our proposed initialization achieves an improvement of 44% with respect to the baseline system in contrast to 28.8% and 24.8% for 'flat start' and uniform segmentation (see Table 2). This improvement of our system is not uniform as we can observe in Table 1(c). A large portion of the improvement occurs if the number of training samples is very low. In this case, the sequence of states provided by the MSA algorithm is truncated to a fixed predefined number of hidden states. This kind of fixing the length of the HMM is not optimal for our bootstrapping and it is only superior in the first experiment when the number of training samples is very low. In contrast to this, the second experiment

Author's Personal Copy

(Table 1(d)) outperforms the other alignments presented. Here it is also shown that although the first phase of the unsupervised initialization is common for all the initialization algorithms proposed, 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), merging only the states that are present in all the sequences ($MSA_{\cup}$) or instead taking all the different states contained in the sequences in order to keep the maximum information ($MSA_{\cap}$). The improvement obtained is more than 50% in continuous speech and even better for isolated speech recognition using MSA. Regarding the consistency of the results, we can observe in Fig. 13(d) that our initialization approach provides substantial and consistent improvements in continuous speech and almost as good results as in isolated learning. In both cases the error bars overlap only in few cases.

In the last stage the word-models are optimized using large margin discriminative training. We investigated here two algorithms, one where only the means μ of the GMMs are updated (LM_{μ}) and another were also the variances σ of the GMMs are adapted ($LM_{\mu,\sigma}$). We compared the improvement obtained (using our strategies) by updating the means and the variances of the GMMs in contrast to adapting only the means (reducing the computational cost). We observe that the combination of $LM_{\mu,\sigma}$ and the CBS strategy provides the best results however it has a high computational cost. A compromise between the recognition scores and the computation time has to be achieved. The improvements obtained in the large margin discriminative training stage are also large and reliable for isolated and continuous speech as Fig. 13(c) and (d) show. Here no overlap of the error bars between the improvements obtained in this stage and the results obtained in the initialization step occurs.

We observe that thanks to our proposed system we obtain ever slightly better results in isolated digit recognition with only half the number of training samples (1.1% with 5 training samples compared to 1.2% with 10 training samples). Another important observation of the present work is that although all the words are learned in isolation, they are then recognized in continuous speech thereby yielding very promising results. When only 10 training samples are used, our framework has only 1.7% WER, in contrast to the 9.6% for the baseline system and the 0.5% obtained if all the samples (in isolation and in continuous speech) are employed.

In contrast to the systems of Roy (2003) and Bosch et al. (2009) our approach does so far not target the learning of new words embedded in continuous speech. Yet for the efficient learning with very few training samples, the main focus of our approach, it shows significant improvements over state-of-the-art methods and achieves very encouraging results.

Our target for the future is to go one step further integrating the system in a natural environment where the words are learned and recognized in continuous speech, similar to the systems of Roy (2003) and Bosch et al. (2009). Additionally, some questions are still open. The most relevant is how to select the most representative samples of a word model in order not to save all the samples for the discriminative stage. Some researchers such as Wu et al. (2007) and Lin and J. (2011) have started to investigate this kind of task. However, they are not yet optimally solved in those cases where the number of samples is limited.

6. Conclusion

We presented an HMM framework able to efficiently deal with a speaker-independent incremental word learning system in an interactive scenario. The system consists of three main stages. The first stage is the initialization of the estimates, where we have presented a bootstrapping technique that makes use of a multiple sequence alignment method to initialize the parameters and determines the number of hidden states of each model at the same time. In comparison with our previous work (Ayllon Clemente et al., 2010b; Ayllon Clemente et al., 2010a), here we compared our algorithm to other state-of-the-art methods as flat start and uniform segmentation, and to other alignments methods not only BVA, but $MSA_{\cup}$ and $MSA_{\cap}$.

The next stage performs re-estimation of the HMM parameters. Once the parameters of the system are initialized, these are recomputed using ML and MAP estimation with the proposed variance floor estimation. It scales them depending on the number of training samples. Here we have realized a more extensive comparison of our variance floor estimation with other techniques that were not reflected in the previous work.

Finally, in the last stage we investigate large margin discriminative training for improving the generalization of the system. Here the GMMs of the word-model HMMs are optimized using several different strategies. Updating the means and the variances in contrast to previous work where only the means were updated and applying the CBS strategy, we report 70–80% improvement over the whole system on the TIDIGITS database. Additionally all the results were not only tested in an isolated digit framework, but also in continuous speech. Moreover, we reduce the number of training samples from 10 (1.2%) to 5 (1.1%) on isolated digit recognition and obtain an improvement of more than 80% for 10 training samples on continuous recognition.

Thus, we conclude that a large reduction of the number of training data samples is obtained by improving the estimation of the variance floor, the initialization of the models and the discriminative training. So, the proposed algorithms enable a significant reduction of the tutoring time while maintaining the generalization performance. We consider this as an important step towards efficient language acquisition. However human infants still outperform all technical systems by far.

Author's Personal Copy

Acknowledgments

We would like to thank Gerhard Sagerer, Heiko Wersing, Samuel Ngouoko, Frank Joublin, Giles Endicott and especially Alexander Denecke for many fruitful discussions and support.

References

Ayllon Clemente, I., Heckmann, M., Denecke, A., Wrede, B., Goerick, C., 2010a. Incremental word learning using large-margin discriminative training and variance floor estimation. In: Proceedings of INTERSPEECH.

Ayllon Clemente, I., Heckmann, M., Sagerer, G., Joublin, F., 2010b. Multiple sequence alignment based bootstrapping for improved incremental word learning. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Bakis, R., 1976. Continuous speech word recognition via centisecond acoustic states. In: Proceedings of the ASA Meeting (Washington, DC).

Bertsekas, D.P., 1999. Nonlinear Programming. Cambridge, MA.

Bilmes, J., 2002. What HMMS can do. Tech. rep., University of Washington, Department of Electrical Engineering.

Bimbot, F., Hutter, H.-P., Jaboulet, C., Koolwaaij, J., Lindberg, J., Pierrot, J.-B., 1998. An overview of the cave project – research activities in speaker verification.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, New York, Inc.

Boves, L., ten Bosch, L., Moore, R.K., 2007. Acorns – towards computational modeling of communication and recognition skills. In: Proceedings of the IEEE International Conference on Cognitive Informatics, IEEE Computer Society, Washington, DC, USA, pp. 349–356.

Brandl, H., 2010. A computational model to unsupervised childlike speech acquisition. Ph.D. thesis, Bielefeld University.

Brandl, H., Wrede, B., Joublin, F., Goerick, C., 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, pp. 31–36.

Chang, T.-H., Luo, Z.-Q., Deng, L., Chi, C.-Y., 2008. A convex optimization method for joint mean and variance estimation of large-margin cdhmm. In: Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Charles W. Bailey, J., 1992. Truly intelligent computers. University of Houston Libraries.

Devijver, P.A., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice Hall.

Dong, Y., Deng, L., He, X., Acero, A., 2008. Large-margin minimum classification error training: a theoretical risk minimization perspective. In: Computer, Speech and Language, vol. 22. pp. 415–429.

Fink, G.A., 2008. Markov Models for Pattern Recognition: From Theory to Applications. Springer.

Gales, M., 1999. Semi-tied covariance matrices for hidden markov models. IEEE Transactions on Speech and Audio Processing 7, 272–281.

Ganapathiraju, A., Hamaker, J., Picone, J., 2000. Hybrid SVM/HMM architectures for speech recognition. In: Proceedings of the Speech Transcription Workshop.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L., 1993. Darpa timit acoustic-phonetic continuous speech corpus CD-ROM, Technical report, Linguistic Data Consortium.

Gauvain, J.-L., Lee, C.-H., 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. IEEE Transactions on Speech and Audio Processing, vol. 2, pp. 291–298.

Geiger, J.T., Lakhal, M.A., Schuller, B., Rigoll, G., 2011. Learning new acoustic events in an hmm-based system using map adaptation. In: Proceedings of INTERSPEECH.

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., 2005. Hidden markov models for automatic annotation and content-based retrieval of images and video. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 544–551.

He, X., Deng, L., Chou, W., 2008. Discriminative learning in sequential pattern recognition – a unifying review for optimization-oriented speech recognition. IEEE Signal Processing Magazine 25, 14–36.

Hermansky, H., Morgan, N., 1994. Rasta processing of speech. IEEE Transactions on Speech and Acoustics 2, 587–589.

Huang, X., Acero, A., Hon, H.-W., 2001. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR.

Itaya, Y., Zen, H., Nankaku, Y., Miyajima, C., Tokuda, K., Kitamura, T., March, 2005. Deterministic annealing em algorithm in acoustic modeling for speaker and speech recognition. IEICE - Transactions on Information and Systems, 425–431.

Iwahashi, N., 2006. Robots that learn language: developmental approach to human–machine conversations. In: Vogt, P. et al. (Eds.), Proceedings of the International Workshop on the Emergence and Evolution of Linguistic Communication. Springer, pp. 143–167.

Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. The rapid generation of mutation data matrices from protein sequences. Bioinformatics – computer Applications in the The Biosciences 8, 275–282.

Juang, B.H., Hou, W., Lee, C.H., 1997. Minimum classification error rate methods for speech recognition. In: IEEE Transactions on Speech and Audio Processing, pp. 257–265.

Kadous, M.W., 2002. Temporal classification: Extending the classification paradigm to multivariate time series. Ph.D. thesis, School of Computer Science & Engineering, University of New South Wales.

Lee, C.H., Giachin, E., Rabiner, L.R., Pieraccini, R., Rosenberg, A.E., 1992. Improved acoustic modeling for large vocabulary continuous speech recognition. Computer Speech and Language 6 (2), 103–127.

Leonard, R., 1984. A database for speaker-independent digit recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Li, X., Jiang, H., 2006. Solving large margin estimation of HMMS via semidefinite programming. In: Proceedings of INTERSPEECH.

Li, X., Jiang, H., Liu, C., 2005. Large margin hmms for speech recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Lin, H., J., B., 2011. Optimal selection of limited vocabulary speech corpora. In: Proceedings of INTERSPEECH.

Liu, F.-H., 1994. Environmental adaptation for robust speech recognition. Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittburgh, Pennsylvania.

Markov, K., Nakamura, S., 2007. Never ending learning with dynamic hidden markov model. In: Proceedings Interspeech, ISCA.

McDermott, E., 1997. Discriminative training for speech recognition. Ph.D. thesis, Waseda University.

Melin, H., 1998. Optimizing variance flooring in HMM-based speaker verification.

Melin, H., Koolwaaij, J., Lindberg, J., Bimbot, F., 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., Lindberg, J., 1999. Variance flooring, scaling and tying for text-dependent speaker verification. In: Proceedings of the European Conference on Speech Communication and Technology. pp. 5–9.

Minematsu, N., 2010. A modulation-demodulation model of speech communication. In: Proceedings of the International Conference on Speech Prosody.

Author's Personal Copy

Mokbel, C., Collin, O., 1999. Incremental enrollment of speech recognizers. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Moore, R.K., 2007. Spoken language processing: Piecing together the puzzle. Speech Communications 49, 418–435, may.

Morgan, N., Boulard, H., Renals, S., Cohen, M., Franco, H., 1993. Hybrid neural network/hidden markov model systems for continuous speech recognition. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 7 (4), 899–916.

Nabney, I.T., 2002. NETLAB: algorithms for pattern recognition. Advances in Pattern Recognition Series. Springer.

Nathan, K., Senior, A., Subrahmonia, J., 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, pp. 3502–3505.

Needleman, S.B., Wunsch, C.D., 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., Rigoll, G., 1998. Controlling the complexity of hmm systems by regularization. In: Neural Information Processing Systems Foundation, denver, USA.

Neukirchen, C., Willett, D., Rigoll, G., 1998. Soft state-tying for hmm-based speech recognition. In: Proceedings of ICSLP. pp. 2999–3002.

Nocedal, J., Wright, S.J., 1999. Numerical Optimization. Springer Verlag, NY.

Rabiner, L.R., 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286.

Rosenberg, A.E., Siohan, O., Parthasarathy, S., 1998. Speaker verification using minimum verification error training. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Roy, D., 2003. Grounded spoken language acquisition: experiments in word learning. IEEE Transactions on Multimedia, 197–209.

Scholkopf, B., Smola, A.J., 2001. In: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA, USA.

F.Sha, L.K.Saul, 2007. Large margin hidden markov models for automatic speech recognition. In: Advances in Neural Information Processing Systems. MIT Press.

Sim, K.C., Gales, M.J.F., 2006. Minimum phone error training of precision matrix models. IEEE Transactions on Audio, Speech and Language Processing 14, 882–889.

Siu, M., Jonas, M., Gish, H., 1999. Using a large vocabulary continuous speech recognizer for a constrained domain with limited training. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. pp. 105–108.

Smith, K., 2002. Hidden markov models in bioinformatics with application to gene finding in human DNA 308–761. In: Machine Learning Project.

Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. Journal of Molecular Biology 147 (1), 195–197, March.

Stuttle, M.N., 2004. A gaussian mixture model spectral representation for speech recognition. Ph.D. thesis, Hughes Hall and Cambridge University Engineering Department.

Bosch, L., Boves, L., 2008. Language acquisition: The emergence of words from multimodal input. In: Proceedings of the International Conference on Text, Speech and Dialogue. TSD '08. Springer-Verlag, Berlin, Heidelberg, pp. 261–268.

Bosch, L., Boves, L., Van Hamme, H., Moore, R.K., 2009. A computational model of language acquisition: the emergence of words. Fundamenta Informaticae 90 (3), 229–249.

Theodoridis, S., Koutroumbas, K., 2008. In: Pattern Recognition. Academic Press, 4th Edition.

Van hamme, H., 2008. Hac-models: a novel approach to continuous speech recognition. In: Proceedings Interspeech, ISCA.

Van Segbroeck, M., Van hamme, H., 2009. Unsupervised learning of time-frequency patches as a noise-robust representation of speech. Speech Communications 51, 1124–1138, November.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley, N.Y.

Wu, Y., Zhang, R., Rudnicky, A., 2007. Data selection for speech recognition.

Young, S.J., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2006. The HTK Book, version 3.4. Cambridge University Engineering Department, Cambridge, UK.