A FAST ALGORITHM FOR UNSUPERVISED INCREMENTAL SPEAKER ADAPTATION

Michael Schüßler[1] *Florian Gallwitz*[2] *Stefan Harbeck*[2]

[1] Bayerisches Forschungszentrum für wissensbasierte Systeme (FORWISS)
Forschungsgruppe Wissensverarbeitung
Am Weichselgarten 7, 91058 Erlangen–Tennenlohe, Germany
E-mail: schuess@forwiss.uni-erlangen.de

[2] Universität Erlangen-Nürnberg,
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstraße 3, 91058 Erlangen, Germany

ABSTRACT

Speaker adaptation algorithms often require a rather large amount of adaptation data in order to estimate the new parameters reliably. In this paper, we investigate how adaptation can be performed in real–time applications with only a few seconds of speech from each user. We propose a modified Bayesian codebook reestimation which does not need the computationally intensive evaluation of normal densities and thus speeds up the adaptation remarkably, e.g. by a factor of 18 for 24–dimensional feature vectors. We performed experiments in two real–time applications with very small amounts of adaptation data, and achieved a word error reduction of up to 11%.

1 INTRODUCTION

Speaker adaptation has been a field of intensive research for several years. Great progress has been made in the development of theoretically well–founded algorithms as well as in the achieved experimental results. Approaches based on optimality criteria such as Maximum Likelihood (ML) and Maximum a posteriori (MAP) have received the most attention in the last few years.

Motivated by this progress we investigated the performance of these methods under difficult conditions, where the system is only used for a short time (e.g. one dialog) by each speaker and where no enrollment speech can be collected off–line. A typical example for this situation is the train timetable information system EVAR developed at our institute [3], which is accessible via public telephone line since January 1994. In this task, the best use of speaker information is certainly made by applying *incremental* adaptation after each utterance. Adaptation methods can only use the results of an automatic labeling of the previous utterance(s); thus we are dealing with *unsupervised* adaptation. Since our speech recognition system is based on semi–continuous Hidden Markov Models (SCHMM), we concentrate on adaptation of the codebook parameters which offer good possibilities for fast adaptation.

A number of investigations [6, 8] have shown that with little adaptation data, good results are achieved by MAP reestimation of the codebook mean vectors. Also, a combination of a linear codebook transform with the MAP reestimation has proven to perform better than either of the two approaches alone [12, 8]. Therefore, we chose to investigate each approach separately first, and then to combine the optimized methods.

A common problem of both ML and MAP adaptation approaches is that the resulting estimation formulas have a relatively high computational cost due to the evaluation of high–dimensional Gaussian densities. We therefore investigated how the estimation could be simplified and found a modification, which is based on a theoretical consideration and at the same time speeds up the computation rapidly.

The rest of this paper is organized as follows: In section 2 we shortly review the ML estimation of linear codebook transforms. In section 3 we introduce a modified Bayesian estimation which will be called *conservative estimation*. Section 4 treats the issues related to the combination of the two adaptation schemes in the scenario of unsupervised and incremental adaptation. Experimental results are presented in section 5.

2 ACOUSTIC ADAPTATION

Acoustic adaptation methods attempt to compensate for external influences on the speech signal by performing a transformation of the feature space, or accordingly, of the codebook densities. The idea to perform acoustic adaptation by estimating a codebook transformation with a Maximum–Likelihood (ML) approach was first presented by [1] and has been applied to several kinds of transformations [2, 7, 12]. The transformation parameters Θ are obtained by maximizing the likelihood of observing the adaptation sample X:

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \mathrm{p}(X|\Theta). \tag{1}$$

The most general transformation which has been investigated so far is a linear transformation of the codebook densities (means *and* covariances). However, there is no closed form solution for the estimation unless the HMM–system works with *diagonal* covariance matrices [2].

Yet a solution is possible if only the means are transformed. The goal is to estimate a transformation matrix $\underline{A}$ and a translation vector $\underline{b}$ which transform the means of the K codebook densities $\mathcal{N}(\underline{x}|\underline{m}_k, \underline{S}_k)$ according to $\hat{\underline{m}}_k = \underline{A}\underline{m}_k + \underline{b}$. This analysis has been carried out by [7]; the estimation requires solving the following system of N linear equations, where N is the number of coefficients that make up the codebook transformation:

$$\sum_{k=1}^{K} n_k \underline{S}_k^{-1} \underline{A}\underline{m}_k \underline{m}_k^{\top} + n_k \underline{S}_k^{-1} \underline{b}\underline{m}_k^{\top} = \sum_{k=1}^{K} n_k \underline{S}_k^{-1} \underline{\mu}_k \underline{m}_k^{\top}$$

The work presented in this paper was partly supported by the DFG (German Research Foundation) under contract number 810 830-0.

$$\sum_{k=1}^{K} n_k \underline{S}_k^{-1} \underline{A} \underline{m}_k + n_k \underline{S}_k^{-1} \underline{b} = \sum_{k=1}^{K} n_k \underline{S}_k^{-1} \underline{\mu}_k. \qquad (2)$$

Although a more compact notation was used in [7], we prefer to write the linear equations in the form of a matrix and a vector equation, since it visualizes the structure of the equation system. The variables n_k and $\underline{\mu}_k$ are calculated as in [2] from the observation sequence via Baum–Welch or Viterbi algorithm.

This method for acoustic adaptation has the advantage that the transformation matrix can be restricted to any number of coefficients according to the expected amount of adaptation data and the allowed computation time. We have developed an efficient computation scheme for the coefficients of the linear equation system which guarantees that no computation is done more than once.

3 PHONE SPECIFIC ADAPTATION USING MODIFIED BAYESIAN ESTIMATION

In contrast to acoustic adaptation, phone specific adaptation methods perform an individual reestimation of codebook densities or even HMM parameters. Bayesian adaptation has received a great deal of attention since Gauvain and Lee [5] developed formulas to adapt *all* parameters of continuous density HMMs. This is certainly the case because the method has optimal properties and leads to significant improvements. In cases where adaptation data is sparse, usually only the codebook mean vectors are adapted by

$$\widehat{\underline{m}}_k = \frac{\tau_k \underline{m}_k + \sum_{t=1}^{T} \zeta_t(k)\, \underline{x}_t}{\tau_k + \sum_{t=1}^{T} \zeta_t(k)} \qquad (3)$$

This reestimation formula is very intuitive, because it is basically an interpolation between a weighted mean of the observations $\underline{x}_t$ and an a priori vector $\underline{m}_k$, which can be chosen as the mean vector from the speaker independent codebook. The parameter τ_k controls the adaptation speed, while $\zeta_t(k)$ is computed via

$$\zeta_t(k) = \mathrm{p}(\omega_t = k | \underline{x}, \underline{\lambda}) = \frac{c_{ik}\, \mathcal{N}(\underline{x}_t | \underline{m}_k, \underline{S}_k)}{\sum_{l=1}^{K} c_{il}\, \mathcal{N}(\underline{x}_t | \underline{m}_l, \underline{S}_l)}, \qquad (4)$$

if the optimal state sequence with states i is computed by the Viterbi algorithm. Here, c_{ik} denotes the output probability for codebook class k in state i. It is easy to see that the evaluation of the high dimensional normal densities for every observation vector is computationally very demanding.

An issue that has not been addressed yet in the context of Bayesian (MAP) adaptation for SCHMM is that the optimality of the codebook–HMM combination is violated when adapting the codebook alone while leaving the HMM state output probabilities unchanged. This has been reported for Maximum Likelihood (ML) adaptation [9], but as the difference between ML and MAP estimation lies only in the chosen a priori distribution, the same problem arises in the MAP case. The estimation according to equations (3) and (4), which we will call *conventional estimation* in the following, has the effect that the "'phonetic meaning"' of the codebook classes for the HMM is changed. This effect is best illustrated by the following example: Consider a speaker who typically pronounces an /a/ like an /o/. Due to the evaluation of the Gaussian densities in equation (4), an observation of this vowel will only give a small contribution to the reestimation of the codebook densities that represent the class /a/, but will contribute strongly to the reestimation of the densities that represent the class /o/. In normal HMM training, this would be compensated by the reestimation of the state output probabilities; however, in adaptation this step is not possible because of the limited size of the observation sequence. Thus, the codebook–HMM combination is no longer optimal.

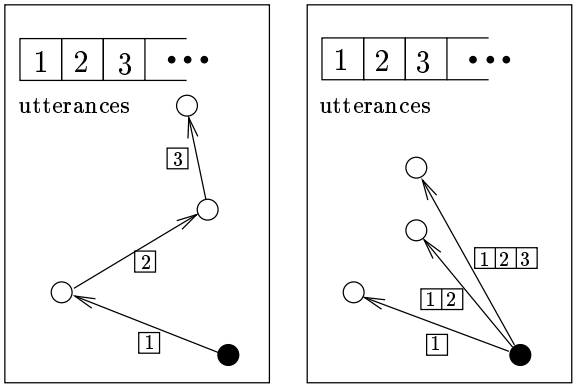

Figure 1. Illustration of incremental adaptation schemes

We can avoid these problems by introducing a modified estimation

$$\zeta_t(k) = c_{ik}. \qquad (5)$$

By omitting the normal densities, we ensure that the estimation changes the codebook densities representing the class /a/ as was intended. This modification was proposed for ML adaptation in [9], but we can obviously apply it to MAP adaptation with the same desired effect. The method has been reported to give significantly better results in ML adaptation than the conventional estimation [9].

Although we have motivated the modified method from a theoretical point of view, it also offers some very desirable properties for practical use. Most importantly, it needs much less computation time than the conventional estimation and is therefore much more suited for use in real–time applications. A second advantage shows up when it is combined with the estimation of a linear codebook transform in an incremental adaptation scenario as shown in Figure 2. This is explained in the next section.

4 COMBINING ACOUSTIC AND PHONE SPECIFIC ADAPTATION

While acoustic adaptation attempts to reduce variations that have an influence on the whole feature space, e.g. causing a shift or rotation of all feature vectors, phone specific adaptation aims to cover individual pronunciations by adapting each codebook class separately. Thus, combining both approaches should lead to a further improvement since they handle different sources of speaker variation.

In incremental adaptation, we can basically distinguish between two ways of using the collected adaptation data which are illustrated in Figure 1. One way is to use only the current utterance to reestimate the most recently adapted codebook. The drawback of this method is that if an utterance is rather short, the estimation is unreliable and may give bad results.

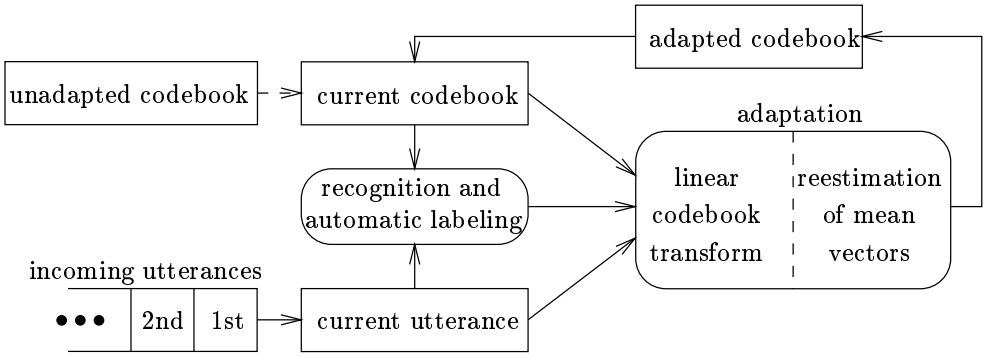

Figure 2. Illustration of unsupervised incremental adaptation

The alternative is to always adapt the speaker *independent* codebook using the whole adaptation sample collected so far. Depending on the adaptation method, however, this approach may cause problems if the computation must be done for the whole sample every time. This is what happens when we combine the conventional acoustic and phone specific adaptation schemes: the acoustic adaptation can re–use the values $\zeta_t(k)$ computed from previously observed samples because the adaptation is always performed based on the speaker independent codebook. However, the phone specific adaptation is then based on the acoustically adapted codebook, which is different after every new observation. Thus, the values $\zeta_t(k)$ have to be recomputed for the *whole* sample, which is prohibitive in a real–time application.

If we now consider the proposed conservative estimation in the same scenario, we see that the values $\zeta_t(k) = c_{ik}$ do *not* depend on the previously adapted codebook. Thus we can re–use the values $\zeta_t(k)$ from the previously observed samples and need to compute only those of the most recent observation.

5 EXPERIMENTAL RESULTS

A series of experiments has been carried out to evaluate the performance of the suggested methods under realistic conditions. We only give a short description of our speech recognizer here; a more detailed description can be found in [4].

For the results presented in this paper, a short time analysis of the speech signal was performed every 10 ms, yielding a 24–dimensional feature vector that consists of twelve cepstral coefficients and their first order derivatives. Word recognition was based on semi–continuous Hidden Markov Models using polyphone models as subword units [10] and a codebook af 256 classes with full covariance matrices. We performed a one-pass recognition using a bigram language model and skipped the second pass that uses higher order polygram language models, because our aim was to compare the improvement achieved by adapting the acoustic models.

We used two data bases that contain sentences from different applications. One test set (T1) is part of the EVAR sample collected at our instistute, which contains real dialogs with our train timetable information system (cf. section 1). The test set comprises 234 dialogs; an average dialog consists of ten utterances which amount to a total of only some 30 seconds of recorded speech per speaker.

The second test set (T2) is taken from the data base of dialogs in the VERBMOBIL project [11]. These are dialogs between two humans who try to arrange an appointment. The test set contains utterances of eight speakers with a total average length of roughly 1 1/2 minutes, split into ten utterances on the average. The word error rate (= 100%−word accuracy) of speaker independent recognition is 24% on T1 and 46% on T2, which means that the automatic labeling procedure produces a lot of wrong labels.

We also used a small validation sample V1 to perform some preliminary experiments and to adjust the parameters of the adaptation methods. V1 is taken from the EVAR dialogs, but is disjunct from T1. No optimization was done on the test samples.

First, we compare the runtime of the different adaptation methods. Both acoustic and phone specific adaptation comprise an estimation step which is identical for both adaptation methods, and the computation of the new mean vectors. The estimation step consists basically of a weighted summation over the observed feature vectors $\sum_{t=1}^{T} \zeta_t(k)\, \underline{x}_t$; so its computation time depends on the length of the adaptation sample. We measured the runtimes using a sample of 1000 frames equalling ten seconds of speech.

Table 1 shows the runtimes for the single computation steps on a HP 9000/735. We see that conventional estimation requires about 15 seconds of computation time, whereas conservative estimation is 18 times faster, taking less than one second. It is also worth noting that the complexity of conventional estimation depends quadratically on the dimension of the feature space, while conservative estimation is independent of it. Comparing the different adaptation methods, we see that phone specific adaptation is very fast, whereas the estimation of a large transformation matrix for acoustic adaptation is obviously prohibitive in a real–time application.

In a first series of experiments we investigated the use of acoustic adaptation by estimation of a linear transform. Since our preliminary experiments showed that estimating a full 24×24 transformation matrix consumes too much computation time, it was necessary to reduce the number of parameters. The dominating part of the computation is the solution of a linear equation system with N parameters (equation 2) that has a complexity of $\mathcal{O}(N^3)$.

There are several possible ways of reducing the number of parameters in the computation. We found that a good compromise is to estimate a full linear transformation for the *stationary* features, which are the first twelve features in our feature vectors. For the other features, only the translation parameters are estimated. We also applied a thresholding rule which keeps the parameter values in a

	conventional estimation	conservative estimation	acoustic adaptation 24×24–dim	acoustic adaptation 12×12–dim	phone specific adaptation
time in sec	14.4	0.84	105	3.14	0.02

Table 1. Runtimes of the different computations. Each adaptation method consists of an estimation step and an adaptation step.

word error rates in %	T1	T2
no adaptation	22.89	46.03
acoustic adaptation conventional estim.	22.01	43.39
phone specific adaptation conventional estim., $\tau = 5$	22.24	41.75
phone specific adaptation conservative estim., $\tau = 15$	22.04	43.99
combined acoustic and phone specific adaptation	21.90	41.10
improvement in %	4.3	10.7

Table 2. Experimental results for the different adaptation methods.

reasonable range.

These constraints were developed in the experiments on the validation sample V1. They resulted in a 4% reduction of the word error rate on T1 and a 6% reduction on T2. Table 2 shows those results; adaptation was performed on the whole sample from each speaker.

In a second series of experiments, we evaluated the phone specific adaptation scheme for conventional and conservative estimation. An unsatisfactorily solved problem is still the choice of the prior parameters in Bayesian adaptation; since we only adapt the codebook means, we need to choose only the parameters τ_k. It is possible to estimate each τ_k separately using the method of moments [6], but this is very time consuming since it requires the training of speaker dependent HMMs from large samples. We chose to estimate a common parameter $\tau = \tau_k$ for all codebook classes using the validation sample; we only distinguish between $\tau_{convent}$ for conventional and $\tau_{conserv}$ for conservative estimation.

The results (Table 2) show that conservative estimation performed better than conventional estimation on T1, giving a 4% improvement. On the other hand, conventional estimation gave better results for T2 with a 10% reduction of error rate. This behaviour may be due to the greater length of the adaptation samples in T2; also, the parameter value $\tau = 15$ may not be optimal for the VERBMOBIL application since the validation sample is taken from the EVAR dialogs.

Finally, we combined the linear codebook transform with the Bayesian adaptation methods and applied it in the scenario of incremental adaptation as illustrated in Figure 2. Note that each adapted codebook is first used for the recognition of the *following* utterance, so the first utterance of a speaker is always recognized with the unadapted system only. It should be stressed that these are exactly conditions as they appear in a real–time application. We observed a 4% word error reduction on T1 and an 11% reduction on T2.

REFERENCES

[1] S. J. Cox and J. S. Bridle. Unsupervised Speaker Adaptation by Probabilistic Spectrum Fitting. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 294–297, Glasgow, 1989.

[2] V. Digalakis, D. Rtischev, and L. Neumeyer. Speaker Adaptation using Constrained Estimation of Gaussian Mixtures. *IEEE Trans. on Speech and Audio Processing*, 3(5):357–365, 1995.

[3] W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E. Schukat-Talamazzini. A Spoken Dialogue System for German Intercity Train Timetable Inquiries. In *Proc. European Conf. on Speech Technology*, pages 1871–1874, Berlin, 1993.

[4] F. Gallwitz, E. Schukat-Talamazzini, and H. Niemann. Integrating Large Context Language Models into a Real Time Word Recognizer. In N. Pavesic and H. Niemann, editors, *3rd Slovenian–German and 2nd SDRV Workshop*. Faculty of Electrical and Computer Engineering, University of Ljubljana, Ljubljana, Apr. 1996.

[5] J. Gauvain and C. Lee. Bayesian Learning of Gaussian Mixture Densities for Hidden Markov Models. *Proc. DARPA Speech Natural Language Workshop*, pages 272–277, 1992.

[6] Q. Huo, C. Chan, and C. Lee. Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition. *IEEE Trans. on Speech and Audio Processing*, 3(5):334–345, 1995.

[7] C. J. Leggetter and P. C. Woodland. Flexible Speaker Adaptation using Maximum Likelihood Linear Regression. In *ARPA SLT Workshop*, pages 110–115, 1995.

[8] L. Neumeyer, A. Sankar, and V. Digalakis. A Comparative Study of Speaker Adaptation Techniques. In *Proc. European Conf. on Speech Technology*, pages 1127–1130, Madrid, 1995.

[9] S. Rieck, E. Schukat-Talamazzini, and H. Niemann. Speaker Adaptation using Semi-Continuous Hidden Markov Models. In *Proc. 11th IAPR Int. Conf. on Pattern Recognition*, volume III, pages 541–544, The Hague, Netherlands, 1992.

[10] E. Schukat-Talamazzini, T. Kuhn, and H. . Niemann. Speech Recognition for Spoken Dialog Systems. In H. Niemann, R. De Mori, and G. Hahnrieder, editors, *Progress and Prospects of Speech Research and Technolog y*, number 1 in Proceedings in Artificial Intelligence, pages 110–120. Infix, 1994.

[11] H. Tillmann and B. Tischer. Collection and Exploitation of Spontaneous Speech Produced in Negotiation Dialogues. In *ESCA Workshop on Spoken Language Systems*, pages 217–220, Vigsø, June 1995.

[12] Y. Zhao. An Acoustic-Phonetic-Based Speaker Adaptation Technique for improving Speaker-Independent Continous Speech Recognition. *IEEE Trans. on Speech and Audio Processing*, 2(3):380–394, 1994.