INTERNATIONAL COMPUTER SCIENCE INSTITUTE
1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

REMAP: RECURSIVE ESTIMATION AND MAXIMIZATION OF A POSTERIORI PROBABILITIES
Application to Transition-Based Connectionist Speech Recognition

Hervé Bourlard [†], Yochai Konig[†,‡], and Nelson Morgan[†,‡]

International Computer Science Institute (ICSI), Berkeley, California[†]
EECS Department, University of California, Berkeley, California[‡]

TR-94-064
March 1995

Abstract

In this paper, we describe the theoretical formulation of REMAP, an approach for the training and estimation of posterior probabilities using a recursive algorithm that is reminiscent of the EM (Expectation Maximization) algorithm (Dempster *et al.* 1977) for the estimation of data likelihoods. Although very general, the method is developed in the context of a statistical model for transition-based speech recognition using Artificial Neural Networks (ANN) to generate probabilities for hidden Markov models (HMMs). In the new approach, we use local conditional posterior probabilities of transitions to estimate global posterior probabilities of word sequences given acoustic speech data. Although we still use ANNs to estimate posterior probabilities, the network is trained with targets that are themselves estimates of local posterior probabilities. These targets are iteratively re-estimated by the REMAP equivalent of the forward and backward recursions of the Baum-Welch algorithm (Baum *et al.* 1970; Baum 1972) to guarantee regular increase (up to a local maximum) of the global posterior probability. Convergence of the whole scheme is proven.

Unlike most previous hybrid HMM/ANN systems that we and others have developed, the new formulation determines the most probable word sequence, rather than the utterance corresponding to the most probable state sequence. Also, in addition to using all possible state sequences, the proposed training algorithm uses posterior probabilities at both local and global levels and is discriminant in nature.

Contents

1 Introduction

The ultimate goal in speech recognition is to determine the sequence of words that has been uttered. Classical pattern recognition theory shows that the best possible system (in the sense of minimum probability of error) is the one that chooses the word sequence with the maximum probability (conditioned on the evidence). If word sequence i is represented by the statistical model M_i, and the evidence (which for our purposes is acoustical) is represented by X, then we wish to choose the sequence that corresponds to the largest $P(M_i|X)$. In (Bourlard & Morgan 1994), summarizing earlier work (such as (Bourlard & Wellekens 1989)) we showed that it was possible to compute the global a posteriori probability $P(M|X)$ of a discriminant form of Hidden Markov Model (HMM) M given a sequence of acoustic vectors X. This was done in the framework of hybrid speech recognition systems using HMMs together with an Artificial Neural Network (ANN), or more particularly a Multi-Layer Perceptron (MLP), to estimate the HMM (local) emission probabilities. We had two goals in doing this:

1. To use more discriminant models that are trained according to the Maximum A Posteriori (MAP) criterion instead of the commonly used Maximum Likelihood (ML) criterion.
2. To define an approach to properly interface ANNs (and in particular, MLPs) with HMMs. In this framework it was shown that it is possible to train systems minimizing common cost functions to generate posterior probabilities of output classes conditioned on the input pattern. However this required the definition of a new HMM formalism to accommodate such probabilities.

However, in order to get reasonable results in our late-80's efforts, we had to simplify the original scheme. We now view these changes as being a consequence of our limited understanding, rather than any fundamental limitation. Despite the restricted implementations (which will be briefly described in Section 6 of this paper), we still were able to alleviate some drawbacks of the typical HMM approach, including:

1. strong distributional assumptions
2. lack of discrimination
3. little incorporation of time correlations

Despite the potential improvements over these limitations, hybrid HMM/MLP procedures still estimated probabilities for likelihood-based models. Additionally, for these models, transition and emission probabilities were described independently of each other. Nonetheless, simple systems based on this approach have performed very well on large vocabulary continuous speech recognition (Renals *et al.* 1992), generally doing as well as far more detailed and complex conventional systems.

Recent work at ICSI has provided us with further insight into the discriminant HMM, particularly in the light of recent work on transition based models (Konig & Morgan 1994; Morgan *et al.* 1994). This new perspective has motivated us to further develop the original Discriminant HMM theory (Bourlard & Morgan 1994), in which an MLP is trained to optimize the full a posteriori probabilities of Markov models given the acoustic data via conditional transition probabilities, i.e., probabilities of the next state given the current state and the current acoustic vector. This approach uses posterior probabilities at both local and global levels and is more discriminant in nature. It also has the potential of using some information about the language model (i.e., HMM topologies and transition probabilities), as contained in the training data.

In this paper, we introduce the Recursive Estimation-Maximization of A posteriori Probabilities (REMAP) training algorithm for hybrid HMM/MLP systems. The proposed algorithm models a window of possible transitions rather than picking a single time point as a transition target. Furthermore, the algorithm incrementally increases the posterior probability of the correct model, while reducing the posterior probabilities of all other models. Thus, it brings the overall system closer to the optimal Bayes classifier.

If you are familiar with HMMs and with neural networks as statistical estimators, you may want to skip the Background section of this paper; however, we still recommend that you read the next two short sections in order to understand the motivations and notation for the newer material presented in the rest of the document.

2 Motivations

As noted above, the current work is motivated by a desire to train and use statistical recognition systems that are discriminant at the global (i.e., utterance) level. However, any real system will also have some underlying focus or perspective that permits some simplifying assumptions. In our recent work, we have concentrated on the view of speech as a sequence of transitions. Perceptually, transitions are commonly viewed as the most significant aspect of speech. However, in nearly all current HMM-based speech recognizers, we find:

1. There is a lack of balance between transition probabilities (which are actual probabilities and whose values are scaled differently depending on the branching factor of HMM topologies) and emission probabilities which are likelihoods. In addition to this, given the usual assumption of independence for feature vector components, the data log likelihoods are proportional to the dimension of the feature space. As a consequence of both of these factors, transition probabilities usually have a much smaller range of values, and do not strongly affect recognition performance.[1] Sev-

[1]Actually, this problem originates from unrealistic assumptions that are made in HMM theory when factoring emission-on-transition probabilities into emission densities and transition probabilities that are independent of the acoustic data.

eral “patches” have been developed to try to minimize the impact of this problem, including:

   (a) A minimum duration phoneme model, which appears to work at least as well as more complex duration models (e.g., Gamma or Poisson-distributed durations)
   (b) Log scaling (raising to a power) of transition probabilities and language model probabilities so that they are no longer probabilities, but are more balanced with emission likelihoods. Thus, a clean mathematical theory is no longer preserved.

2. There have been attempts to model transitions by transforming non-stationary features into stationary ones. A partial solution to this problem is to use time derivative features (Furui 1986). In general, though, the problem of modeling (non-stationary) transitions is still an open one. Another step in this direction was to use RASTA processing to emphasize transitions (Hermansky *et al.* 1992). While this is sometimes helpful in reducing errors due to mismatches between training and testing conditions, the resulting observation sequence is a representation that has emphasized the regions of strong change and de-emphasized temporal regions without significant spectral change. This is a mismatch to the underlying speech model in standard HMMs, which has been designed to represent piecewise stationary signals.

While psychoacoustic experiments suggest that transitions (in the sense of temporal regions of significant spectral change) are important to speech perception, the discriminant HMM theory (Bourlard & Morgan 1994) affirms that recognition should actually be based on probabilities of transitions (in the sense of changes of model state) conditioned on observations. As shown in this paper, it is actually possible to train and to use this kind of model. While state transitions are not the same thing as observation transitions, state transition models do have the potential of alleviating the stationarity assumptions implicitly made in all current HMMs, and so there is good reason to think that they can represent spectral transitions better.

3 Definitions and Notation

We first define notation and basic terms:

- A set of HMM states $Q^* = \{q_1, \ldots, q_K\}$, from which phone and word models will be built. Each state class will be associated with a specific probability density function (PDF) or with specific statistical properties (see “conditional transition probabilities” in 5.3).

- $X = \{x_1, \ldots, x_N\}$ is a sequence of acoustic vectors that is associated with a specific utterance.

- A sub-sequence of acoustic vectors that is local to the current vector, extending c frames into the past and d frames into the future: $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$.

- The set of possible elementary speech unit HMMs: $\mathcal{M} = \{m_1, \ldots, m_u, \ldots, m_U\}$. For large vocabularies (and in our case), these elementary speech units are often phones or phone-like units. Each of those speech units are then assumed to be composed of a succession of a few discrete stationary states from Q^*. Usually, each speech unit m_u is represented in terms of a Markov chain (see next section) built up from a few elementary (stationary) states q_k from Q^*. However, in the case of the hybrid systems described here that we have used over the last few years, we have not observed any benefit in using multiple states per phone for the context-independent phone models that we have generally used. In this particular case, there is a one-to-one relation between states q_k's and phones. This is simpler to describe than multi-density phone models and will be used for the theory presented here, without loss of generality.

- A specific word or sentence model M_i is then represented as a sequence of elementary units m_u of $\mathcal{M}$ and, consequently, as a sequence of L_i discrete stationary states q_ℓ of Q^*, with $L_i \leq N$ (and, in general, $L_i \neq K$). Of course, we can have multiple instances of the same phone and state in M_i.

- M_i is defined for $i \in \mathcal{I} = \{1, 2, \ldots, I\}$, the set of possible Markov model indices; I is the number of possible Markov models (i.e., in the case of continuous speech, number of possible sentences allowed by the grammar, though this is generally infinite).

- M_{w_j}, $w_j \in \mathcal{I}$, is the Markov model associated with a specific training sequence X_j, $j = 1, \ldots, J$.

- The parameter set describing all models is defined as $\Theta = \{\lambda_1, \ldots, \lambda_u, \ldots, \lambda_U\}$, in which λ_u represents only the parameters present in m_u. Of course, the different m_u, for $u = 1, \ldots, U$ can share some common parameters. In the hybrid systems discussed in this paper, all HMMs will share the same set of parameters Θ through a common neural network, which will be parameterized in terms of Θ.

- The set of parameters that are only present in M_{w_j} will be denoted Θ_{w_j}, which is a subset of Θ.

- q^n = the HMM-state at time n.

- q_k^n means that state q_k has been occurred at time n.

- A HMM state sequence of length N: $Q = \{q^1, \ldots, q^n, \ldots, q^N\}$, $q^n \in Q^*$; a HMM state subsequence: $Q_m^n = \{q^m, q^{m+1} \ldots, q^n\}$.

- Γ_i (Γ) a path of length N (associated with a specific Q) in M_i (M).

- $P(\cdot)$ will represent probabilities, while $p(\cdot)$ will represent probability density functions (PDFs) and likelihoods.

Throughout much of this paper, the following two statistical properties (valid for both probabilities and likelihoods) will be extensively used:

$$P(a,b) = P(a|b)P(b) = P(b|a)P(a) \tag{1}$$

$$P(a) = \sum_{\ell} P(a,b_{\ell}) \tag{2}$$

if events b_ℓ are mutually exclusive and $\sum_\ell P(b_\ell) = 1$.

4 Background

Whenever a new discovery is reported to the scientific world, they say first, 'It is probably not true.' Thereafter, when the truth of the new proposition has been demonstrated beyond question, they say, 'Yes, it may be true, but it is not important.' Finally, when sufficient time has elapsed fully to evidence its importance, they say, 'Yes, surely it is important, but it is no longer new.'

– Michel Eyquem Montaigne, 1533 - 1592 –

4.1 Hidden Markov Models (HMMs)

In this section we give a short review of the classical HMM approach to speech recognition. For a more complete explanation, see (Huang *et al.* 1990; Levinson *et al.* 1983; Rabiner 1989).

4.1.1 Brief Description

One of the greatest difficulties in speech recognition is to model the inherent statistical variations in speaking rate and pronunciation. An efficient approach consists of modeling each speech unit (e.g., words, phones, triphones, or syllables) by an HMM (Jelinek 1976; Rabiner 1989). A number of large-vocabulary, speaker-independent, continuous speech recognition systems have been based on this approach.

In order to implement practical systems based on HMMs, a number of simplifying assumptions are typically made about the signal. For instance, although speech is a non-stationary process, HMMs model the sequence of feature vectors as a piecewise stationary process. That is, an utterance $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ is modeled as a succession L discrete stationary states $q_\ell \in Q^*$, with instantaneous transitions between these states. In this case, a HMM is defined (and represented) as a stochastic finite state automaton with a particular topology (generally strictly left-to-right, since speech is sequential). The approach defines two concurrent stochastic processes: the sequence of HMM states (modeling the temporal structure of speech), and a set of state output processes (modeling the [locally] stationary character of the speech signal). The HMM is called a "hidden" Markov model because there is an underlying stochastic process (i.e., the sequence of states) that is not observable, but that affects the observed sequence of events. It is called "Markov" because the statistics of the current state are modeled as being dependent only on the current and the previous state (for the first-order Markov case).

Ideally, there should be a HMM for every possible utterance. However, this is clearly infeasible for all but extremely constrained tasks; generally a hierarchical scheme must be adopted to reduce the number of possible models. First, a sentence is modeled as a sequence of words. To further reduce the number of parameters (and, consequently, the required amount of training material) and to avoid the need of a new training each time a new word is added to the lexicon, sub-word units are usually preferred to word models. Although there are good linguistic arguments for choosing units such as syllables or demi-syllables, the unit most commonly used is the phone (or context-dependent versions such as the triphone). This is the unit that we have generally used in our work, resulting in a selection of between 50 and 70 subword models. In this case, word models consist of concatenations of phone models (constrained by pronunciations from a lexicon), and sentence models consist of concatenations of word models (constrained by a grammar).

Once the topology of the HMMs has been defined (usually by an ad hoc procedure), the HMM training and decoding criterion is based on the posterior probability $P(M_j|X,\Theta)$ that the acoustic vector sequence X has been produced by M_j given the parameter set Θ. In the following, this will be referred to as the Bayes or the Maximum A posteriori (MAP) criterion.

During *training*, we want to determine the set of parameters $\hat{\Theta}$ that will maximize $P(M_{w_j}|X_j,\Theta)$ for all training utterances X_j, $j = 1,\ldots,J$, associated with M_{w_j}[2], i.e.,

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \prod_{j=1}^{J} P(M_{w_j}|X_j,\Theta) \tag{3}$$

During *recognition* of an unknown utterance X, we have to find the best model M_j, $j \in \mathcal{I}$, that maximizes $P(M_j|X,\Theta)$ given a fixed set of parameters Θ and an observation sequence X. An utterance X will then be recognized as the word sequence associated with model M_j such that:

$$M_j = \underset{M_i}{\text{argmax}}\ P(M_i|X,\Theta) \tag{4}$$

Ideally we thus want to optimize (3) during training, and this will be the main aim of this work. However, in standard HMMs, this problem is usually simplified by using Bayes' rule which expresses $P(M_i|X,\Theta)$ as

$$P(M_i|X,\Theta) = \frac{p(X|M_i,\Theta)P(M_i|\Theta)}{p(X|\Theta)} \tag{5}$$

and separates the probability estimation process into two parts: (1) the *language modeling* which does not depend on the acoustic data and (2) the *acoustic modeling*.

[2] M_{w_j} represents the model associated with the specific acoustic sequence X_j that is known at training time.

4.1.2 Language Modeling

The goal of the language model is to estimate prior probabilities of sentence models $P(M_i|\Theta)$. However, this language model is usually assumed to be independent of the acoustic model parameters and is described in terms of an independent set of parameters Θ^*. At training time, Θ^* is learned separately, which is sub-optimal but convenient. These language model parameters are commonly estimated from large text corpora or from a given finite state automaton from which N-grams (i.e., the probability of a word given the (N-1) preceding words) are extracted. Typically, only bi-grams and tri-grams are currently used.

It has to be noted here that, according to what is trained and what M_i represents, we get a different meaning for the language model; in some cases that language model could preferably be learned directly from the acoustic data. For more discussion about this see Section 4.1.6 on "Priors and HMM Topology".

4.1.3 Acoustic Modeling

The goal of acoustic modeling is to estimate the data-dependent probability densities $\frac{p(X|M_i,\Theta)}{p(X|\Theta)}$. In mainstream approaches to this process, parameters from other models do not affect the estimates for any particular model. In this case, since $p(X|M_i,\Theta)$ is conditioned on M_i it only depends on the parameters of M_i. Therefore, it can be rewritten as $p(X|M_i,\Theta_i)$.

Given a transcription in terms of the speech units being trained, the acoustic parameter set Θ estimation is trained according to

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \frac{p(X_j|M_{w_j},\Theta_{w_j})}{p(X_j|\Theta)} \tag{6}$$

for all training utterances X_j known to be associated with a Markov model M_{w_j} obtained by concatenating the elementary speech unit models associated with X_j. Since the models are mutually exclusive and $\sum_{i\in\mathcal{I}} P(M_i|\Theta) = 1$ (i.e., what has been pronounced actually corresponds to one of the models[3]), the denominator in (5) and (6) can be rewritten as:

$$p(X_j|\Theta) = \sum_{i=1}^{I} p(X_j|M_i,\Theta_i)P(M_i|\Theta_i) \tag{7}$$

where the summation extends over all possible (rival) sequences of elementary HMMs. In practice, the second factor in (7) is defined by the language model $P(M_i|\Theta^*)$.[4]

At *recognition time*, $p(X_j|\Theta)$ is a constant, since the model parameters are fixed. However, at *training time*, the parameters of the models are being adapted by the training algorithm; therefore (7) and (6) depend on the parameters of all models. Of course, this is also the case when one tries to optimize (3) directly (see Section 11).

[3]This is an issue when there can be utterances that are outside of the lexicon.

[4]In Section 11, we show that summing over all possible models or over all possible rival models ($i \neq \ell$) is equivalent.

Maximization of (6) is equivalent to maximization of a related discriminant criterion referred to as *mutual information*[5] (Cover & Thomas 1991)

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \ \log \frac{p(X_j|M_{w_j}, \Theta_{w_j})}{p(X_j|\Theta)} \tag{8}$$

Several algorithms have been developed to optimize (6) or (8) (Bahl *et al.* 1986; Brown 1987; Chow 1990; Normandin *et al.* 1994). See Section 11 for further discussion and comparison with other discriminant algorithms or the work presented here.

Since optimization of (3), (6) or (8) in the whole parameter space is not easy, the problem is usually simplified by disregarding the conditional dependence of X on Θ during training. In this case, training according to (3), (6) or (8) is equivalent to

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \ \prod_{j=1}^{J} p(X_j|M_{w_j}, \Theta_{w_j}) \tag{9}$$

When used for training, this is usually called the Maximum Likelihood (ML) criterion, emphasizing that optimization (i.e, maximization of $p(X_j|M_{w_j}, \Theta_{w_j})$) is performed in the parameter space of the Probability Density Function (PDF) or likelihood.

At *recognition* time, $P(M_i|X, \Theta)$ is estimated for all possible M_i allowed by the language model. In this case $p(X|\Theta)$ is actually a constant, since the parameters are fixed and X given. Then solution to (4) is equivalent to

$$M_j = \underset{M_i}{\text{argmax}} \ p(X|M_i, \Theta_i)P(M_i|\Theta^*) \tag{10}$$

in which $p(X|M_i, \Theta_i)$ and $P(M_i|\Theta^*)$ are estimated separately from the acoustic and language models.

4.1.4 Likelihood Estimation and Training

Both training and recognition thus require the estimation of the likelihood $p(X|M_i, \Theta_i)$ which is given by:

$$p(X|M_i, \Theta_i) = \sum_{\{\Gamma_i\}} p(X, \Gamma_i|M_i, \Theta_i) \tag{11}$$

in which $\{\Gamma_i\}$ represents the set of all possible paths of length N in M_i. If q_ℓ^n denotes the state q_ℓ observed at time $n \in [1, N]$, it is easy to show [see, e.g., (Bourlard & Morgan 1994)] that $p(X|M_i, \Theta_i)$ can be calculated by the forward recurrence of the popular *forward-backward algorithm* (Baum *et al.* 1970; Baum 1972; Liporace 1982)

$$p(X_1^n, q_\ell^n|M_i, \Theta_i) = \sum_{k=1}^{L_i} p(X_1^{n-1}, q_k^{n-1}|M_i, \Theta_i)p(x_n, q_\ell^n|X_1^{n-1}, q_k^{n-1}, M_i, \Theta_i) \tag{12}$$

[5]See Section 11 for further discussion about this.

in which $p(X, q_\ell^n|M_i, \Theta_i)$ represents the likelihood that X is produced by M while associating x_n with state q_ℓ; X_1^n stands for the partial sequence $\{x_1, x_2, \ldots, x_n\}$ of acoustic vectors.

Sometimes it is desirable to replace the full likelihood by a *Viterbi* approximation in which only the most probable state sequence capable of producing X is taken into account. In this case, the sum in (11) is replaced my a *max* operator and likelihood $p(X|M_i, \Theta_i)$ is approximated by:

$$\overline{p}(X|M_i, \Theta_i) = \max_{\{\Gamma_i\}} p(X, \Gamma_i|M_i, \Theta_i) \qquad (13)$$

which can be calculated by a Dynamic Programming (DP) recurrence (called the Viterbi search or *Viterbi algorithm*):

$$\overline{p}(X_1^n, q_\ell^n|M_i, \Theta_i) = \max_k \left[\overline{p}(X_1^{n-1}, q_k^{n-1}|M_i, \Theta_i)p(x_n, q_\ell^n|X_1^{n-1}, q_k^{n-1}, M_i, \Theta_i)\right] \qquad (14)$$

For both "full" likelihood and Viterbi approximation, probabilities $p(X|M_i, \Theta_i)$ and $\overline{p}(X|M_i, \Theta_i)$ can be expressed in terms of $p(x_n, q_\ell^n|X_1^{n-1}, q_k^{n-1}, M_i, \Theta_i)$, where X_k^n is the partial acoustic vector sequence $\{x_k, x_{k+1}, \ldots, x_n\}$.

Recapitulating, some of the features commonly associated with the estimation and training of HMMs, include:

- Assumption of piecewise stationarity, i.e., that speech can be modeled by a Markov state sequence, for which each state has stationary statistics,
- Optimizing the language model $P(M_i|\Theta^*)$ separately from the acoustic model,
- Disregarding the dependence of the estimate of $p(X)$ on the model parameters during training. The acoustic models are then defined and trained on the basis of likelihoods $p(X|M_i, \Theta_i)$ (i.e., production-based models) instead of a posteriori probabilities (i.e., recognition-based models) or MMI criteria, which limits the discriminant properties of the models.

Additionally, several additional assumptions are usually required to make the estimation of $p(X|M_i, \Theta_i)$ [or its Viterbi approximation $\overline{p}(X|M_i, \Theta_i)$] tractable (Bourlard & Morgan 1994):

- Acoustic vectors are not correlated (i.e., observation independence). The current acoustic vector x_n is assumed to be conditionally independent of the previous acoustic vectors (e.g., X_1^{n-1}). To limit the impact of this assumptions, acoustic vectors at time n are usually complemented by their first and second time derivatives (Furui 1986; Poritz & Richter 1986) computed over a span of a few frames, allowing very limited acoustical context modeling. Another solution to limit this assumption is to consider a few adjacent frames (typically 3-5 frames in total) on which linear discriminant analysis is performed to reduce the dimension of the acoustic features (Haeb-Umbach & Ney 1992).

- Markov models are first-order Markov chains, i.e., the probability that the Markov chain is in state q_ℓ at time n depends only on the state of the Markov chain at time $n-1$, and is conditionally independent of the past (both the past acoustic vector sequence and the states before the previous one).

Given these assumptions, $p(X|M_i, \Theta_i)$ and $\overline{p}(X|M_i, \Theta_i)$ can be estimated (Bourlard & Morgan 1994) by replacing $p(x_n, q_\ell^n | X_1^{n-1}, q_k^{n-1}, M_i, \Theta_i)$ in (12) and (14) by the product of *emission-on-transition probability densities* $p(x_n | q_\ell^n, q_k^{n-1}, M_i, \Theta_i)$ and *transition probabilities* $p(q_\ell^n | q_k^{n-1}, M_i, \Theta_i)$. Often, emission-on-transition probability densities are further simplified (to reduce the number of free parameters) by assuming that the current acoustic vector x_n depends only on the current state of the process q_ℓ^n, which reduces the former to *emission probability densities* $p(x_n|q_\ell)$.

HMM training then is simplified to be estimation of transition probabilities and emission PDFs associated with each state (or with each transition, in the case of emission on transitions). Additionally, one has to make distributional assumptions about the emission PDF, e.g., independence of discrete features or a mixture of multivariate Gaussian distributions with diagonal-only covariances of continuous features.

The most popular approach to iteratively maximize

$$\prod_{j=1}^{J} p(X_j | M_{w_j}, \Theta_{w_j}) \tag{15}$$

has been described in a number of classic papers (Baum & Petrie 1966; Baum *et al.* 1970; Baum 1972; Liporace 1982). Starting from initial guesses Θ^0, the model parameters are iteratively updated according to the "Forward-Backward" algorithm [or equivalently the Expectation-Maximization (EM) algorithm (Dempster *et al.* 1977)] so that (15) is maximized at each iteration. This kind of training algorithm, often referred to as Baum-Welch training in the particular case of HMMs, can also be interpreted in terms of gradient techniques (Levinson *et al.* 1983; Levinson 1985). Although this algorithm is not described here, we strongly recommend these references to readers who are not familiar with them since the ideas expressed there will be extended to posterior probabilities and hybrid systems in this paper. For recognition, powerful algorithms referred to as Stack-Decoding or A* decoding have been developed to find the N-best models M_i maximizing $p(X|M_i)$ or $P(M_i|X)$ if there is a grammar [see, e.g., (Bahl *et al.* 1983)].

In the case of Viterbi criterion, the parameters of the models are optimized iteratively to find the best parameters and the best state sequence (i.e., the best segmentation in terms of the speech units used) maximizing

$$\prod_{j=1}^{J} \overline{p}(X_j | M_{w_j}, \Theta_{w_j}) \tag{16}$$

Each training iteration consists of two steps. In the first step, we use the old parameter values (or initial values) to determine the new best path matching the training sentences against the associated sequence of Markov models [by using (14)]. In the second step, we use this path to re-estimate the new parameter values; backtracking of the optimal paths provides us with the number of observed transitions between states (to update the transition probabilities) and the acoustic vectors that have been observed on each state (to update the parameters describing the emission probabilities). This process can be proved to converge to a local minimum. For recognition, algorithms based on DP have been developed to find the best word sequence model M_i which maximizes $\overline{p}(X|M_i)$ (Vintsyuk 1971; Ney 1984).

4.1.5 HMM Advantages and Drawbacks

Standard HMM procedures, as defined above, have been very useful for speech recognition, and a number of laboratories have demonstrated large-vocabulary (1,000-65,000 words), speaker-independent, continuous speech recognition systems based on HMMs (Lee 1989; Kubala *et al.* 1988). HMMs can deal efficiently with the temporal aspect of speech (including temporal distortion or time warping) as well as with frequency distortion. There are powerful training and decoding algorithms that permit efficient training on very large databases, and recognition of isolated words as well as continuous speech. Given their flexible topology, HMMs can easily be extended to include phonological rules (e.g., building word models from phone models) or syntactic rules. For training, only a lexical transcription is necessary (assuming a dictionary of phonological models); explicit segmentation of the training material is not required.

However, the assumptions that permit HMM optimization and improve their efficiency also, in practice, limit their generality. As a consequence, although the theory of HMMs can accommodate significant extensions (e.g., correlation of acoustic vectors, discriminant training, ...), practical considerations such as number of parameters and train-ability limit their implementations to simple systems usually suffering from several drawbacks including:

- Poor discrimination due to training algorithms that maximizes likelihoods instead of a posteriori probabilities (i.e., the HMM associated with each speech unit is trained independently of the other models). Discriminant learning algorithms do exist for HMMs (Section 11), but in general they have not scaled well to large problems.

- A priori choice of model topology and statistical distributions, e.g., assuming that the probability density functions associated with the HMM state can be described as multivariate Gaussian densities or as mixtures of multivariate Gaussian densities, each with a diagonal-only covariance matrix (i.e., possible correlation between the components of the acoustic vectors is disregarded).

- Assumption that the state sequences are first-order Markov chains.[6]

[6] This limitation remains valid for our hybrid HMM/MLP system, with the exception of the most recent developments briefly described later in this report.

- Typically, very limited acoustical context is used, so that possible correlation between successive acoustic vectors is not modeled very well. As previously mentioned, a solution that has been adopted in standard HMMs with relative success has been to complement acoustic features by their first and second time derivatives (Furui 1986; Poritz & Richter 1986) computed over a span of a few frames. Another solution which sometimes leads to some improvements is to consider a few adjacent frames (typically 3-5 frames in total) on which linear discriminant analysis is performed to reduce the dimensionality of the acoustic features while minimizing the intra-class variance and maximizing the inter-class variance (Haeb-Umbach & Ney 1992). Other approaches of interest were the use of autoregressive HMMs, as described in (Juang & Rabiner 1985; Poritz 1982), and the work of (Wellekens 1987), who explicitly modeled the correlation across several frames with a multivariate, full covariance matrix, Gaussian density defined over two consecutive acoustic vectors.[7] However, these last two solutions apparently did not lead to conclusive experimental results for reasons that have never been clearly identified.[8]

Much ANN-based ASR research has been motivated by these problems.

4.1.6 Priors and HMM Topology

As shown in the previous section, the prior probabilities of models are not used during likelihood training (or, in other words, are trained independently of the acoustic models or fixed by a priori knowledge). It is usually assumed that $P(M_i|\Theta)$ in (5) and (7) can be calculated separately (i.e., without acoustic data). In continuous speech recognition, M_i usually represents a sequence of word models for which the probability $P(M_i)$ can be estimated from a language model, usually formulated in terms of a stochastic grammar. Likewise, each word model is represented in terms of a HMM that combines phone models according to the allowed pronunciations of that word; these multiple pronunciations can be learned from the data, from phonological rules, or from both. Each phone is also represented by a HMM for which the topology is usually chosen a priori independently of the data (or, sometimes, in a very limited way, e.g., to reflect minimum or average durations of the phones). Therefore, the grammar, the lexicon, and the phone models *together* comprise the language model, specifying prior probabilities for sentences [$P(M_i)$], words, phones, and HMM states [$P(q_k)$]. These priors are encoded in the topology and associated transition probabilities of the sentence, word and phone HMMs. Usually, it is preferable to infer these priors from large text corpora, due to insufficient speech training material to derive so many parameters from the speech data. However, as seen later (see Sections 5.4 and 11), neural networks and discriminant training implicitly make use of these priors. As a consequence,

[7] This can be shown equivalent to estimating a multivariate autoregressive process (Wellekens 1987).

[8] Some plausible explanations to this discrepancy between theory and practical results include: (1) increase of number of parameters, and (2) estimating autoregressive models implicitly assumes some "smoothness" properties of the signal, which is not always true in the case of speech (and, consequently, what is gained on the one hand is lost on the other).

if the priors observed on the training data are not the same as the priors that are given by the HMM topology (and which have been a priori given or trained from an independent knowledge source), there will be a mismatch that will impact the recognition performance of the global level. Thus, it would be preferable to learn the topology of the HMMs directly from the data. This has been done in a limited way in (Wooters 1993).

4.2 Artificial Neural Networks (ANNs)

4.2.1 Multilayer Perceptrons (MLPs)

In this paper, our discussion of neural networks for speech will be limited to the Multi-Layer Perceptron (MLP), a form of ANN that is commonly used for speech recognition. However, the analyses that follow are generally extensible to other kinds of ANN, e.g., a recurrent neural network (Robinson 1994).

MLPs have a layered feedforward architecture with an input layer, zero or more hidden layers, and an output layer. Each layer computes a set of linear discriminant functions (Duda & Hart 1973) (via a weight matrix) followed by a nonlinear function, which is often a sigmoid function

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{17}$$

As discussed in (Bourlard & Morgan 1994), this nonlinear function performs a different role for the hidden and the output units. On the hidden units, it serves to generate high order moments of the input; this can be done effectively by many nonlinear functions, not only by sigmoids. On the output units, the nonlinearity can be viewed as a differentiable approximation to the decision threshold of a threshold logic unit or perceptron (Rumelhart *et al.* 1986), i.e., essentially to count errors. For this purpose, the output nonlinearity should be a sigmoid or sigmoid-like function. Alternatively, a function called the *softmax* can be used. For an output layer of K units, this function would be defined as

$$f(x_i) \;=\; \frac{\exp(x_i)}{\sum_{n=1}^{K} \exp(x_n)} \tag{18}$$

It can be proved that MLPs with enough hidden units can (in principle) provide arbitrary mappings $g(x)$ between input and output. The MLP parameter set Θ (the elements of the weight matrices) are trained to associate a "desired" output vector with an input vector. This is generally achieved via the Error Back-Propagation (EBP) algorithm (Rumelhart *et al.* 1986) that uses a steepest descent procedure to iteratively minimize a cost function in their parameter space. Since in our approach the HMMs will be described by the parameters of the neural network, we also denote the MLP parameter space by Θ.

Popular cost functions are, among others, the Mean Square Error (MSE) criterion:

$$E = \sum_{n=1}^{N} \parallel g(x_n, \Theta) - d(x_n) \parallel^2 \tag{19}$$

or the relative entropy criterion[9]:

$$E_e = \sum_{n=1}^{N} \sum_{k=1}^{K} d_k(x_n) \ln \frac{d_k(x_n)}{g_k(x_n, \Theta)} \tag{20}$$

where $g(x_n, \Theta) = (g_1(x_n, \Theta), \ldots, g_k(x_n, \Theta), \ldots, g_K(x_n, \Theta))^t$ represents the actual MLP output vector (depending on the current input vector x_n and the MLP parameters Θ), $d(x_n) = (d_1(x_n), \ldots, d_k(x_n), \ldots, d_K(x_n))^t$ represents the desired output vector (as given by the labeled training data), K the total number of classes, and N the total number of training patterns.

MLPs, as well as other neurally-inspired architectures, have been used for many speech-related tasks. For instance, for some problems the entire temporal acoustic sequence is processed as a spatial pattern by the MLP. For isolated word recognition, for instance, each word can be associated with an output of the network. However, this approach has not been useful for continuous speech recognition and will not be discussed further here.

4.2.2 Motivations

ANNs have several advantages that make them particularly attractive for ASR, e.g.:

- They can provide discriminant learning between speech units or HMM states that are represented by ANN output classes. That is, when trained for classification (using common cost functions such as MSE or relative entropy), the parameters of the ANN output classes are trained to minimize the error rate while maximizing the discrimination between the correct output class and the rival ones. In other words, ANNs not only train and optimize the parameters of each class on the data belonging to that class, but also attempt to reject data belonging to the other (rival) classes. This is in contrast to the likelihood criterion, which does not lead to minimization of the error rate.

- Because ANNs can incorporate multiple constraints and find optimal combinations of constraints for classification, features do not need to be assumed independent. More generally, there is no need for strong assumptions about the statistical distributions of the input features (as is usually required in standard HMMs).

- They have a very flexible architecture which easily accommodates contextual inputs and feedback, and both binary and continuous inputs.

[9] In a number of references, including (Bourlard & Morgan 1994), this criterion is defined differently. In particular, the desired outputs are sometimes assumed to be independent, binary random variables and as a result this criterion gets a different form (which is sometimes called the cross entropy (Richard & Lippmann 1991)). However, viewing the network outputs as a posterior distribution over the values of one random variable (class conditioned on acoustic data), a discrete version of the classical definition of relative entropy may be used, as given here.

- ANNs are typically highly parallel and regular structures, which makes them especially amenable to high-performance architectures and hardware implementations.

A general formulation of statistical ASR can be summarized simply by a question: how can an input sequence (e.g., a sequence of spectral vectors) be explained in terms of an output sequence (e.g., a sequence of phones or words) when the two sequences are not synchronous (since there are multiple acoustic vectors associated with each pronounced word or phone)? It is true that neural networks are able to learn complex mappings between two vector variables. However, a connectionist formalism is not very well suited to solve the sequence-mapping problem. Most early applications of ANNs to speech recognition have depended on severe simplifying assumptions (e.g., small vocabulary, isolated words, known word or phone boundaries). We shall see here that further structure (beyond a simple MLP) is required to perform well on continuous speech recognition, and that HMMs provide one solution to this problem. First, the relation between ANNs and HMMs must be explored.

4.3 MLPs as Statistical Estimators

MLPs can be used to classify speech classes such as words. However, MLPs classifying complete temporal sequences have not been successful for continuous speech recognition. In fact, used as spatial pattern classifiers, they are not likely to work well for continuous speech, since the number of possible word sequences in an utterance is generally infinite. On the other hand, HMMs provide a reasonable structure for representing sequences of speech sounds or words. One good application for MLPs can be to provide the local distance measure for HMMs, while alleviating some of their typical drawbacks (e.g., lack of discrimination, assumptions of no correlation between acoustic vectors).

4.3.1 Posterior Probability Estimation

For statistical recognition systems, the role of the local estimator is to approximate probabilities or probability density functions. In particular, given the basic HMM equations, we would like to estimate something like $p(x_n|q_k)$, which is the value of the probability density function (pdf) of the observed data vector given the hypothesized HMM state. The MLP can be trained to produce the *posterior* probability $P(q_k|x_n)$ of the HMM state give the acoustic data. This can be converted to emission probabilities density function values using Bayes' rule.

Several authors (Bourlard & Wellekens 1989; Bourlard & Morgan 1994; Gish 1990; Richard & Lippmann 1991) have shown that ANNs can be trained to estimate *a posteriori* probabilities of output classes conditioned on the input pattern. Recently, this property has been successfully used in HMM systems, referred to as *hybrid HMM/ANN* systems, in which ANNs are trained to estimate local probabilities $P(q_k|x_n)$ of HMM states given the acoustic data (see, e.g., (Lubensky *et al.* 1994)).

Since MLPs required supervised training, all these systems have been used so far in the framework of Viterbi training, which provided the segmentation of the training sentences in terms of q_k's and, hence, MLP training targets. The principle of these systems are briefly recalled here.

Let c_k, with $k = 1, \ldots, K$, be the output classes of an MLP. Since we will use the MLP for probability estimation associated with each HMM state q_k ($k = 1, \ldots, K$), there is a one-to-one equivalence between the q_k's and the c_k's that are associated with the discrete stationary states of Q^*. Also, we associate the parameter set Θ as defined for HMMs with the MLP parameter set.

The output activation of the k-th MLP output class for a given set of parameters Θ and an input x_n is denoted $g_k(x_n, \Theta)$. Since MLP training is supervised we will also assume the training set consists of a sequence of N acoustic vectors $\{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ labeled in terms of q_k's. At time n, the input pattern of the MLP is acoustic vector x_n, and is associated with a state q_k^n.

For these popular MLP cost functions, it can be proved [see, e.g., (Bourlard & Wellekens 1989; Bourlard & Morgan 1994; Gish 1990; Richard & Lippmann 1991)] that the optimal MLP output values are estimates of the probability distribution over classes conditioned on the input $\hat{P}(q_k|x_n)$, i.e.:

$$g_k(x_n, \Theta^{opt}) = \hat{P}(q_k|x_n) \tag{21}$$

if:

1. the MLP contains enough parameters to be able to reasonably approximate the input/output mapping function,
2. the network is not over-trained (which can be assured by stopping the training before the decline of generalization performance on an independent cross-validation set),
3. the training does not get stuck at a local minimum.

In (21), Θ^{opt} represents the parameter set minimizing (19) or (20).

It has been experimentally observed that, for systems trained on a large speech corpus, the outputs of a properly trained MLP do in fact approximate posterior probabilities, even for error values that are not precisely the global minimum.

This conclusion can easily be extended to other cases. For example, if we provide the MLP input not only with the acoustic vector x_n at time n, but also with some acoustic context $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$, the output values of the MLP will estimate

$$g_\ell(x_n, \Theta^{opt}) = \hat{P}(q_\ell^n|X_{n-c}^{n+d}), \quad \forall \ell = 1, \ldots, K \tag{22}$$

This is what has been used in our previous hybrid system (briefly summarized later in this section) to take partial account of the correlation of the acoustic vectors.

If the previous class is also provided to the input layer (leading to a quasi-recurrent network), the MLP output values will be estimates of

$$g_\ell(x_n, \Theta^{opt}) = \hat{P}(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}), \quad \forall k, \ell = 1, \ldots, K \tag{23}$$

It will be shown in Section 5 that this is a form of the local probability the hybrid HMM/MLP theory tells us to use. This will be referred to as "conditional transition probability" and will be the major thread throughout this paper.

Again, this conclusion remains valid for other kinds of networks, given similar training conditions. For example, recurrent networks (Robinson 1994) and radial basis function networks (Renals *et al.* 1991) can also be used to estimate posterior probabilities.

There is another important generalization of this property that will be essential later in this report. If the ANNs are trained with an estimate of the posterior probabilities of the output states (as opposed to the "1-from-K" binary output targets used for a classification mode training), then (21) remains valid. In other words, if the targets come from some independent "expert", the net will learn to produce posterior probabilities as well.[10] Although this property is mentioned in, e.g., (Bourlard & Wellekens 1989; Bourlard & Morgan 1994; Richard & Lippmann 1991), it has never been systematically used in hybrid HMM/MLP systems because of the lack of a full algorithm for the convergence to better probabilities. Such an algorithm has now been developed, and will be presented in this report.

4.3.2 Estimating HMM Likelihoods with MLP

Since the network outputs approximate Bayesian probabilities, $g_k(x_n, \Theta)$ is an estimate of

$$P(q_k|x_n) = \frac{p(x_n|q_k)P(q_k)}{p(x_n)} \tag{24}$$

which implicitly contains the a priori class probability $p(q_k)$. It is thus possible to vary the class priors during classification without retraining, since these probabilities occur only as multiplicative terms in producing the network outputs. As a result, class probabilities can be adjusted during use of a classifier to compensate for training data with class probabilities that are not representative of actual use or test conditions (Richard & Lippmann 1991).

Thus, (scaled) likelihoods $p(x_n|q_k)$ for use as emission probabilities in standard HMMs can be obtained by dividing the network outputs $g_k(x_n)$ by the relative frequency of class q_k in the training set, which gives us an estimate of:

$$\frac{p(x_n|q_k)}{p(x_n)} \tag{25}$$

During recognition, the scaling factor $p(x_n)$ is a constant for all classes and will not change the classification. It could be argued that, when dividing by the priors, we are using a scaled likelihood, which is no longer a discriminant criterion. However, this need not be true, since the discriminant training has affected the parametric optimization for the system that is used during recognition. Thus, this permits use of the standard HMM formalism, while taking advantage of ANN characteristics.

[10] Actually, it is easy to prove that, for the popular MLP cost functions, $g(x_n)$ will be an estimate of $E\{d(x_n)|x_n\}$, where E stands for the expected value.

5 Discriminant HMM/MLP Hybrid

In this section we present an overview of a form of HMM that has discriminant properties. The estimation properties of MLPs that were described in the previous section make them useful for this part of the overall system. Much of this section is similar to previous expositions on the subject, such as can be found in (Bourlard & Morgan 1994). However, the reader may find it useful to see our current perspective on this older approach, as it provides a basis for understanding the new approach as described in the sections that follow.

5.1 Motivations

In earlier work, multilayer perceptrons (MLP) (Bourlard & Morgan 1994) and recurrent neural networks (Robinson 1994) have been used to estimate local probabilities or likelihoods for HMMs. The interest in this scheme was partially based on the availability of locally discriminant training algorithms for the network, since according to the earlier theory (Bourlard & Wellekens 1989), globally discriminant systems (i.e., ones trained to accept correct utterances and reject incorrect ones) could be derived from these local probability estimators.

However, in the years following the original theoretical formulations, simplified systems were derived to benefit from the general character of the scheme (for instance, to reduce the dependence on distributional assumptions for the observation space, and to make the probability estimates more discriminant). These simplified approaches did not make use of the full power of the initial scheme. Nonetheless, for controlled tests they displayed some significant strengths. The basic scheme consisted of training neural networks to estimate probabilities of HMM states, and then using simple functions of these probabilities to label the training data using Viterbi decoding (dynamic programming). This procedure was repeated iteratively to train the system. The Viterbi procedure was then used with probabilities from the trained networks during recognition.

The remainder of this section will describe the original theory, but with the benefit of hindsight from our more recent developments.

5.2 Global Posterior Probability Estimation

If $X = \{x_1, x_2, \ldots, x_N\}$ is a sequence of acoustic vectors and M_i a HMM, the optimal training and recognition criterion (actually minimizing the probability of errors) should be based on the posterior probabilities $P(M_i|X, \Theta)$.

In standard HMMs, using Bayes' rule, $P(M_i|X, \Theta)$ is usually expressed in terms of $p(X|M_i, \Theta)$ as

$$P(M_i|X, \Theta) = \frac{p(X|M_i, \Theta)P(M_i|\Theta)}{p(X|\Theta)} \tag{26}$$

which, as discussed in Section 4.1, separates the probability estimation process into language modeling and acoustic modeling *in one particular way*.

However, when one wants to use posterior probabilities for the "acoustic" decoding[11] it is necessary to decompose $P(M|X)$ differently. In (Bourlard & Morgan 1994) we showed that it was possible to estimate $P(M_i|X,\Theta)$ according to

$$P(M_i|X,\Theta) \;=\; \sum_{\ell_1=1}^{L_i} \cdots \sum_{\ell_N=1}^{L_i} P(q^1_{\ell_1},\ldots,q^N_{\ell_N},M_i|X,\Theta) \tag{27}$$

for all state sequences $\{q^1_{\ell_1},\ldots,q^N_{\ell_N}\} \in \Gamma_i$, the set of all possible paths in M_i. Generally, this probability is approximated by considering only the best path (Viterbi approximation), in which case $P(M_i|X,\Theta)$ is approximated by:

$$\overline{P}(M_i|X,\Theta) \;=\; \max_{\ell_1,\ldots,\ell_N} P(q^1_{\ell_1},\ldots,q^N_{\ell_N},M_i|X,\Theta) \tag{28}$$

However, it is shown in this paper [see Section 8] that it is actually possible to compute the full posterior probability (27) by a new form of "forward-backward" algorithm [see, e.g., (Baum 1972) for likelihoods].

In both cases, the right hand side can be factored using

$$P(q^1,\ldots,q^N,M_i|X,\Theta) \;=\; P(q^1,\ldots,q^N|X,\Theta)\; P(M_i|X,q^1,\ldots,q^N,\Theta) \tag{29}$$

which separates [in a different way than (26)] the a posteriori probability estimation into two parts:

1. the acoustic model $P(q^1,\ldots,q^N|X,\Theta)$,

2. the language model $P(M_i|X,q^1,\ldots,q^N,\Theta)$; as seen later on, according to what we actually encode into the acoustic models, this factor will represent phonological, lexical and/or syntactical information.

5.3 Acoustic Model

Probability $P(q^1,\ldots,q^N|X,\Theta)$ of the acoustic models in (29) can be factored as follow:

$$\begin{aligned} P(q^1,\ldots,q^N|X,\Theta) &= P(q^1|X,\Theta)\; P(q^2|X,q^1,\Theta)\ldots \\ &\quad \ldots P(q^N|X,q^1,\ldots,q^{N-1},\Theta) \qquad (30) \\ &= \prod_{n=1}^{N} P(q^n|X,Q_1^{n-1},\Theta) \qquad (31) \end{aligned}$$

where q^n represents the state observed at time n and Q_1^N the state sequence associated with X_1^N. Probabilities $P(q^1,\ldots,q^N|X,\Theta)$ can thus be calculated from "local" probabilities

[11] Again, the distinction between "acoustic" modeling and "language" modeling is very ambiguous here since this definition depends mainly on which priors one wants to learn from the acoustic data and which priors one wants to learn independently of the acoustic data. Ideally, of course, we want to learn as much as possible from the acoustic data, as long as it generalizes well to the test data. See Section 11 for further discussion about this.

$P(q^n|X, Q_1^{n-1}, \Theta)$; these local probabilities may be simplified by relaxing the conditional constraints, for example by assuming dependency only on the previous state (first-order Markov model assumption) and on a temporal window X_{n-c}^{n+d} around the acoustic vector at time n (acoustic correlation limited to the contextual window). We then can approximate these local contributions by

$$P(q^n|X, q^1, \ldots, q^{n-1}, \Theta) \;=\; P(q^n|X_{n-c}^{n+d}, q^{n-1}, \Theta) \tag{32}$$

where input contextual information is taken into account. These probabilities can be estimated at the outputs of an MLP with contextual input and output feedback. If input contextual information is neglected ($c = d = 0$), (32) becomes:

$$P(q^n|x_n, q^{n-1}, \Theta)$$

In (Bourlard & Morgan 1994) we showed that those probabilities could be estimated by an MLP (as represented in Figure 1) with X_{n-c}^{n+d} as input, complemented by a K-binary input vector representing the state q_k^{n-1} ($\forall k \in [1, K]$) and K output units, one output for each HMM state, generating estimates of

$$P(q_\ell|X_{n-c}^{n+d}, q_k^-, \Theta) \tag{33}$$

in which q_k^- stands the state hypothesized at the previous time step. Given an HMM topology, probability (33) is estimated by running the neural network for each hypothesized state q_k^n.

Since these probabilities cannot be split further into something equivalent to emission and transition probabilities (as it is the case with likelihoods), we refer to these local probabilities as *conditional transition probabilities*.

As a consequence, probabilities $P(q^1, \ldots, q^n, \ldots, q^N|X, \Theta)$ of paths Q_1^N given X, as appearing in (27), (28) and (29) are given by

$$P(q^1, \ldots, q^n, \ldots, q^N|X, \Theta) \;=\; \prod_{n=1}^{N} P(q_{\ell_n}^n|x_n, q_{\ell_{n-1}}^{n-1}, \Theta) \tag{34}$$

Such probabilities should be useful for better models of the speech recognition process since, as opposed to standard HMM emission and transition probabilities, they intrinsically have the following properties:

- They model a recognition process instead of a production process,
- They can model non-stationary processes, since they are dependent on the previous state, and on the current acoustic vector,
- They potentially put more emphasis on transitions.

5.4 Priors, Transition Probabilities and Language Model

The second factor in (29) represents a phonological and lexical step; once the sequence of states is known, the model M_i associated with X can be found from the state sequence without an explicit dependence on X so that

$$P(M_i|X, q^1, \ldots, q^N, \Theta) \;=\; P(M_i|q^1, \ldots, q^N, \Theta) \tag{35}$$

For example, if the states represent phonemes, this probability must be estimated from phonological knowledge of the vocabulary in a separate process without any reference to the input vector sequence. Neglecting this probability is equivalent to assuming that, given a sequence of states, it is possible to recover the model that generated it.

However, that conclusion is too facile and, although this factor has been neglected in most of our previous work, we recently realized that it actually contained all the information about the language model and training/test transition probabilities. This second factor $P(M_i|q^1, \ldots, q^N, \Theta)$ may indeed be rewritten as:

$$P(M_i|q^1, \ldots, q^N, \Theta) \;=\; \frac{P(q^1, \ldots, q^N|M_i, \Theta)P(M_i|\Theta)}{P(q^1, \ldots, q^N|\Theta)} \tag{36}$$

and, using the same assumptions as for the acoustic models, we get:

$$P(M_i|q^1, \ldots, q^N, \Theta) \;=\; \left[\prod_{n=1}^{N} \frac{P(q^n|q^{n-1}, M_i, \Theta)}{P(q^n|q^{n-1}, \Theta)}\right] P(M_i|\Theta) \tag{37}$$

Finally, taking (37), (36), and (35) into account in (29) and (27) we get:

$$P(M_i|X) = \sum_{\ell_1,\ldots,\ell_N} \prod_{n=1}^{N} \left[P(q^n|X_{n-c}^{n+d}, q^{n-1}, \Theta)\frac{P(q^n|q^{n-1}, M_i, \Theta)}{P(q^n|q^{n-1}, \Theta)}\right] \; P(M_i|\Theta) \tag{38}$$

in which

$$P(q^n|X_{n-c}^{n+d}, q^{n-1}, \Theta) \tag{39}$$

represents the acoustic contribution (using conditional transition probabilities obtained at the output of the MLP),

$$P(q^n|q^{n-1}, \Theta) \tag{40}$$

are the transition probabilities observed on the training data, independently of the model, and

$$P(q^n|q^{n-1}, M_i, \Theta) \tag{41}$$

are the transition probabilities as given by the Markov models used during recognition. Note that ignoring the dependence on the previous state yields the approach that we implemented in the past, in which we divided by the class priors (as given by the relative phone frequencies in the training data) and multiplied by the transition probabilities form our phonological models. This use of scaled likelihoods compensated for a mismatch between the training data and the Markov models used for recognition. In fact, (38) provides us with a convenient way to split MAP probabilities into acoustic and prior contributions depending on what we want to learn from the "acoustic" training data.

5.5 MAP Constraints

It is thus possible to estimate the global posterior probability $P(M_i|X)$ of a Markov model M_i given the acoustic vector sequence X as follows:

$$P(M_i|X) = \sum_{\ell_1=1}^{L_i} \cdots \sum_{\ell_N=1}^{L_i} P(q_{\ell_1}^1, \ldots, q_{\ell_N}^N, M_i|X) \tag{42}$$

$$= \sum_{\ell=1}^{L_i} P(q_\ell^n, M_i|X), \forall n \in [1, N] \tag{43}$$

However, we also have:

$$P(q_\ell^n, M_i, X) = P(q_\ell^n, M_i|X)p(X) = P(X, q_\ell^n|M_i)P(M_i) \tag{44}$$

or

$$P(q_\ell^n, M_i|X) = \frac{P(M_i)}{p(X)} P(X, q_\ell^n|M_i) \tag{45}$$

which is nothing other than the likelihood used in standard HMMs multiplied by a scaling factor. This kind of scaling was already used in (Devijver 1985; Levinson *et al.* 1983) to avoid numerical problems (because of the product of probabilities), where it was shown that this led to the same forward and backward recurrences of the standard Baum-Welch algorithm (within a normalization factor).

Maximization of $p(X|M_i)$ and of $P(M_i|X)$ thus seems to lead to the same estimation formulas. However, we should not conclude from this that the discriminant approach does not change anything since, during training, re-estimation of the parameters will have to take the following major constraint of the MAP approach into account:

$$\sum_{i=1}^{I} P(M_i|X) = 1 \tag{46}$$

where the sum over i represents the sum over all possible Markov models. Here lies the difference between an ML and an MAP criterion. Any modification of the parameters of a model M_i must be complemented by a modification of all the parameters of the other models so as to preserve this constraint, thus making the MAP procedure discriminant. Thus, even if the estimation formulas are the same, the re-estimation (maximization and update) formulas will have to be different to take the constraint (46) into account. In the following, we define the key parameters of such discriminant HMMs and we show which constraint they must meet to guarantee (46). It is shown that this constraint is automatically met when using MLPs to estimate these parameters.

It is also important to show that, in this case, if the "local" constraint:

$$\sum_{k=1}^{K} p(q_k^n|x_n, q_\ell^{n-1}) = 1 \tag{47}$$

is met (which will be the case, at least approximately, with sigmoidal MLP outputs[12]), the constraint (46) on the global MAP probabilities is also met. Indeed, if Γ denotes the set of all possible paths $\{q^1, \ldots, q^n, \ldots, q^N\}$ in all possible Markov models M_i, we have:

$$\begin{aligned}
\sum_i P(M_i|X) &= \sum_\Gamma \sum_i P(M_i, \Gamma|X) \\
&= \sum_\Gamma \sum_i P(M_i|X, \Gamma) P(\Gamma|X) \\
&= \sum_\Gamma P(\Gamma|X) \sum_i P(M_i|X, \Gamma) \\
&\qquad \text{and, assuming } \sum_i P(M_i|X,\Gamma) = 1\text{:} \\
&= \sum_{\ell_1=1}^{K} P(q_{\ell_1}^1|x_1) \left(\sum_{\ell_2=1}^{K} P(q_{\ell_2}^2|x_2, q_{\ell_1}^1) \ldots \left(\sum_{\ell_N=1}^{K} P(q_{\ell_N}^N|x_N, q_{\ell_{N-1}}^{N-1}) \right) \ldots \right) \\
&= 1
\end{aligned}$$

It is however important to remember that this property is valid only if one considers all possible paths through the models.

Besides the advantage of forcing discrimination, numerical problems that plague the classical HMM are avoided when using discriminant models: namely, the lack of balance between the transition probability values (which only depend on the topology of the model) and the emission probability values (which decrease with the dimension of the input feature space).

5.6 MAP Estimation and Training

Most of the ideas in this section have already been presented in (Bourlard & Morgan 1994). When that book was written, however, we did not know how to perform "full" MAP estimation and training (i.e., taking all possible paths into account). Thus, we proposed a Viterbi-like algorithm using conditional transition probabilities as local probabilities in a DP procedure taking the following form:

$$\overline{P}(Q_1^n, q_\ell^n|X_1^n) = \max_k \left[\overline{P}(Q_1^{n-1}, q_k^{n-1}|X_1^{n-1}) P(q_\ell^n|q_k^{n-1}, x_n) \right] \tag{48}$$

where the index k points to all possible states preceding q_l and $\overline{P}(Q_1^n, q_\ell^n|X_1^n)$ denotes the cumulative best path probability of reaching state q_l given the partial sequence X_1^n. Note that this DP procedure finds the most likely acoustic model, i.e., the left factor of (29). During training, this type of DP was used to get a segmentation of the training data and, consequently, to provide us with targets for the training of the MLP. While preserving local discrimination, it was clear that constraint (46) was no longer guaranteed. Also,

[12]This constraint is precisely met in the case of a softmax output layer, since the outputs are normalized to sum to 1.

discrimination was performed at the state level only. Still, many problems were faced with even this simplified approach, as briefly summarized in the next section, and so we were not prepared at that time to face the more global difficulty.

However, while recently working on transition-based recognition systems, we realized that the original discriminant HMM/MLP theory appeared to lead to the right formalism. However, the results of some simple pilot experiments suggested that it was necessary to be able to handle "full" MAP training and recognition of hybrid HMM/MLP systems for transition-based to work (see Section 6). The development of this MAP estimation procedure is the main point of this report, and is described in Section 8. We first provide a bit more necessary background in the next two Sections.

6 Early Experiments with HMM/MLP Systems

6.1 Brief Description

In 1988, we attempted to directly use conditional transition probabilities (23) in a discriminant HMM Viterbi training for the recognition of continuous speech. Unfortunately, our early results were quite poor. As a consequence, we simplified our system significantly so that we could begin to trace the sources of error. Over the next few years, we discovered key practical points that are summarized in (Bourlard & Morgan 1994). These led to a number of modifications to the basic HMM/MLP system that were initially required to get acceptable recognition performance:

- No transition-based emission probabilities are used; i.e., probabilities like

  $$P(q_k|X_{n-c}^{n+d}) \tag{49}$$

  are estimated at the output of the MLP that are independent of the previous state.

- Division by the prior probabilities of the output classes to get scaled likelihood

  $$\frac{P(q_k|X_{n-c}^{n+d})}{P(q_k)} = \frac{p(X_{n-c}^{n+d}|q_k)}{p(X_{n-c}^{n+d})}$$

  for use in standard likelihood recognizers. However, as predicted, this doesn't seem to be required if HMM topologies are trained to better fit the training data (which is the case, for instance, when using multiple pronunciations models that are learned from the data (Stolcke & Omohundro 1993)). In this case, HMM topology is no longer independent of the acoustics so the division by the priors is not required (Wooters 1993).

- It was not clear how to use information about the language model (at the word level) in discriminant HMMs. Once we divided the MLP outputs by the prior probabilities, we were estimating scaled likelihoods, for which standard statistical language models apply. In particular, the scaled likelihoods could be multiplied by language priors in order to get estimates of the global posterior probability, as is done in standard HMM systems.

- Only Viterbi approximation of full posterior probabilities was used for training (and recognition). As shown later, this limitation can lead to major problems when working with transition-based systems. However, particularly given a good initialization of the estimators, this approximation was most likely not a bad one for the simplified system.

6.2 Some Results

Results of a large number of experiments using embedded Viterbi training of neural networks (as probability estimators for HMMs) have been presented in the literature (Wooters 1993; Renals *et al.* 1994) and we will not substantially repeat them here. In general, these results have shown that for controlled conditions (i.e., ones in which the only system change is the choice of estimator between a discriminant neural network approach and a more conventional estimator), estimators with relatively few parameters can provide significant improvements. For instance, in a recent study by Nynex on connected digit recognition (Lubensky *et al.* 1994), an MLP with 11,631 parameters provided equivalent results to a Gaussian mixture estimator with 112,640 parameters (about .8% word error on the TI-Digits database). In that same experiment, the Gaussian mixture estimator with 28,160 parameters had roughly double the error rate of the MLP-based system.

For large-vocabulary continuous speech recognition, there are only a few controlled experiments that we know of. However, in one that was done in a collaboration between ICSI and SRI, we again found an equivalence between MLP-based performance and that of mixture Gaussian systems with many more parameters, and a halving of error for the MLP case when the number of parameters was equivalent (Renals *et al.* 1992).

Often, however, both researchers and users are simply interested in the best possible performance, and are not concerned with controlled experiments. Given this motivation, a number of researchers have found that the blending of probabilities or log probabilities from the two types of systems has given better results than were obtainable from either one alone. For instance, in the Nynex experiment, mixing the log probabilities from the best mixture Gaussian system (which had a .7% error rate) with the MLP system (which had a .8% error rate) led to a system with a significantly lower error rate (roughly .6% error).[13]

[13]While all of these error rates may seem extremely low, this database was artificially generated and is far easier to recognize than similar sources obtained from natural telephone recitations of connected digit strings. Additionally, many practical applications require the correct recognition of long strings of digits, for instance corresponding to a credit card number. A 15-digit string that is recognized with a .6% digit error rate will be incorrect roughly 8% of the time.

6.3 Discussion

This brief survey of results suggests the utility of discriminant HMM hybrid systems, even in the simplified form of our earlier HMM/MLP hybrid that:

- contained no transition information,
- was only locally discriminant,
- and used only Viterbi-like training.

However, the above limitations may have constrained our system performance when comparing with unconstrained traditional systems, namely ones with detailed levels of context and with huge numbers of parameters. It is likely that in order to show significant performance advantages for discriminant HMMs we will need to conform better to the original discriminant HMM theory. Recently we have developed algorithms which in principle should permit us to do this, and their explanation is the major task for the rest of this report.

7 Transition-based Recognition Systems

7.1 Motivations

In a series of experiments that were conducted at ICSI and elsewhere, it was shown that approaches to feature extraction that emphasized transitions provided improved robustness to some types of acoustical interference (Hermansky *et al.* 1992). However, we also sometimes observed degradations in recognition performance with these measures for tests that were good matches to training conditions. Consequently, we began looking for statistical models that were more fundamentally based on transition information.

In an earlier study we showed that given accurate transition information we can significantly improve the recognition performance (Konig & Morgan 1994). Specifically, we studied a time-index model that explicitly conditions the emission probability of a state on the time index, where time index is defined as the number of frames between entering a state and the current frame.

In a recent study (Goldenthal 1994), Goldenthal got a consistent improvement in phone recognition results when enhancing his statistical trajectory models with explicit transition models. He used a set of 200 *canonical* transitions that were created by clustering all the transitions in the training set. Each canonical transition modeled the trajectory of a fixed number of frames centered about the transition boundary.

The studies mentioned above indicate the importance of transition information and transition modeling to an improved recognition performance. Segment-based models have also been used to attempt to model the non-stationarity of speech without explicit dependence on transition information. In segment-based models the basic unit is a sequence of acoustic vectors emitted in a given speech unit (a "segment"), as opposed to a single acoustic vector as used for HMMs (Digalakis 1992; Ostendorf & Roukos 1989; Ghitza & Sondhi 1993).

Given this background, we began to experiment with simple forms of statistical training that modeled transitions. The rest of this section reports on some pilot experiments with this motivation.

7.2 Early Experiments

Given the theoretical properties of the Discriminant HMM/MLP model described earlier, we felt that empirical evaluations of this model would be a first good step in improving our understanding of transition-based systems. In particular, we began to empirically evaluate conditional transition probabilities as used in Discriminant HMM/MLP systems on phoneme classification and phonemic frame classification tasks.

As presented in the initial theory (Bourlard & Morgan 1994) our paradigm for training (and recognition) was to use the Viterbi approximation, i.e., to consider only the most probable state sequence in assigning phonetic labels to acoustic frames. We chose to estimate the local discriminant probabilities (23) by an MLP as represented in Figure 1. In this case, the previous state is coded as additional binary inputs, one for each possible previous state. For every hypothesized previous state we set the corresponding input to one and the rest to zero. As already discussed in Section 5.3, the set of possible previous states (or the set of possible successor states for a given q_k at the input) will be given by the topology of the HMMs (and by the currently hypothesized states of the matching process).

In a Viterbi training (as used so far) we know the correct previous state (again by considering only the most probable state sequence), either by having a hand-segmented database such as TIMIT, or by running an automatic alignment (Forced-Viterbi) on the training data. During recognition, the MLP outputs will have to be hypothesized for every possible previous state (possibly constrained by a particular HMM topology or a language model).

We chose the TIMIT corpus (Garofolo 1988) for our experiments because it is phonetically balanced, and in addition there are time-aligned phonetic transcriptions of all the sentences in the database. The experiments were done on a 200 sentence development set that was selected from the official training set and was not used for the training. We used 3300 sentences for training and 396 sentences for cross-validation (where the 200 sentence development set is a subset of the cross-validation set). No language model was used in these experiments. All of our results are on the full 61 TIMIT phone set. Our phone models were simple one state per phone models.

The net that estimates the local discriminant probabilities (as shown in Figure 1) had 1000 hidden units, 61 outputs (the size of the phone set). There were 295 inputs to the net, including 234 that consisted of 9 frames of 26 features each (PLP12 + log gain + delta features for each of these 13) (Hermansky 1990), and 61 binary inputs that represent the possible previous state. With the exception of these binary inputs, this net was the same as the hybrid HMM/MLP system as described in (Bourlard & Morgan 1994). Our

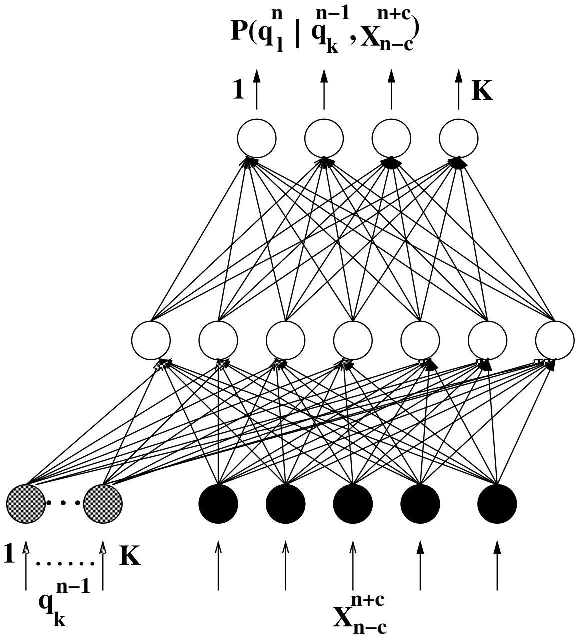

Figure 1: **Estimating Local Discriminant Probabilities**

reference HMM/MLP system (Bourlard & Morgan 1994) had 36.3% phone error on this task. When evaluating the Discriminant HMM on this task the error rate was 40.4%. This was an intriguing negative result; *increasing* the input information led to a *decrease* in generalization performance. Why should this be so? Although it is difficult to draw firm conclusions from a negative result, it can at least inspire directions of inquiry. This result motivated the error analysis as described in the following section.

7.3 Error Analysis

As shown in the following, error analysis suggests two potential reasons[14] for the observed performance lost: (1) poor transition detection, and (2) mismatch between the input space distribution of the MLP during training and recognition.

The first potential problem is missing transitions; i.e., implicitly the net is a transition detector because when it determines that the current state is different from the previous it signals a transition, and transition detection between phonemes is known to be a hard problem (see (Glass 1988)). In order to test this assumption we compared the performance of the MLP described above on two kinds of acoustic frames: *transition frames* that start a new segment, i.e., their phonetic label is different from the previous frame, and all other frames, i.e., *self-loop frames. While presenting the correct previous state*, the frame level performance on the development set was:

1. Self-loop frames: 85.5% of correct phonemic classification.

[14]Other than bugs.

2. Transition frames: 39.2% of correct transition detection and classification.

Transitions thus seems harder to detect and classify than "steady-state" frames. However, we suspect that this is not (only) due to the properties of transitions but to two problems related to the training and testing procedure:

1. We have much less training data for transition frames that for "steady-state" frames (less than one-sixth). Thus the classifier will tend to focus its learning ability on the steady-state phonetic classification.

2. Our training procedure assumes that a single frame is the transition and that its neighbors are not transitions. This does not make sense in terms of the acoustic phonetics, since many spectral transitions are gradual. This makes a difficult classification function for a network to learn.

Additionally, we think that another possible source of the observed degradation of performance is the potential disparity between training and recognition input populations. During training we only present to the net "correct" pairs of acoustic vectors and correct previous state, while in recognition we expect the net to generalize to all possible combinations of acoustic vectors and previous states. Some of these recognition inputs can be completely meaningless, e.g., like the combination of the acoustics of a middle of a vowel and a previous state that corresponds to a plosive. The net is not trained on anything close to these "impossible" pairs, but through the vagaries of interpolation could end up having the highest MLP outputs during recognition. This problem is often referred to as the "lack of negative training example" and sometimes can be partially overcome by presenting additional negative training examples to the net (Zavaliagkos *et al.* 1994).

In order to test this hypothesis we computed the frame level performance of the net on the development set for the following two cases:

1. Presenting the correct previous state, we got 79.4% frames correct.

2. Presenting all possible previous states and taking as the winner the output with the highest activity, i.e., taking for every frame the maximum of 61 by 61 probabilities (61 outputs for each possible previous state), and checking if it was the correct pair of previous state and current state. In this case we got 15.9% correct, which was the weighted average of 18.3% correct on self-loop frames and 0.4% correct on transition frames.

These results seem to suggest that, even for "steady-state" frames, there is a problem of mismatch between the space of training and testing for hypothesized inputs. Of course, the performance is also hurt by the problems mentioned earlier (a difficult classification problem with insufficient examples for many of the classes).

All the problems identified here motivated the REMAP training and recognition algorithm for HMM/MLP hybrids that is presented below. Specifically, from the first hypothesis we learned that "hard" transitions are difficult to detect. As we will see, the full MAP training will provide the nets with soft targets and soft decisions, i.e., with conditional probabilities of transitions. Furthermore, by considering all possible paths and transitions, we will reduce the mismatch between training and recognition. A formalism will be introduced that automatically considers negative training examples without the need for explicit enumeration of impossible input combinations.

8 REMAP Training of HMM/MLP Hybrids

8.1 Motivations

The discriminant HMM/MLP theory as described above uses transition-based probabilities as the key building block for acoustic recognition. However, it is well known that estimating transitions accurately is a difficult problem (Glass 1988). Due to the inertia of the articulators, the boundaries between phones are blurred and overlapped in continuous speech (Deng & Sun 1994). It is also likely that some time variability exists in human perception of the onset of a new phonetic region. Consequently, we would like to have a "window" of possible transitions instead of a single transition. Ideally the width of the transition window should depend on the specific bi-phone and on the speaker. Thus we need an automated way of estimating the transition windows to be used as targets in the MLP training.

Targets are typically obtained using an automatic alignment procedure incorporating a Viterbi approximation. However, this procedure yields rigid transition targets, and thus suffers from the problems mentioned above. Furthermore, our preliminary experiments with this procedure yielded poor transition detection performance.

Another related problem in our Viterbi-based MLP training procedure is a disparity between the training input space of the MLP and the input space used in recognition. Specifically, in training the network only processes input consisting of "correct" pairs of acoustic vectors and correct previous state, while in recognition we expect the net to generalize to all possible combinations of acoustic vectors and previous states. However, some of the hypothesized inputs may correspond to an impossible condition that has thus never been observed, such as the acoustics of the temporal center of a vowel in combination with a previous state that corresponds to a plosive. It is unfortunately possible that the interpolative capabilities of the network may not be sufficient to give these "impossible" pairs a sufficiently low probability during recognition.

One possible solution to these problems is to use a full MAP algorithm to find transition probabilities at each frame for all possible transitions by a forward-backward (Dempster *et al.* 1977) like algorithm taking all possible paths into account. Furthermore the MAP algorithm increases the discriminant power of the model by increasing the a posteriori probability of the correct model and reducing the posterior probabilities of all other models. Thus, it might improve the approximation to an optimal Bayes classifier.

In the rest of this section, we describe a set of procedures that we have derived in order to train and use the desired discriminant probability estimators in a full MAP framework. This comprises the heart of our new developments (from a mathematical perspective).

8.2 Problem Formulation

Full MAP training of the discriminant HMM (as defined earlier) requires solution to the following problem. Given a trained MLP at iteration t providing a parameter set Θ^t and, consequently, estimates of $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}, \Theta^t)$, how can we determine new MLP targets which:

1. will be smooth estimates of the probability of a transition $q_k^{n-1} \rightarrow q_\ell^n, \forall k, \ell \in [1, K]$ and $\forall n \in [1, N]$,
2. when training the MLP for iteration $t+1$, will lead to new estimates of Θ^{t+1} and $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}, \Theta^{t+1})$ that are guaranteed to incrementally increase the global posterior probability $P(M_i|X, \Theta)$?

In Appendix A, we prove that a re-estimate of MLP targets that guarantee convergence to a local maximum of

$$\prod_{j=1}^{J} P(M_{w_j}|X_j, \Theta^t) \tag{50}$$

is given by[15]:

$$P^*(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}, M) = P(q_\ell^n|X, q_k^{n-1}, \Theta^t, M) \tag{51}$$

where we have restricted the targets to be a mapping from the previous state and the local acoustic data to the current state, thus making the estimator realizable by an MLP with a local acoustic window. Thus, we will want to estimate the transition probability conditioned on the local data (as MLP targets) by using the transition probability conditioned on all of the data, as determined by procedures to be shown below.

In Appendix A, we further prove that alternating MLP target estimation (the "estimation" step) and MLP training (the "maximization" step) is guaranteed to incrementally increase (50) over t.[16]

Since we know (see Section 4.3.1) how to train an MLP to lead to (good) estimates of posterior probabilities (whatever the MLP targets are "1-from-K" binary vector or themselves estimates of posterior probabilities), the remaining problem is to find an efficient algorithm to express $P(q_\ell^n|X, q_k^{n-1}, M)$ in terms of $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1})$ so that the next iteration's targets can be found.

By simple statistical rules [as recalled in (1) and (2)], we have:

$$P(q_\ell^n|X, q_k^{n-1}, M) \quad = \quad \frac{p(X, q_k^{n-1}, q_\ell^n, M)}{p(X, q_k^{n-1}, M)} \tag{52}$$

[15]In most of the following, as well as in Appendix A, we consider only one particular training sequence X associated with one particular model M. It is, however, easy to see that all of our conclusions remain valid for the case of several training sequences X_j, $j = 1, \ldots, J$. A simple way to look at the problem is to consider all training sequences as a single training sequence obtained by concatenating all the X_j's with boundary conditions at every possible beginning and end point.

[16]Note here that one "iteration" does not stand for one iteration of the MLP training but for one estimation-maximization iteration for which a complete MLP training will be required.

$$= \frac{p(X, q_k^{n-1}, q_\ell^n, M)}{\sum_\ell \; p(X, q_k^{n-1}, q_\ell^n, M)} \tag{53}$$

in which

$$p(X, q_k^{n-1}, q_\ell^n, M) \;=\; \sum_i p(X, \Gamma_i, q_k^{n-1}, q_\ell^n, M) \tag{54}$$

is equal to the sum of the probabilities of all possible paths Γ_i in a particular M visiting q_k at time $n-1$ and q_ℓ at time n. In a similar way, the denominator represents the sum of the probabilities of all possible paths in M visiting q_k at time $n-1$.

We can break the probability appearing in (53) into two factors as follows:[17]

$$\begin{aligned} p(X, q_k^{n-1}, q_\ell^n, M) &= p(X_1^{n-1}, q_k^{n-1}, M) p(X_n^N, q_\ell^n | X_1^{n-1}, q_k^{n-1}, M) \\ &= \alpha_{n-1}(k) \beta_n(k, \ell) \end{aligned} \tag{55}$$

in which:

- $\alpha_{n-1}(k) \;=\; p(X_1^{n-1}, q_k^{n-1}, M)$ is the "forward" probability, defined as the probability of observing the partial acoustic sequence X_1^{n-1} and being in class q_k at time $n-1$.
- $\beta_n(k, \ell) \;=\; p(X_n^N, q_\ell^n | X_1^{n-1}, q_k^{n-1}, M)$ is the "backward" probability, defined as the probability of observing the rest of the sequence and starting from state q_ℓ at time n given that we have already observed X_1^{n-1} and that we were in class q_k at time $n-1$.

Since we cannot afford to compute and memorize all possible paths in (54) to compute $P(q_\ell^n | X, q_k^{n-1}, M)$, we need to find recursions to compute the forward and the backward probabilities in terms of local probabilities $P(q_\ell^n | X_{n-c}^{n+d}, q_k^{n-1})$ generated by the MLP. As done with standard HMMs, these local probabilities will be assumed independent of M. Also, in the following, to simplify notation, we will disregard the dependency on M, keeping in mind that all the following α's and β's will be computed using specific models M_{w_j} associated with training sentences X_j.

8.3 Forward Recursion

By applying simple statistical rules, we have:

$$\begin{aligned} \alpha_{n+1}(l) &= p(X_1^{n+1}, q_\ell^{n+1}) \\ &= \sum_k \; p(X_1^n, x_{n+1}, q_k^n, q_\ell^{n+1}) \\ &= \sum_k \; p(X_1^n, q_k^n) \; p(x_{n+1}, q_\ell^{n+1} | X_1^n, q_k^n) \\ &= \sum_k \alpha_n(k) \; P(q_\ell^{n+1} | X_1^n, x_{n+1}, q_k^n) \; p(x_{n+1} | X_1^n, q_k^n) \\ &\simeq \sum_k \alpha_n(k) \; P(q_\ell^{n+1} | X_{n-c}^{n+d}, q_k^n) \; c_{n+1}(k) \end{aligned} \tag{56}$$

[17]This probabilistic factorization and the resulting recurrence procedures were strongly inspired by the alpha-beta factors of the forward-backward algorithms used for data likelihood estimation. The new factorization and procedures turns out to be quite different, however.

where $c_{n+1}(k) = p(x_{n+1}|X_1^n, q_k^n)$; the sum over k extends over all possible previous states. It will be shown later that this factor does not appear in the training procedure. During recognition, if not neglected, it could be estimated via a predictive neural net or an autoregressive model.

Initialization of this forward recursion is done according to

$$\alpha_0(i) = p(q_I^0, x_0) = 1 \tag{57}$$

in which q_I is the non-emitting initial state. This initialization is similar to what is used in the case of the traditional forward-backward algorithm.

8.4 Backward Recursion

In a similar way, we have:

$$\begin{aligned}
\beta_n(j,k) &= p(X_n^N, q_k^n | X_1^{n-1}, q_j^{n-1}) \\
&= \sum_\ell p(x_n, X_{n+1}^N, q_k^n, q_\ell^{n+1} | X_1^{n-1}, q_j^{n-1}) \\
&= \sum_\ell p(X_{n+1}^N, q_\ell^{n+1} | X_1^n, q_j^{n-1}, q_k^n)\ p(x_n, q_k^n | X_1^{n-1}, q_j^{n-1}) \\
&\qquad \text{if 1st order HMMs :} \\
&\simeq \sum_\ell\ \beta_{n+1}(k,\ell)\ P(q_k^n | X_1^n, q_j^{n-1})\ p(x_n | X_1^{n-1}, q_j^{n-1}) \\
&\simeq p(q_k^n | X_{n-c}^{n+d}, q_j^{n-1})\ c_n(j) \sum_\ell\ \beta_{n+1}(k,\ell)
\end{aligned} \tag{58}$$

Note that the last simplification, i.e., $P(q_k^n|X_1^n, q_j^{n-1}) \simeq p(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})$ is not necessary in the case that we use a recurrent net to estimate the conditional transition probabilities. However, in practice it might not make a difference if we use an MLP to estimate the conditional transition probabilities, i.e., to do the last simplification. Initialization of this backward recursion can be done according to

$$\beta_{N+1}(L,F) = p(X_{N+1}^N, q_F^{N+1} | X_1^N, q_L^N) = 1 \tag{59}$$

in which q_F is the (non-emitting) final state and q_L represents any last emitting HMM state.

8.5 MLP Output Targets Update

Given (55) and the recursions for these two factors, at each time the complete probability specified in (52) and (53) can be obtained by computing their product (for each permissible value of current and previous state):

$$p(X, q_k^{n-1}, q_\ell^n) = \alpha_{n-1}(k)\ \beta_n(k,\ell) \tag{60}$$

and

$$
\begin{aligned}
p(X, q_k^{n-1}) &= \sum_\ell p(X, q_k^{n-1}, q_\ell^n) \\
&= \sum_\ell \alpha_{n-1}(k)\ \beta_n(k,\ell) \\
&= \alpha_{n-1}(k)\ \sum_\ell\ \beta_n(k,\ell)
\end{aligned}
$$

Using this in (53), we get:

$$
\begin{aligned}
P(q_k^n|X, q_j^{n-1}) &= \frac{p(X, q_j^{n-1}, q_k^n)}{p(X, q_j^{n-1})} \\
&= \frac{\beta_n(j,k)}{\sum_\ell \beta_n(j,\ell)} \\
&= \frac{\sum_\ell \beta_{n+1}(k,\ell)\ P(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})\ c_n(j)}{\sum_h \sum_\ell\ \beta_{n+1}(h,\ell)\ P(q_h^n|X_{n-c}^{n+d}, q_j^{n-1})\ c_n(j)} \\
&= \frac{\sum_\ell\ \beta_{n+1}(k,\ell)\ P(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})}{\sum_h \sum_\ell\ \beta_{n+1}(h,\ell)\ P(q_h^n|X_{n-c}^{n+d}, q_j^{n-1})}
\end{aligned}
\tag{61}
$$

This final form of the equation shows that the probabilities required to determine MLP targets can be obtained from the previous MLP outputs and the beta recursions alone. Note also that we can compute $P(q_k^{n-1}|X, \Theta^t)$, i.e., the posterior probability of being in class q_k at time $n-1$ to be used in the training of the MLP as specified below, according to the following:

$$
\begin{aligned}
P(q_k^{n-1}|X, \Theta^t) &= \frac{P(q_k^{n-1}, X|\Theta^t)}{P(X|\Theta^t)} \\
&= \frac{\alpha_{n-1}(k) \sum_\ell \beta_n(k,l)}{\sum_k \alpha_{n-1}(k) \sum_\ell \beta_n(k,l)}
\end{aligned}
\tag{62}
$$

8.6 REMAP Training Algorithm

The general scheme of the MAP Forward-Backward training of hybrid HMM/MLP systems can be summarized as follow:

1. Start from some initial net providing $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}, \Theta^t)$, $t=0$, $\forall$ possible (k,ℓ)-pairs[18].

[18]This can be done, for instance, by training up such a net from a hand-labeled database like TIMIT or from some initial forward-backward estimator of equivalent local probabilities (usually referred to as "gamma" probabilities in the Baum-Welch procedure).

2. Run backward recurrences to compute MLP targets $P(q_\ell^n|X, q_k^{n-1}, \Theta^t)$, $\forall$ possible (k, ℓ)-pairs. In Appendix A, we show that these new estimates of "local" posteriors guarantee the increase of the global MAP. Also as part of the forward and backward recurrences we compute $P(q_k^{n-1}|X, \Theta^t)$ (62), i.e., the posterior probability of being in class q_k at time $n-1$ to be used in the training of the MLP as specified below.

3. For every x_n (or X_{n-c}^{n+d}) in the training database choose q_k^{n-1} according to $P(q_k^{n-1}|X, \Theta^t)$, train the MLP to minimize the relative entropy between the outputs and targets equal to $P(q_\ell^n|X, q_k^{n-1}, \Theta^t)$. (See Appendix A for the theoretical explanation.) This provides us with a new set of parameters Θ^t, for $t = t+1$.

4. Iterate from 2 until convergence.

This procedure is thus composed of two steps: an Estimation (E) step, corresponding to step 2 above, and a Maximization (M) step, corresponding to step 3 above. In this regards, it is reminiscent of the EM algorithm originally introduced in (Dempster *et al.* 1977). This latter algorithm is, however, an iterative approach to maximum likelihood estimation while the approach proposed here is an iterative approach to maximum a posteriori probability estimation. Also, in the standard EM algorithm, the M step involves the actual maximization of the likelihood function. In some instances of the EM algorithm, usually referred to as Generalized EM (GEM) algorithm, the M step does not actually maximize the likelihood but simply increases it (by using, e.g., a gradient procedure). Similarly, REMAP increases the global posterior function during the M step (in the direction of targets that actually maximize that global function), rather than actually maximizing it.

Although EM or GEM algorithms can also be applied to Hierarchical Mixtures of Experts (HME) (Jordan & Jacobs 1994), we note here that the formalism and goals are different for these approaches than for the methods proposed here.

8.7 Remark

From a practical point of view, it is worth noting that this discriminant training is more convenient than Maximum Mutual Information (MMI) training (see Section 11). Indeed, MMI requires maximization of

$$\log\left(\frac{p(X_j|M_{w_j}, \Theta_{w_j})}{\sum_{i=1}^{I} p(X_j|M_i, \Theta_i)}\right) \tag{63}$$

where estimation of the denominator requires running several forward recurrences of each of the possible rival models. A common approximation evaluates an estimate of the denominator by running the forward recurrence through the N-best sentence models matching X_j or through an ergodic word model containing all possible phonemes.

8.8 REMAP Recognition

During recognition with full MAP models, we wish to evaluate the probability of each contending model (e.g., Markov models for a sequence of words) given the new acoustic data, so that we can choose the model with highest probability. This can be simply determined from the final value of the alpha recursion over the input data, as evaluated for each possible model (for instance using an equivalent to the stack decoder algorithm for searching the space of possible word sequences). This can be shown simply as follows, using the notation of the previous sections:

$$
\begin{aligned}
P(M_i|X) &= \frac{p(M_i, X)}{p(X)} \\
&= \frac{p(M_i, q_{F_i}^{N+1}, X)}{p(X)} \\
&= \frac{1}{p(X)} p(X_1^{N+1}, q_{F_i}^{N+1}|M) P(M_i) \\
&= \frac{1}{p(X)} \alpha_{N+1}^i(F_i) P(M_i)
\end{aligned}
\tag{64}
$$

in which:

- F_i is the non-emitting final state associated with M_i,
- $\alpha_{N+1}^i(F_i)$ represents the results of the forward recursion ran through M_i only,
- $P(M_i)$ is given by the language model.

As in the more common ML approach to this problem, the scaling factor $P(X)$ can be ignored during recognition. Thus, the candidate models can be evaluated from a product of their priors (which typically come from language models) and the final value for the alpha (forward) recursion. As done with standard likelihood-based recognizer, recognition based on (64) can be performed by an A*-like algorithm (Paul & Necioglu 1993). In the case of continuous speech recognition, and given the definition of the forward recursion, the $P(M_i)$ factor will usually contain only information about the syntax at the word level (usually N-grams) and the (state) transition probabilities introduced in (38) can be included in the α recursion.

During recognition, both (38) or (64) could be used. However, in our definition of the α recursion (56) we implicitly ignore the effect of the model M on the conditional state transition probabilities by assuming that

$$P(q_\ell^{n+1}|X_{n-c}^{n+d}, q_k^n, M) \simeq P(q_\ell^{n+1}|X_{n-c}^{n+d}, q_k^n) \tag{65}$$

8.9 Summary

In this section, we have described a new procedure to estimate and globally train discriminant (posterior) probabilities of Markov models given a full utterance X, using local posterior probabilities as obtained at the output of a neural network. In the particular instantiation of this algorithm that is presented here, we used local conditional transition probabilities to estimate global posterior probabilities. It is, however, clear that this algorithm can (in principle) be realized in other ways. As with the venerable Baum-Welch procedure, we have shown that there is an efficient procedure to iteratively converge to local probabilities that are guaranteed to maximize a global discriminant criterion (which in our case is MAP). Also, there is a convenient way to incorporate prior knowledge about the language.

The new algorithm presented here

1. can be used in a new form of hybrid HMM/ANN that retains the advantages of standard HMM/ANN hybrids (local discrimination, lack of distributional assumptions, etc.), while using "full" posterior probabilities for training and recognition;

2. still uses neural networks (in our case MLPs, although recurrent networks or TDNNs could be used) to estimate local posterior probabilities (conditional transition probabilities); but the new networks are trained with probabilistic targets that are themselves estimates of local posterior probabilities (conditioned on the acoustic data and the previous state);

3. iteratively estimates these optimal targets using an algorithm that is similar in spirit to the forward-backward recursions of the Baum-Welch algorithm; and

4. can be proved to iteratively increase the global posterior probability (this is proved in Appendix A.)

We note in passing that the proposed algorithm differs from an approach in which local likelihoods (or transition probabilities) would be computed from the standard forward-backward algorithms and then multiplied by the prior probabilities of states to get posterior targets for neural net training. Although such an algorithm could potentially provide the network with posterior probabilities estimated from full global probabilities, it is not discriminant (in the sense that the training steps are not chosen to minimize a globally discriminant criterion, resulting in different targets).

Section 9 generalizes the algorithm presented in this section to M-th order Markov models. Since it is expected that this approach will prove most important for transition-based recognition systems, Section 10 adapts the new algorithm to a particular kind of transition-based recognition system that we have recently been studying.

We conclude this section with a discussion of the naming convention for the underlying models used in REMAP. It could be argued that the new approach is simply an alternate algorithm to compute parameters for a particular kind of HMM. On the other hand, it could be argued that our hidden Markov models are no longer "hidden". We believe that both interpretations are valid. A "hidden" Markov process is usually defined as two correlated Markovian processes, where one corresponding to the state sequence that is not observed directly, but that affects the second Markovian process associated with the observed acoustic vector sequence. Although most early papers (Baum & Petrie 1966) referred to this kind of model as a "probabilistic function of a Markov chain", (a very precise but somewhat cumbersome definition), L.P. Neuwirth (Neuwirth 1970; Poritz 1988) coined the popular term "hidden Markov model". If we limit the definition of "hidden Markov processes" to the initial definition it is clear that our new model is still a hidden Markov chain. In fact, $p(q_\ell^n|x_n, q_k^{n-1})$ can indeed be inverted with Bayes' Rule to be a probabilistic function of a Markov chain. If, however, we define hidden Markov models as models in which transition probabilities of the underlying (hidden) state sequence can be expressed independently of the observed acoustic vector sequence, then the MAP-based Markov model is no longer hidden.

In any event, whatever the new stochastic models might be called, they are defined by their topologies and by conditional transition probabilities that can be estimated by an MLP.

9 M-th order REMAP Training

In the previous section, we developed procedures for training estimators of the posterior probabilities of models of speech utterances. These procedures relied on the factorization of (29) and (30) (factorizing the probability of a state sequence into the product of a series of state probabilities conditioned on previous states). They further relied on an assumption that the local state posteriors could be evaluated without any explicit dependence on states earlier than the immediately preceding state. That is, we assume that

$$P(q^n|X, Q_1^{n-1}, \Theta) \simeq P(q^n|X, q^{n-1}, \Theta) \tag{66}$$

While this first-order Markov assumption does not seem too bad, it is also possible to derive analogous procedures using explicit representations of higher order dependencies, namely

$$P(q^n|X, Q_1^{n-1}, \Theta) \simeq P(q^n|X, Q_{n-M}^{n-1}, \Theta) \tag{67}$$

In this section we will simply give the final form of the recursions and training procedure of the M-th order case, without their derivation. Once the 1st order derivation is understood, however, the Mth order case follows in a straightforward manner.[19]

9.1 Forward Recursion

Defining the "forward" probability α as:

$$\alpha_n(k_1, k_2, \ldots, k_M) = P(X_1^n, q_{k_1}^n, q_{k_2}^{n-1}, \ldots, q_{k_M}^{n-(M-1)}) \tag{68}$$

[19]We leave it as an exercise for the reader to show that this is so. We could have put the derivation in this TR also, but it was too large to fit in the margins.

and, using similar methods to those shown previously, we can derive the recursion

$$\alpha_{n+1}(k_0, k_1, \ldots, k_{M-1})$$
$$= \sum_{k_{M-1}} \alpha_n(k_0, k_1, \ldots, k_{M-1})\; P\left(q_\ell^{n+1} | X_1^n, x_{n+1}, q_{k_0}^n, q_{k_1}^{n-1}, \ldots, q_{k_{M-1}}^{n-(M-1)}\right)$$
$$P\left(x_{n+1} | X_1^n, q_{k_0}^n, q_{k_1}^{n-1}, \ldots, q_{k_{M-1}}^{n-(M-1)}\right) \tag{69}$$

For the case of $M = 1$, this recursion reduces to the relation that was derived previously.

9.2 Backward Recursion

Similarly, the "backward" probability β can be defined as

$$\beta_n(k_0, k_1, \ldots, k_M) \;=\; p(X_n^N, q_{k_0}^n | X_1^{n-1}, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M}) \tag{70}$$

Again, using similar methods to those shown in Section 8.4, we can derive the recursion

$$\beta_n(k_0, k_1, \ldots, k_M)$$
$$\simeq \sum_\ell \beta_{n+1}(\ell, k_0, k_1, \ldots, k_{M-1}) P(q_{k_0}^n | X_1^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})$$
$$P(x_n | X_1^{n-1}, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M}) \tag{71}$$

in which the only approximation is due to an assumption of an M-th order model.

As with the forward recursion, for the case of $M = 1$, this recursion reduces to the relation that was derived previously.

9.3 MLP Output Targets Update

Plugging the $M-th$ order relations into the expression of the desired posterior probability, we can get the general case for the required recursion for the target values:

$$P(q_{k_0}^n | X, Q_{n-M}^{n-1}, \Theta)$$
$$= \frac{\sum_\ell \; \beta_{n+1}(\ell, k_0, k_1, \ldots, k_{M-1})\; P(q_{k_0}^n | X_1^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})}{\sum_{k_0} \sum_\ell \; \beta_{n+1}(\ell, k_0, k_1, \ldots, k_{M-1})\; P(q_{k_0}^n | X_1^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})} \tag{72}$$

where the predictor terms drop away as they did in the 1st order case.

9.4 M-th order REMAP Training Algorithm

Using the same local acoustic input as described for the 1st order case, we summarize the full procedure as follows:

1. Start from some initial net providing $P(q_{k_0}^n | X_{n-c}^{n+d}, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M}, \Theta^t)$, $t = 0$, $\forall$ possible $(M+1)$-tuples $(q_{k_0}^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})$,

2. Run backward recursions to compute MLP targets $P(q_{k_0}^n|X, q_{k_1}^{n-1}, \ldots, q_{k_M}^{n-M}, \Theta^t)$, $\forall$ possible $(M+1)$-tuples $(q_{k_0}^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})$,

3. For every x_n in the training database, train MLP with output targets equal to $P(q_{k_0}^n|X, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M}, \Theta^t)$, $\forall$ possible M-tuples $(q_{k_0}^n, q_{k_1}^{n-1}, q_{k_2}^{n-2}, \ldots, q_{k_M}^{n-M})$ at the input or for a limited subset as imposed by the HMM topology. This provides us with a new set of parameters Θ^t, for $t = t + 1$.

4. Iterate from 2 until convergence.

The corresponding recognition algorithm is precisely the same as given for the 1st-order case, with the exception that the alpha definition has changed.

9.5 Discussion

The 1st order REMAP algorithm, as defined earlier, turns out to be a convenient special case of the general M-th order algorithm. However, in general it should be expected to be much more difficult to implement higher order systems. While the recursions are not much more complicated, each probabilistic estimate now is dependent on many more terms, so that training may be difficult in general.

Nonetheless, the abstract M-th order case may be of some significance, because other assumptions than the 1st order one may prove to be practical. In the following section, we briefly discuss some perceptually oriented assumptions that could be made, and show the effect they could have on the REMAP procedure.

10 Stochastic Perceptual Auditory-Event-Based Models (SPAMs)

10.1 General Description

In (Morgan *et al.* 1994), we describe a statistical model of speech that is broadly based on perceptual constraints. In this case, the model is constrained to consist of a sequence of auditory events or *avents*, separated by relatively stationary periods (ca. 50-150 ms). Avents correspond to times when the spectrum and amplitude are rapidly changing, which are believed to be the most important regions for phonetic discrimination (Furui 1986).

In this new formalism, states *receive* observations rather than *emit* them. Each state corresponds to an avent from the avent set $Q^* = \{q_0, q_1, \ldots, q_K\}$, in which q_0 represents a non-perceiving (or non-avent) state. In this framework, q_k^n refers to *avent* q_k being perceived at time n. In our current experiments, we are initially assuming that the avents occur during times of significant spectral change, and as such probably correspond to something like truncated diphones. In this approach, all of the stationary regions are tied to the same class (q_0). Markov-like recognition models can use avents as time-asynchronous observations.

A statistical model M_i of a word or a sentence is then defined as a sequence of avent states (q_k's, with $k \neq 0$), with no self-loop, with the non-avent state in between (including a self-loop). In a manner that is analogous to the discriminant HMMs of the earlier sections, we can compute (Morgan *et al.* 1994) $P(M_i|X, \Theta)$ from local probabilities like

$$p(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-\Delta(n)}, \Delta(n)), \quad \left\{ \begin{array}{c} \forall \ell = 0, 1, \ldots, K \\ \forall k = 1, 2, \ldots, K \end{array} \right\} \tag{73}$$

in which $\Delta(n)$ represents the time difference between time frame n and the time that the previous avent was observed. We note that a related variable frame-rate analysis method was already proposed in (Russell *et al.* 1990). In that method, when consecutive acoustic frames were "too" similar, only the first frame and the length of the (dropped) sequence as an additional input variable (our $\Delta(n)$) were passed to the training and recognition process. This idea is similar to SPAMs except that, in our case, we are actually using a recognition model. Also, the assignment of frames to "non-perceiving states" q_0 (essentially, which frames to drop) is based on a global criterion and not on local decisions. However, both approaches will emphasize dynamic portions of the speech signal, incorporate implicit duration modelling, and remove correlation between successive frames. These effects make the recognition process more consistent with HMM assumptions. In (Russell *et al.* 1990), this was shown to improve recognition performance.

Discriminant models can be trained to distinguish among all classes (including the non-avent class). The training data can be automatically aligned using dynamic programming, and the discriminant system (e.g., a neural network) can be iteratively trained towards the optimal segmentation.

In addition to using a discriminant recognition model, this approach should focus modeling power on the perceptually-dominant and information-rich portions of the speech signal, which may also be the parts of the speech signal with a better chance to survive in adverse environmental conditions (since timing between auditory events seems to be an important aspect of robustness to noise). The tying of non-avent states also permits statistical models that include dependencies on states long before the current one, without exploding the parameter requirements.

10.2 REMAP for SPAMs

The SPAM assumptions can be considered as a special case of the M-th order models described earlier, with a few modifications:

1. States are defined to be either an avent class or the single non-avent class that corresponds to all "non-informative" or stationary states. With the exception of this latter class, then, the states may be viewed as themselves representative of spectral transitions.

2. All of the M predecessor states that are not avents provide no information about the current state other than the time between the previous avent and the current state.

3. If two avents fall within the most recent M states, we ignore the older one (assuming that the effects from the more recent one will be more important).

Given these modifications, all of the recursions and relations given in the previous section can be rewritten in a form that is more compact than the general M-th order case. This compactness also is indicative of the reduced computation and difficulty in training, since the dependency on past states is reduced to a single state, as in the 1st order case. The only additional complexity is the addition of explicit dependencies on duration between avents. However, the details of the derivation are somewhat different from the earlier one, and so will be provided here with some explanation.

We define

$$\alpha_{n-1}(\Delta(n), \ell) \;=\; p(X_1^{n-1}, q^{n-1}, \ldots, q_\ell^{n-\Delta(n)}) \tag{74}$$

where the subscript for the state at time $n-1$ is left unspecified, since it is either an avent state (in which case it is state ℓ), or a non-avent state (in which case it is state 0).

We further define a backward factor

$$\beta_n(\Delta(n), j, k) \;=\; p(X_n^N, q_k^n | X_1^{n-1}, q^{n-1}, \ldots, q_j^{n-\Delta(n)}) \tag{75}$$

And then we will compute MLP targets from a ratio of terms that will be computed from their product, as before.

10.2.1 Forward recursion

Keeping in mind that $\Delta(n+2)$ refers to the time difference between $n+2$ and the time that the most recent avent was observed, we require a forward probability

$$\begin{aligned} \alpha_{n+1}(\Delta(n+2), \ell) \;&=\; p(X_1^{n+1}, q_\ell^{n+1}, q^n, \ldots, q_k^{n+2-\Delta(n+2)}) \\ &=\; \sum_{\Delta(n+1), j} p(X_1^{n+1}, q_\ell^{n+1}, q^n, \ldots, q_k^{n+2-\Delta(n+2)}, q_j^{n+1-\Delta(n+1)}) \end{aligned} \tag{76}$$

But if q^{n+1} is an avent, then $\Delta(n+2) \;=\; 1$, and $q_k^{n+2-\Delta(n+2)} \;=\; q^{n+1}$. If q^{n+1} is not an avent, then $\Delta(n+2) \;=\; \Delta(n+1)+1$, and $q_k^{n+2-\Delta(n+2)} \;=\; q^{n+1-\Delta(n+1)}$. Therefore, we can remove the $n+2$ term, and we get

$$\begin{aligned} \alpha_{n+1}(\Delta(n+2), \ell) \;&=\; \sum_{\Delta(n+1), j} p(X_1^{n+1}, q_\ell^{n+1}, q^n, \ldots, q_j^{n+1-\Delta(n+1)}) \\ &=\; \sum_{\Delta(n+1), j} p(X_1^n, q^n, \ldots, q_j^{n+1-\Delta(n+1)}) p(x_{n+1}, q_\ell^{n+1} | X_1^n, q^n, \ldots, q_j^{n+1-\Delta(n+1)}) \end{aligned} \tag{77}$$

The first term is just the previous value for α , and in the second term we can omit explicit mention of states other than $q^{n+1-\Delta(n+1)}$, replacing them by the dependence on $\Delta(n+1)$, as defined by the SPAM assumptions. This gives

$$\begin{aligned}\alpha_{n+1}(\Delta(n+2),\ell) &= \sum_{\Delta(n+1),j} \alpha_n(\Delta(n+1),j)p(x_{n+1},q_\ell^{n+1}|X_1^n,\Delta(n+1),q_j^{n+1-\Delta(n+1)}) \\ &= \sum_{\Delta(n+1),j} \alpha_n(\Delta(n+1),j)p(q_\ell^{n+1}|X_1^{n+1},\Delta(n+1),q_j^{n+1-\Delta(n+1)}) \\ &\qquad p(x_{n+1}|X_1^n,\Delta(n+1),q_j^{n+1-\Delta(n+1)}) \end{aligned} \tag{78}$$

in which the last step was a simple factorization that is the same step that is taken for the 1st order case of Section 8.

Note that as in that case, X_1^{n+1} can be approximated by X_{n-c}^{n+d}, and that the final term is an autoregressive factor that is a SPAM form of the earlier case. As before, this factor will not appear in the training, but will (ideally) be used in the recognition.

10.2.2 Backward recursion

The backward recursion can be obtained as follows:

$$\begin{aligned}\beta_n(\Delta(n),j,k) &= p(X_n^N,q_k^n|X_1^{n-1},q^{n-1},\ldots,q_j^{n-\Delta(n)}) \\ &= \sum_{\Delta(n+1),\ell}\sum_m p(X_n^N,q_k^n,q_\ell^{n+1},q_m^{n+1-\Delta(n+1)}|X_1^{n-1},q^{n-1},\ldots,q_j^{n-\Delta(n)}) \\ &= \sum_{\Delta(n+1),\ell}\sum_m p(X_{n+1}^N,q_\ell^{n+1}|X_1^n,q_m^{n+1-\Delta(n+1)}q_k^n,q^{n-1},\ldots,q_j^{n-\Delta(n)}) \\ &\qquad p(x_n,q_k^n,q_m^{n+1-\Delta(n+1)}|X_1^{n-1},q^{n-1},\ldots,q_j^{n-\Delta(n)}) \end{aligned} \tag{79}$$

However, it turns out that m is entirely determined by $\Delta(n)$, j, and k, since

$$m = \begin{cases} k & \text{if } \Delta(n+1)=1 \\ j & \text{otherwise} \end{cases}$$

Therefore we can remove the inner sum. Additionally, we can remove some other terms. For the first factor, we can remove the dependence on $q^{n-\Delta(n)}$, since it is either identical to or older than the other avent state, and we are assuming a dependence that only goes back one avent into the past. For the second factor, the state $q^{n+1-\Delta(n+1)}$ provides no new information, since it either corresponds to q^n or $q^{n-\Delta(n)}$, depending on whether or not an avent occurs at time n.

Taking into account these points, we get

$$\begin{aligned}\beta_n(\Delta(n),j,k) &= \sum_{\Delta(n+1),\ell} p(X_{n+1}^N,q_\ell^{n+1}|X_1^n,q_k^n,q^{n-1},\ldots,q_m^{n+1-\Delta(n+1)}) \\ &\qquad p(x_n,q_k^n|X_1^{n-1},q^{n-1},\ldots,q_j^{n-\Delta(n)})\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\Delta(n+1),\ell} \beta_{n+1}(\Delta(n+1), m, \ell) p(q_k^n | X_1^n, q^{n-1}, \ldots, q_j^{n-\Delta(n)}) \\
&\qquad p(x_n | X_1^{n-1}, q_k^n, q^{n-1}, \ldots, q_j^{n-\Delta(n)}) \\
&= \sum_{\Delta(n+1),\ell} \beta_{n+1}(\Delta(n+1), m, \ell) p(q_k^n | X_1^n, \Delta(n), q_j^{n-\Delta(n)}) \\
&\qquad p(x_n | X_1^{n-1}, \Delta(n), q_j^{n-\Delta(n)}) \qquad (80)
\end{aligned}
$$

where the second last step gave the factorization between posterior and predictor probabilities, and the last step summarized the non-avent states by their duration, which is consistent with the SPAM assumptions .

10.3 MLP Output Targets Update

Plugging the SPAM recursions into the expression of the desired posterior probability, we can get the required recursion for the SPAM MLP target values:

$$
\begin{aligned}
P(q_k^n | X, q_j^{n-\Delta(n)}, \Delta(n)) &= \frac{\beta_n(\Delta(n), j, k)}{\sum_k \beta_n(\Delta(n), j, k)} \\
&= \frac{\sum_{\Delta(n+1),\ell} \beta_{n+1}(\Delta(n+1), m, \ell)\ P(q_k^n | X_1^n, q_j^{n-\Delta(n)}, \Delta(n))}{\sum_k \sum_{\Delta(n+1),\ell} \beta_{n+1}(\Delta(n+1), m, \ell)\ P(q_k^n | X_1^n, q_j^{n-\Delta(n)}, \Delta(n))} \qquad (81)
\end{aligned}
$$

where the predictor terms drop away as they did in the 1st order case, and where the dependence on the data sequence can be approximated by a local window as before.

10.4 Discussion

Unlike the general M-th order case, the SPAM version of the REMAP estimations is only expanded out by a factor corresponding to the range of permitted durations to the most recent avent. Thus, in principle one can derive important correlations between the states over a significant region of time (e.g., 100 msec) without a crippling increase in computational or training data requirements. This simplicity is dependent, of course, on the SPAM assumptions, which may not be good ones. However, this section provides an alternate to the 1st-order approach. At this stage we have no way of knowing which will provide the more useful model of speech nonstationarities for recognition. Nonetheless, this section (and the M-th order case described previously) show that there are a number of modifications and extensions to the basic 1st order REMAP procedure that can be considered.

11 Related Discriminant Approaches

As briefly recalled at the beginning of this paper, there are two conceptual problems with the maximum likelihood approach (apart from all the hypotheses that are usually done in systems based on standard HMMs):

- It is implicitly assumed that the model (with all its assumptions relative to its topology and probability density functions) is accurate and reflects the structure of the data (although the data might not adhere to the constraints imposed by the HMMs). If we had enough training data, it would probably be preferable to infer all the parameters of the models (including topology and non-parametric probability density functions) directly from the data. This can be seen as implicitly using a Bayes or MAP criterion (i.e., maximizing $P(M|X)$) during training instead of ML. Since MAP includes the effects of prior information, the language model would also be inferred from the training data. However, it appears that this would require a prohibitive amount of training data.
- By training with ML instead of MAP, we strongly reduce the discriminant properties of HMMs. Ideally, each HMM should be trained not only to generate high probabilities for its own class, but also to discriminate against rival models.

Both of these two points (but particularly the second one) are related to the discussion that follows on discriminant criteria for HMM training.

The ML criterion can lead to optimal recognition performance only if the model is an exact statistical model of the source. It has indeed been shown that ML will give the optimal estimator under a certain set of conditions (Nadas 1983). However, these conditions are rarely (if ever) satisfied in speech recognition. With an inaccurate model, the best we can do is to optimize its ability to distinguish between the underlying source symbols, which is typically achieved by replacing the ML criterion by a discriminant The source symbols of a language model can be defined at many possible levels, including sentences, words, phones or HMM-states, each of them leading to a different discriminant criterion.

In this section, we briefly discuss related discriminant approaches that have been proposed for speech recognition and compare them with the approach described in this paper.

11.1 Maximum Mutual Information (MMI)

Initially introduced in (Bahl *et al.* 1986; Brown 1987), this method aims at maximizing the mutual information (Cover & Thomas 1991) between a set of (sentence) models M_{w_j} and the associated sequences of acoustic vectors X_j. This mutual information is then defined as (Cover & Thomas 1991):

$$I(M_{w_j}, X_j|\Theta) = \sum_{M_{w_j}, X_j} p(M_{w_j}, X_j|\Theta) \log \frac{p(M_{w_j}, X_j|\Theta)}{P(M_{w_j}|\Theta)p(X_j|\Theta)} \tag{82}$$

$$= E_{p(M_{w_j}, X_j|\Theta)} \left\{ \log \frac{p(M_{w_j}, X_j|\Theta)}{P(M_{w_j}|\Theta)p(X_j|\Theta)} \right\} \tag{83}$$

where Θ is the whole parameter space (for all models) in which optimization is performed and the sum over M_{w_j} X_j represents a sum over all training utterances. $E_{p(M_{w_j}, X_j|\Theta)}$ stands for the expected value according to the mass function $p(M_{w_j}, X_j|\Theta)$.

For one particular (M_{w_j}, X_j) set, we then have:

$$\begin{aligned} I(M_{w_j}, X_j|\Theta) &= \log \frac{p(M_{w_j}, X_j|\Theta)}{P(M_{w_j}|\Theta)p(X_j|\Theta)} \\ &= \log \frac{p(X_j|M_{w_j}, \Theta)}{\sum_{i=1}^{I} p(X_j|M_i, \Theta)P(M_i|\Theta)} \end{aligned} \tag{84}$$

As already mentioned in Section 4.1, the language model parameters Θ^* are often assumed independent of the acoustics parameters Θ and are estimated independently from a (large) text copra. Furthermore, the likelihoods $p(X_j|M_i, \Theta)$ depend only on the parameters Θ_i present in M_i. As a consequence, (84) can be rewritten as

$$I(M_{w_j}, X_j|\Theta) = \log \frac{p(X_j|M_{w_j}, \Theta_{w_j})}{\sum_{i=1}^{I} p(X_j|M_i, \Theta_i)P(M_i|\Theta^*)} \tag{85}$$

in which the contribution of each term in the denominator is weighted according to the prior probability of the associated sentence as given by the language model and which is independent of the acoustic training data.

It is obvious that (84) and (85) are discriminant criteria. In (Bahl *et al.* 1986) it is shown that it is possible to get some kind of re-estimation recursion of local probabilities but, unfortunately, there is no proof that the recursion converges and there is no guarantee that the new estimates of (e.g., transition) probabilities are positive. As a consequence, a local gradient ascent method is usually used for optimization and the standard (likelihood-based) forward-backward recurrences are used to estimate the gradient. This is similar to the Alpha-Nets presented in (Bridle 1990)[20] in which the gradient of the mutual information criterion takes the form of the backward recurrence used in the Baum-Welch algorithm. In the framework of hybrid HMM/ANN systems, this MMI criterion has been used in (Bengio *et al.* 1992), in which the ANN generates the sequence of acoustic vectors for the HMM and is trained to optimize the (global) MMI. In that paper, it is shown that it is possible to compute the gradient of the HMM training criterion (MMI or ML) with respect to the parameters of the ANNs.

However, in addition to "theoretical" problems, this algorithm suffers from a "practical" problem for continuous speech recognition. Indeed, optimization of Θ to maximize (84) requires not only a forward recurrence for the numerator, but also many forward recurrences for the denominator to estimate the contribution of all possible rival model.

Several solutions have been proposed to alleviate this problem, including:

1. If phoneme models are trained, the use of a "looped" phonetic model, i.e., a word model that allows any possible phoneme sequence (Merialdo 1988). This model may generate all possible phoneme sequences and, by running the forward recursion through it, may provide the summed probability in the denominator of (84).

[20] And which are not "regular" nets.

2. Estimating the denominator in (84) by running an N-best algorithm providing the N-best (rival) sentences through which we run the forward recursion.

With the algorithm we propose in this paper, in addition to all the advantages of "standard" HMM/MLP hybrids (i.e., local nonlinear discrimination, time correlation and no significant assumptions about probability density functions), we do not need to run multiple forward recursions. Also, all probabilities will always be estimates of actual (local and global) posterior probabilities, will be positive and will sum to one.

11.2 MAP Probability

As discussed in Section 4.1 , the ideal criterion to be maximized during training of HMMs is the posterior probability $P(M_{w_j}|X_j)$ that a Markov model M_{w_j} generates the acoustic vector sequence X_j. According to Bayes' rule, this probability can be written as

$$P(M_{w_j}|X_j,\Theta) \;=\; \frac{p(X_j|M_{w_j},\Theta)P(M_{w_j}|\Theta)}{p(X|\Theta)} \tag{86}$$

where Θ is the parameter set in which optimization is performed. As already mentioned in Section 4.1, the language model parameters Θ^* are often assumed independent of the acoustic parameters Θ. Furthermore, the likelihoods $p(X|M_{w_j},\Theta)$ depend only on the parameters Θ_{w_j} present in M_{w_j}. As a consequence, (86) may be written as:

$$P(M_{w_j}|X_j,\Theta) \;=\; \frac{p(X_j|M_{w_j},\Theta_{w_j})P(M_{w_j}|\Theta^*)}{\sum_{i=1}^{I} p(X_j|M_i,\Theta_i)P(M_i|\Theta^*)} \tag{87}$$

As shown in Section 4.1, restricting maximization of $P(M_{w_j}|X_j,\Theta)$ to the subspace of the M_{w_j} parameters leads to the maximum likelihood criterion usually used in standard HMM training.

On the other hand, maximization of (87) with respect to the whole parameter space is equivalent to maximization of

$$\frac{1}{1+\frac{\sum_{i\neq w_j} p(X_j|M_i,\Theta_i)P(M_i|\Theta^*)}{p(X_j|M_{w_j},\Theta_{w_j})P(M_{w_j}|\Theta^*)}}$$

or maximization of

$$\frac{p(X_j|M_{w_j},\Theta_{w_j})P(M_{w_j}|\Theta^*)}{\sum_{i\neq w_j} p(X_j|M_i,\Theta_i)P(M_i|\Theta^*)} \tag{88}$$

which leads to discriminant models since it implies that the contribution of the numerator should be enhanced while the contribution of all possible rival models, represented by the denominator, should be reduced.

If the parameters of the language model Θ^* are assumed independent (and estimated independently) of the acoustic parameters, $P(M_{w_j}|\Theta^*)$ is then a constant during acoustic training and can be disregarded in the maximization process. Maximization of (87) is then equivalent to maximization of

$$\frac{p(X_j|M_{w_j}, \Theta_{w_j})}{\sum_{i=1}^{I} p(X_j|M_i, \Theta_i)P(M_i|\Theta^*)} \tag{89}$$

which is nothing else than the exponential of the mutual information criterion (see next section). Given (88), the sum over i in (89) can include all models allowed by the grammar (including the correct) or only the rival models.

In conclusion, *if the parameters of the language model are estimated independently of the acoustic parameters*, maximization of (86) is then simply equivalent to maximization of mutual information, and there is no advantage in optimizing posterior probabilities instead of mutual information.[21] However, in general the model priors (language model) not only refers to syntactical information at the word level, but also to prior information at, for instance, the word and phoneme levels. In those cases, one could attempt to optimize the language model *together* with the acoustic model, which is exactly what the full MAP optimization is doing.

11.3 Embedded Viterbi

A simple, frame-based, discriminative criterion is the Viterbi approximation of the "full" MAP presented in this paper. In this case, each HMM-state is considered as a separate source symbol and the training algorithm attempts to maximize the posterior probability of detecting the correct state q_k^n given x_n. The Viterbi algorithm will then minimize

$$\log \overline{P}(M|X) = \sum_{n=1}^{N} \log P(q^n|x_n) \tag{90}$$

This is what has been used for several years in standard hybrid HMM/MLP systems (Wooters 1993; Renals *et al.* 1994; Renals *et al.* 1992; Robinson 1994). In this case, the Markov models are only used to find state targets that are then used to estimate local posterior probabilities, which can be done, e.g., by an MLP. This kind of algorithm is very simple and basically performs discrimination between individual states, rather than between phonemes, words or sentences. Although it is possible to prove (see Section 5.5 and (Bourlard & Morgan 1994)) that preserving the MAP constraints at the local level also preserves the MAP constraint at the global (sentence) level, it is not certain that improving discrimination at the local level implies improving discrimination at the global level, i.e., iteratively increasing the global a posteriori probability.

11.4 Generalized Probabilistic Descent (GPD)

Generalized Probabilistic Descent (GPD) is another discriminant approach that is sometimes used to train speech recognition systems. GPD is actually very close in spirit to MMI,

[21] Apart, of course, from the advantages of using an MLP and acoustic context to estimate local probabilities.

although it permits generalization to different kinds of training criteria (Katagiri *et al.* 1991).

The general idea of GPD is actually simple and can be summarized as follows. Given the whole set of parameters Θ, define a discriminant function associated with each (word or sentence) model M_i as $g_i(X;\Theta)$. This discriminant function can be any differentiable distance function or probability distribution. Several instances of this are discussed in (Katagiri *et al.* 1991), each of them leading to different interpretations (as is also the case for MMI and MAP training). However, often the discriminant function is defined as:

$$g_i(X;\Theta) = -\log p(X|M_i,\Theta) \tag{91}$$

Here again, (91) can be considered as the "full" (word or sentence) likelihood, the best-path (Viterbi) approximation (referred to as "segmental GPD training") or any intermediate solution like a sum over the S-best matching path scores. Another solution could be to define $g_i(X;\Theta)$ as the MMI in (85). However, since this will then be used in a discriminant measure (as defined below) taking all the classes into account, it can be easily shown that using MMI or full likelihoods as discriminant functions results in the same misclassification measure.

Classification (i.e., recognition) will then be based on that discriminant function according to the (usual) rule

$$X \in M_j \text{ if } j = \operatorname*{argmax}_i \, g_i(X;\Theta) \tag{92}$$

Given this discriminant function, we can define a misclassification measure that will measure the "distance" between one specific class and all the others. Here again, several measures can be used, each of them leading to different interpretations. However, one of the most general ones given in (Katagiri *et al.* 1991) is:

$$d_j(X;\Theta) = g_j(X,\Theta) - \log\left[\frac{1}{I-1}\sum_{i\neq j}\exp(\eta g_i(X;\Theta))\right]^{1/\eta} \tag{93}$$

in which I represents the total number of possible reference models. It is easy to see that if $\eta = 1$, (93) is then equivalent to (85) in which all the priors are assumed equal to $1/I$.

The error measure (93) could be used as the criterion for optimization by a gradient-like procedure, which would result in something very similar to MMI training. However, the goal of GPD is to minimize the actual misclassification rate, which can be achieved by passing $d_j(X;\Theta)$ through a nonlinear, nondecreasing, differentiable function F (such as a sigmoid function) and then minimizing

$$E(\Theta) = \sum_j \sum_{X\in M_j} F(d_j(X;\Theta)) \tag{94}$$

Other functions can be used to approximate the error rate. For example, we can also assign zero cost when an input is correctly classified and a unit cost when it is not properly classified, which is then another formulation of the minimum Bayes risk.

As briefly shown above, this approach is certainly very general and includes several discriminant approaches as particular cases. For some problems such as continuous speech recognition, however, this approach has the same potential difficulty as MMI, i.e., the need to estimate "scores" (whatever they might be) of both the correct model and for all possible rival models.

11.5 Discussion

A wide range of discriminant approaches to speech recognition have been studied by researchers. A significant difficulty that has remained in applying these approaches to continuous speech recognition has been the requirement to run computationally intensive algorithms on all of the rival sentences. Since this is not generally feasible, compromises must always be made in practice. For instance, estimates for all rival sentences can be derived from a list of the "N-best" utterance hypotheses, or by using a model of all possible phonemes.

It may be that the fundamental potential practical advantage to REMAP is that it is a discriminant training algorithm that does not require the explicit enumeration of possible rivals in order to get a globally and locally discriminant training. In any event, it seems that it is at least important to explore full MAP approaches, which actually appear to be more straightforward than MMI (for continuous speech). Additionally, of course, we retain the advantages that we have described about using MLPs for the local estimation (freedom from distributional assumptions, and ability to learn high order correlations between flexibly chosen features).

12 Conclusions

The writing of a paper can have many purposes. One of the most important for us was to write clearly in one place the sets of ideas that we have been working on these last few months. In this sense we have by definition succeeded, since we now know where to look for the information that we held so clearly in our heads a few weeks ago.

Additionally, however, reports have a more social role; others may be also interested in learning about our work. Thus we wish to communicate our ideas to other readers as well. This is a much harder task. There is a great deal of detail here, and it may be hard for someone who is not working in an extremely similar area to easily follow all of it. It is unfortunately quite likely that many readers may have become lost in the detail.

Therefore, as a public service, we offer this brief summary of the major conclusions (we think) that we have reached concerning our recent work on recursive estimation and maximization of a posteriori probabilities (for speech recognition):

- We have a method to estimate and train full MAPs (for sequences).
- This can be used in a new form of hybrid HMM/MLP in which, in addition to the advantages of standard HMM/MLP hybrids, we use "full" posterior probabilities for

training and recognition.

- We still use neural nets (in our case MLPs, though recurrent nets or TDNNs could be used) to estimate local posterior probabilities (conditional transition probabilities), but our nets are trained with probabilistic targets that are themselves estimates of local posterior probabilities (conditioned on the acoustic data and the previous state).
- We have a way, similar in spirit to the forward-backward recursions of the Baum-Welch algorithm, to estimate these optimal targets given a previously trained neural network.
- We have a convergence proof that guarantees iterative increase of the global posterior probability (see Appendix A).
- This method is valid for any hybrid HMM/MLP system but, in this paper, was developed in the framework of "Discriminant HMMs" using conditional transition probabilities.
- It is expected that this approach will prove most important for transition-based recognition systems.
- This approach can be generalized easily to other transition-based recognizers like SPAMs.
- We don't have any major experimental results yet, but this should come soon...
- It is however possible that this new approach (even just the theory) opens several new research paradigms (we may have just opened Pandora's box...)

Acknowledgments

We would like to thank John Lazzaro and Steve Renals for their comments and corrections on earlier drafts. Particularly for the SPAM-related work, we are indebted to Steve Greenberg and Hynek Hermansky for pushing us in the direction of interest in transitions. The earlier (mid-80's) work in discriminant HMMs on which this is based was done in collaboration with Chris Wellekens. We gratefully acknowledge the support of the Office of Naval Research, URI No. N00014-92-J-1617 (via UCB), ESPRIT project 6487 (WERNICKE) (through ICSI), and ICSI in general for supporting this work.

BAHL, L. R., F. JELINK, & R. MERCER. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-5.179–190.

BAHL, L.R., P.F. BROWN, P.V DE SOUZA, & R.L. MERCER. 1986. Maximum mutual information estimation of hidden Markov model parameters. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 49–52, Tokyo, Japan.

BAUM, L. E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3.1–8.

——, & T. PETRIE. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 36.1554–1563.

——, T. PITRIE, G. SOULES, & N. WEISS. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions in Markov chains. *The Annals of Mathematical Statistics* 41.164–171.

BENGIO, Y., R. DE MORI, G. FLAMMIA, & R. KOMPE. 1992. Global optimization of a neural network-hidden Markov model hybrid. *IEEE trans. on Neural Networks* 3.252–258.

BOURLARD, H., & N. MORGAN. 1994. *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers.

——, & C. J. WELLEKENS. 1989. Links between Markov models and multilayer perceptrons. In *Advances in Neural Information Processing Systems 1*, ed. by D.J. Touretzky, 502–510, San Mateo. Morgan Kaufmann.

BRIDLE, J.S. 1990. Alpha-nets: A recurrent "neural" network architecture with a hidden Markov model interpretation. *Speech Communication* 9.83–92.

BROWN, P. F., 1987. *The Acoustic-Modelling Problem in Automatic Speech Recognition*. Pittsburgh, PA: CMU dissertation.

CHOW, YEN-LU. 1990. Maximum mutual information estimation of HMM parameters for continuous speech recognition using the n-best algorithm. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 701–704, Albuquerque, New Mexico.

COVER, THOMAS M., & JOY A. THOMAS. 1991. *Elements of Information Theory*. New York: John Wiley and Sons, Inc.

DEMPSTER, A. P., N. M. LAIRD, & D. B. RUBIN. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B* 34.1–38.

DENG, L., & D.X. SUN. 1994. A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *JASA* 95.2702–2719.

DEVIJVER, P.A. 1985. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters* 3.369–373.

DIGALAKIS, V.V., 1992. *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*. Boston University dissertation.

DUDA, R. O., & P.E. HART. 1973. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc.

FURUI, S. 1986. Speaker independent isolated word recognizer using dynamic features of speech spectrum. *IEEE Trans. on Acoustics, Speech, and Signal Processing* 34.52–59.

GAROFOLO, J. S., 1988. *Getting Started with the DARPA TIMIT CD-ROM: an Acoustic Phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburgh, Maryland.

GHITZA, O., & M.M. SONDHI. 1993. Hidden Markov models with templates as non-stationary states: an application to speech recognition. *Computer Speech and Language* 2.101–119.

GISH, H. 1990. A probabilistic approach to the understanding and training of neural network classifiers. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 1361–1364, Albuquerque, NM.

GLASS, J.R., 1988. *Finding Acoustic Regularities in Speech Applications to Phonetic Recognition*. M.I.T dissertation.

GOLDENTHAL, W. D., 1994. *Statistical Trajectory Models for Phonetic Recognition*. M.I.T dissertation.

HAEB-UMBACH, R., & H. NEY. 1992. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 1, 13–16, San Francisco, USA.

HERMANSKY, H. 1990. Perceptual linear predictive (PLP) analysis of speech. *JASA* 87.

——, N. MORGAN, A. BAYYA, & P. KOHN. 1992. RASTA-PLP speech analysis technique. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 1, 121–124, San Francisco, CA. IEEE.

HUANG, X.D., Y. ARIKI, & M.A. JACK. 1990. *Hidden Markov Models For Speech Recognition*. Edinburgh University Press.

JELINEK, F. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE* 64.532–556.

JORDAN, M. I., & R. A. JACOBS. 1994. Hierachical mixtures of experts and the EM algorithm. *Neural computation* 6.181–214.

JUANG, B. H., & L.R. RABINER. 1985. Mixture autoregressive hidden Markov models for speech signals. *IEEE ASSP Magazine* 6.1404–1413.

KATAGIRI, S., C.H. LEE, & JUANG B.H. 1991. New discriminative training algorithms based on the generalized probabilistic decent method. In *Proc. of the IEEE Workshop on Neural Netwroks for Signal Processing*, ed. by B.H. Juang, S.Y. Kung, & C.A. Kamm, 299–308.

KONIG, Y., & N. MORGAN. 1994. Modeling dynamics in connectionist speech recognition - the time index model. In *Proceedings Int'l Conference on Spoken Language Processing*, 1523–1526, Yokohama, Japan.

KUBALA, G. F., Y. CHOW, A. DERR, M. FENG, O. KIMBALL, J. MAKHOUL, P. PRICE, J. ROHLICEK, S. ROUCOS, R. SCHWARTZ, & J. VANDEGRIFT. 1988. Continuous speech recognition results of the BYBLOS system on the DARPA 1000-word Resource Management databasement. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 291–294.

LEE, K. F. 1989. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic Publishers.

LEVINSON, S. E., L. R. RABINER, & M. M. SONDHI. 1983. An introduction to the application of the theory of probabilistic functions on a Markov process to automatic speech recognition. *Bell System Technical Journal* 62.243–272.

LEVINSON, S.E. 1985. Structural methods in automatic speech recognition. *Proceedings of the IEEE* 73.1625–1650.

LIPORACE, L. A. 1982. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Trans. on Information Theory* IT-28.729–734.

LUBENSKY, DAVID M., AYMAN O. ASADI, & JAYANT M. NAIK. 1994. Connected digit recognition using connectionist probability estimators and mixture-gaussian densities. In *Proceedings Int'l Conference on Spoken Language Processing*, 295–298, Yokohama, Japan.

MERIALDO, B. 1988. Phonetic recognition using hidden Markov models and maximum mutual information training. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 111–114, New York.

MORGAN, N., H. BOURLARD, S. GREENBERG, & H. HERMANSKY. 1994. Stochastic perceptual auditory-event-based models for speech recognition. In *Proceedings Int'l Conference on Spoken Language Processing*, 1943–1946, Yokohama, Japan.

NADAS, A. 1983. A decision-theortic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 31.814–817.

NEUWIRTH, L.P., 1970. Unpublished lectures.

NEY, H. 1984. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.263–271.

NORMANDIN, Y., R. LACOUTURE, & R. CARDIN. 1994. MMIE training for large vocabulary continuous speech recognition. In *Proceedings Int'l Conference on Spoken Language Processing*, 1367–1370, Yokohama, Japan.

OSTENDORF, M., & S. ROUKOS. 1989. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE ASSP trans.* 37.1857–1869.

PAUL, D., & B. NECIOGLU. 1993. The Lincoln large-vocabulary stack-decoder HMM CSR. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 2, 660–663, Minneapolis.

PORITZ, A. B. 1982. Linear predictive hidden Markov models and the speech signal. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 1291–1294, Paris, France.

——, & A.L. RICHTER. 1986. On hidden Markov models in isolated word recognition. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, volume 1 of *Tokyo, Japan*, 705–708.

PORITZ, A.B. 1988. Hidden Markov models: A guided tour. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 7–13, New-York,.

RABINER, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.257–285.

RENALS, S., N. MORGAN, & H. BOURLARD. 1991. Probability estimation by feed-forward networks in continuous speech recognition. Technical Report TR-91-030, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704.

——, ——, ——, M. COHEN, & H. FRANCO. 1994. Connectionist probability estimators in HMM speech recognition. *IEEE Transactions on Speech, and Audio Processing* 2.161–174.

——, N. MORGAN, M. COHEN, H. FRANCO, & H. BOURLARD. 1992. Connectionist probability estimation in the DECIPHER speech recognition system. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 601–604, San Francisco, California. IEEE.

RICHARD, MICHAEL D., & RICHARD P. LIPPMANN. 1991. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation* 3.461–483.

ROBINSON, TONY. 1994. An application of recurrent nets to phone probability estimation. *IEEE transactions on Neural Networks* 5.298–305.

RUMELHART, D. E., G. E. HINTON, & R. J. WILLIAMS. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing. Explorations of the Microstructure of Cognition*, ed. by D. E. Rumelhart & J. L. McClelland, volume 1: Foundations. MIT Press.

RUSSELL, M.J., K.M. PONTING, S.M. PEELING, S.R. BROWNING, J.S. BRIDLE, & R.K. MOORE. 1990. The ARM continuous speech recognition system. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, Albuquerque, NM, 69–71.

STOLCKE, A., & S. OMOHUNDRO. 1993. Hidden Markov Model induction by Bayesian model merging. In *Advances in Neural Information Processing Systems 5*, 11–18. San Mateo, Ca.: Morgan Kaufman.

VINTSYUK. 1971. Element-wise recognition of continuous speech composed of words from a specified dictionary. *kibernetica* 133–143.

WELLEKENS, C. J. 1987. Explicit time correlation in hidden Markov models for speech recognition. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, 384–386, Dallas, USA.

WOOTERS, CHARLES C., 1993. *Lexical Modeling in a Speaker Independent Speech Understanding System*. Berkeley, CA: University of California dissertation.

ZAVALIAGKOS, G., Y.ZHAO, R. SCHWARTZ, & J. MAKHOUL. 1994. A hybrid segmental neural net/hidden Markov model system for continuous speech recognition. *IEEE Transactions on Speech and Audio Processing* 2.151–160.

A Convergence Proof of REMAP HMM/MLP Training

A.1 Introduction

Below we prove convergence (at least to a local minimum) of the 1st order REMAP training algorithm described in Section 8. We show that for any training sentence X[22], an iteration consisting of

1. estimating new MLP training targets from a previously trained MLP via backward recursion, and

2. training the MLP with the new targets

will increase the global MAP probability of the sentence model given the sequence of acoustic vectors, i.e. $P(M|X)$. It is easy to see that this proof can be generalized to several training sentences since this is then simply equivalent to training on a long sentence built up by concatenating all training sentences (with additional start and end point constraints).

The proof has three main steps:

1. Defining an auxiliary function such that maximizing that function is equivalent to maximizing the global posterior probability of the correct model and, (since such probabilities must sum to 1 for the complete set of possible models) minimizing the posterior probabilities of the rival models.

2. Finding new targets for training the MLP that maximize the auxiliary function.

3. Showing that training the MLP with those new targets (using a weighted relative entropy error criterion) leads to an increase in the value of the auxiliary function.

Note that while this paper has largely assumed the use of an MLP for the required probability estimation, other gradient-trained estimators (such as a recurrent network) could also be used.

A.2 Definitions

Let us define an auxiliary function[23] $R(\upsilon_1, \upsilon_2)$ as a function of two probability sets $\upsilon_1, \upsilon_2 \in \Upsilon$, where Υ is the set of all conditional transition probability sets that have been defined in this paper as the set of probabilities $P(q_\ell^n|x_n, q_k^{n-1})$ for all permitted values of k, ℓ, and n. Each set thus contains the K^2N possible condition transition probabilities, where K is the number of states in the model, and N is the number of acoustic vectors. Note that the probability sets can be a function of a probability estimator. In our case, these

[22] To simplify our notations, all the following proofs will be done for only one training sentence X associated with the Markov model M, but it is easy to see that all the proofs remain valid in the case of several training sentences X_j associated with M_{w_j}, for $j = 1, \ldots, J$.

[23] This auxiliary function is usually denoted $Q(\cdot)$. However, to avoid any possible ambiguity with an HMM state sequence, we denote it $R(\cdot)$ in this paper.

probabilities are estimated by an MLP with parameter (weight) set Θ, in which case we denote the probabilities as $P(q_\ell^n|x_n, q_k^{n-1}, \Theta)$. In this case, the probability set υ also becomes a function of Θ and is then denoted by $\upsilon(\Theta)$.

The auxiliary function $R(\upsilon_1, \upsilon_2)$ is then defined as

$$R(\upsilon_1, \upsilon_2) \;=\; \frac{1}{P(M|X, \upsilon_1)} \sum_\Gamma P(M, \Gamma|X, \upsilon_1) \; \log P(M, \Gamma|X, \upsilon_2) \tag{95}$$

where Γ is a legal path (state sequence) in model M and $P(M, \Gamma|X, \upsilon_i)$ represents the probability of a specific path Γ in M given a probability set υ_i.

A.3 Theorem 1

Theorem 1:

IF $R(\upsilon_1, \upsilon_2) \geq R(\upsilon_1, \upsilon_1)$
THEN $P(M|X, \upsilon_2) \geq P(M|X, \upsilon_1)$.

In other words, if we can find a new set of probabilities υ_2 increasing R, the new set of probabilities will also increase the posterior probability of the model M.

Proof:

$$\begin{aligned}
&\log \frac{P(M|X, \upsilon_2)}{P(M|X, \upsilon_1)} \\
&\quad = \log \left[\sum_\Gamma \frac{P(M, \Gamma|X, \upsilon_1)}{P(M|X, \upsilon_1)} \frac{P(M, \Gamma|X, \upsilon_2)}{P(M, \Gamma|X, \upsilon_1)} \right] \\
&\quad \geq \sum_\Gamma \frac{P(M, \Gamma|X, \upsilon_1)}{P(M|X, \upsilon_1)} \log \left[\frac{P(M, \Gamma|X, \upsilon_2)}{P(M, \Gamma|X, \upsilon_1)} \right] \\
&\quad \text{(because of Jensen's inequality and concavity of log function)} \\
&\quad = R(\upsilon_1, \upsilon_2) - R(\upsilon_1, \upsilon_1)
\end{aligned}$$

(Note that the random variable used for the Jensen's inequality is a $\frac{P(M, \Gamma|X, \upsilon_2)}{P(M, \Gamma|X, \upsilon_1)}$ which is a deterministic function of the random variable Γ). As a consequence, we have:

$$\log \frac{P(M|X, \upsilon_2)}{P(M|X, \upsilon_1)} \geq R(\upsilon_1, \upsilon_2) - R(\upsilon_1, \upsilon_1) \tag{96}$$

which proves the theorem. If a new set of probabilities υ_2 that makes the right-hand side of (96) positive can be found, then the model re-estimation algorithm can be guaranteed to increase the posterior probability of the model to $P(M|X, \upsilon_2)$.

A.4 Theorem 2

The question that arises from the first theorem is how to find a new set of probabilities υ_2 that increases the value of the auxiliary function $R(\cdot)$ and, consequently, the posterior probability of the correct model (and therefore also minimizes the posterior probability of the rival models).

Theorem 2:

Given υ_1 a fixed set of probabilities that is estimated by an MLP with a fixed set of weights Θ, we show that $R(\upsilon_1, \upsilon_2)$ attains its maximum value when the conditional transitional probabilities $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \in \upsilon_2$ are defined as[24]

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta), M) \tag{97}$$

Proof:

We now treat the conditional transitional probabilities $P_{\upsilon_2}(\cdot)$ as the variables for the optimization. To maximize $R(\cdot)$ in that transition probability space we thus have to solve K^2N equations of the form:

$$\frac{\partial R(\upsilon_1(\Theta), \upsilon_2)}{\partial P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} = 0 \tag{98}$$

under the KN constraints

$$\sum_{j=1}^{K} P_{\upsilon_2}(q_j^n|x_n, q_k^{n-1}) = 1, \ \forall k = 1, \ldots, K; \forall n = 1, \ldots, N \tag{99}$$

Using Lagrange multipliers $\Lambda = (\lambda_{1,1}, \ldots, \lambda_{1,N}, \ldots, \lambda_{K,1}, \ldots, \lambda_{K,N})^t$, maximization of $R(\cdot)$ as defined in (95) under the constraints specified in (99) is then equivalent to maximization of

$$R^*(\upsilon_1, \upsilon_2, \Lambda) \;=\; R(\upsilon_1, \upsilon_2) + \sum_{k,n} \lambda_{k,n} \left(1 - \sum_{j=1}^{K} P_{\upsilon 2}(q_j^n|x_n, q_k^{n-1})\right) \tag{100}$$

So we have K^2N unknowns that are the conditional transition probabilities in υ_2 and KN unknowns that are the Lagrange multipliers. Furthermore, we have the same number of equations as we compute the partial derivative of $R^*(\cdot)$ relative to each unknown and equalize it to zero. Fortunately it turns out that we can solve each of the K^2N equations described above independently and find solutions that satisfy the KN constraints.

Considering a specific transition (q_k^{n-1}, q_ℓ^n), we then have:

$$\frac{\partial R^*(\cdot)}{\partial \lambda_{k,n}} = 0 \tag{101}$$

[24] Of course, all x_n's in the following should be replaced by X_{n-c}^{n+d} if local contextual input is used, or X_1^n for a recurrent network.

which returns the constraint (99). For the partial derivative of $R^*(\cdot)$ with respect to $P_{\upsilon_2}(\cdot)$, we first use the following decomposition:

$$P(M, \Gamma|X, \upsilon_2) = P(\Gamma|X, \upsilon_2)P(M|\Gamma, X, \upsilon_2) \tag{102}$$

According to (34), the first factor in (102) can be expressed as

$$P(\Gamma|X, \upsilon_2) \ = \ \prod_{n=1}^{N} P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \tag{103}$$

Also, the second factor in (102) can be assumed independent of the conditional transition probabilities (i.e., given a state sequence, the probability of the model does not depend on the transition probabilities), in which case we have:

$$P(M|\Gamma, X, \upsilon_2) = P(M|\Gamma, X) \tag{104}$$

Taking partial derivatives, then, the second term in (102) has no effect, since it can be assumed to have no dependence on $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})$, and since it only appears as an additive term once the logarithmic function has been applied.

We then get

$$\begin{aligned}
&\frac{\partial R^*(\cdot)}{\partial P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} \\
&\quad = \frac{1}{P(M|X, \upsilon_1)} \sum_{\Gamma_{k,\ell,n}} \left[P(M, \Gamma_{k,\ell,n}|X, \upsilon_1) \frac{1}{P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} \right] - \lambda_{k,n} \\
&\quad = 0
\end{aligned} \tag{105}$$

where $\Gamma_{k,l,n}$ stands for those paths containing the transition (q_k^{n-1}, q_ℓ^n). Solving (105) gives us:

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \ = \ \frac{1}{\lambda_{k,n}} \frac{\sum_{k,\ell,n} P(M, \Gamma_{k,\ell,n}|X, \upsilon_1)}{P(M|X, \upsilon_1)} \tag{106}$$

We now have to find the value (or "a" value) of $\lambda_{k,n}$ that guarantees that the new estimates of $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})$ will meet the constraint. It is possible to find it without directly solving the set of equations. It is indeed easy to show that:

$$\begin{aligned}
\frac{\sum_{k,\ell,n} P(M, \Gamma_{k,\ell,n}|X, \upsilon_1)}{P(M|X, \upsilon_1)} &= \frac{\sum_{\forall\Gamma} P(M, \Gamma, q_\ell^n, q_k^{n-1}|X, \upsilon_1)}{P(M|X, \upsilon_1)} \\
&= \frac{P(M, q_\ell^n, q_k^{n-1}|X, \upsilon_1)}{P(M|X, \upsilon_1)} \\
&= P(q_\ell^n, q_k^{n-1}|X, M, \upsilon_1) \\
&= P(q_\ell^n|q_k^{n-1}, X, M, \upsilon_1)P(q_k^{n-1}|X, M, \upsilon_1)
\end{aligned} \tag{107}$$

Since the second factor in (107) is a function of k and n only we can set $\lambda_{k,n}$ to $P(q_k^{n-1}|X, M, \upsilon_1)$ which then gives us, according to (106):

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, M, \upsilon_1) \tag{108}$$

This is a valid solution since the constraint

$$\sum_{\ell=1}^{K} P(q_\ell^n|X, q_k^{n-1}, M, \upsilon_1) = 1, \forall k \text{ and } \forall n$$

given in (99) is automatically met.[25] In order to verify that we got a maximum point we have to compute the Hessian matrix. It is easy to see by looking at (105) that all the non diagonal elements are zero. In computing the diagonal elements we get

$$\begin{aligned} &\frac{\partial^2 R^*(\cdot)}{\partial P_{\upsilon_2}^2(q_\ell^n|x_n, q_k^{n-1})} \\ &\quad = -\frac{1}{P(M|X,\upsilon_1)} \sum_{\Gamma_{k,\ell,n}} \left[P(M, \Gamma_{k,\ell,n}|X, \upsilon_1) \frac{1}{P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} \right] \end{aligned} \tag{109}$$

and it is obvious that for probabilities (i.e., positive numbers) we get negative diagonal elements. Thus, we found a maximum point. This proves Theorem 2 which, briefly, says the following. A trained MLP with a fixed set of parameters (MLP weights) Θ provides us with estimates of conditional transition probabilities $P(q_\ell^n|x_n, q_k^{n-1}, \upsilon_1(\Theta))$ (estimated on a given training data set $X = \{x_1, \ldots, x_n, \ldots, x_N\}$) $\forall n = 1, \ldots, N$ and $\forall k, \ell = 1, \ldots, K$. Given these estimates, obtained at the output of the MLP, it is possible to compute re-estimates of the conditional transition probabilities $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta), M)$ by the backward recursion given in (58) to increase the global posterior probability of the correct model M over $P(M|X, \upsilon_1)$.

A.5 Theorem 3

As opposed to the "standard" EM algorithm (Baum *et al.* 1970; Baum 1972), Theorems 1 and 2 are not enough to prove convergence of the training process for two reasons:

1. The MLP training is usually minimizing a function (e.g., least mean square or relative entropy) that is different from the function optimized in Theorem 2. As a consequence, we have to prove convergence through the same auxiliary function $R(\cdot)$.

2. Theorem 2 gives us new ("optimal") values (MLP targets) for the conditional transition probabilities which are going to be used to train the MLP. If the cost function can be

[25] We cannot prove that this is a unique solution since most of the equations are nonlinear but we know this is a least one valid solution.

trained to reach its optimal minimum, the MLP will just "learn" the targets and we will have

$$g_\ell(x_n, q_k^{n-1}, \Theta) = p(q_\ell^n | X, q_k^{n-1}, M, v_1) \tag{110}$$

which, by Theorem 2, is known to increase $R(\cdot)$ and, consequently, $P(M|X)$. In this case, of course, we proved that MLP training is increasing $P(M|X)$ and we do not need anything more. However, in general, the nets will not be trained to their optimal minimum because of

- "overlapping" of input patterns (e.g., two instances of the same pattern with two different classifications).
- use of cross-validation (early stopping) (Bourlard & Morgan 1994) to avoid over fitting and to get better estimates of actual probabilities.

Below we describe a training procedure for the MLP and a corresponding error criterion. We show that by minimizing this criterion we are maximizing the auxiliary function $R(\cdot)$. Thus by Theorem 1 we increase the posterior probability of the correct sentence. By this we show convergence (at least to a local minimum) on the training set.

Specifically, given a trained MLP with a set of weights Θ^t which provides a set of conditional transition probabilities $v_1(\Theta^t)$ and given a sequence of acoustic vectors X and a model M, we can compute (by using the Discriminant HMM backward recursion) a new set of probabilities

$$T \;=\; \{P_T(q_\ell^n | x_n, q_k^{n-1}) = P(q_\ell^n | X, q_k^{n-1}, v_1(\Theta^t), M); \forall k, l = 1, \ldots, K; \forall n = 1, \ldots, N\} \tag{111}$$

which will be used as targets to adapt further the MLP weights to a new set of weights Θ^{t+1} and, consequently, a new set of conditional transition probabilities $v_2(\Theta^{t+1})$.

In the following we prove this property in the case of a weighted relative entropy E_e, similar to a common cost function for MLP training.[26] In this case, given a sequence of acoustic vectors X and a model M and the current set of parameters Θ^t, the parameters Θ^{t+1} of the MLP are trained to minimize

$$E_e(\Theta^{t+1}) = \mathcal{E}_{P(x_n, q_k^{n-1} | M, X, \Theta^t)} \sum_{\ell=1}^{K} P_T(q_\ell^n | x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n | x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \tag{112}$$

where $g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})$ is the ℓ-th output of the MLP given weight set Θ^{t+1} and inputs (x_n, q_k). Note that the expected value $\mathcal{E}_{P(x_n, q_k^{n-1} | M, X, \Theta^t)}$ is taken according to the distribution of the input variables that in the case of the Discriminant HMM are the concatenation of

[26]Relative entropy is a particularly common error criterion for classification and probability estimation tasks, and we have used it for all of the speech training systems that we have developed over the last few years. The new criterion will actually only differ in that the expectation leading to its formulation will be taken with respect to the entire network input space, which includes a choice for the previous state.

the acoustic input and the previous state. In this case (112) can also be expressed as:

$$\begin{aligned}
E_e(\Theta^{t+1}) &= \sum_{x_n, q_k^{n-1}} P(x_n, q_k^{n-1}|M, X, \Theta^t) \sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, x_n, \Theta^t) P(x_n|X, M, \Theta^t) \\
&\qquad \left[\sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \qquad (113)
\end{aligned}$$

Given that $P(q_k^{n-1}|M, X, x_n, \Theta^t) = P(q_k^{n-1}|M, X, \Theta^t)$ and assuming that $P(x_n|X, M, \Theta^t) = \frac{1}{N}$, i.e., sampling the acoustic vector sequence uniformly

$$\begin{aligned}
E_e(\Theta^{t+1}) &= \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, \Theta^t) \\
&\qquad \left[\sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \\
&\qquad \text{replacing } P_T(\cdot) \text{ with its definition } (111) \text{ we get :} \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, \Theta^t) P(q_\ell^n|X, q_k^{n-1}, v_1(\Theta^t), M) \\
&\qquad \left[\sum_{\ell=1}^{K} \log \frac{P(q_\ell^n|X, q_k^{n-1}, v_1(\Theta^t), M)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \qquad (114)
\end{aligned}$$

It is easy to see that the above criterion will reach its global minimum when the outputs of the MLP will be equal to the targets [27]. Note that the relative entropy between two probability mass functions is always greater or equal to zero (Cover & Thomas 1991).Thus, given that the targets are posterior probabilities, a network trained to the global minimum of error criterion (114) will estimate the posteriors.

An important point is that previous state q_k^{n-1} is not part of the features that are extracted from the speech waveform. Thus the scaling factor $P(q_k^{n-1}|X, M, v_1(\Theta^t))$ is needed to compute the expected value over the "extended" input space. There are several ways to implement this scaling, one is to choose the previous state uniformly and to scale the error signal that is back propagated by this factor. An alternative way in stochastic gradient descent training (online training) is to implement this criterion by first choosing the acoustic frame x_n at random from the training test, and then choosing the previous state according to $P(q_k^{n-1}|X, M, v_1(\Theta^t))$.

[27] In the case that both the target probability and the net output are zero, this still holds given $\lim_{\epsilon \to 0} \epsilon \log \frac{\epsilon}{\epsilon} = 0$

Theorem 3

When we minimize the weighted relative entropy criterion (20) with the target set T (which is calculated from a probability set v_1) we maximize the auxiliary function $R(\cdot)$. Specifically the new set of probabilities v_2, implemented by the trained MLP satisfies the following:

$$E_e(\Theta^{t+1}) \leq E_e(\Theta^t) \Longrightarrow R(v_1(\Theta^t), v_2(\Theta^{t+1})) \geq R(v_1(\Theta^t), v_1(\Theta^t)) \tag{115}$$

Proof:

$$\begin{aligned}
E_e(\Theta^{t+1}) - E_e(\Theta^t) &= \\
&= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} P(q_k^{n-1}|X, M, v_1(\Theta^t)) \\
&\quad \sum_{\ell=1}^{K} P(q_\ell^n|X, q_k^{n-1}, M, v_1(\Theta^t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^t)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \\
&= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K} P(q_\ell^n, q_k^{n-1}|X, M, v_1(\Theta^t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^t)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}
\end{aligned} \tag{116}$$

Below we show that the change in the auxiliary function $R(\cdot)$ is with the same magnitude (within a scaling factor that is fixed for an utterance) but with the opposite sign.

$$\begin{aligned}
&R(v_1(\Theta^t), v_2(\Theta^{t+1})) - R(v_1(\Theta^t), v_1(\Theta^t)) = \\
&= \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log\left[\frac{P(M, \Gamma|X, v_2)}{P(M, \Gamma|X, v_1)}\right] \\
&= \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log\left[\frac{P(M|\Gamma, X, v_2)P(\Gamma|X, v_2)}{P(M|\Gamma, X, v_1)P(\Gamma|X, v_1)}\right] \\
&\qquad \text{given a state sequence, the probability of the model does not depend on the transition} \\
&\qquad \text{probabilities as stated in (104)} \\
&= \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log\left[\frac{P(\Gamma|X, v_2}{P(\Gamma|X, v_1)}\right] \\
&\qquad \text{following the same decomposition as in (103)} \\
&= \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log\left[\frac{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_{t+1})}{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_t)}\right] \\
&= \sum_{\Gamma} P(\Gamma|M, X, v_1) \log\left[\frac{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_{t+1})}{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_t)}\right] \\
&\qquad \text{rearranging the terms in the summation} \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{\Gamma_{k,\ell,n}} P(\Gamma_{k,\ell,n}|M, X, v_1) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)}
\end{aligned}$$

where $\Gamma_{k,\ell,n}$ stands for those paths containing the transition (q_k^{n-1}, q_ℓ^n)

$$= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{\Gamma} P(\Gamma, q_k^{n-1}, q_\ell^n | M, X, \upsilon_1(\Theta_t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K} P(q_k^{n-1}, q_\ell^n | M, X, \upsilon_1(\Theta_t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)} \tag{117}$$

A closer look at the last equation shows the term that we got for the difference in the auxiliary function $R(\cdot)$ is with opposite sign and proportional magnitudes (by $\frac{1}{N}$) to the difference in E_e in (116) Thus, minimizing the cost function E_e (as part of the MLP training) is equivalent to maximizing the auxiliary function $R(\cdot)$. Thus, we have proved Theorem 3, and in fact showed that to minimize error criterion (114) is equivalent (within a scaling factor) to maximizing the auxiliary function. In combination with the previous Theorems, this confirms that a network trained using error criterion (114) and targets defined by Theorem 2 will increase the auxiliary function. This in turn means that the global probability of the correct model will be increased.

In practice we compute the change of the error measure on a cross-validation set to guide the training schedule of the MLP, e.g., for deciding the learning rate and the stopping point.

A.6 Summary and Discussion

Like the EM algorithm, REMAP training consists of two major steps: estimation (which in this case is estimating new targets for the MLP), and maximization (which here consists of adapting the MLP weights to maximize performance on the new set of targets). Here we have proved three theorems that together show that each iteration of REMAP training increases the posterior probability of the training sentence. It is assumed that the training set is a good sample of the overall input space, and cross-validation techniques could be used to check that we have not over-fit to the training data, e.g., by computing the change of the posterior probability on an independent set after every iteration of the REMAP algorithm. In principle, REMAP should ultimately provide improved recognition accuracy for practical systems. However, as with all other gradient-based optimization techniques, we will be vulnerable to potential difficulties with local minima. Further, in order to derive the training and recognition algorithms described here, we still have had to make a number of simplifying assumptions. While we do not think that they will be serious limitations, it will take experimentation to learn the practical tradeoffs. More generally, "there's many a slip twixt the cup and the lip."