A Constrained Baum-Welch Algorithm for Improved Phoneme Segmentation and Efficient Training

David Huggins-Daines, Alexander I. Rudnicky

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
{dhuggins,air}@cs.cmu.edu

Abstract

We describe an extension to the Baum-Welch algorithm for training Hidden Markov Models that uses explicit phoneme segmentation to constrain the forward and backward lattice. The HMMs trained with this algorithm can be shown to improve the accuracy of automatic phoneme segmentation. In addition, this algorithm is significantly more computationally efficient than the full Baum-Welch algorithm, while producing models that achieve equivalent accuracy on a standard phoneme recognition task.

Index Terms: speech segmentation, speech synthesis, phoneme recognition, hidden markov models

1. Introduction

Hidden Markov Models have been shown useful for both segmentation and recognition on word and phoneme tasks. However, for phoneme segmentation, such as used in building speech databases for unit-selection speech synthesis[1], the labels (phone segmentations) produced using the standard HMM-based alignment tools for speech recognition typically require extensive hand-correction[2].

As well, while they may be internally consistent, the automatically generated labels do not always reflect human labelers' ideas of the boundaries between phonemes. This is particularly noticeable when context-dependent triphone models are used for automatic segmentation; past research[3] has indicated that context-independent models are preferable for segmentation tasks (though [4] presents a novel way of compensating for the effects of context-dependency). We hypothesize that, given that the state sequence, and thus by extension the phoneme segmentation, is considered to be a hidden variable in training, the Viterbi segmentations generated by HMM-based tools reflect boundaries that are optimal only in a maximum-likelihood sense.

This raises the question of whether it is possible to use linguistic knowledge, available in the form of manually generated phoneme labels, to improve the quality of the models, and in doing so improve both phoneme segmentation and recognition. In this paper, we describe an algorithm for doing so, and provide results that show a marked improvement on an objective measure of segmentation accuracy, as well as a significant reduction in computational complexity.

2. Algorithm

In the standard EM algorithm for training HMMs[5], the state sequence required to calculate the likelihood function is considered to be *missing* or *hidden* data. Thus, in order to obtain a maximum likelihood estimate λ of the model parameters, we must calculate the conditional expectation of the likelihood given a current set of parameters λ'. This is the objective function $Q(\lambda, \lambda')$ which is then maximized in successive iterations (where Q is the set of state sequences and O is the observation sequence):

$$Q(\lambda, \lambda') = \sum_{q \in Q} log P(O, q|\lambda) P(O, q|\lambda')$$

Calculation of this expectation involves summation over all possible state sequences Q, which can be achieved in quadratic time[1] using dynamic programming techniques, namely the forward and backward algorithms. To speed computation, implementations of Baum-Welch often constrain the set of state sequences using a beam search algorithm. This is accomplished by maintaining a list of active states at each timepoint and storing only those entries in the forward or backward lattice that are non-zero or above the pruning threshold determined by the beam.

In our modified Baum-Welch algorithm, we introduce a phone segmentation which constrains the set of possible state sequences to those that pass through the given phoneme sequence. This can be represented as an extra variable S in the formulation of the objective function $Q(\lambda, \lambda')$. We consider S to be known *a priori*, though it is concievable that it could be estimated jointly with the other parameters, as in the *hidden model sequence* approach to pronunciation modeling[6]. The objective function then becomes:

$$Q(\lambda, \lambda') = \sum_{q \in Q} log P(O, q|\lambda, S) P(O, q|\lambda', S)$$

Assuming independence between the parameters λ and the phone sequence S, in addition to the standard HMM conditional independence assumptions, the joint likelihood of the data and a state sequence q becomes:

$$\begin{aligned} P(O, q|\lambda, S) &= P(O|q, \lambda, S) P(q|\lambda, S) \\ &= P(O|q, \lambda, S) P(q|\lambda) P(q|S) \\ &= \prod_{t=1}^{T} P(o_t|q_t) P(q_t|q_{t-1}) P(q_t|s_t) \end{aligned}$$

In our experiments, the probability $P(q_t|s_t)$ is fixed at zero or one depending on whether the state q_t belongs to the phone s_t, though it is also conceivable that these probabilities could be

[1]In practice, it often requires cubic time, as noted below.

estimated independently, or that some other function indicating the correspondence between phones and states could be used. There are, however, distinct computational advantages to using a zero-one function in the current case.

For an arbitrary topology, the forward algorithm has a complexity of $O(|Q|^2T)$ in time and $O(|Q|T)$ in memory. In practice, for training speech recognition systems using a left-to-right topology, the worst-case complexities are $O(T^3)$ and $O(T^2)$, respectively, since the number of states in the sentence HMM grows linearly with the length of the observation, and the number of active states grows linearly with the number of timepoints previously evaluated. However, when a phoneme sequence is used to constrain evaluation, only those states belonging to the current phone can ever be active. The number of active states at each timepoint is thus constant in the expected number of states in the current phone $E[|P_t|]$, and the worst-case time complexity is $O(E^2[|P_t|]T) = O(T)$.

3. Experimental Results

We implemented this technique in the trainer for the SPHINX-III continuous speech recognition system[7], and tested it for both phoneme segmentation and phoneme decoding on several corpora of phonetically labelled speech data. The **TIMIT** dataset[8] consists of 6300 phonetically balanced sentences from 630 speakers, and contains officially designated training and test sets comprising 3.13 and 1.14 hours of data, respectively. The **F2B** dataset consists of one female speaker from the Boston University Radio speech corpus[9], for whom a full set of phonetic labels is available. It consists of 111 news items, for a total of 0.93 hours of audio. The **SWB** dataset is the subset of Switchboard conversational telephone speech data phonetically transcribed at ICSI as part of the Switchboard Transcription Project[10]. It contains 1287 sentences and 0.81 hours of audio. Finally, the **kaltext4** dataset is a set of 534 phonetically balanced sentences (0.54 hours) from a single speaker collected at Cepstral LLC for use in a unit-selection speech synthesis engine.

We performed some normalization on the phonesets used in the various datasets to make them consistent with each other, as well as to make the resulting phoneset as close to the one used in training our standard acoustic models. This involved removing a fair amount of phonetic detail in some cases, such that the resulting phoneset represents a phonemic rather than a phonetic level of transcription.

For the datasets without a designated training and test set, we held out 10% of the utterances for testing. From the remaining data, we trained acoustic models using both standard Baum-Welch and phoneme-constrained Baum-Welch. Phoneme segmentation was done using the `sphinx3_align` tool for Viterbi alignment. Phoneme decoding used the `sphinx3_allphone` decoder with a trigram model for phone transition probabilities trained from the testing portion of the dataset. For each dataset, we trained context-independent (CI) phone models using a three-state left-to-right topology. We also trained context-dependent (CD) triphone models using 2000 tied states. Except where noted, we used 16 Gaussian mixture components for the output distributions for CI models and 8 Gaussians for CD models.

The results of phoneme segmentation, shown in Table 1, were evaluated by calculating the RMS error in milliseconds of the predicted versus the reference labels. The phone sequences used as input to segmentation were taken from the reference labels, with the timing information left out, thus ensuring a one-to-one alignment between the reference and hypothesis labels. For phoneme recognition, we used phoneme error rate, as calculated using the standard dynamic programming alignment algorithm. These results are shown in Table 2.

Dataset	Model	Algorithm	RMSE (ms)
TIMIT	CD	Baum-Welch	31.035
TIMIT	CD	constrained	20.715
TIMIT	CI	Baum-Welch	29.258
TIMIT	CI	constrained	20.260
SWB	CD	Baum-Welch	107.730
SWB	CD	constrained	84.160
SWB	CI	Baum-Welch	43.252
SWB	CI	constrained	30.873
F2B	CD (2 Gau)	Baum-Welch	28.733
F2B	CD (2 Gau)	constrained	22.685
F2B	CI (2 Gau)	Baum-Welch	28.168
F2B	CI (2 Gau)	constrained	19.679
kaltext4	CD (2 Gau)	Baum-Welch	33.870
kaltext4	CD (2 Gau)	constrained	29.891
kaltext4	CI	Baum-Welch	25.873
kaltext4	CI	constrained	17.998

Table 1: Phoneme segmentation results

Dataset	Model	Algorithm	PER (%)
TIMIT	CD	Baum-Welch	31.64
TIMIT	CD	constrained	34.13
TIMIT	CI	Baum-Welch	36.72
TIMIT	CI	constrained	41.44
SWB	CD	Baum-Welch	54.49
SWB	CD	constrained	55.18
SWB	CI	Baum-Welch	53.35
SWB	CI	constrained	53.94
F2B	CD	Baum-Welch	22.87
F2B	CD	constrained	26.22
F2B	CI	Baum-Welch	26.38
F2B	CI	constrained	28.99
kaltext4	CD	Baum-Welch	30.67
kaltext4	CD	constrained	31.81
kaltext4	CI	Baum-Welch	31.03
kaltext4	CI	constrained	35.16

Table 2: Phoneme recognition results

The amount of CPU time used in training one iteration of TIMIT context-independent phone models is shown in Figure 1. Training was done on a 1.6GHz AMD Sempron64 based workstation with 1GB of RAM, running Linux 2.6.15. The number for Baum-Welch with one Gaussian is an average over five iterations of initial training - since we used a flat initialization of the model parameters, the beam remains extremely wide until the parameters begin to asymptote.

For comparison purposes, we also applied several existing phoneme-segmentation algorithms to the **kaltext4** dataset, and compared them with the segmentations obtained by "cross-labeling" this dataset using the **TIMIT** models. The algorithms we compared are the `dtw`, `sphinxtrain` (SPHINX-II semi-continuous HMM), and `ehmm` (continuous HMM)[11] meth-

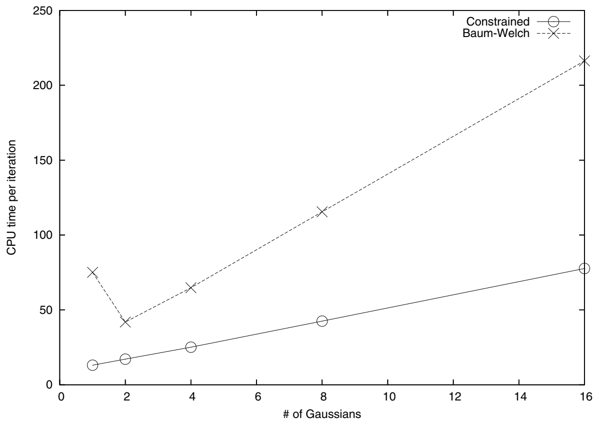

Figure 1: CPU time required to train TIMIT CI models

ods implemented in the FestVox toolkit for building synthesis voices[12]. We also trained `kaltext4` models, using only the word transcriptions of the dataset in the same manner as the `sphinxtrain` method. The results of this comparison are shown in Table 3.

For this experiment, it was necessary to perform a dynamic programming alignment and calculate the RMS error over only the aligned segments. This is because, in real-world phoneme segmentation tasks, phonemes and silences in particular may be inserted, deleted, or substituted by human labelers to match what was actually spoken. For the input to our segmentation algorithm, we used the phone sequence predicted by the Festival text-to-speech engine, which does not always match the reference labels. Of these labeling algorithms, only `ehmm` is able to insert and delete silences with any degree of accuracy, and therefore its slightly higher error rate may be misleading. We plan to incorporate more intelligent silence handling into the SPHINX-III aligner in the near future. It is worth noting that SPHINX-III using context-independent models also ran much faster than any of the comparison methods.

Method	# Alignments	RMSE (ms)
`TIMIT` (CD, Baum-Welch)	14869	190.167
`dtw`	14869	75.499
`sphinxtrain`	14770	38.660
`kaltext4` (CD, 2 Gau)	14869	35.468
`kaltext4` (CI, 4 Gau)	14869	34.735
`ehmm`	14902	32.244
`TIMIT` (CD, constrained)	14869	32.222
`TIMIT` (CI, constrained)	14869	30.419
`TIMIT` (CI, Baum-Welch)	14869	30.263

Table 3: Cross-labeling results on **kaltext4**

Though we expected that the unconstrained context-dependent models would perform worse than the constrained ones in the "cross-labeling" task, the magnitude of the difference is surprising. It appears that there are a large number of catastrophic labeling errors in the alignment for these models, where one phone extends over a long sequence of heterogeneous speech data, as shown in Figure 2. These are the same type of errors that show up in the speaker-dependent, context-independent, and constrained models when too many Gaussian mixtures are trained, indicating that this is most likely a parameter estimation issue.

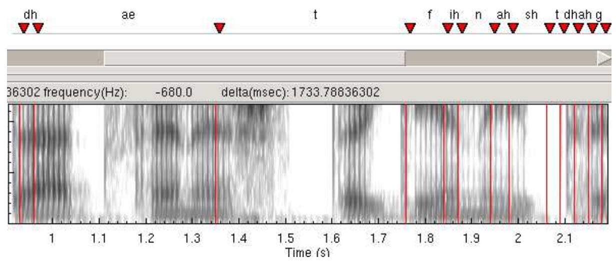

Figure 2: Bad alignments using CD TIMIT models

4. Discussion

The phoneme-constrained training algorithm consistently improved phoneme labeling in our experiments, particularly when the training and test set are matched. It also reduced the "gap" in performance between context-dependent and context-independent models for labeling, though the context-independent models still perform better. This does not necessarily imply that context-dependent models are not useful; it may be that they are simply overfitted to the data, given the small size of the datasets used. The results on the "cross-labeling" task seem to confirm this.

Initially, it appears that using phoneme-constrained training degrades performance for phoneme recognition tasks. There are two possible explanations for this; either the human-labeled phoneme boundaries are suboptimal for training models for recognition, or the training process is overly constrained, leading to inferior estimates of the model parameters.

To test this, we ran a "bootstrapping" experiment, where initial context-independent models with a single Gaussian per state were trained using unconstrained Baum-Welch, and the resulting Viterbi phoneme segmentations were then used to constrain the context-dependent and multiple-Gaussian training. The results of this experiment are shown in Table 4. In addition to TIMIT, we also applied this technique to the Wall Street Journal connected-word dictation task, using the phone segmentations produced through forced recognition using the initial context-independent models and a multiple-pronunciation dictionary.

Dataset	Model	PER/WER (%)
TIMIT	Baum-Welch	31.64
TIMIT	bootstrap	31.25
TIMIT	constrained	34.13
WSJ0 (devel5k)	Baum-Welch	8.64
WSJ0 (devel5k)	bootstrap	8.98
WSJ0 (test5k)	Baum-Welch	11.11
WSJ0 (test5k)	bootstrap	11.97

Table 4: Bootstrapping constrained context-dependent models

The more or less comparable performance here seems to indicate that it is the human-labeled phoneme boundaries, rather than any data sparsity problem, which lead to poorer recognition performance when phoneme-constrained training is used. However, since the initial unconstrained models are used to initialize the context-dependent training, this may be providing some additional robustness in parameter estimation. Also, forward evaluation more frequently fails to reach the final state in phoneme-constrained Baum-Welch, thus reducing the effective amount of training data. This is a particularly serious problem for datasets such as **F2B** which contain very long utterances.

5. Conclusion

The phoneme-constrained training technique appears to improve the performance of phoneme segmentation, particularly in the case where the training and test data are matched. On the other hand, it does not improve the accuracy of phoneme or connected word recognition, though with the bootstrap method, it does not severely degrade them either.

In another apparent duality, it is evident that different training techniques for HMMs are required for the segmentation vs. recognition tasks. Context-dependent models and state tying do not appear to be effective for phonetic segmentation. Using phoneme-constrained training mitigates their negative effects somewhat, though context-independent models are still superior, both in accuracy and efficiency of training and segmentation.

In future work, we plan to investigate refinements to the bootstrap training technique. By dumping Viterbi alignments automatically from the Baum-Welch estimation program, it should be possible to eliminate several steps of training. We also intend to investigate the evolution of the optimal phone boundaries across iterations of training, to determine if there is some point at which they can be fixed in place without degrading the recognition accuracy of the resulting models.

6. Acknowledgements

This research was supported by DARPA grant NG CH-D-03-0010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred. We would also like to thank Cepstral LLC for the use of the **kaltext4** dataset and Kevin Lenzo for recording it, as well as John Kominek, Alan Black, and Laura Tomokiyo for discussions on the problem of phonetic segmentation for speech synthesis.

7. References

[1] John Kominek and Alan Black, "CMU ARCTIC databases for speech synthesis," Tech. Rep. CMU-LTI-03-177, CMU Language Technolgies Institute, 2003.

[2] John Kominek, Christina Bennett, and Alan Black, "Evaluating and correcting phoneme segmentation for unit selection synthesis," in *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.

[3] John Kominek and Alan Black, "A family-of-models approach to HMM-based segmentation for unit selection speech synthesis," in *Proceedings of ICSLP 2004*, Jeju, Korea, 2004.

[4] Doroteo Torre Toledano, Luis A. Hernández Gómez, and Luis Villarrubia Grande, "Automatic phonetic segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, November 2003.

[5] Jeff Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov Models," Tech. Rep., International Computer Science Institute, 1997, ICSI-TR 97-021.

[6] T. Hain, *Hidden Model Sequence Models for Automatic Speech Recognition*, Ph.D. thesis, Cambridge University, 2001.

[7] Michael Seltzer and Rita Singh, *Instructions for using the Sphinx3 trainer*, http://www.speech.cs.cmu.edu/sphinxman/fr4.html.

[8] John S. Garofalo, Lori F. Lamel, William M. Fisher, Johnathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM*, Linguistic Data Consortium, 1993.

[9] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University Radio News Corpus," Tech. Rep. ECS-95-001, Boston University, March 1995.

[10] S. Greenberg, "Insights into spoken language gleaned from phonetic transcription of the switchboard corpus," in *Proceedings of ICSLP 1996*, Philadelphia, PA, 1996, vol. supplement.

[11] Kishore Prahallad, Alan Black, and Ravishankar Mosur, "Sub-phonetic modeling for capturing pronunciation variation in conversational speech synthesis," in *Proceedings of ICASSP 2006*, Toulouse, France, 2006.

[12] Alan Black and Kevin Lenzo, *Building Voices in the Festival Speech Synthesis System*, http://www.festvox.org/festvox/index.html.