Methods of Unsupervised Adaptation in Online Speech Recognition

Lukáš Machlica, Zbyněk Zajíc, Aleš Pražák

University of West Bohemia in Pilsen,
Faculty of Applied Sciences, Department of Cybernetics,
Univerzitní 22, 306 14 Pilsen
machlica@kky.zcu.cz, zzajic@kky.zcu.cz, aprazak@kky.zcu.cz

Abstract

This paper deals with adaptation techniques based on maximum likelihood linear transformations, which are well suited for the task of on-line recognition. When transcriptions are available before the system starts running, we are speaking about supervised adaptation. In unsupervised adaptation the transcriptions have to be computed in the first pass of the recognition process. This is often the case in on-line recognition, where data are gathered continuously. Because the system does not work perfectly it is suitable to assign a certainty factor (CF) to each of the transcriptions. Only data that transcriptions have high CF are used for the adaptation.

In the on-line recognition, the adaptation (in the sense of updating transformation formulas) has to be performed iteratively whenever the amount of recognized data reaches the prespecified level. When small amount of adaptation data is available, it is suitable to involve regression trees to cluster similar model parameters. It is quite useful to adapt both speech and silence parameters. Because speech and silence have very different characteristics, we have separated them into two different clusters. Presented methods have been tested on short term recordings and results have proved the suitability of the proposed approach.

1. Introduction

After twenty years of intensive research in the field of the speech recognition, the technology has become usable in common applications. In this paper we focus mainly on adaptation techniques in the on-line recognition of speech [1],[2].

The Hidden Markov Model (HMM) with output probabilities described by Gaussian Mixture Models (GMMs) has been proved as an efficient tool in the speech recognition [3]. To train the HMM, it is necessary to have large amount of data from many speakers. The final model, denoted as Speaker Independent (SI), is able to recognize speech from any speaker. When the speakers identity is known, we could acquire additional lowering of the error rate by using a model trained on the data from a particular speaker. Such a model is called the Speaker Dependent (SD) model. The main problem by the construction of the SD model is the need of a large database of utterances from one speaker. This problem is in praxis often non-solvable. The solution is provided by the adaptation of an acoustic model, as described in Section 2. The choice of the adaptation technique depends on the actual problem, in systems for on-line recognition it is restricted mainly by the time consumption of adopted methods. Thus, in our system we have utilized linear transformations based on Maximum Likelihood (ML), where a transformation is computed for all the parameters in a given cluster – see Section 3. Such an approach can be applied in situations when only small amount of adaptation data are at hand. As the characteristics of speech and silence are very different, they should be considered separately. Therefore, we proposed a simple division of speech and silence parameters of the acoustic model as described in Section 3.2.

In on-line recognition the reference transcriptions of adaptation data are not available. In this case we speak about unsupervised adaptation mentioned in Section 4. Problems related to the on-line adaptation are summarized in Section 5. The detailed description of our system with proposed experiments can be found in the last part of the paper (Section 6). The results prove the suitability of adopted methods and adaptation techniques improve the recognition.

2. Adaptation techniques

The difference between the adaptation and ordinary training methods stands in the prior knowledge about the distribution of model parameters, usually derived from the SI model [4]. The adaptation adjusts the model so that the probability of the adaptation data would be maximized. This is equivalent to

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\boldsymbol{\lambda})p(\boldsymbol{\lambda}), \qquad (1)$$

where $p(\boldsymbol{\lambda})$ stands for the prior information about the distribution of the vector $\boldsymbol{\lambda}$ containing model parameters, $\boldsymbol{O}^i = \{\boldsymbol{o}_1^i, \boldsymbol{o}_2^i, \ldots, \boldsymbol{o}_T^i\}, i = 1, \ldots, E$ is the sequence of T feature vectors related to one speaker, $\boldsymbol{\lambda}^*$ is the best estimation of parameters of the SD model. We will focus now on HMMs with output probabilities of states represented by GMMs. GMM of the $j-th$ state is characterized by a set $\boldsymbol{\lambda}_j = \{\omega_{jm}, \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}\}_{m=1}^{M_j}$, where M_j is the number of mixtures, ω_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{C}_{jm}$ are weight, mean and variance of the $m-th$ mixture, respectively. Let us define some statistics used by the description of adaptation techniques:

$$\gamma_{jm}(t) = \frac{\omega_{jm} p(\boldsymbol{o}(t)|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}(t)|jm)} \qquad (2)$$

stands for the $m-th$ mixtures' posterior,

$$c_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t) \qquad (3)$$

is the soft count of mixture m,

$$\varepsilon_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}(t)}{\sum_{t=1}^{T} \gamma_{jm}(t)} \qquad (4)$$

represents the average of features in frames which align to mixture m in the j-th state and note that $\boldsymbol{\sigma}_{jm}^2$ = diag$(\boldsymbol{C}_{jm})$ is the diagonal of the covariance matrix $\boldsymbol{C}_{jm}$.

The most know adaptation methods are Maximum A-posteriori Probability (MAP) [5] and linear transformations based on the Maximum Likelihood (ML) [6]. The first approach is mainly used in cases when big amount of training data is available, because each of model parameters demands sufficient amount of data to be adapted. The latter method has the advantage that model parameters are clustered in an convenient way (see Section 3), hence several parameters share the same transformation. Thus less amount of data is needed.

2.1. Linear Transformations based on Maximum Likelihood

These methods are focused on the adaptation of means and variances of GMMs (the mixture weights are of no interest) and are based on the minimization of the auxiliary function [7]:

$$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}}) = const - \frac{1}{2}\sum_{jm}\sum_{t}\gamma_{jm}(t)(const_{jm}+ \\ + \log|\bar{\boldsymbol{C}}_{jm}| + (\boldsymbol{o}(t)-\bar{\boldsymbol{\mu}}_{jm})^{T}\bar{\boldsymbol{C}}_{jm}^{-1}(\boldsymbol{o}(t)-\bar{\boldsymbol{\mu}}_{jm}))\ . \quad (5)$$

The number of available model parameters is reduced via clustering (see Section 3) of similar model components. The transformation is the same for all of the parameters from the same cluster $K_n, n = 1,\ldots,N$. Hence, less amount of adaptation data is needed.

Further, we distinguish unconstrained and constrained transformations. In the unconstrained case, means and variances from the particular cluster are adapted utilizing two distinct matrices, whereas in the constrained case the same matrix is used for both.

2.1.1. Maximum Likelihood Linear Regression (MLLR)

MLLR can be regarded as an unconstrained adaptation, means and variances from the same cluster K_n are transformed separately according to formulas:

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{A}_{(n)}\boldsymbol{\mu}_{jm} + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}_{jm}\ , \quad (6)$$

$$\bar{\boldsymbol{C}}_{jm} = \boldsymbol{H}_{(n)}\boldsymbol{C}_{jm}\boldsymbol{H}_{(n)}^{T}\ , \quad (7)$$

where $\bar{\boldsymbol{\mu}}_{jm}$, $\bar{\boldsymbol{C}}_{jm}$ are the new adapted mean and covariance of the $m-th$ mixture in the $j-th$ state of the HMM, respectively. $\boldsymbol{A}_{(n)}$ is the regression matrix related to the cluster K_n, $\boldsymbol{b}_{(n)}$ is the additive vector and $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$, $\boldsymbol{\xi}_{jm} = [\boldsymbol{\mu}_{jm}^{T}, 1]^{T}$ is the original mean extended by 1 and $\boldsymbol{H}_{(n)}$ represents the transformation matrix for the covariance.

First, let's focus on the transformation matrix of means. It can be shown [8] that the part of the function (5) that changes with the current transform $\boldsymbol{W}_{(n)}$ is:

$$Q_{W_{(n)}} = \boldsymbol{w}_{(n)i}^{T}\boldsymbol{k}_{(n)i} - 0.5\boldsymbol{w}_{(n)i}^{T}\boldsymbol{G}_{(n)i}\boldsymbol{w}_{(n)i}\ , \quad (8)$$

where the column vector $\boldsymbol{w}_{(n)i}, i = 1,\ldots,I$ is the transpose of the i-th row of $\boldsymbol{W}_{(n)}$ and $I = \dim(\boldsymbol{o})$,

$$\boldsymbol{k}_{(n)i} = \sum_{jm\in K_n}\frac{c_{jm}\boldsymbol{\xi}_{jm}\varepsilon(\boldsymbol{o})_{jm}(i)}{\sigma_{jm}^{2}(i)}\ , \quad (9)$$

and

$$\boldsymbol{G}_{(n)i} = \sum_{jm\in K_n}\frac{c_{jm}\boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^{T}}{\sigma_{jm}^{2}(i)}\ . \quad (10)$$

And finally, after maximization of equation (8) with respect to the row of the mean transformation matrix $\boldsymbol{W}_{(n)}$ we obtain:

$$\frac{\partial Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})}{\partial\boldsymbol{W}_{(n)}} = 0 \Rightarrow \boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1}\boldsymbol{k}_{(n)i}\ . \quad (11)$$

Similar procedure can be utilized also for the derivation of the transformation matrix $\boldsymbol{H}_{(n)}$ for covariances, and can be found in e.g. [7],[6].

2.1.2. Feature Maximum Likelihood Linear Regression (fMLLR)

Compared to the MLLR case, fMLLR is a constrained adaptation. Hence, means and covariances from the same cluster K_n are transformed with the same matrix, what gives us the ability to transform directly the features $\boldsymbol{o}_t$ instead of the model parameters $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{C}_{jm}$ [8]. The feature vectors are transformed according to the formula

$$\bar{\boldsymbol{o}}_t = \boldsymbol{A}_{(n)}\boldsymbol{o}_t + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}(t)\ , \quad (12)$$

where $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ stands for the transformation matrix corresponding to the $n-th$ cluster K_n, $\boldsymbol{\xi}(t) = [\boldsymbol{o}_t^{\mathrm{T}}, 1]^{\mathrm{T}}$ represents the extended feature vector. In analogy with the previous section, it is possible to rearrange the auxiliary function (5) into the form [8]

$$Q_{\boldsymbol{W}_{(n)}}(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}}) = \log|\boldsymbol{A}_{(n)}| - \sum_{i=1}^{I}\boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{k}_i - 0.5\boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{G}_{(n)i}\boldsymbol{w}_{(n)i}, \quad (13)$$

where

$$\boldsymbol{k}_{(n)i} = \sum_{m\in K_n}\frac{c_m\mu_{mi}\varepsilon_m(\boldsymbol{\xi})}{\sigma_{mi}^{2}}\ , \quad (14)$$

$$\boldsymbol{G}_{(n)i} = \sum_{m\in K_n}\frac{c_m\varepsilon_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}})}{\sigma_{mi}^{2}}\ , \quad (15)$$

$$\varepsilon_m(\boldsymbol{\xi}) = \left[\varepsilon_m^{\mathrm{T}}(\boldsymbol{o}), 1\right]^{\mathrm{T}}\ , \quad (16)$$

and

$$\varepsilon(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}})_m = \begin{bmatrix} \varepsilon_m(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) & \varepsilon_m(\boldsymbol{o}) \\ \varepsilon_m(\boldsymbol{o})^{\mathrm{T}} & 1 \end{bmatrix}\ . \quad (17)$$

To find the solution of equation (13) we have to express $\boldsymbol{A}_{(n)}$ in terms of $\boldsymbol{W}_{(n)}$, e.g. use the equivalency $\log|\boldsymbol{A}_{(n)}| = \log|\boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{v}_{(n)i}|$, where $\boldsymbol{v}_{(n)i}$ stands for transpose of the $i-th$ row of cofactors of the matrix $\boldsymbol{A}_{(n)}$ extended with a zero in the last dimension. After the maximization of the auxiliary function (13) we receive

$$\frac{\partial Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})}{\partial\boldsymbol{W}_{(n)}} = 0 \Rightarrow \boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1}\left(\frac{\boldsymbol{v}_{(n)i}}{\alpha_{(n)}} + \boldsymbol{k}_{(n)i}\right)\ , \quad (18)$$

where $\alpha_{(n)} = \boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{v}_{(n)i}$ can be found as the solution of the quadratic function

$$\beta_{(n)}\alpha_{(n)}^{2} - \alpha_{(n)}\,\boldsymbol{v}_{(n)i}^{\mathrm{T}}\boldsymbol{G}_{(n)i}^{-1}\boldsymbol{k}_{(n)i} - \boldsymbol{v}_{(n)i}^{\mathrm{T}}\boldsymbol{G}_{(n)i}^{-1}\boldsymbol{v}_{(n)i} = 0\ , \quad (19)$$

where

$$\beta_{(n)} = \sum_{m\in K_n}\sum_{t}\gamma_m(t)\ . \quad (20)$$

Two different solutions $\boldsymbol{w}_{(n)i}^{1,2}$ are obtained, because of the quadratic function (19). The one that maximizes the auxiliary function (13) is chosen. The log likelihood for fMLLR can be computed as

$$\log \mathcal{L}\left(\boldsymbol{o}_t|\boldsymbol{\mu}_m, \boldsymbol{C}_m, \boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}\right) =$$
$$= \log \mathcal{N}\left(\boldsymbol{A}_{(n)}\boldsymbol{o}_t + \boldsymbol{b}_{(n)}; \boldsymbol{\mu}_m, \boldsymbol{C}_m\right) + 0.5 \log |\boldsymbol{A}_{(n)}|^2 \, . \tag{21}$$

The estimation of $\boldsymbol{W}_{(n)}$ is an iterative procedure. Matrices $\boldsymbol{A}_{(n)}$ and $\boldsymbol{b}_{(n)}$ have to be correctly initialized first, e.g. $\boldsymbol{A}_{(n)}$ can be chosen as a diagonal matrix with ones on the diagonal and $\boldsymbol{b}_{(n)}$ can be initialized as a zero vector. The estimation ends when the change in parameters of transformation matrices is small enough (about 20 iterations are sufficient [8]).

3. Similar components clustering

The benefit of MLLR like methods is given by the possibility to cluster similar parameters (mixture components) of the model. Many clustering methods were already developed and successfully tested [9]. The number of clusters depends on the amount of adaption data, which is in our case a-priori unknown. Hence, hierarchical clustering approach, where the number of classes is established when all the data were introduced, is most suitable. Very common hierarchical methods utilized in the adaptation are regression trees (RT). RT is designed in advance, but the exact number of final classes will be determined according to the amount of adaptation data. An example is depicted in Figure 1. RT can be based e.g. upon phonetic knowledge [10] or distances in the acoustic space [12]. In this work we focus on the latter approach.

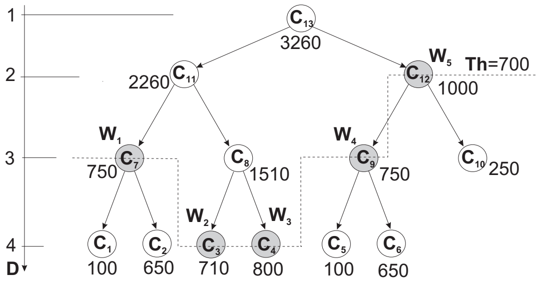

Figure 1: Example of a binary regression tree. The numbers assigned to nodes are the actual occupation counts of nodes (clusters). Nodes C_1 and C_2 have occupations lesser then the occupation threshold $Th = 700$, therefore for all the components of mixtures located in C_1 and C_2 will be used the transformation defined for node C_7. D denotes the depth in the tree.

3.1. Regression Tree (RT)

The model parameters, which are close in the acoustic space, are clustered utizing a given criterion, e.g. Euclidean distance as done in [12] or maximizing the likelihood of the adaptation data as done in [11]. The construction of RT, considering a HMM with output probabilities described by GMMs, can be done as

- each of the final leaves (clusters) of the tree contains just a single mixture component,
- leaves are merged together according to the Euclidean distance of GMM means,

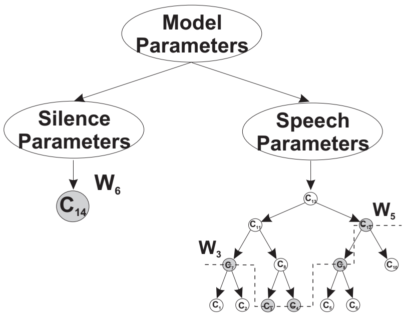

Figure 2: An extension of the regression tree depicted in Figure 1, where an extra node C_{14} (sil-node) was added exclusively for silence parameters of the HMM.

- finally, the root node containing all components from all mixtures is obtained.

The final set of classes is established during the adaptation process according to the amount of adaptation data in each of the clusters (*cluster occupation*) in RT utilizing an empirical threshold Th (see Figure 1). The amount of adaptation data in a cluster is determined as the sum of all soft counts (defined in (3)) of all mixtures belonging to this cluster. Hence, the occupation of the $n - th$ cluster K_n is given as $occup(n) = \sum_{jm \in K_n} c_{jm}$. The number of transformations equals the cardinality of the final set of classes. If considering Figure 1 and the occupation threshold of a cluster set to $Th = 700$, the final set would contain five classes C_3, C_4, C_7, C_9, C_{12}. Each of the transformation matrices is computed only from data aligning to mixtures of the given class.

3.2. Adaptation of Silence

If regression trees based on Euclidean distance as defined in Section 3.1 are used, the speech and silence segments usually share the same cluster. For clarification, silence segments are those parts in the acoustic signal with absence of the speaker's voice. In cases where only adaptation data containing small amount of silence frames are available, a situation may occur that the states of silence, presented in the HMM, are bended toward the speech data. This can happen mainly when the channel of training data of the SI model and the channel of actual speaker data significantly differ. Hence, the silence segments can be more often recognized as speech, thus the error rates increase. Generally, the speech and silence are so much different that the idea to separate them is straightforward.

In order to solve the stated problem, another *sil-node* containing all the mixtures belonging to states in the HMM that represent the silence has to be involved as depicted in Figure 2. At first, all the HMM states related to silence are set apart (they are associated to the sil-node) and the remaining set of states is used to construct a regression tree based upon the rules specified in the Section 3.1.

It should be mentioned that the adaptation is performed only when sufficient amount of data is available for both silence and speech parameters. Details will be discussed in Section 6.1.

4. Unsupervised Adaptation

In the case when the acoustic model is build upon a set of HMMs, where each HMM represents an elementary linguistic unit (e.g. monophone, triphone, syllable, etc.), a transcription of adaptation utterances of spoken speech is ncessary. Such a transcription has to be available before the adaptation process. If a reference transcription is at hand (e.g. obtained by an annotator) we speak about supervised adaptation. For an annotator, it is intractable to assign a phonetic label to each frame at each time. Usually, only beginning and end of a sentence are marked and the transcription of the sentence has to be aligned automatically (e.g. using Viterbi algorithm [4]).

In the case of unsupervised adaptation, transcriptions have to be computed in the first pass of the recognition process utilizing the not adapted SI model. Such a Recognition System (RS) replaces the role of the annotator. Because RS does not work perfectly, it is suitable to assign a Certainty Factor (CF) to each of the transcriptions. CF for any particular word sequence is extracted from the lattice and can be computed as in [13]. We use only the best path in the lattice. Only the data which transcriptions have high CF, greater than an empirically specified threshold, are used for the adaptation. Still some problems may occur. Even if the CF of a word is high, the boundaries (time labels) of the word can be inaccurate, because of low values of CF of neighborhood words. Hence, it is useful to take into account the left and right context of each word in the sense of CF. We are seeking for a sequence of three words, where each of them has a CF higher than the threshold and for adaptation we consider only the middle one. As an alternative for sequence of three words, mainly when such triplets occur only very rare, a lattice representation of each utterance may be used [14].

5. Adaptation in On-line Recognition

In the on-line recognition an unknown person speech has to be transcribed. At the beginning, a Speaker Independent (SI) model is used. The effort is to utilize the increasing amount of speaker utterances to improve the SI model using adaptation. There are several issues concerning the on-line adaptation.

The reference transcriptions are unavailable, therefore the unsupervised adaptation has to be employed (see Section 4). These can be done as a parallel process, where two problems are solved separately:

- recognition of the actual sentence,
- adaptation of the acoustic model according to the previous sentences, already recognized by RS.

Hence, the acoustic model is iteratively adapted so that the subsequent recognition becomes more accurate.

Next important point is the time consumption. Acoustic models involved in RS comprises huge amount of states with output probabilities represented by GMMs with lots of mixtures. Thus, the modification of such an acoustic model is unreasonable. Much more preferable is to transform directly the acoustic features, hence only a few transformation matrices has to be stored. A method satisfying these requirements is e.g. fMLLR introduced in Section 2.1.2 or their modifications presented in [15],[16].

Another question concerns the moment when transformation matrices should be updated again. Hence, when should be the SI model adapted for the first time and when should it be further updated. It is appropriate to wait until the increase of information is sufficient so that the newly formed transformation matrices are well-conditioned and the new iteration reasonably improves the recognition. The solution of the stated problem will be given in Section 6.1.

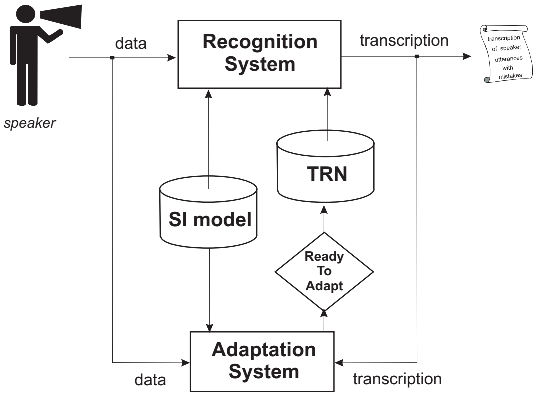

Figure 3: Scheme of the adaptation process in the on-line recognition. $SImodel$ represents a speaker independent acoustic model and TRN stands for a feature transformation that depends on the actual speaker. The block $ReadyToAdapt$ decides when a new transformation have to be computed.

6. Experiments

6.1. System Description

The analogue input speech signal is digitized at 44.1 kHz sampling rate and 16-bit resolution format. The aim of the front-end processor is to convert continuous acoustic signal into a sequence of feature vectors. We performed experiments with MFCC and PLP parameterizations, see [17] for methodology. The best results were achieved using 19 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features. Feature vectors are computed at the rate of 100 frames per second.

The acoustic model was trained on 100 hours of parliament speech records with manual transcriptions. We used 42 Czech phonemes. As the number of Czech triphones is too large, phonetic decision trees were used to tie their states. The recognition of Czech Parliament meetings works with 5 385 different HMM states of a speaker and a gender independent acoustic model. Note that only diagonal covariances are assumed.

The language models were trained on about 25M tokens of transcriptions of normalized Czech Parliament meetings (Chamber of Deputies only). The vocabulary size is almost 175 000 words including the names of parliament members in five classes for different grammatical cases. Class-based 2-gram language model was used for on-line recognition while confidence factors were computed using class-based 3-gram language model in real-time. The SRI Language Modeling Toolkit [18] was used for training.

Our on-line system is represented by a feedback connection of a Recognition System (RS) and an Adaptation System (AS), both of them operate in parallel (see Fig. 3). We are using fMLLR adaptation approach as described in the Section 2.1.2. Parameters of HMM are divided into two sets – for silence and for speech. Further, speech parameters are splitted using regression tree (RT) proposed in Section 3 utilizing Euclidean distance and occupation threshold $Th = 1000$.

The process of the on-line adaptation is as follows. An input speaker utterance is recognized using RS. At the beginning, the RS is represented only by the SI model missing any information about the actual speaker. The utterance and its output transcription, with a Certainty Factor (CF) assigned to each word, are used in AS to computie the adaptation statistics (see Section 2). Just reliably transcribed data are used for adaptation. Thus, AS utilizes only words that satisfy following constraints:

- the CF of such a word is greater then the pre-specified threshold $T_{CF} = 0.98$,
- the confident of adjacent left and right words is at least equal to T_{CF},

for details see Section 4. Note that the recognition of upcoming utterances is still running in parallel, whereas the adaptation process is performed.

In our system two thresholds for two node occupations (see Section 3.1) were set as silence and speech parameters were considered separately. The first threshold for silence parameters (sil-node – see Section 3.2) was set to $T_{sil} = 500$. As the adaptation process was performed iteratively (see Section 5), the second threshold T_{speech}, for speech parameters, had to be updated after each adaptation. The initial value of T_{speech}, used for the first adaptation pass, was chosen as $T_{speech}(0) = 1000$ and its further values were set according to the depth (D – see Figure 1) in RT. Hence $T_{speech}(k+1) = T_{speech}(k) * (2^{D_{\text{act}}} + 1)$, where $k = 1, \ldots, K$ represents the $k - th$ iteration and D_{act} is the actual depth in RT determined in dependence on the amount of adaption data:

$$D_{\text{act}} = \log_2 \frac{\sum_{\forall jm} c_{jm}}{Th} , \tag{22}$$

where c_{jm} was defined in (3). When both thresholds (T_{speech} and T_{sil}) were reached, hence a sufficient amount of (transcribed) adaptation data was accumulated ($ReadyToAdapt = True$ – see Figure 3), then the transformation matrices (TRNs) have been (re)computed.

After first transformation matrices have been computed, following utterances were recognized using the adapted model (SI model + TRNs). Thus, the system (and output transcriptions) become more accurate. New threshold T_{speech} was set and statistics were further accumulated till the next adaptation. The iterative adaptation ended when the maximal occupation of all possible nodes in RT was reached. We assume that no following adaptation (with unchanged RT) would increase the performance of the recognition system. However, in our tests such a situation never happened (see Section 6.2).

For our experiments we have used Intel Core 2 Duo, 2.40 GHz and 3 GB RAM. The recognition ran on both processors until the adaptation matrices were computed. In order to update the transformation matrices one of the processors was used, whereas the recognition was still running on the other processor. The time consumption of the on-line recognition is measured according to the real-time factor computed as

$$T_{real} = \frac{\text{time spent on recognition}}{\text{time duration of the speech recording}} . \tag{23}$$

6.2. Test Data

The experiments were focused mainly on situations where low amount of adaptation data is available. Two sets of testing data were prepared. The Set No.1 contained 10 parliament speakers with speech recorded directly from TV. Hence, same conditions were preserved as for data used to train the acoustic model. The Set No.2 contained another 10 speakers recorded in the office with completely different operating conditions (mainly channel dissimilarity) than training data. In both sets, each speaker was represented by 5-8 minutes of utterances. The adaptation process was iterative one (see Section 6.1). Enough data for the first adaptation pass were available after circa 3 minutes of speech and for the second pass after circa 6 minutes. Note that at most 6 minutes of speech were used for adaptation (rather less).

system	Corr[%]	Acc[%]
$SImodel$	89.05	86.37
$Adapt_{RT}$	89.49	87.31
$Adapt_{SIL+RT}$	89.86	87.35

Table 1: Correctness (Corr)[%] and Accuracy (Acc)[%] of transcribed words for set No.1.

system	Corr[%]	Acc[%]
$SImodel$	71.63	64.37
$Adapt_{RT}$	76.99	72.40
$Adapt_{SIL+RT}$	76.98	72.27

Table 2: Correctness (Corr)[%] and Accuracy (Acc)[%] of transcribed words for set No.2.

6.3. Results

The results of the experiment on set No.1, No.2 are shown in Table 1 and Table 2, respectively. The accuracy and correctness of the baseline system (recognition done utilizing only the SI model) can be found in the first row. The other columns contain results obtained by the system with the iterative adaptation using the fMLLR approach. The terms $Adapt_{RT}$ and $Adapt_{SIL+RT}$ denote systems without and with separation of speech and silence parameters of the HMM, respectively. In comparison to the baseline system, the adaptation methods increase both Correctness (Corr) and Accuracy (Acc) of recognized words.

Important factor in the on-line recognition is the time consumption (see (23)). We have measured the average time consumption of the recognition system based upon

- SI model: $T_{real} = 2.06$,
- SI model + TRNs: $T_{real} = 2.47$.

In order to further reduce the real-time factor, computer with more than two cores should be used.

7. Discussion

The results demonstrate improvement of the RS already for low amount of adaptation data. As could be anticipated and was proved by proposed experiments, the improvement in the recognition is much more significant in the case when channels in training and testing utterances differ (set No.2). Further, when the special node (sil-node) for silence parameters is involved (see Section 3.2), an slight improvement in the case of set No.1 can be observed. This is not the case for set No.2. However, because of the expected independence of silence and speech segments, the results remained basically unchanged.

8. Conclusion

In this paper methods for adaptation in task of on-line recognition were presented. Linear transformation methods based on maximum likelihood were discussed. In our system we have utilized in advance the fMLLR approach. The clustering of similar model components was divided into two separate tasks – speech vs. silence parameters, and the regression tree was involved. The on-line recognition demanded the unsupervised adaptation approach using certainty factor. In description of our system we assumed that there is no change in the speaker in the whole process of recognition. Hence, the change of speaker identity should be handled by the user. Such systems are well suited e.g. to replace the court reporter. In the future work, it would be convenient to extend the system with automatic speaker change detection.

9. Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, project No. 2C06020, by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707 and by the Ministry of Education of the Czech Republic, project No. MŠMT LC536.

10. References

[1] Bnhalmi, A., Kocsor, A., 2008. An On-line Speaker Adaptation Method for HMM-based Speech Recognizers, *Acta Cybernetica, vol. 18, pp. 379–390*.

[2] Barras,C., Meignier,S., Gauvain, J.L., 2004. Unsupervised Online Adaptation for Speaker Verification over the Telephone, *Speaker Odyssey, Toledo, pp. 157-160*.

[3] Rabiner, L.R., 1990. A tutorial on hidden Markov models and selected applications in speech recognition, *Readings in speech recognition, pp. 267-296, ISBN:1-55860-124-4*.

[4] Psutka, J., Müller, L., Matoušek, J., Radová, V., 2007. Mluvíme s počítačem česky, *Academia, Praha 2007, ISBN:80-200-1309-1*.

[5] Alexander, A., 2005. Forensic automatic speaker recognition using bayesian interpretation and statistical compensation for mismatched conditions, *Ph.D. thesis in Computer Science and Engineering, Indian Institute of Technology, Madras, pp. 27-29*.

[6] Ganitkevitch, J., 2005. Speaker Adaptation using Maximum Likelihood Linear Regression. *Rheinish-Westflesche Technische Hochschule Aachen, the course of Automatic Speech Recognition, www-i6.informatik.rwth-aachen.de/web/Teaching/Seminars/SS05/ASR*.

[7] Gales, M.J.F., 1997. Maximum Likelihood Linear Transformation for HMM-based Speech Recognition, *Tech. Report, CUED/FINFENG/TR291, Cambridge Univ.*.

[8] Povey, D., Saon, G., 2006. Feature and model space speaker adaptation with full covariance Gaussians, *Interspeech 2006, paper 2050-Tue2BuP.14*.

[9] Theodoridis, S., Koutroumbas, K., 1999. Pattern Recognition, *Academic Press*.

[10] Stolcke, A., Kajarekar, S., Ferrer, L., Shriberg, E., 2007. Speaker recognition with session variability normalization based on MLLR adaptation transforms, *IEEE International Conference on Spoken Language Processing, vol. 15, pp. 1987-1998*.

[11] Gales, M.J.F., 1996. The generation and use of regression class trees for MLLR adaptation, *Cambridge University Engineering Department*.

[12] Young, S., Evermann, G., Gales, M., and col., 2006. The HTK Book (for HTK 3.4). *User's manual, Cambridge University Engineering Department*.

[13] Uebel, L.F., Woodland, P.C., 2001. IMPROVEMENTS IN LINEAR TRANSFORM BASED SPEAKER ADAPTATION, *IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, pp. 49-52*.

[14] Uebel, L.F., Woodland, P.C., 2001. Speaker adaptation using lattice-based MLLR, *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition, Sophia Antipolis, France, pp. 57-60*.

[15] Varadarajan, B., Povey, D., Chu, S.M., 2008. Quick fmllr for speaker adaptation in speech recognition, *IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, pp. 4297-4300*.

[16] Lf,J., Gollan,Ch., Ney, H., 2008. Speaker Adaptive Training Using Shift-MLLR, *Interspeech, Brisbane, pp. 1701-1705*.

[17] Psutka, J., Mller, L., Psutka, J.V., 2001. Comparison of MFCC and PLP Parameterization in the Speaker Independent Continuous Speech Recognition Task, *EUROSPEECH 7th European Conference on Speech Communication and Technology*.

[18] Stolcke, A., 2002. SRILM - An Extensible Language Modeling Toolkit, *ICSLP 2002, 7th International Conference on Spoken Language Processing*.