MINIMUM SEGMENTATION ERROR BASED DISCRIMINATIVE TRAINING FOR SPEECH SYNTHESIS APPLICATION

Yi-Jian Wu[†‡] *Hisashi Kawai*[†] *Jinfu Ni*[†] *Ren-Hua Wang*[‡]

†ATR, Spoken Language Translation Laboratories
‡University of Science and Technology of China

E-mail: †{yijian.wu, hisashi.kawai, jinfu.ni}@atr.co.jp, ‡rhw@ustc.edu.cn

ABSTRACT

In the conventional HMM-based segmentation method, the HMM training is based on MLE criteria, which links the segmentation task to the problem of distribution estimation. The HMMs are built to identify the phonetic segments, not to detect the boundary. This kind of inconsistency between training and application limited the performance of segmentation. In this paper, we adopt the discriminative training method and introduce a new criterion, named Minimum Segmentation Error (MSGE), for HMM training. In this method, a loss function directly related to the segmentation error is defined. By minimizing the overall empirical loss with the Generalized Probabilistic Descent (GPD) algorithm, the segmentation error is also minimized. From the results on both Chinese and Japanese data, the accuracy of segmentation is improved. Moreover, this method is robust even when we do not have enough knowledge on HMM modeling, e.g. the number of states is not optimized.

1. INTRODUCTION

Recently, corpus-based concatenative speech synthesis has become popular due to its high quality, which critically depends on the accuracy of the phonetic labeling of the corpus. Since the labeling task needs a lot of human effort and a long time, especially for the large corpus, automatic segmentation (AS) has been very important for corpus-based speech synthesis, providing consistent and accurate phonetic labeling with high efficiency. Many methods have been proposed for the AS task,[1][2][3] and the HMM-based method adopted from automatic speech recognition is now the most popularly used. Although the current results of HMM-based method are quite impressive, there are also shortcomings that prevent them from achieving even better performance.

The conventional method of training the HMM is adopted from speech recognition, which is based on Maximum Likelihood Estimation (MLE) criteria (via a powerful training algorithm, Expectation Maximization algorithm). In other words, this training method links the segmentation task to the problem of distribution estimation, and the HMMs are built to identify the phonetic segments, not to detect the boundary between the phonetic segments. This kind of inconsistency between the training and the application of HMM limits the segmentation performance.

The discriminative training method and the criteria of Minimum Classification Error (MCE) based on the Generalized Probabilistic Descent (GPD) framework has been successful in training HMM for speech recognition [5][6], and to a certain extent segmentation can be regarded as a state recognition task with known transcription. This prompts us to apply the discriminative training method and the corresponding criteria for the segmentation task. In this paper, a new criteria, called minimum segmentation error (MSGE), is proposed to train the HMM under the GPD framework. In this method, we defined a loss function directly related to segmentation errors. By minimizing the overall empirical loss under the GPD framework, the segmentation errors could also be minimized.

This paper is organized as follows. In section 2, we briefly review the GPD framework for parameter optimization. In section 3, the MSGE-based HMM training procedure, including a loss function definition and the parameters updating schedule, is presented in detail. Next, the segmentation accuracy of the HMMs trained by MSGE criteria is evaluated on both Chinese and Japanese data in section 4. Finally, we give our conclusion in section 5.

2. GENERALIZED PROBABILISTIC DESCENT

For a given loss function $\ell(X,\Lambda)$, where X is a feature vector and Λ represents the system parameters, we want to optimize Λ to minimize the overall expectation loss:

$$L(\Lambda) = E[\ell(X,\Lambda)] = \int \ell(X,\Lambda) p(X) dX \,, \tag{1}$$

where $p(X)$ is *a priori* distribution. Since we do not know the *a priori* distribution, we cannot evaluate the expected loss directly. The Generalized Probabilistic Descent (GPD) algorithm[4] is a very powerful algorithm that can be used to accomplish this task. In a GPD framework, the target loss function is minimized according to an iterative procedure

$$\Lambda_{t+1} = \Lambda_t - \varepsilon_t U_t \nabla \ell(X_t,\Lambda)\Big|_{\Lambda=\Lambda_t} \,, \tag{2}$$

where U_t is a positive definite matrix, X_t is the tth training sample used in the sequential training process, and ε_t is a sequence of positive numbers that satisfies the conditions:

$$i)\sum_{t=1}^{\infty} \varepsilon_t \to \infty \,,\; ii)\sum_{t=1}^{\infty} \varepsilon_t^2 < \infty \,. \tag{3}$$

In the above, an infinite number of training samples is required for convergence. In practice, only a finite number of samples are available. However, we can minimize the empirical loss

$$L_0(\Lambda) = \frac{1}{N}\sum_{i=1}^{N} \ell(X_i,\Lambda) = \int \ell(X,\Lambda) p_N(X) dX \tag{5}$$

under the GPD framework. With sufficient training samples, the empirical loss converges to the actual expected loss. It should be noted that the GPD framework is a general framework for various definitions of loss function. A more detailed introduction and discussion of GPD algorithm can be found in the literature [4][6].

3. MINIMUM SEGMENTATION ERROR

The conventional measurement of segmentation error is usually defined as the time difference in boundary location between human labeling and automatic labeling, i.e. error length. According to this definition, the segmentation errors are discrete (in frame scale) and not explicitly related to the parameters of the HMM. Therefore, the gradient-based optimization methods cannot be used to minimize the segmentation errors directly. Here, we introduced a new measurement, named error degree, for segmentation error. Under the new measurement, a meaningful loss function is defined, which is directly related to segmentation errors and can be minimized by using the GPD algorithm.

3.1. Measurement for segmentation

Usually, the HMM-based segmentation is a state alignment procedure performed by the Dynamic Programming algorithm (e.g. Viterbi). For simplification, we look into the segmentation procedure of a sample X that consists of two connected segment units X_1 and X_2, i.e. $X = \{X_1, X_2\}$. In the DP algorithm, the likelihood of the best state alignment is calculated by

$$g_b(X;\Lambda) = \max_{Q} g(X,Q;\Lambda) = g(X,\overline{Q}_b;\Lambda), \qquad (6)$$

where $\overline{Q}_b$ is the optimal state sequence with maximum likelihood, which is calculated as

$$\begin{aligned} g(X,\overline{Q};\Lambda) &= \log P(X,\overline{Q};\Lambda) \\ &= \sum_{t=1}^{T}[\log a_{\overline{q}_{t-1}\overline{q}_t} + \log b_{q_t}(x_t)] + \log \pi_{\overline{q}_0}, \end{aligned} \qquad (7)$$

where $a_{\overline{q}_{t-1}\overline{q}_t}$ and $b_{q_t}(x_t)$ are transition probability and output probability distribution, respectively.

With the optimal state alignment, the corresponding phonetic boundary is labeled at time t', which satisfies the condition that $\overline{q}_{t'-1}$ is the final state of first unit and $\overline{q}_{t'}$ is the first state of the next unit. If the boundary is not the same as the humanly labeled boundary, i.e. the correct boundary, the optimal state alignment is regarded as "incorrect" state alignment. Also, the "correct" state alignment is defined as the optimal state alignment with the correct phonetic boundary restriction, which satisfies

$$\begin{aligned} g_c(X;\Lambda) &= g_1(X_1,\overline{Q}_{c1};\Lambda) + g_2(X_2,\overline{Q}_{c2};\Lambda) \\ &= g(X,\overline{Q}_c;\Lambda), \end{aligned} \qquad (8)$$

where $\overline{Q}_{c1}$ and $\overline{Q}_{c2}$ are respectively the optimal state sequences of X_1 and X_2, and $\overline{Q}_c = \{\overline{Q}_{c1}, \overline{Q}_{c2}\}$.

Accordingly, we defined *error degree* as the difference in likelihood between the incorrect and the correct state sequence, i.e.

$$E_d = g_b(X,\Lambda) - g_c(X,\Lambda). \qquad (9)$$

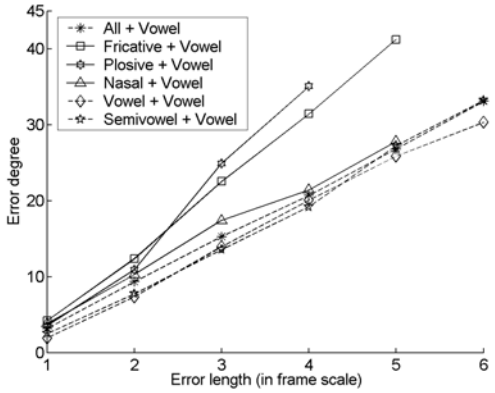


Figure 1. Correlation between error degree and error length

where $g_b(X,\Lambda)$ and $g_c(X,\Lambda)$ are the likelihood of incorrect and correct state sequences, respectively. When the segmentation is correct, i.e. $\overline{Q}_b = \overline{Q}_c$, E_d is equal to 0. If E_d is larger than 0, this indicates that the segmentation is incorrect and the value of E_d reflects how large the segmentation error is in some aspect. In order to find the meaning of error degree in depth, we analyzed the correlation between error degree and error length.

The HMMs trained by MLE criteria were used to segment the Japanese training data (the details of the data information can be found in section 4.2). The correlation between error degree and error length was analyzed from all segmentation errors, and the correlations of some typical boundaries are shown in figure 1. From the figure, error degree is nearly linear with error length, and for different boundary types, the slope is different, i.e. the correlation is context dependent. For the boundary between plosive and vowel, or fricative and vowel, the slope is relative large, i.e. error degree is sensitive to error length. For the boundary between vowel and vowel, or semivowel and vowel, the slope is relative small, i.e. error degree is less sensitive to error length. This characteristic is identical to the requirement of concatenative speech synthesis, which is quite sensitive to the segmentation accuracy of plosive segments, since a plosive segment with an imprecise boundary might result in two bursts or no burst in synthetic speech, and less sensitive to the accuracy of vowel segment. In this sense, error degree is a meaningful factor for measuring the segmentation error. Because of the correlation between error degree and error length, minimization of error degree is also related to minimizing error length.

3.2. Loss function definition

To consider both explicit error length and inherent error degree, we defined the loss function as

$$\ell(\Lambda) = E_\ell^\alpha E_d = E_\ell^\alpha (g_b(X,\Lambda) - g_c(X,\Lambda)), \qquad (10)$$

where E_ℓ is error length and α is a positive number. In this loss function, E_ℓ^α is regarded as a constant number in the optimization procedure by the GPD algorithm, so the loss function can be differentiated with respect to the parameters. The meaning of E_ℓ^α can be explained as follows.

On the one hand, it indicates the consideration of explicit error length. When α is larger than 0, the loss of the training data with large error length is large, and accordingly the model

parameters are updated on a large scale, which means there is more focus on eliminating large errors. From this point of view, the loss function provides a flexible way to optimize the parameter for the different focus. On the other hand, E_ℓ^α means the weight of the training data, i.e. the same performance can be achieved by repeating the training data E_ℓ^α times when the loss function is defined as E_d only.

This definition of loss function is much more meaningful, reflecting both the explicit error length and the inherent error degree. Moreover, by this definition, the loss function is continuous, differentiable, and directly related to the parameters of HMM. By using the gradient-based optimization method (e.g. GPD), the loss function can be minimized, which relates to a minimization of the segmentation error.

3.3. Parameter updating

Next, we optimized the parameters under this loss function by the GPD algorithm. For a state j of HMM h which has M mixtures, the output probability distribution is

$$b_{h,j}(x_t) = \sum_{m=1}^{M} c_{h,j,m} b_{h,j,m}(x_t) = \sum_{m=1}^{M} c_{h,j,m} G\left[x_t; \mu_{h,j,m}, R_{h,j,m}\right], \quad (10)$$

where $b_{h,j,m}(\cdot)$ is the output probability of one mixture, $G[]$ is a normal Gaussion distribution, and $c_{h,j,m}$, $\mu_{h,j,m} = [\mu_{h,j,m,l}]_{l=1}^{D}$ and $R_{h,j,m} = [\sigma_{h,j,m,l}]_{l=1}^{D}$ are mixture weights, mean vector and covariance matrix, respectively.

It should be noted that the HMM as a probability measure has some original constraints, such as: 1) the function is positive; 2) $\sum_m c_{h,j,m} = 1$ for all h, j, and 3) $\sigma_{h,j,m,l} > 0$. In order to maintain these constraints during parameter adaptation, we should take some parameter transformations as follows:

$$c_{h,j,m} \to \widetilde{c}_{h,j,m} \text{ where } c_{h,j,m} = \frac{\exp(\widetilde{c}_{h,j,m})}{\sum_k \exp(\widetilde{c}_{h,j,m})} \quad (11)$$

$$\mu_{h,j,m} \to \widetilde{\mu}_{h,j,m} = \mu_{h,j,m} R_{h,j,m}^{-1} \quad (12)$$

$$R_{h,j,m} \to \widetilde{R}_{h,j,m} = \log(R_{h,j,m}) \quad (13)$$

The transformation in (12) is important for designing the step size for convergence. More discussion about the parameter transformation can be found in [6].

For a sample X_n in the training set, the adaptation of the parameter is

$$\Lambda_{h,j,m}(n+1) = \Lambda_{h,j,m}(n) - \varepsilon \frac{\partial \ell(X_n; \Lambda)}{\partial \Lambda_{h,j,m}}\Big|_{\Lambda=\Lambda_n}, \quad (14)$$

where

$$\frac{\partial \ell(X;\Lambda)}{\partial \Lambda_{h,j,m}} = E_\ell^\alpha \frac{\partial (g_b(X,\Lambda) - g_c(X,\Lambda))}{\partial \Lambda_{h,j,m}} = E_\ell^\alpha \sum_{t=1}^{T} (\delta(q_{bt} - j) - \delta(q_{ct} - j)) b_{h,j}^{-1}(x_t) \frac{\partial b_{h,j}(x_t)}{\partial \Lambda_{h,j,m}}, \quad (15)$$

where $\delta(\cdot)$ denotes the Kronecher delta function. For the mean vector, the updating rule is

$$\frac{\partial b_{h,j}(x_t)}{\partial \widetilde{\mu}_{h,j,m}} = c_{h,j,m} b_{h,j,m}(x_t) R_{h,j,m}^{-1} (x_t - \mu_{h,j,m}) \cdot \quad (16)$$

Finally,

$$\mu_{h,j,m}(n+1) = \widetilde{\mu}_{h,j,m}(n+1) R_{h,j,m} \cdot \quad (17)$$

Similarly, for the covariance matrix $R_{h,j,m}$, the updating rule is

$$\frac{\partial b_{h,j}(x_t)}{\partial \widetilde{R}_{h,j,m}} = c_{h,j,m} b_{h,j,m}(x_t) \cdot \left(R_{h,j,m}^{-1} R_{h,j,m}^{-1} (x_t - \mu_{h,j,m})(x_t - \mu_{h,j,m})^T - I_D\right), \quad (18)$$

where I_m is a identity matrix. Finally,

$$R_{h,j,m}(n+1) = \exp\{\widetilde{R}_{h,j,m}(n+1)\} \cdot \quad (19)$$

Also, the mixture weight is updated as

$$\frac{\partial b_{h,j}(x_t)}{\partial \widetilde{c}_{h,j,m}} = b_{h,j,m}(x_t) c_{h,j,m} (1 - c_{h,j,m}) \quad (20)$$

Finally

$$c_{h,j,m}(n+1) = \frac{\exp(\widetilde{c}_{h,j,m}(n+1))}{\sum_k \exp(\widetilde{c}_{h,j,m}(n+1))} \cdot \quad (21)$$

The meaning of the updating rule can be explained as follows. In equation (15), $\delta(q_{bt} - j) - \delta(q_{ct} - j)$ is equal to zero when $q_{bt} = q_{ct}$, or equal to 1 when $q_{bt} \neq q_{ct}$ and $q_{bt} = j$, or equal to –1 when $q_{bt} \neq q_{ct}$ and $q_{ct} = j$, which indicates, for an input vector, if the best state alignment differs with the correct state alignment, the updating rule is to move the parameters of the incorrect state model far away from the vector and to move the parameters of the correct state model close to the vector.

4. EVALUATION AND DISCUSSION

We trained the HMMs by using MLE and MSGE criterion and then compared the segmentation accuracies of these two methods. The MLE-based HMM training is performed by the HTK tools.[8] In MSGE-based training, the HMMs are initialized by the results of MLE-based training. The performance was evaluated on both Chinese and Japanese data.

4.1. MSGE-based HMM training on Chinese

The training and testing data consists of 1000 and 680 sentences, including 27,312 and 15,872 phones, respectively. All of the data had been carefully hand-labeled by the same labeler.

The phone set used here has 60 phonemes, including 21 initials, 37 finals, pause and silence. Monophone HMMs are adopted and the numbers of states are three for initials, pause and silence, and five for finals. The number of mixture components set to five for each phoneme. The acoustic features are 16-order MFCC and energy and the delta coefficients. The analysis window size and shift are 20 ms and 5 ms, respectively.

From the result of close and open test in figure 2(a), the MSGE-based discriminative training is convergent after 10-20 iterations. As can be seen in Table 1, the accuracy of segmentation improved after MSGE-based training, especially for the errors less than 5ms. We also examined the effect of error length on loss function by training with different α values. When α increases from 0 to 1, which means we have more focus on larger errors, the percentage of error less than 30 ms increased 0.13%, whereas the percentage of error less than 5ms decreased 0.83%.

The details on accuracy with different phonetic boundaries are shown in Table 2. After MSGE-based training, the average error of the CV-boundary decreased from 4.51 ms to 3.60 ms, i.e. a reduction of 19.7%, whereas that of the VV-boundary decreased 9.6%. Since we noted that concatenative speech synthesis is much more sensitive to the accuracy of the CV-boundary and insensitive to the VV-boundary, this improvement appears to be reasonable for speech synthesis.

4.2. MSGE-based HMM training on Japanese

The training and testing data consists of 2263 and 501 phonetically balanced sentences, including 185,404 and 30,706 phones respectively, and all of the data had been hand-labeled. The phone set used here includes 60 phonemes. Monophone HMMs are also used, and the numbers of states and mixture components are three and five, respectively for each phoneme. The configuration of the acoustic feature analysis is the same as that used in the former experiment.

The convergence of MSGE-based discriminative training on Japanese data can be found in figure 2(b). From Table 3, the segmentation accuracy for Japanese was improved after MSGE-based training, and the effect of E_ℓ with different α values is similar to that for Chinese. In Table 4, the largest improvement also occurred in the accuracy of the CV-boundary, where the average error reduced from 7.85 ms to 4.84 ms, i.e. a reduction of 38%.

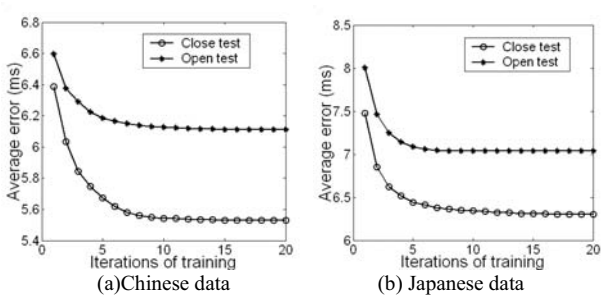
Comparing the results for Japanese and Chinese data, we found that the improvement for Japanese is much larger than that for Chinese. One reason is that the HMM modeling in Japanese is not optimized, that is, it simply uses 3-state model for all phonemes. Therefore, the segmentation accuracy of the baseline trained by MLE criteria for Japanese is much worse than that for Chinese. Nevertheless, the difference in accuracy between Japanese and Chinese data is reduced after MSGE-training. This indicates that the MSGE-based training method can work well even when the HMM modeling is not optimized. Furthermore, it can compensate for the inaccuracy of the HMM modeling to a certain extent.

5. CONCLUSION

In this paper, we proposed minimum segmentation error (MSGE) based discriminative training method for automatic segmentation. In this method, a meaningful loss function was defined to directly relate to the segmentation errors. By minimizing the overall empirical loss with the GPD algorithm, the segmentation error was also minimized. We investigated its performance both on Chinese and Japanese. From the results, the accuracy of segmentation is largely improved after MSGE-based training, even when the HMM modeling is not optimal. As the improvement in eliminating large errors, e.g. larger than 30 ms, is very limited. Further research and experiments on eliminating the large errors by using the explicit duration model are in progress.

ACKNOWLEDGEMENT

This research was supported in part by the Telecommunications Advancement Organization of Japan.



(a)Chinese data (b) Japanese data

Figure 2. Convergence of MSGE-based discriminative training

Table 1. Segmentation accuracy for Chinese

	Percentage of the accuracy (%)				Aver (ms)
	≤5ms	≤10ms	≤20ms	≤30ms	
MLE	70.44	86.89	95.58	97.75	6.856
MSGE(α=0)	76.01	88.70	95.74	97.85	6.112
MSGE(α=1)	75.18	88.65	95.81	97.98	6.174

Table 2. Accuracy with different phonetic boundaries (Chinese)

	Average error (ms)			
	CC	CV	VC	VV
MLE	×	4.51	5.69	11.99
MSGE(α=0)	×	3.60	5.37	10.83

Table 3. Segmentation accuracy for Japanese

	Percentage of the accuracy (%)				Aver (ms)
	≤5ms	≤10ms	≤20ms	≤30ms	
MLE	60.84	79.64	92.07	96.31	8.666
MSGE(α=0)	70.15	84.46	94.00	97.29	7.035
MSGE(α=1)	69.68	84.40	94.24	97.43	7.084

Table 4. Accuracy with different phonetic boundaries (Japanese)

	Average error (ms)			
	CC	CV	VC	VV
MLE	5.18	7.85	7.16	11.31
MSGE(α=0)	4.64	4.84	6.45	9.59

REFERENCES

[1] P. Carvalho, I. Trancoso and L. Oliveira, "Automatic Segment Alignment for Concatenative Speech Synthesis in Portuguese", in Proc. RECPAD'98 – 10th Portuguese Conference on Pattern Recognition, Lisboa, 1998.

[2] Y.J. Kim and A. Conkie, "Automatic segmentation combining an HMM-based approach and spectral boundary correction", in ICSLP 2002, pp145-148, 2002

[3] A. Sethy, S. Narayanan, "Refined speech segment- ation for concatenative speech synthesis", in ICSLP 2002, pp149-153, 2002

[4] J.R. Blum, "Multidimensional stochastic approxi- mation methods," Ann. Math. Stat, vol. 25, pp.737-744, 1954

[5] E. McDermott, "Discriminative training for speech recognition", Dissertation for doctor degree, March 1997

[6] W. Chou, Discriminant-Function-Based Minimum Recognition Error Rate Pattern-Recognition Approach to Speech Recognition, in Proc. IEEE, Vol.88, No.8, pp1201-1223, Aug. 2000

[7] B.H. Juang, W. Chou, and C.H. Lee, "Minimum classification error rate methods for speech recognition", IEEE Trans. Speech Audio Processing, vol.5, pp257-265, May 1997

[8] S.Young, D. Kershaw, J.Odell, D. Ollason, V. Valtchev and P. Woodland, "The HTK Book," Entropic Ltd. 1999