Maximum Entropy Direct Model as a Unified Model for Acoustic Modeling in Speech Recognition

Hong-Kwang Jeff Kuo, Yuqing Gao
IBM T. J. Watson Research Center, 1101 Kitchawan Road, Route 134, Yorktown Heights, NY 10598. {hkuo,yuqing}@us.ibm.com

Abstract

Traditional statistical models for speech recognition have been dominated by generative models such as Hidden Markov Models (HMMs). We recently proposed a new framework for speech recognition using maximum entropy direct modeling, where the probability of a state or word sequence given an observation sequence is computed directly from the model. In contrast to HMMs, features can be non-independent, asynchronous, and overlapping. In this paper, we discuss how to make the computationally intensive training of such models feasible through parallelizing the IIS (Improved Iterated Scaling) algorithm. The direct model significantly outperforms traditional HMMs in word error rate when used as stand-alone acoustic models. Modest improvements over the best HMM system are seen when combined with HMM and language model scores. The maximum entropy model can potentially incorporate non-independent features such as acoustic phonetic features in a way that is robust to missing features due to mismatch between training and testing.

1. Introduction

Statistical models for speech recognition usually model speech as a sequence of observations, \( O = o_1 o_2 \ldots o_t \) (acoustic features), produced by an unobservable “true” state sequence, \( \hat{S} = \hat{s}_1 \hat{s}_2 \ldots \hat{s}_t \) (sequence of sub-phones, phonemes, or words). The states take on values in a finite state space \( V \). The goal of a speech recognizer is to find the state sequence \( \hat{S} \) with the maximum posterior probability given an observed sequence \( O \):

\[
\hat{S} = \arg\max_s p(S|O)
\]

In past decades, Hidden Markov Models (HMMs) have been studied extensively and have proven to be a very effective modeling technique. When solving Equation 1 with the HMM, \( \hat{S} \) is obtained by maximizing the joint probability [1]:

\[
\hat{S} = \arg\max_S \frac{p(O|S) \cdot p(S)}{p(O)} = \arg\max_s p(O|S) \cdot p(S)
\]

This approach uses Bayes’ rule to compute \( p(S|O) \) through a generative model \( p(O|S) \). The HMM parameters defining \( p(O|S) \) are estimated to maximize the likelihood of the training observations \([1, 10]\).

Direct modeling attempts to model the posterior probability \( P(S|O) \) directly. This approach has been used for statistical natural language understanding [8], for information extraction and segmentation [7], and only recently for acoustic modeling [6, 5]. We have previously shown that the maximum entropy direct model outperforms the HMM as a stand-alone acoustic model [5], possibly due to the discriminative nature of the maximum entropy model. More importantly, the model is a new framework that allows us to flexibly incorporate linguistic and other types of features in a unified model for speech recognition. The maximum entropy model allows asynchronous and overlapping non-independent features to be incorporated formally, unlike the case for HMMs. Thus it will be possible to take advantage of suprasegmental features like prosodic features and a multitude of other features such as acoustic phonetic features [12], linguistic features, etc. We expect features from different levels of linguistic hierarchies (e.g. [11, 3]) to play an important role. The direct model can also be framed as a joint acoustic and language model. However, joint estimation would require a large amount of parallel speech and text data, clearly a challenge for data collection.

Because the computational requirements for training maximum entropy models are quite intensive, in this paper, we discuss how to parallelize the Improved Iterated Scaling (IIS) training algorithm in order to train large models on a large amount of training data, running hundreds of iterations within a reasonable length of time. We present results of how this parallelization allows us to achieve better model convergence on complex model and data. Experiments show that the direct model significantly outperforms the HMM as a stand-alone acoustic model. Modest improvements over the best HMM speech recognizer were observed when the direct model was combined with the HMM. In addition, we discuss how the maximum entropy direct model may be able to robustly combine different features, such as context, suprasegmental, and acoustic phonetic distinctive features, in a unified fashion.

2. Maximum Entropy Direct Models

Please refer to [5] for details about the model, as well as the training and decoding. Here a brief description is included.

An HMM [10] is a discrete stationary process on a state space with a limited horizon: \( p(s_t|s_{t-1}) = p(s_t|s_{t-1}) \). Each state\( s_t \) emits an observable symbol \( o_t \) at time \( t \) according to some probability distribution \( p(o_t|s_t) \). There are no dependencies between \( o_t \) and previous states, given the current state.

In direct models, separate transition and observation probabilities are replaced with one function, \( p(s_t|o_1, o_2, \ldots, o_t, s_{t-1}) \). In this paper, we concentrate mainly on the Maximum Entropy Markov Model (MEMM) [7]:

\[
p(s_t|o_1, o_2, \ldots, o_t, s_{t-1}) = p(s_t|o_t, s_{t-1})
\]

The Maximum Entropy model can be expressed as an exponential model:

\[
p(s_t|\alpha) = \frac{e^{\sum_i (\lambda_i \cdot f_i(s_t, \alpha_i))}}{Z_s(\alpha_t)}
\]

Here the summation is over all features of the model and \( \alpha_t \) denotes the context - everything the probability is conditioned on. In the MEMM, \( \alpha_t = \{o_t, s_{t-1}\} \). \( Z_s(\alpha_t) \) is a normalization factor that depends on the conditioning event \( \alpha_t \).
The acoustic feature vector for every speech frame is quantized into a list of Gaussian IDs ranked in the order of their Gaussian likelihoods $P(o_t|g_t)$, as is done in a rank-based HMM speech recognition system [4]. Observation features are selected based on the IDs of the highest ranking Gaussians at each frame:

$$f_{<g,s>} (o_t, s) = \begin{cases} \alpha > 0 & \text{if } g \in \sigma_t \text{ and } s = s_t \\ 0 & \text{otherwise} \end{cases}$$ (5)

Essentially, the feature $f_{<g,s>}$ checks if $g$ appears as a top ranking Gaussian in the observation vector $\sigma_t$. $\alpha$ is the weight of $g$, which can be chosen to be a function of the rank of $g$. In this paper we tried using the top 10 or 20 Gaussians.

Features for $s \rightarrow s$ transitions are defined as:

$$f_{<g,s>} (o_t, s) = \begin{cases} 1 & \text{if } s = s_{t-1} \text{ and } s = s_t \\ 0 & \text{otherwise} \end{cases}$$

### 2.1. Parallelization of Improved Iterative Scaling

Maximum entropy models are often trained using an iterative algorithm known as Improved Iterative Scaling (IIS): [9]

**Step 0.** For each feature $f$ calculate its empirical expectation $E_f = \frac{\sum_{g \in \sigma_{t \in \mathcal{S}}} f(\sigma_t, g) \cdot \frac{1}{N_{\sigma,g}}}{N_{\sigma,g}}$. Here $N$ is the number of examples in the training set and $N_{\sigma,g}$ is the number of occurrences of the pair $\sigma, g$ in the training set.

**Step 1.** Start with arbitrary values for $\lambda_i$, for example 0.

**Step 2.** For each feature $f_i$ calculate the $\Delta \lambda_i$ given by the following equation [2]:

$$\sum_{g \in \sigma_{t \in \mathcal{S}}} p(\sigma_t | \lambda) \cdot \lambda(\sigma_t, g) \cdot f_i(\sigma_t, g) \cdot e^{\Delta \lambda_i \cdot f_i(\sigma_t, g)} = \hat{E}_{f_i}$$ (6)

Here $f_i(\sigma_t, g) = \sum_{\mathcal{S}} f_i(\sigma_t, g)$. The equation can be solved either with a Newton or Bisection method.

**Step 3.** Update the values of parameters $\lambda_i^{new} = \lambda_i + \Delta \lambda_i$.

**Step 4.** Go to Step 2 and repeat till the convergence.

The idea behind IIS is to speed up convergence by taking different step sizes depending on $f_i^*(\sigma_t, g)$. The equation in Step 2 can be thought of as a polynomial equation in $e^{\lambda_i}$, assuming $f_i^*(\sigma_t, g)$ are integers. If $f_i^*(\sigma_t, g)$ are not integers, because the $\lambda_i$ weights in Equation 5 for the ranked Gaussians are not necessarily integral, discretization is often done, and counts of features are accumulated in buckets (corresponding to the polynomial coefficients) in order to construct the polynomial. Thus each iteration of IIS can be thought of as having two phases: collecting statistics in the buckets and solving the equation in Step 2. The first phase can be partly parallelized by having different CPUs/computers work on different blocks of the training data. Then the buckets are combined to create the polynomial equations, which are solved in order to update the parameters. This is the approach we implemented to be able to more quickly train models with large numbers of parameters on a large amount of training data.

Clearly there are some tradeoffs in choosing the number of parallel processors, since there is overhead in communicating the results. A simple way of communicating the results is through shared memory or filesystem. In our present implementation on a group of Linux servers, each server/processor reads in the model and works on a separate block of training data. The bucket statistics collected by each server/processor are written out to a shared filesystem. Once every block of data has been processed, another process combines the output to compute the coefficients of the polynomial, solves the equation for $\Delta \lambda_i$, and updates the model. A convergence criterion is tested and if further iterations are needed, the cycle repeats itself. Note that the amount of disk space needed grows linearly with the number of parallel processes, the number of buckets, and the number of parameters. Having too many processes may actually slow down the progress because all the processes compete for the available disk I/O bandwidth. In practice, the disk writing activity of the parallel processes may have to be modulated by granting write privileges to only a certain number of processes at any time.

### 3. Experimental Results

The experiments were carried out on the DARPA Communicator data, collected within the domain of travel reservation. There are a total of about 380 hours of speech over various telephone lines, of which 240K utterances (150-200 hours) were used for training the MEMM. In contrast, all the available training speech data were used to train the HMM. The baseline HMM is a state-of-the-art context-dependent model that has a total of about 45K Gaussians. The number of parameters in this HMM is about 3.6M (45K × 80) (40 dimensional LDA rotated vectors: 40 mean and 40 diagonal covariance parameters.) There are a total of 52 phones, each with 3 states, hence a total of 156 context-independent phone states.

For the training data, each speech frame is forced aligned with a corresponding phone state using the HMM. For each speech frame, the acoustic feature vector is also compared with all the Gaussians in the HMM to generate a ranked list of Gaussian IDs. The context-independent phone state sequence and Gaussian IDs are extracted as features used to train the MEMM.

We performed re-scoring experiments to compare the performance of the HMM and the new MEMMs for word recognition. An initial N-best list is generated using an HMM decoder, with 300 best hypotheses for each sentence. The test set consists of 1173 sentences from a NIST evaluation. Each of the N-best hypotheses for each sentence can be converted deterministically into phone sequences, then state sequences, with optional silence states between words. The speech is forced aligned to the state sequence using the MEMM acoustic model to compute the probability of the best path for that hypothesis. Each hypothesis is assigned the log probability as its acoustic score, and the hypothesis with the best score is taken as the recognized string.

![Figure 1: Word error rate (WER) of MEMM acoustic models without incorporating any language model. In comparison, the WER for a state-of-the-art HMM acoustic model is 32.7%.

Figure 1 shows the word error rate (WER) of different...](image-url)
acoustic models when used to re-score the N-best hypotheses without incorporating the language model probabilities. The baseline HMM, which has approximately 3.6M parameters, yielded a WER of 32.7%, represented by a triangle in the figure. The solid curve shows the WER results for MEMMs (using top 10 Gaussian features) recently presented at the ASRU2003 workshop [5] in December 2003, whereas the dashed curve shows new results from a significantly larger number of training iterations, made possible by the more efficient parallelized implementation of the IIS algorithm. As an example, previously a model may be trained for only 30 iterations, whereas the new model may now be trained for over 200 iterations. Using the top 10 Gaussians as observation features, an MEMM with 1.8M parameters previously achieves a WER of 28.6%, a relative improvement from the baseline HMM of 12.5%. The new MEMM achieves a WER of 27.1%, a relative improvement of 6% over the old MEMM and 17% over the HMM. This result indicates that the MEMM is superior to the HMM as a stand-alone acoustic model.

The new implementation of the IIS training algorithm parallelizes the computation across different CPUs/computers, reducing both the processing time and memory usage within each computer. Table 1 shows an example, with 150 hours of speech for training a 1M-parameter MEMM, using one computer, about 1.6GB memory is used (1.6GB for training data plus 200MB for storing intermediate statistics assuming 100 buckets per parameter). Each iteration takes about 2 hour 15 minutes on a 1GHz Pentium machine, for a total of 20 days for 200 iterations. In contrast, the new parallelized implementation spreads the computation among 12 computers, for example, and reduces the time per iteration to 40 minutes, for a total of about 6 days for 200 iterations. The memory used by each computer is about 333MB (133MB(1.6GB/12) for training data plus 200MB for intermediate statistics). Using more computers can further reduce the computation time, but at some point, it becomes I/O (input/output) bound, i.e. the bottleneck is not the computation of each processor, but the communication between processors. Where the communication is through a shared filesystem, the amount of temporary disk space required also increases with the number of parallel processes; for example, with 12 processes, each having 200MB of intermediate statistics, a total of 2.4GB disk space is needed on the shared filesystem. Nevertheless, parallelizing the computation is quite effective to achieve convergence in a reasonable amount of time, and the results above show that it is important to achieve convergence (relative difference of about 6% in WER).

Using the top 20 ranked Gaussians, instead of just 10, as possible features, there is an absolute improvement of 1.5–2.0% in the WER. Due to memory limitations in the previous implementation, we did not train such models with over 500K parameters [5]; these results are now presented in Figure 2. Using a longer state context history as features, the performance also improved, as shown by the square data points in Figure 2. The performance seems to saturate above about 1 million parameters, possibly due to over-training and the lack of training data.

Given that the MEMM outperforms HMM as a stand-alone acoustic model, we want to investigate how the MEMM would perform when combined with a language model and with the HMM. Strictly speaking, it does not make sense to combine the MEMM posterior probability with the language model probability. However, experiments show that good performance can be achieved using a heuristic approach proposed previously [5] of normalizing the dynamic range of the MEMM scores to fit the range of the HMM scores within the N-best hypotheses for each sentence. Figure 3 shows the results of combining HMM with MEMM, when combined with language model log probabilities. The WER of a state-of-the-art HMM recognizer for this task is 16.9%. The language model is a combination of n-gram, class n-gram, concept n-gram, and semantic structured language models [3]. With combined HMM and MEMM scores, the WER improves, by as much as 9% (from 16.9% to 15.4%). This result is slightly better than that reported recently at the ASRU2003 workshop [5].
4. Discussion

The maximum entropy direct model is a posterior probability model (c.f. Equation 4) that is conditioned on the observed features. Therefore, during decoding, the model is potentially robust to features being absent due to mismatch conditions. As a simple example, Figure 4 shows the effects of a mismatch between training and testing. A model that is trained on features based on the top 20 Gaussians at each frame is tested on test data where only the top 10 Gaussians are present at each frame. Figure 4 shows that there is little to no performance degradation due to this mismatch, i.e. the performance is almost as good as the model that is trained on the top 10 Gaussians.

Importantly, such robustness of the Maximum Entropy model to the absence of features can lead to a different paradigm of incorporating a multitude of feature detectors. For example, it has been difficult to incorporate distinctive features into the statistical speech recognition framework. With this new maximum entropy direct model, such features can be more easily built on top of our current model (which is based on conventional mel cepstra features). We can expect the combined model to be no worse than our current model, which we have shown is already superior to the HMM as an acoustic model. Each feature detector can abstain when it is unsure about whether the feature is present. In the extreme case, when none of the feature detectors fire, the model will do as well as our current model. Consider an example of how different features can be useful for robustness. The spectrum of the burst after the release of a stop consonant contains information about the size of the cavity anterior to the constriction and thus the place of articulation (whether it is a /pl/, /tl/, or /k/). Such information can be used advantageously in clean speech (quiet conditions). However, in noisy conditions, such information is not reliable and should be discarded. Instead, the formant transition, which is more robust to noise, can be used. It may be possible to use the maximum entropy direct model as a unifying framework to incorporate such features and knowledge to build a speech recognizer that is more robust to varying conditions.

Many other open research issues remain, including how to formulate the direct model for variable length units rather than at the frame level, how to take advantage of overlapping features, and how to incorporate the language model into the direct model to achieve a joint acoustic and language model. It is fair to say that we do not yet fully understand this model (in contrast to the HMM, which has been studied for many years), and thus more investigation by the research community is desirable.

5. Conclusions

For over two decades, statistical models for speech recognition have been dominated by the hidden Markov model, which is formulated as a generative model of speech. Recently, we have proposed a maximum entropy direct model that directly models the posterior probability [5]. Even with very simple features such as ranked Gaussians and the prior state, this model has performed surprisingly well compared to the HMM, despite being trained on much less training data and having only the top ten ranked Gaussians as features. Training the maximum entropy direct model requires significant computational and memory requirements, and we have addressed how to make it feasible to train models with a large number of parameters on a large amount of data. The direct modeling approach is a drastic departure from generative models such as HMMs and offers interesting research opportunities. We believe that it holds even greater promise when additional features such as phone, word, parse tree contexts, semantic coherence, and other suprasegmental features are incorporated. We are also excited about the prospects of using this model to incorporate discrete features such as acoustic phonetic distinctive features in a unified framework for speech recognition.

6. References