A Direct Criterion Minimization based fMLLR via Gradient Descend

Jan Vaněk and Zbyněk Zajíc

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

Abstract. Adaptation techniques are necessary in automatic speech recognizers to improve a recognition accuracy. Linear Transformation methods (MLLR or fMLLR) are the most favorite in the case of limited available data. The fMLLR is the feature-space transformation. This is the advantage with contrast to MLLR that transforms the entire acoustic model. The classical fMLLR estimation involves maximization of the likelihood criterion based on individual Gaussian components statistic. We proposed an approach which takes into account the overall likelihood of a HMM state. It estimates the transformation to optimize the ML criterion of HMM directly using gradient descent algorithm.

Keywords: ASR, adaptation, fMLLR

1 Introduction

Nowadays, systems of speech recognition are based on Hidden Markov Models (HMMs) with output probabilities described mainly by Gaussian Mixture Models (GMMs) [1]. To recognize the speech from a recording one could train a Speaker Dependent (SD) model for each of the speakers present in the recording. However, this is in praxis often intractable because of the need of a large database of utterances coming from one speaker. Instead, so called Speaker Independent (SI) model is trained from large amount of data collected from many speakers, and subsequently, the SI model is adapted to better capture the voice of the talking person. Thus, a SD model is acquired.

More precisely, the adaptation adjusts the SI model so that the probability of the adaptation data would be maximized. Well known adaptation methods are Maximum A-posteriori Probability (MAP) technique [3] and Linear Transformations based on Maximum Likelihood (LTML), as model adaptation Maximum Likelihood Linear Regression (MLLR). In the ASR systems where the speaker changes quickly the adaptation of acoustic feature then updating of an acoustics model is less time consuming, such method is called feature Maximum Likelihood Linear Regression (fMLLR). In this paper we have chosen out of LTML based adaptations preferably the feature transformations which are well suited for on-line adaptation, see [12].

The classical approach to the estimation of the fMLLR approach using row-by-row estimation of the adaptation matrix. Data are accumulated with respect to individual Gaussians. In our proposed method a direct minimization of a criterion function is applied. Our criterion is based on likelihood of whole HMM states. The adaptation parameters are estimated via gradient descend method [13]. We used Newton's method with

diagonal Hessian matrix to speed-up a convergence of the estimation process. Moreover, we modified the ML criterion to be less sensitive to the phones length.

This paper is organized as follows. In Section 2 is described an idea of speaker adaptation. Particular techniques for feature adaptation, fMLLR approach, is presented in Section 3. The proposed approach for finding the fMLLR adaptation matrices using gradient techniques is discussed in Section 4. Experimental results are presented in Section 5.

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 [2]. The adaptation adjusts the model in order to maximize the probability of adaptation data. Hence, the new, adapted parameters can be chosen as

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

where $p(\boldsymbol{\lambda})$ stands for the prior information about the distribution of the vector $\boldsymbol{\lambda}$ containing model parameters, $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ 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 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.

The most know adaptation methods are Maximum A-posteriori Probability (MAP) [4] and Linear Transformations based on the Maximum Likelihood (LTML) [7]. The benefit of MAP technique is in the convergence of such adapted model SA into the model SD, but in the task of limited amount of adaptation data is inappropriate.

The advantage of LTML techniques over the MAP technique is that the number of available model parameters is reduced via clustering of similar model components [9]. The transformation is the same for all the parameters from the same cluster $K_n, n = 1, \ldots, N$. Hence, less amount of adaptation data is needed. In the extreme case, co called global adaptation, only one adaptation matrix for all model components is computed from all the adaptation data. The first of the methods introduced by Leggeter in [5] is known as Maximum Likelihood Linear Regression (MLLR) and was further investigated by Gales, who introduced feature MLLR (fMLLR). The main difference between these two approaches stands in the area of their interest. MLLR transforms means and covariances of the model, whereas fMLLR transforms directly the acoustic feature vectors. The MLLR method is out of our interest and the adaptation formulas can be found in [5].

3 Feature Maximum Likelihood Linear Regression (fMLLR)

The method is based on the minimization of the auxiliary function [7]:

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

where $\bar{\boldsymbol{o}}(t)$ represents the feature vector transformed according to the formula:

$$\bar{\boldsymbol{o}}_t = \boldsymbol{A}_{(n)} \boldsymbol{o}_t + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)} \boldsymbol{\xi}(t) \,, \tag{3}$$

where $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ stands for the transformation matrix corresponding to the $n-th$ cluster K_n and $\boldsymbol{\xi}(t) = [\boldsymbol{o}_t^{\mathrm{T}}, 1]^{\mathrm{T}}$ represents the extended feature vector.

The standard implementation of fMLLR (or other adaptation based on linear transformation) requires four steps [6]:

1. **Alignment** of the adaptation utterance to HMM states. This can be done by *forced-alignment(Vitterby algorithm)* or more time demanding but more accurate *forward-backward algorithm* [2]. Both approaches need transcription of adaptation utterance. This transcription can by done as reference transcription (supervised adaptation) or can by required from the first pass or ASR (unsupervised adaptation). The result of alignment is probability $p(\boldsymbol{o}(t)|jm)$ that feature $o(t)$ is generated by $m-th$ mixture of the $j-th$ state of the HMM. Posterior probability $\gamma_{jm}(t)$ of feature $o(t)$ is given as

$$\gamma_{jm}(t) = \frac{\omega_{jm} p(\boldsymbol{o}(t)|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}(t)|jm)} \tag{4}$$

2. **Computation**
of the soft count c_{jm} of mixture m and the first and the second statistics moment, $\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o})$ and $\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$, of features which align to mixture m in the j-th state of the HMM

$$c_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t) \tag{5}$$

is the soft count of mixture m,

$$\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t) \boldsymbol{o}(t)}{\sum_{t=1}^{T} \gamma_{jm}(t)} \,, \quad \boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t) \boldsymbol{o}(t) \boldsymbol{o}(t)^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_{jm}(t)} \tag{6}$$

Note that $\boldsymbol{\sigma}_{jm}^2$ = diag($\boldsymbol{C}_{jm}$) is the diagonal of the covariance matrix $\boldsymbol{C}_{jm}$.

3. **Accumulation**
of the statistics matrices $\boldsymbol{G}_{(n)i}$ and $\boldsymbol{k}_{(n)i}$ for each cluster (n) of similar model components [9] and for i-row of the adaptation matrix $\boldsymbol{W}_{(n)}$

$$\boldsymbol{k}_{(n)i} = \textstyle\sum_{m \in K_n} \frac{c_m \mu_{mi} \boldsymbol{\varepsilon}_m(\boldsymbol{\xi})}{\sigma_{mi}^2} \,, \quad \boldsymbol{G}_{(n)i} = \textstyle\sum_{m \in K_n} \frac{c_m \boldsymbol{\varepsilon}_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}})}{\sigma_{mi}^2} \tag{7}$$

where

$$\boldsymbol{\varepsilon}_m(\boldsymbol{\xi}) = \left[\boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}), 1\right]^{\mathrm{T}} \ , \ \ \boldsymbol{\varepsilon}_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{\varepsilon}_m(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) & \boldsymbol{\varepsilon}_m(\boldsymbol{o}) \\ \boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}) & 1 \end{bmatrix} . \tag{8}$$

4. **Iterative update**
of estimated matrix $\boldsymbol{W}_{(n)}$. The auxiliary function (2) can be rearranged into the form [8]

$$Q_{\boldsymbol{W}_{(n)}}(\boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}}) = \log|\boldsymbol{A}_{(n)}| - \sum_{i=1}^{I} \boldsymbol{w}_{(n)i}^{\mathrm{T}} \boldsymbol{k}_i - 0.5 \boldsymbol{w}_{(n)i}^{\mathrm{T}} \boldsymbol{G}_{(n)i} \boldsymbol{w}_{(n)i} \ , \tag{9}$$

To find the solution of equation (9) 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 (9) we receive

$$\frac{\partial Q(\boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}})}{\partial \boldsymbol{W}_{(n)}} = 0 \Rightarrow \boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1} \left(\frac{\boldsymbol{v}_{(n)i}}{\alpha_{(n)}} + \boldsymbol{k}_{(n)i} \right) \ , \tag{10}$$

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

$$\beta_{(n)} \alpha_{(n)}^2 - \alpha_{(n)} \, \boldsymbol{v}_{(n)i}^{\mathrm{T}} \boldsymbol{G}_{(n)i}^{-1} \boldsymbol{k}_{(n)i} - \boldsymbol{v}_{(n)i}^{\mathrm{T}} \boldsymbol{G}_{(n)i}^{-1} \boldsymbol{v}_{(n)i} = 0 \, , \tag{11}$$

where

$$\beta_{(n)} = \sum_{m \in K_n} \sum_{t} \gamma_m(t) \, . \tag{12}$$

Two different solutions $\boldsymbol{w}_{(n)i}^{1,2}$ are obtained, because of the quadratic function (11). The one that maximizes the auxiliary function (9) is chosen. Note that an additional term appears in the log likelihood for fMLLR because of the feature transforms, hence:

$$\log \mathcal{L}\left(\boldsymbol{o}_t | \boldsymbol{\mu}_m, \boldsymbol{C}_m, \boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}\right) = \log \mathcal{N}\left(\boldsymbol{A}_{(n)} \boldsymbol{o}_t + \boldsymbol{b}_{(n)}; \boldsymbol{\mu}_m, \boldsymbol{C}_m\right) + 0.5 \log|\boldsymbol{A}_{(n)}|^2 . \tag{13}$$

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].

4 Gradient descent fMLLR

Classical fMLLR is based on a row-by-row estimation of the adaptation matrix W with respect to data accumulated for each Gaussian. The main difference in our gradient descend fMLLR technique is a direct minimization of a criterion function [6]. From classical fMLLR described above, only the first step of the estimation - *alignment* - is

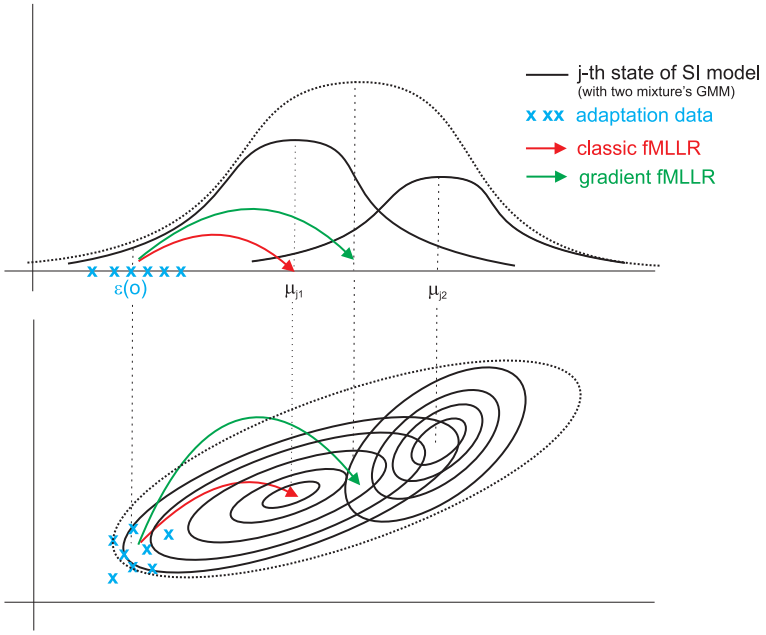

Fig. 1. Visualization of the fMLLR adaptation base on classical estimation and on our proposed estimation using gradient approach.

identical. The rest of the estimation is modified to direct minimization of the criterion function.

We do not consider individual Gaussians only. We consider negative Maximal Likelihood (ML) criterion that is based on likelihood of whole HMM states (see Figure 1). In contrast with classical fMLLR approach, adapted data are transformed into the center of the HMM state instead of the center of the Gaussian only.

The same approach can be used for various alternative differentiable criteria (e.g. Maximal Mutual Information or other discriminative ones). The minimization of the criteria formally written is similar to the equation (1)

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\lambda}), \tag{14}$$

where $\mathcal{F}(\boldsymbol{O}, \boldsymbol{\lambda})$ is the criterion function which is the negative ML criterion in our case

$$\mathcal{F}(\boldsymbol{O}, \boldsymbol{\lambda}) = -p(\boldsymbol{O}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}). \tag{15}$$

We choose the gradient descend method to optimize parameters λ because it is the most general optimization technique. Therefore, it can be used with various criteria and it can be used to optimize even other parameters, not only the fMLLR transformation matrix. So, the same framework can be developed further. In our case of ML criterion, even second derivatives - diagonal Hessian - can be easily calculated and the second order Newton optimization method can by employed to reduce a number of the optimization iterations.

For single Gaussian case, the partial derivation of the one element a_{ij} of the transformation matrix A is

$$\frac{\partial \mathcal{F}}{\partial a_{ij}} = \frac{\mu_i - \bar{o}_i(t)}{\sigma_i^2} o_j(t), \tag{16}$$

and the diagonal Hessian element - the second partial derivation is

$$\frac{\partial^2 \mathcal{F}}{\partial a_{ij}^2} = -\frac{o_j^2(t)}{\sigma_i^2}. \tag{17}$$

The partial derivations for the fMLLR vector b are

$$\frac{\partial \mathcal{F}}{\partial b_i} = \frac{\mu_i - \bar{o}_i(t)}{\sigma_i^2} \tag{18}$$

and

$$\frac{\partial^2 \mathcal{F}}{\partial b_i^2} = -\frac{1}{\sigma_i^2}. \tag{19}$$

Besides the sum of partial derivations over all data, the $\log(\det(A))$ derivation needs to be added. The derivation is equal to $inv(A)^T$. The second derivative of $\log(\det(A))$ is computed numerically.

The total partial derivations for entire HMM is a sum of all the individual Gaussians with using the same γ_{jm} as in the equations (5) and (6).

Then, the new estimate of A is

$$A_{(n+1)} = A_{(n)} - \alpha \frac{1}{2} \frac{\frac{\partial \mathcal{F}}{\partial A_{(n)}}}{\frac{\partial^2 \mathcal{F}}{\partial A_{(n)}^2}}, \tag{20}$$

where α is a stabilization constant from interval $\langle 0, 1 \rangle$. The stabilization together with an iterative approach must be used because we use only the diagonal Hessian which is inaccurate. The used γ_{jm} are also dependent on the derived parameters, but it makes the derivations too complicated. Therefore, we ignore theirs influence and the gammas are treated as fixed constants. It brings additional inaccuracy which involves a need of iterative stabilized approach.

4.1 Modified ML criterion

A classic ML criterion has uniform influence over all processed feature-vectors. It means that long phones or non-speech models have a higher total influence than shorter phones. Therefore, we modified the criterion to compute per-state means of the ML criterion and than the total sum is calculated from the means. But, some states with a few accumulated feature-vectors may disturb the final estimates. We proposed a smooth fade-out of the low-occupied states via soft threshold τ. The per-state means are summed with using a state weight w_j

$$w_j = \frac{\sum_{m=1}^{M} c_{jm}}{\tau + \sum_{m=1}^{M} c_{jm}}. \tag{21}$$

The same weights are used to compute first and second order of the partial derivatives.

5 Experiments

5.1 SpeechDat-East (SD-E) Corpus

For experiment purposes we used the Czech part of SpeechDat-East corpus [10]. In order to extract the features Mel-frequency cepstral coefficients (MFCCs) were utilized, 11 dimensional feature vectors were extracted each 10 ms utilizing a 32 ms hamming window, Cepstral Mean Normalization (CMN) was applied, and Δ, Δ^2 coefficients were added. A 3 state HMM based on triphones with 2105 states total and 8 GMM mixture components with diagonal covariances in each of the states was trained on 700 speakers with 50 sentences for each speaker (cca 5 sec. on a sentence). Using the same data UBM containing 256 mixture components was trained, and subsequently all the GMMs of individual development speakers were MAP adapted. To test the systems performance different 200 speakers from SD-E were used with 50 sentences for each speaker, however a maximum of 12 sentences was used for the adaptation. A language model based on trigrams used in the recognition [11]. The vocabulary consisted of 7000 words.

5.2 Results

The results of the experiment are shown in Table 1. The first part of the table contains the Accuracy (Acc) of the baseline system (recognition done utilizing only the SI model).

	supervised	unsupervised[%]
SI model	74.27	74.27
classic fMLLR	78.67	77.37
gradient fMLLR	78.99	77.66

Table 1. Accuracy (Acc)[%] of transcribed words for each type of the adaptation.

As can be seen from Table 1, the proposed gradient fMLLR approach performed better than classical fMLLR. The margin is not large but significant and it is obtained for both cases, supervised as well as unsupervised adaptation.

6 Conclusion

We proposed an approach which takes into account the overall likelihood of a HMM state. It estimates the transformation to optimize the ML criterion of HMM directly using the gradient descent algorithm. The criterion is based on likelihood of whole HMM states. It is better than the classical fMLLR which considers a likelihood of individual Gaussians only. The experiment results show improvement over the classical fMLLR method. Additional advantage of our approach is a compatibility with other differentiable criteria, especially the discriminative ones.

Acknowledgements

This research was supported by .....

References

1. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Readings in speech recognition, pp. 267–296 (1990).
2. Psutka, J., Müller, L., Matoušek, J., Radová, V.: Mluvíme s počítačem česky, Academia, Praha ISBN:80-200-1309-1 (2007).
3. Gauvain, L., Lee, C.H.: Maximum A-Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. In: IEEE Transactions SAP, pp. 2:291–298 (1994).
4. Alexander, A.: Forensic Automatic Speaker Recognition using Bayesian Interpretation and Statistical Compensation for Mismatched Conditions. In: Ph.D. thesis in Computer Science and Engineering, pp. 27-29, Indian Institute of Technology, Madras (2005).
5. Leggeter, C. J., Woodland P. C.: Maximum Likelihood Linear Regression for Speaker Adaption of Continuous Density Hidden Markov Models. In.: Computer Speech and Language, pp. 9:171-185 (1995).
6. Balakrishnan, S. V.: Fast incremental adaptation using maximum likelihood regression and stochastic gradient descent. In: EUROSPEECH, pp. 1521–1524 (2003).
7. Gales, M.J.F.: Maximum Likelihood Linear Transformation for HMM-based Speech Recognition. Tech. Report, CUED/FINFENG/TR291, Cambridge Univ. (1997).
8. Povey, D., Saon, G.: Feature and Model Space Speaker Adaptation with Full Covariance Gaussians. In: Interspeech, paper 2050-Tue2BuP.14 (2006).
9. Gales, M.J.F.: The Generation and use of Regression class Trees for MLLR Adaptation, Cambridge University Engineering Department (1996).
10. Pollak, P., et al.: SpeechDat(E) - Eastern European Telephone Speech Databases, XLDB - Very Large Telephone Speech Databases (ELRA), Paris (2000).
11. Pražák, A., Psutka, J., Hoidekr, J., et al.: Automatic online subtitling of the Czech parliament meetings, Lecture Notes in Artificial Intelligence, pp. 501–508 (2006).
12. Machlica, L., Zajíc, Z., Pražák, A.: Methods of Unsupervised Adaptation in Online Speech Recognition. In: Specom, St.Petersburg (2009).
13. Visweswariah, K., Gopinath, R.: Adaptation of front end parameters in a speech recognizer, In: Interspeech, pp. 21–24 (2004).