Improved HMM/SVM Methods for Automatic Phoneme Segmentation

Jen-Wei Kuo, Hung-Yi Lo, and Hsin-Min Wang

Institute of Information Science, Academia Sinica, Taipei, Taiwan

{rogerkuo, hungyi, whm}@iis.sinica.edu.tw

Abstract

This paper presents improved HMM/SVM methods for a two-stage phoneme segmentation framework, which tries to imitate the human phoneme segmentation process. The first stage performs hidden Markov model (HMM) forced alignment according to the minimum boundary error (MBE) criterion. The objective is to align a phoneme sequence of a speech utterance with its acoustic signal counterpart based on MBE-trained HMMs and explicit phoneme duration models. The second stage uses the support vector machine (SVM) method to refine the hypothesized phoneme boundaries derived by HMM-based forced alignment. The efficacy of the proposed framework has been validated on two speech databases: the TIMIT English database and the MATBN Mandarin Chinese database.

Index Terms: automatic phoneme segmentation, duration model, HMM, minimum boundary error, support vector machine

1. Introduction

The development of speech technology relies heavily on corpus-based methodologies in which phoneme transcription and segmentation usually play indispensable roles. For example, the development of a text-to-speech (TTS) system, requires a precisely segmented speech database for training data-driven prosodic models. However, manual segmentation of speech signals is extremely time consuming and costly. To reduce the human effort and speed up the labeling process, several approaches try to utilize automatic phoneme segmentation techniques to provide initial phoneme segmentation for subsequent manual segmentation and verification. Such methods include dynamic time warping (DTW) [1], methods that utilize specific features and algorithms [2], HMM-based Viterbi forced alignment [3, 4], and two-stage approaches [5, 6]. HMM-based segmentation is the most widely used, while the support vector machine (SVM) method with some discriminative features is useful for post-correction [6].

This paper presents a HMM/SVM-based two-stage framework for phoneme segmentation. The first stage performs HMM-based forced alignment according to the minimum boundary error (MBE) criterion. The objective is to align a phoneme sequence of a speech utterance with its acoustic signal counterpart based on MBE-trained HMMs and explicit phoneme duration models. The second stage uses SVM to refine the hypothesized phoneme boundaries derived by HMM-based forced alignment, based on some discriminative features and MFCCs. The proposed framework has been extensively evaluated on the TIMIT English database, and applied to semi-automatic phonemic labeling of speech utterances selected from the MATBN Mandarin Chinese database [7].

2. HMM-based phoneme segmentation

2.1. Minimum Boundary Error (MBE) training

Given a training set of observation sequences $\{\mathcal{O}^1, \ldots, \mathcal{O}^R\}$, the MBE criterion for acoustic model training tries to minimize the expected boundary errors in the sequences as follows:

$$\mathcal{F}_{MBE} = \sum_{r=1}^{R} \sum_{S_i^r \in \mathbf{\Phi}^r} P(S_i^r|\mathcal{O}^r) ER(S_i^r, S_c^r), \tag{1}$$

where $\mathbf{\Phi}^r$ is a set of possible phoneme alignments for the training observation utterance $\mathcal{O}^r$; S_i^r is one of the hypothesized alignments in $\mathbf{\Phi}^r$; S_c^r is the manually labeled phoneme alignment; $P(S_i^r|\mathcal{O}^r)$ is the posterior probability of alignment S_i^r, given the training observation sequence $\mathcal{O}^r$; and $ER(S_i^r, S_c^r)$ denotes the "boundary error" of S_i^r compared with the manually labeled phoneme alignment S_c^r. For each training observation sequence $\mathcal{O}^r$, $\mathcal{F}_{MBE}$ gives the weighted average boundary error of all hypothesized alignments. However, Eq. (1) cannot be used directly because, in practice, $P(S_i^r|\mathcal{O}^r)$ is unknown. For simplicity, the plug-in approximation of the posterior probability is used. We assume the prior probability of alignment S_i^r is uniformly distributed, and the likelihood $p(\mathcal{O}^r|S_i^r)$ of alignment S_i^r is governed by the acoustic model parameter set Λ. Therefore, Eq. (1) can be rewritten as:

$$\mathcal{F}_{MBE} = \sum_{r=1}^{R} \sum_{S_i^r \in \mathbf{\Phi}^r} \frac{p_\Lambda(\mathcal{O}^r|S_i^r)^\xi}{\sum_{S_k^r \in \mathbf{\Phi}^r} p_\Lambda(\mathcal{O}^r|S_k^r)^\xi} ER(S_i^r, S_c^r), \tag{2}$$

where ξ is a scaling factor that prevents the denominator $\sum_{S_k^r \in \mathbf{\Phi}^r} p_\Lambda(\mathcal{O}^r|S_k^r)$ from being dominated by only a few alignments. If ξ is set to zero, all the hypotheses are equally weighted. The boundary error $ER(S_i^r, S_c^r)$ of the hypothesized alignment S_i^r is calculated as the sum of the boundary errors of the individual phonemes in S_i^r, i.e., $ER(S_i^r, S_c^r) = \sum_{n=1}^{N^r} er(q_n^i, q_n^c)$, where N^r is the number of total phonemes in $\mathcal{O}^r$; q_n^i and q_n^c are the n-th phonemes in S_i^r and S_c^r, respectively; and $er(q_n^i, q_n^c)$ is the phoneme boundary error calculated as $\frac{1}{2} \times (|s_n^i - s_n^c| + |e_n^i - e_n^c|)$, where s_n^i and e_n^i are, respectively, the hypothesized start time and end time of phoneme q_n^i; and s_n^c and e_n^c correspond to the human-labeled start time and end time, respectively. Since $\mathbf{\Phi}^r$ contains a huge number of hypothesized phoneme alignments, for efficiency, we restrict the hypothesized space $\mathbf{\Phi}^r$ to the set of alignments constructed from a phoneme lattice like the example shown in Fig. 1. To minimize Eq. (2), we adopt the extended Baum-Welch (EB) algorithm for optimization.

2.2. MBE segmentation

The MBE alignment approach is a promising realization of the minimum *Bayes-Risk* classifier for the automatic phoneme

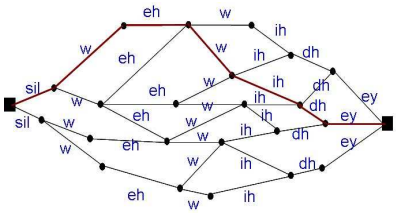

Figure 1: An illustration of the phoneme lattice for the speech utterance "Where were they?"

segmentation task. The latter can be considered as an action, $\alpha_S(\mathcal{O})$, taken to identify a certain alignment, S, from all the phoneme alignments of a given utterance $\mathcal{O}$. Let the function $L(S, S_c)$ be the loss incurred when the action $\alpha_S(\mathcal{O})$ is taken, given that the true alignment is S_c. During the classification stage, we do not know the true alignment in advance, i.e., any arbitrary alignment S_j could be true. The MBR classifier is designed to select the action whose conditional risk, $\mathcal{R}(\alpha_S|\mathcal{O}) = \sum_{S_j \in \mathbf{\Phi}} L(S, S_j) P(S_j|\mathcal{O})$, is minimal, i.e., the best alignment based on the MBR criterion is found by:

$$S^* = \arg\min_S \sum_{S_j \in \mathbf{\Phi}} L(S, S_j) P(S_j|\mathcal{O}). \quad (3)$$

By replacing the loss function in Eq. (3) with the boundary error function defined in Sec 2.1, the best alignment based on the MBE criterion is found by:

$$\begin{aligned} S^* &= \arg\min_S \sum_{S_j \in \mathbf{\Phi}} ER(S, S_j) P(S_j|\mathcal{O}) \\ &= \arg\min_S \sum_{S_j \in \mathbf{\Phi}} \sum_{n=1}^{N} er(q_n, q_n^j) P(S_j|\mathcal{O}), \end{aligned} \quad (4)$$

where N is the number of phonemes in utterance $\mathcal{O}$; and q_n and q_n^j are the n-th phonemes in the alignments S and S_j, respectively. To simplify the implementation, we restrict the hypothesized space $\mathbf{\Phi}$ to the set of alignments constructed from the phoneme lattice, which can be generated by a conventional beam search.

Let the ***cut*** $\mathbf{C_n}$ be the set of phoneme arcs of the n-th phoneme in the utterance. For example, in Fig. 1, there are four phoneme arcs for the second phoneme "w" in $\mathbf{C_2}$ and six phoneme arcs for the third phoneme "eh" in $\mathbf{C_3}$. From the figure, it is obvious that each hypothesized alignment will pass a single phoneme arc in each ***cut*** $\mathbf{C_n}$, $n = 1, 2, \ldots, N$. Based on this observation, Eq. (4) can be rewritten as:

$$S^* = \arg\min_S \sum_{n=1}^{N} \sum_{q_{n,m} \in \mathbf{C_n}} \rho_{q_{n,m}} er(q_n, q_{n,m}), \quad (5)$$

where $q_{n,m}$ is the m-th phoneme arc in $\mathbf{C_n}$, and $\rho_{q_{n,m}} = \sum_{\{S_j \in \mathbf{\Phi} | q_{n,m} \in S_j\}} P(S_j|\mathcal{O})$ is equivalent to the probability of $q_{n,m}$ given the utterance $\mathcal{O}$, which can be calculated easily by applying a forward-backward algorithm to the phoneme lattice. In this way, MBE forced alignment can be performed efficiently on the phoneme lattice via a Viterbi search.

2.3. Applying the phoneme duration model to segmentation

Duration information plays an important role in discriminating between certain words in various languages. For example, in English, it is not easy to distinguish between "ship" and "sheep" without using duration information. However, HMM-based systems, in which the probability of the duration of a state decreases exponentially over time, are known to be deficient in modeling the duration of phonemes. Many duration modeling techniques, such as the hidden semi-Markov model (HSMM) [8], the expanded state HMM (ESHMM) [9], and the post-processor duration model [10], have been proposed to model the duration more accurately. Compared to HSMM and ESHMM, the post-processor duration model used in re-scoring a list of likely hypotheses is more suitable for integration into the MBE segmentation process.

In the implementation, the phoneme duration probability is integrated into the output likelihood of a specific phoneme arc q on the lattice as follows:

$$\hat{p}(q) = p(q) \cdot d(\tau_q)^{\beta}, \quad (6)$$

where $d(\tau_q)$ is the duration probability of q, and β is a scaling factor that controls the duration model's impact. We use a nonparametric probability mass function for duration modeling, which makes no prior assumption about the parametric form of the distribution, and is more computationally efficient than parametric approaches. Phoneme durations from the training data are used to compute the histograms with a bin width of 5 ms.

3. Boundary refinement using SVM

As noted in [6], SVM is useful for refining the initial phoneme boundaries detected by HMM-based segmentation. For each initial boundary, several hypothesized boundaries around it are identified, and each one is examined by a phoneme-transition-dependent SVM classifier; then, the initial boundary is replaced by the most likely boundary.

3.1. Phoneme transition clustering

Ideally, we should be able to train an SVM classifier for each type of phoneme transition. However, this is not feasible because the training data is always limited. Maintaining a balance between the available training data and the model's complexity is critical to the training process. Furthermore, since many phoneme transitions have similar acoustic characteristics, we can partition them into clusters so that the training data can be shared and the phoneme transitions with little training data can be covered by the SVM classifiers of the categories they belong to. We implement phoneme transition clustering in two ways: by K-means clustering and by decision-tree-based clustering.

K-means-based phoneme transition clustering is performed as follows. For each type of phoneme transition, we gather all the feature vectors associated with the human-labeled phoneme boundaries and compute the mean vector. For each one of the four phoneme transition classes, namely *sonorant to non-sonorant*, *sonorant to sonorant*, *non-sonorant to non-sonorant*, and *non-sonorant to sonorant*, we apply the K-means algorithm to cluster the phoneme transitions according to their mean vectors. Note that only phoneme transitions with enough instances are considered in this step. Finally, we assign the phoneme transitions ignored during clustering (due to sparse instances) to the nearest clusters according to the Euclidean distances between their mean vectors and the cluster centers.

The drawback of K-means clustering is that it can not cover phoneme transitions that do not occur in the training data. In contrast, decision-tree-based clustering can generalize to unseen phoneme transitions and take advantage of linguistic

knowledge during clustering. Here, all the questions have the form "Is the left phoneme of the transition a member of set **X** and the right phoneme a member of set **Y**?" The sets **X** and **Y** range from broad phonetic classes, such as sonorant, stop, and unvoiced classes, to distinct phonemes, such as {r} and {s}. In total, 397 phonetic sets are used.

3.2. Support vector machine

Consider the problem of classifying data points into two classes, A_+ and A_-. We are given a training data set $\{(x_i, y_i)\}_{i=1}^m$, where $x_i \subset R^n$ is an input vector variable and $y_i \in \{1, -1\}$ is a class label that indicates which of the two classes, A_+ and A_-, it belongs to. We represent these data points by an $m \times n$ matrix A, in which the i-th row, A_i, corresponds to the i-th data point. The SVM classifier $f(x)$ is of the following form:

$$f(x) = \sum_{i=1}^{m} y_i \alpha_i K(A_i, x) + b, \tag{7}$$

where $K(A_i, x)$ is a kernel function, and α_i and b are parameters to be trained. In this paper, the SVM classifiers with Gaussian kernels are implemented by the SVM tool developed by Lee and Mangasarian [11].

For each K-means-derived cluster or each leaf node of the decision tree, an SVM classifier is trained by using the feature vectors associated with the true boundaries as positive training samples and the randomly selected feature vectors at least 20 ms away from the true boundaries as negative training samples. In the test phase, the feature vectors associated with the speech frames around the hypothesized boundary are examined by the associated SVM classifier. Then, the frame index associated with the feature vector with the maximum classifier output is recognized as the refined boundary.

4. Experiments

We evaluate our approaches on two databases: the TIMIT database and the MATBN (Mandarin Across Taiwan Broadcast News) [7] database.

4.1. Evaluation on the TIMIT database

TIMIT, a well-known read speech corpus with manual acoustic-phonetic labeling, is widely used for the evaluation of automatic speech recognition and phoneme segmentation methods. The TIMIT suggested training and testing sets contain 462 and 168 speakers, respectively. We discard utterances with phones shorter than 10 ms. The resulting training set contains 4,546 sentences, with a total duration of 3.87 hours; while the testing set contains 1,646 sentences, with a total duration of 1.41 hours.

The acoustic models for HMM-based segmentation consist of 50 context-independent phoneme models, each represented by a 3-state continuous density HMM with a left-to-right topology. Each frame of the speech data is represented by a 39-dimensional feature vector comprised of 12 MFCCs and log energy, along with their first and second time derivatives. The frame width is 20 ms and the frame shift is 5 ms.

As in our previous work [6], in the SVM refinement stage, each frame of the speech data is represented by a 45-dimensional feature vector comprised of the above 39 MFCC-based coefficients, plus the zero crossing rate, bisector frequency, burst degree, spectral entropy, weighted entropy, and subband energy. For each hypothesized boundary, the feature vectors of its adjacent left and right frames, together with the symmetrical Kullback-Leibler distance (SKLD) and the spectral feature transition rate (SFTR) between the two feature vectors, are concatenated to form a 92-dimensional augmented vector. The augmented vectors are used as features for phoneme transition clustering and as the input vectors for SVM.

Table 1: Results of HMM-based automatic phoneme segmentation evaluated on the TIMIT database.

Training / Segmentation	Mean boundary distance	%Correct marks (error < tolerance)		
		<5ms	<10ms	<20ms
ML^{10}/ML	9.83 ms	46.69	71.10	88.94
ML^{20}/ML	9.78 ms	46.95	71.23	88.97
ML^{10}+MBE^{10}/ML	7.82 ms	58.48	79.75	92.11
ML^{20}/MBE	8.92 ms	49.93	74.40	90.69
ML^{10}+MBE^{10}/MBE	7.50 ms	58.74	80.51	92.85
ML^{10}+MBE^{10}/MBE_{pdm}	7.14 ms	59.58	81.57	93.74

4.1.1. HMM-based segmentation

Table 1 shows the percentage of phoneme boundaries correctly placed within different tolerances with respect to the human-labeled phoneme boundaries. The experiments were conducted on the test set. The acoustic models were first trained on the training speech according to the human-labeled phoneme transcriptions and boundaries derived by the Baum-Welch algorithm using the ML criterion with 10 iterations, denoted as ML^{10} in Table 1. Then, MBE discriminative training with 10 iterations was applied to further manipulate the ML-trained models, denoted as ML^{10}+MBE^{10} in Table 1. From rows 3 (ML^{20}/ML) and 4 (ML^{10}+MBE^{10}/ML) of the table, we observe that the MBE-trained models significantly outperform the ML-trained models. Clearly, the MBE training is particularly effective in correcting boundary errors in the proximity of human-labeled positions. Comparing the results in rows 3 and 5, and in rows 4 and 6, we also observe that MBE segmentation outperforms conventional ML segmentation, though the improvement is not as significant as that of the MBE-trained models over the ML-trained models. This is because, MBE segmentation, like conventional ML segmentation, is still deficient in the knowledge of the true posterior distribution, even though the MBE criterion accords with the objective of minimizing boundary errors. By comparing rows 6 and 7, where MBE_{pdm} denotes MBE segmentation with explicit phoneme duration models, we observe that the segmentation accuracy can be slightly improved by integrating explicit phoneme duration models into the segmentation process.

4.1.2. Boundary refinement using SVM

We now evaluate the SVM$_{KM}$ classifiers based on K-means clustering and the SVM$_{DT}$ classifiers based on decision-tree-based clustering. By using cross-validation on the TIMIT training data, 151 SVM$_{DT}$ classifiers and 46 SVM$_{KM}$ classifiers are derived for use in the experiments.

Given the boundary of each phoneme transition obtained by HMM-based segmentation, 11 hypothesized boundaries (extracted every 1 ms) around the initial boundary within $\pm$5 ms are examined by the SVM classifier associated with that specific phoneme transition. Table 2 shows the results of SVM-based refinement applied to the initial segmentation derived by the best HMM-based method in Table 1, i.e., "ML^{10}+MBE^{10}/MBE_{pdm}". From Table 2, we observe that, although both SVM$_{KM}$ and SVM$_{DT}$ improve the segmentation accuracy, SVM$_{KM}$ slightly outperforms SVM$_{DT}$.

Table 2: Results of HMM/SVM-based automatic phoneme segmentation evaluated on the TIMIT database.

Methods	Mean boundary distance	%Correct marks (error < tolerance)		
		<5ms	<10ms	<20ms
HMM*	7.14ms	59.58	81.57	93.74
HMM*+SVM$_{KM}$	6.75ms	62.47	84.00	94.33
HMM*+SVM$_{DT}$	6.83ms	62.07	83.70	94.12

Table 3: Results of automatic phoneme segmentation evaluated on the MATBN database.

Training / Segmentation	Mean boundary distance	%Correct marks (error < tolerance)		
		<5ms	<10ms	<20ms
$Unsup.ML/ML$	20.29 ms	16.80	29.68	58.65
$Unsup.ML/MBE$	18.62 ms	18.21	34.10	62.88
ML/ML	13.06 ms	27.67	50.50	83.70
ML/MBE	11.73 ms	30.28	58.25	87.22
ML+MBE/ML	11.99 ms	35.11	59.56	85.01
ML+MBE/MBE	10.91 ms	37.83	63.78	87.53
ML+MBE/MBE_{pdm}	10.29 ms	40.24	66.30	88.43
HMM*+SVM$_{KM}$	9.29 ms	49.02	71.36	89.11

4.2. Evaluation on the MATBN database

The MATBN Mandarin Chinese corpus contains 198 hours of broadcast news from the Public Television Service Foundation (Taiwan). The data includes orthographic transcripts and SGML tagging for annotating acoustic conditions, background conditions, story boundaries, speaker turn boundaries, and acoustic events, such as hesitations and repetitions. We select approximately five hours of speech data from the corpus for further phoneme annotation. To reduce costs, we employ HMM-based segmentation and SVM-based refinement to obtain the initial phoneme segmentation for subsequent manual segmentation and verification. To do this, we divide the speech data into subsets, each containing five minutes of speech. First, we perform unsupervised ML training and forced alignment on the complete set to generate the initial segmentation. When the first subset has been manually verified, it is used for supervised training of the HMMs and SVMs. To prevent over-fitting in HMM training, the remaining unverified data is re-segmented and used to smooth the HMM parameters. Then, based on the new HMMs, we apply forced alignment to the remaining subsets to generate more accurate phoneme boundaries. The above training and segmentation process is repeated until all the subsets have been manually verified. In this way, the accuracy of automatic segmentation can be improved stage by stage, and the overall cost of manual segmentation can be reduced.

Now that the first subset has been processed completely, we evaluate the efficacy of the proposed semi-automatic phoneme segmentation process by applying four minutes of the human-verified speech for supervised training and the remaining one minute for testing. In total, 34 context-independent phoneme HMMs and 14 SVM$_{KM}$ classifiers are used. From Table 3, we observe that the proposed HMM-based segmentation (ML+MBE/MBE_{pdm}, HMM*) significantly outperforms the conventional HMM-based segmentation ($Unsup.ML/ML$). The mean boundary distance achieved is 10.29 ms. By using SVM$_{KM}$ for boundary refinement, the mean boundary distance can be further reduced from 10.29 ms to 9.29 ms. We believe that the segmentation accuracy could be improved even further if more subsets are manually verified, i.e., the cost of labeling one subset could be progressively reduced.

5. Conclusions

We have presented several improved HMM/SVM methods for a two-stage phoneme segmentation framework that imitates the human phoneme segmentation process. In the first stage, HMM-based forced alignment is performed according to the minimum boundary error (MBE) criterion, based on MBE-trained HMMs and explicit phoneme duration models. In the second stage, phoneme-transition-dependent SVM classifiers are used to refine the phoneme segmentation derived by the HMM-based forced alignment step. The efficacy of the proposed framework has been validated on the TIMIT database. We have also applied the framework in a semi-automatic process to facilitate manual labeling of the MATBN Mandarin Chinese database. The preliminary evaluation results are rather promising. The annotation work is ongoing and the results will be made available at a future time.

6. Acknowledgements

This work was funded by the National Science Council, Taiwan, under Grant: NSC95-2221-E-001-035.

7. References

[1] F. Malfrere and T. Dutiot, "High-quality speech synthesis for phonetic speech segmentation," in *Proc. Eurospeech 1997*, pp. 2631–2634.

[2] J. van Santen and R. Sproat, "High accuracy automatic segmentation," in *Proc. Eurospeech 1999*, pp. 2809–2812.

[3] F. Brugnara, D. Falavigna, and M. Omologo, "Automatic segmentation and labeling of speech based on hidden markov models," *Speech Communication*, vol. 12, no. 4, pp. 357–370, 1993.

[4] J.-W. Kuo and H.-M. Wang, "A minimum boundary error framework for automatic phonetic segmentation," in *Proc. ISCSLP 2006, LNAI 4274 Springer*, pp. 399–409.

[5] D. Torre Toledano, M. A. Rodriguez Crespo, and J. G. Escalada Sardina, "Try to mimic human segmentation of speech using HMM and fuzzy logic post-correction rules," in *Proc. the 3th ESCA/COCOSDA International Workshop on Speech Synthesis*, pp. 1263–1266.

[6] H.-Y. Lo and H.-M. Wang, "Phonetic boundary refinement using support vector machine," in *Proc. ICASSP*, 2007.

[7] H.-M. Wang, B. Chen, J.-W. Kuo, and S.-S. Cheng, "MATBN: A Mandarin Chinese broadcast news corpus," *International Journal of Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2, pp. 219–236, 2005.

[8] M. J. Russell and R. K. Moore, "Explicit modelling of state occupancy in hidden markov models for automatic speech recognition," in *Proc. ICASSP 1988*, pp. 5–8.

[9] M. J. Russell and A. E. Cook, "Experimental evaluation of duration modelling techniques for automatic speech recognition," in *Proc. ICASSP 1987*, pp. 2376–2379.

[10] J. Pylkkönen and M. Kurimo, "Duration modeling techniques for continuous speech recognition," in *Proc. Interspeech 2004 - ICSLP*.

[11] Y.-J. Lee and O. L. Mangasarian, "SSVM: A smooth support vector machine," *Computational Optimization and Applications*, vol. 20, pp. 5–22, 2001.