On Adaptively Learning HMM-Based Classifiers Using Split-Merge Operations*

Sang-Woon Kim[1] and Soo-Hwan Oh[2]

[1] *Senior Member, IEEE.* Dept. of Computer Science and Engineering,
Myongji University, Yongin, 449-728 Korea
kimsw@mju.ac.kr

[2] Dept. of Computer Science and Engineering, Myongji University,
Yongin, 449-728 Korea
ohsh@mju.ac.kr

Abstract. In designing classifiers for automatic speech recognitions, one of the problems the user faces is to cope with an unwanted variability in the environment such as changes in the speaker or the acoustics. To overcome this problem, various adaptation schemes have been proposed in the literature. In this short paper, rather than selecting a single acoustic model as being representative of a category, we adaptively find the optimal or near-optimal number of hidden Markov models during the Baum-Welch (BW) learning process through *splitting* and *merging* operations. This scheme is based on incorporating the split-merge operations into the HMM parameter re-estimation process of the BW algorithm. In the *splitting* phase, an acoustic model is divided into *two* sub-models based on a suitable criterion. On the other hand, in the *merging* phase, *two* models are combined into a single one. The experimental results demonstrate that the proposed mechanism can efficiently resolve the problem by adjusting the number of acoustic models while increasing the classification accuracy. The results also demonstrate that the advantage gained in the case of multi-modally distributed data sets is significant.

Keywords: Automatic Speech Recognitions (ASR), Hidden Markov Models (HMM), Baum-Welch (BW) Algorithm, Splitting - Merging Techniques.

1 Introduction

Hidden Markov Models (HMMs) have been proven to be one of the most successful statistical modeling methods in the area of automatic speech recognition systems (ASR), especially of continuous speech recognition [7]. One of the problems in designing classifiers for ASR is that of coping with unwanted variability as is encountered when there are changes in the environment concerning the speaker or the acoustics. To overcome this problem, various adaptation schemes such as the deleted interpolation [2], the speaker adaptation [1], the corrective

* This work was generously supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

M. Ali and R. Dapoigny (Eds.): IEA/AIE 2006, LNAI 4031, pp. 668–673, 2006.
© Springer-Verlag Berlin Heidelberg 2006

training [3], and the model clustering and splitting [6], have been proposed in the literature. Rather than selecting a single acoustic model (e.g., an HMM unit[1]) as representative of a particular category, the above schemes permit more than one acoustic model to be assigned to a category. However, typically the number of acoustic models is randomly determined in advance, and is decided by the number of pattern classes, or by resorting to a clustering of the training samples. The most popular training method for the parameter estimation of the acoustic module is the Baum-Welch (BW) algorithm based on the Maximum Likelihood (ML) criterion [7]. Other approaches are omitted here in the interest of brevity, but can be found in the literature including [4].

Motivated by the methods mentioned above, we investigate an adaptive learning method for HMM-based classifiers by using a splitting-merging technique. Our idea is to incorporate the split-merge operations into the BW learning process without resorting to any particular *transformation*. In the proposed method, the training data set is automatically clustered into multiple subsets through the split-merge operations of the BW learning process. With the *merging* operation, we combine similar data points which are close to their nearest neighbors, into a cluster. As opposed to this, in the *splitting* phase, we distribute two distant points into different clusters. As a criterion of merging or splitting, we utilize the differences in magnitude between the output probabilities of the models for the sample points and their representative values.

The main contribution of this paper is to demonstrate that the performance of HMM-based classifiers can be increased by employing an adaptive learning method - which is crucial in multi-modally distributed data sets. This has been done by incorporating a splitting-merging technique into the BW learning process and by demonstrating its power in classification accuracy. The reader should observe that this philosophy is *quite distinct* from those used in the recently-proposed SAT (Speaker Adaptive Training) [1] or the CAT (Cluster Adaptive Training) [5] strategies.

2 Adaptive Learning of HMM-Based Classifiers

We consider that the problem of attempting to recognize C different speech pattern classes. Then, an HMM-based classifier is designed with C HMMs to separate the C pattern classes. Each of the HMMs evaluate the output probability on the basis of the observed input vector strings. It then selects the largest output probability product and assigns the unknown input pattern to the corresponding class.

Consider a Markov chain with N states $\{q_1, \cdots, q_N\}$ and transition probabilities $P\{q_i \rightarrow q_j\} = a_{ij}$. Let $s(t)$ denote the state at time t. At each $t = 1, \cdots, T$, one of M output symbols or observations, $v_1, \cdots, v_M$, is generated with a probability $P\{v_k|s(t) = q_i\} = b_{ik}$. A hidden Markov model λ is specified by the $N \times N$ matrix $A = [a_{ij}]$, the $N \times M$ matrix $B = [b_{ik}]$, and the initial description

[1] In this paper, the “HMM unit” or “HMM module” represent a computational acoustic unit, which evaluates the output probability of an HMM-based classifier.

$\pi_i = P\{s(0) = q_i\}$. Given a model $\lambda = (A, B, \pi)$, the probability of a sequence of observations, $v_{y_1}, \cdots, v_{y_T}$ (each $y_t \in \{1, \cdots, M\}$), λ can be calculated by the *Forward-Backward* algorithm. In the case of the ML estimate criterion, the goal of the training is then to find the best set of parameters, λ^*, such that $\lambda^* = argmax_\lambda P_\lambda\{y_1^T\}$. The approach to iteratively maximize $P_\lambda\{y_1^T\}$ is referred to as Baum-Welch (BW) algorithm. Starting from initial guesses[2], the model parameters λ are iteratively updated according to the *Forward-Backward* algorithm so that $P_\lambda\{y_1^T\}$ is maximized at each iteration. Details of the algorithms are omitted here, but can be found in the well-known pieces of literature.

The problem we encounter in learning the HMM is to select the number of acoustic models required to optimize the HMM-based classifier for automatic speech recognition, as well as to estimate the parameter sets. We propose a systematic method for efficiently selecting the optimal or near-optimal *number* of HMM modules for each class. The selection is itself an iterative process and is achieved even as the HMM parameter set is estimated using the BW algorithm.

The procedure of the proposed algorithm can be formalized as follows:

1. *Initialization* : For every data sample, j, we initially train an HMM parameter set, λ_j, with the Baum-Welch algorithm. After this learning, the output probability, P_{λ_j}, (for each j), is used as the *representative* value, $P_{\lambda_{j0}}$, of the sample data point in the following steps;

2. *Splitting* : For every cluster, k, we train a model, λ_k, with the BW algorithm. In this learning[3], if the difference in magnitude between the output probability of a sample i, $P_{\lambda_k}(i)$, and its representative value, $P_{\lambda_{i0}}$, is greater than a threshold value ρ, namely, if $\|P_{\lambda_k}(i) - P_{\lambda_{i0}}\| > \rho$, then the data element i which has the greatest value in the cluster k is *split* as a new cluster, and the number of clusters is increased;

3. *Merging* : After clustering all of samples into clusters according to their output probabilities, we again train a HMM to get a parameter set for each cluster. In this learning, we consider all samples i and j of any two clusters, k and l, respectively. If the magnitudes of the different representative values, $P_{\lambda_{i0}}$ and $P_{\lambda_{j0}}$, and the output probabilities, $P_{\lambda_l}(i)$ and $P_{\lambda_k}(j)$, are smaller than the ρ, namely, if $\|P_{\lambda_l}(i) - P_{\lambda_{i0}}\| < \rho$ and $\left\|P_{\lambda_k}(j) - P_{\lambda_{j0}}\right\| < \rho$, then the two clusters, k and l, are *merged* into “a” cluster, and the number of clusters is decreased;

4. *Termination* : If *Splitting* or *Merging* step does not occur any more, then the process terminates. Otherwise, the above Steps 2 and 3 are repeated.

[2] In the *discrete* HMM, it is important to have a reasonable set of initial estimates. Empirical studies showed that we can use a uniform distribution to generate initial estimates.

[3] The learning has two versions: *Top-down* and *Bottom-up*. In the *Top-down* approach, we start the learning with ‘a’ cluster, in which all training samples are included. In the *Bottom-up* approach, on the other hand, initially the number of clusters is equivalent to the number of data samples. In this paper, we tested the experiments using the *Top-down* approach.

Table 1. The experimental learning steps for two speech data sets $C_1 = \{V_{1,i}\}_{i=0}^{9}$ and $C_2 = \{V_{2,i}\}_{i=0}^{9}$. In the notation of "$a : b$", the a's (which are integer values) are the serial numbers of the clusters, and the b's (real values) are the different magnitudes of the *representative* values and the output probabilities. The details of these terms can be found in the text.

# of epoch	data class	$V_{1,0}$ $V_{2,0}$	$V_{1,1}$ $V_{2,1}$	$V_{1,2}$ $V_{2,2}$	$V_{1,3}$ $V_{2,3}$	$V_{1,4}$ $V_{2,4}$	$V_{1,5}$ $V_{2,5}$	$V_{1,6}$ $V_{2,6}$	$V_{1,7}$ $V_{2,7}$	$V_{1,8}$ $V_{2,8}$	$V_{1,9}$ $V_{2,9}$
1	C_1	1: 93.8	1: 63.6	1: 73.1	1: 70.9	1: 38.1	1: 40.1	1: 47.8	1: 41.8	1: 57.9	1: 33.9
	C_2	1: 52.7	1: 44.8	1: 62.3	1: 51.5	1: 49.8	1: 26.6	1: 66.8	1: 56.9	1: 69.8	1: 39.7
2	C_1	2: 62.7	2: 38.4	2: 43.5	2: 44.7	2: 15.4	2: 28.2	1: 40.6	2: 20.1	2: 41.4	2: 21.6
	C_2	1: 51.1	1: 36.5	1: 65.8	1: 52.7	1: 39.8	1: 15.4	1: 48.0	1: 37.9	1: 47.1	1: 25.1
3	C_1	2: 62.7	2: 38.4	2: 43.5	2: 44.7	2: 15.4	2: 28.2	1: 35.8	2: 20.1	2: 41.4	2: 21.6
	C_2	1: 51.3	1: 35.7	3: 0.0	1: 52.4	1: 38.3	1: 13.4	1: 44.4	1: 33.7	1: 43.4	1: 22.9
4	C_1	4: 0.0	2: 37.6	2: 40.5	2: 44.6	2: 14.5	2: 27.3	1: 48.0	2: 21.4	2: 37.0	2: 21.9
	C_2	1: 51.3	1: 35.7	3: 0.0	1: 52.4	1: 38.3	1: 13.4	1: 44.4	1: 33.7	1: 43.4	1: 22.9

3 Experimental Results

The proposed algorithm has been tested and compared with conventional ones. This was first done by performing experiments on a naturally spoken data set, cited from the ETRI (Electronics and Telecommunications Research Institute, http://www.etri.re.kr/). The ETRI data set consists of a total of 1,150 speech patterns, which correspond to the 115 kinds of bi-syllabic words spoken by ten speakers, five males and five females. The details of the pre-processing phases are omitted here, but can be found in the related manuals.

We report the run-time characteristics of the proposed algorithm for the speech data set. First of all, Table 1 shows an intermediate part of the learning steps (processes) for the two speech data sets $C_1 = \{V_{1,i}\}_{i=0}^{9}$ and $C_2 = \{V_{2,i}\}_{i=0}^{9}$. The speech data C_1 represents a Korean trisyllable phrase, "GaGeEa-", which means "at a store" in English, while the speech data C_2 is for a Korean quadrisyllabic phrase, "KwaGeoEaNun", which means "in the past" in English. We started the learning with a cluster each for both C_1 and C_2, i.e., by invoking *Top-down* learning. Here, the employed discrete HMM was the ergodic one, where the number of states and output symbols are 12 and 32, respectively. The threshold value was set as $\rho = 30$ [4].

From Table 1, we can see that optimal or near optimal number of HMM modules can be adaptively found in the learning process. We accomplished this by employing the *splitting-merging* strategy. First of all, consider the results for

[4] We selected this figure as the threshold value, ρ, after doing the experiment several times. It is doubtful whether the same threshold will work for various classes of speech recognition data. The choice of the threshold determines the number of clusters. A more valid choice would most probably be based on using a choice of threshold which would be some function of the data set to incorporate variability like noise conditions and male-female disparity. This choosing problem which will most probably arise in practical cases is currently being investigated.

Table 2. The experimental results for the speech data set. Here, *Acc*'s are the classification accuracies (%), and t_1 and t_2 are the required processing CPU-times (in seconds) for the learning and the classification, respectively. The details of these terms can be found in the text.

Learning	# of States	# of Symbols	# of Clusters	Averaged # of Modules per a class	$t_1(sec)$	$Acc(\%)$	$t_2(sec)$
CLM	8	32	115	1	901	86.51	40
	8	64	115	1	1,526	92.64	70
	8	128	115	1	2,750	94.66	128
PCM	8	32	313	3	1,322	86.16	110
	8	64	357	3	2,158	90.28	216
	8	128	427	4	3,875	92.29	474
ALM	8	32	788	7	2,150	96.15	277
	8	64	955	8	4,279	98.42	578
	8	128	1,033	9	8,272	98.95	1,146

the first iteration (epoch), captioned '1'. All input patterns of C_1 and C_2 classes are regarded as those of the same cluster, namely, '1'. However, in the second iteration, the cluster of having the highest value (in the absolute sense), is *split* into a new cluster, and thus the $V_{1,0}$ of 93.8 is selected as a new cluster numbered as '2'. After this separation, the remaining words are classified into clusters '1' or '2' by invoking a clustering procedure. Identical comments can also be made about the other iteration steps. The reader should observe that the number of clusters, namely, *four*, can be automatically decided within *only* four iterations. As a consequence, we can design an HMM-based classifier that consists of four HMM modules even though the number of pattern classes is *two*.

Table 2 shows the experimental results of the proposed method for the ETRI speech data set. In *CLM* (the *C*onventional *L*earning *M*ethod), the classifiers that were designed processed just *one* HMM module per category as done for conventional classifiers. In *PCM* (the *P*re-*C*lustering Learning *M*ethod), the number of HMM modules is determined by invoking a clustering algorithm before training the models. On the other hand, in the proposed *ALM* (*A*daptive *L*earning *M*ethod), the number of modules is *adaptively* decided in the learning process [5].

From Table 2, we can see that an optimal or near optimal number of HMM modules for an HMM-based classifier can be selected adaptively. Consider the results of the *ALM* method. Here, three kinds of HMM-based classifiers were designed. The numbers of the states, namely, 8, are the same and the numbers of output symbols are 32, 64 and 128, respectively. The cluster numbers obtained from this learning are 788, 955, and $1,033$, respectively. The reader should observe that these results are *automatically* determined (without a user-intervention) from the split-merge processes. Then, a comparison of the *Acc*'s of *CLM*, *ALM*, and *PCM* shows that the HMM-based classifiers adaptively trained with *ALM* outperform the others. The comparison also shows that the results

[5] Evaluation is performed by using the *R*esubstitution (R) method, in which the same samples are used for both designing and testing the classifier.

obtained by PCM (worth than CLM) are not improved, and even worsened by the pre-clustering. Finally, it should be mentioned that the processing CPU-times of the proposed method is "marginally" higher. Approximately 5 to 10 fold increasing in processing time for *ALM* compared to *CLM* is based on the heavy iteration in the split-merge processes.

4 Conclusions

In designing classifiers for automatic speech recognitions, one of the difficult problems encountered is one of coping with an unwanted variability in the environment such as changes in the speaker or the acoustics. In this paper, we have proposed an adaptive learning mechanism to solve the problem using a *splitting-merging* technique. Rather than independently performing the clustering and the learning (estimation) processes, we have suggested a new scheme in which both processes are simultaneously incorporated. The experimental results demonstrate that the proposed scheme can efficiently resolve the problem of the unwanted variabilities by selecting an appropriate number of HMM acoustic models. Especially we emphasize that this adaptive learning method can be used advantageously for multi-modally distributed data sets. However the problems of reducing the processing CPU-time for the adaptive learning, and that of optimizing the experimental parameters of the method, are still open.

Acknowledgements. The authors would like to express their gratitude for the KRF Grant. They are also very grateful to Prof. B. J. Oommen who helped us in the preparation of this manuscript.

References

1. Anastasakos, T., McDonough, J., Schwartz, R., Kakhoul, J.: A compact model for speaker-adaptive training. *Proceedings of ICSLP*, 1137–1140, 1996
2. Bahl, L. R., Jelinek, F., Mercer, R. L.: A maximum likelihood approach to continues speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-5** 179–190, 1983.
3. Bahl, L. R., Brown, P. F., deSouzaBrown, P. V., Mercer, R. L.: A new algorithm for estimation of hidden Markov model parameters. *Proceedings of ICASSP'88*, New York, 493–496, April 1988.
4. Ben-Yishai, A., Burshtein, D.: A discriminative training algorithm for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, **12(3)** 204–217, May 2004.
5. Gales, M. J. F.: Cluster adaptive training for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, **8(4)** 417–428, July 2000.
6. Lee, K. F.: *Automatic Speech Recognition - The development of the SPHINX System.* Kluwer Academic Publishers, Boston, 1989.
7. Nakagawa, S.: A survey on automatic speech recognition. *IEICE Transactions on Information and Systems*, **E85-D(3)** 465–486, March 2002.