Fast Adaptation using Constrained Affine Transformations with Hierarchical Priors

Tor André Myrvoll, Kuldip K. Paliwal (*), Torbjørn Svendsen

(*) Signal Processing Lab
School of Microelectronic Engineering
Griffith University
Brisbane, Australia

Signal Processing Group
Department of Telecommuntcations
NTNU
Trondheim, Norway

Abstract

In this paper we present an approach to transformation based model adaptation that combines a fast, closed form solution to the MAP estimation of our transforms with robust priors. The robust priors are found using the technique of *hierarchical priors*, and a closed form solution is achieved by choosing diagonally constrained affine transformations and a suitable family of prior distributions for these transformations. We show that the method gives results comparable to other algorithms, but with significantly reduced computational complexity and memory demands. Experiments are conducted on the SI Recognition Outlier task from the Wall Street Journal corpus, where speaker independent models have to be adapted to handle speech from non-native speakers.

1. Introduction

An automatic speech recognition (ASR) system has a significant performance degradation when it is deployed under conditions that are different from the training conditions. Assuming that the system has been trained using clean speech obtained under controlled conditions from native speakers, such "real-life" issues as different microphones, non-native speakers and noisy conditions will have an adverse impact on the system performance.

Several approaches to overcome this mismatch problem has been put forward over the past years. One research direction that has shown good results is *model adaptation*, that is, the parameters of the hidden Markov model (HMM) are adapted to reflect the new working conditions. *Maximum a posteriori* (MAP) adaptation [1] was one of the earliest approaches. Though this method is optimal in the maximum likelihood (ML) sense when the amount of adaptation data increases, the convergence is slow. Therefore this method is ill-suited when data is scarce.

Another method that has been shown to yield good results even with little adaptation data is *maximum likelihood linear regression* (MLLR)[2]. The mean vectors of the HMM Gaussian mixtures are collected in clusters, and each cluster is adapted using an affine transformation

$$\hat{\mu} = A\mu + b \quad (1)$$
$$= W\tilde{\mu}, \quad (2)$$

where $W = [A\ b]$ and $\tilde{\mu}$ is the mean vector augmented by a one. In this way all mean vectors are adapted even when no adaptation data is available for some of the mixtures. Given a HMM represented by the parameters Λ and some adaptation data $\mathbf{O}$, the matrix A and vector b are estimated in the ML sense:

$$W = \underset{W'}{\text{argmax}} P(\mathbf{O}|W', \Lambda) \quad (3)$$

The major drawback of MLLR is the sensitivity with respect to the number of clusters. Using too many clusters results in poorly estimated transformations, while using to few gives us a too simple mapping and suboptimal performance. To overcome this problem the *maximum a posteriori linear regression* (MAPLR) method was introduced in [3]. In this method a prior distribution $P(W|\phi)$ is associated with each transformation to alleviate these problems. The parameters ϕ are referred to as the *hyperparameters* of the prior. Using the MAPLR formulation each transformation W is estimated as

$$W = \underset{W'}{\text{argmax}} P(W'|\mathbf{O}, \Lambda) \quad (4)$$
$$= \underset{W'}{\text{argmax}} P(\mathbf{O}|W', \Lambda) P(W'|\phi). \quad (5)$$

The performance of MAPLR is dependent on the quality of the priors used. The recently presented *Structural maximum a posteriori linear regression* (SMAPLR) algorithm uses the concept of hierarchical priors to estimate good priors for the transformations, and has been shown to outperform MLLR on a number of tasks [4, 5].

A common drawback with all the transformation based methods is the computational complexity. For every transformation there are p systems of linear equations having $p + 1$ unknowns that has to be solved. Some of the systems may be poorly conditioned, especially when adaptation data is scarce, so simple approaches like LU factorization should to be replaced by SVD or variations on the conjugate gradient algorithm[6].

In this paper we constrain the transforms to a be linear transformation by a diagonal matrix D and an added translation by a vector b. We show that when suitable priors are chosen for D and b, we can find a simple, closed form solution for the transform estimation problem. Experiments are conducted on the SI Recognition Outliers task (spoke 3 from the Wall Street

We acknowledge the helpful comments made by Dr. Olivier Siohan of Bell Laboratories – Lucent Technologies. This work was done at Griffith University while T. A. Myrvoll was on leave from the Department of Telecommunications, Norwegian University of Science and Technology.

Journal corpus), where a speaker independent model have to be adapted to handle speech from non-native speakers. We show that the results are comparable with the much used, but more complex MLLR algorithm.

2. MAP estimation of constrained affine transformations

2.1. Formulating the estimation problem

Let Λ be the parameters of the HMM to be adapted. In this work we will only adapt the mean vectors of the Gaussian mixtures, so Λ can be considered to be a set of mean vectors. Assuming that we have N disjoint clusters of mean vectors, let λ_n be the n'th such cluster.

For each cluster n we define a corresponding transformation of its member mean vectors:

$$\hat{\mu} = D_n \mu + b_n, \tag{6}$$

where D_n is a diagonal matrix. Given some adaptation data $\mathbf{O}$, and prior densities $P(D_n, b_n|\Phi)$, we can now estimate $\{D_n, b_n\}$ for all n by maximizing the prior probability of the transform with respect to the data:

$$\begin{aligned}\{D_n, b_n\} &= \operatorname*{argmax}_{D'_n, b'_n} P(D'_n, b'_n|\mathbf{O}, \lambda_n) \qquad (7)\\ &= \operatorname*{argmax}_{D'_n, b'_n} P(\mathbf{O}|D'_n, b'_n, \lambda_n) P(D_n, b_b|\Phi)\end{aligned}$$

As direct maximization of (7) is impossible due to the missing data problem, we apply the EM-algorithm[7]. We define the following auxiliary function $Q(\bar{\mathcal{D}}, \bar{\mathcal{B}}|\mathcal{D}, \mathcal{B})$:

$$\begin{aligned}& Q(\bar{\mathcal{D}}, \bar{\mathcal{B}}|\mathcal{D}, \mathcal{B}) \qquad (8)\\ &= \textstyle\sum_{s \in \mathcal{S}} P(\mathbf{O}, s|\mathcal{D}, \mathcal{B}, \Lambda) \log P(\mathbf{O}, s|\bar{\mathcal{D}}, \bar{\mathcal{B}}, \Lambda)\\ &\quad + \log P(\bar{\mathcal{D}}, \bar{\mathcal{B}}|\Phi)\\ &= \textstyle\sum_{m=1}^{M} \sum_{t=1}^{T} -\frac{1}{2}\gamma_m(t)(o_m(t) - D_{f(m)}\mu_m - b_{f(m)})'\\ &\quad \times R_m(o_m(t) - D_{f(m)}\mu_m - b_{f(m)})\\ &\quad + \log P(\bar{\mathcal{D}}, \bar{\mathcal{B}}|\Phi) + C\end{aligned}$$

Here $\mathcal{S}$ is the set of all possible state sequences associated with the adaptation data $\mathbf{O}$, $\mathcal{D}$ and $\mathcal{B}$ are sets of all D_n and b_n respectively, $\gamma_m(t)$ is the probability of observing mixture m at time t and $f(m)$ is an function that associates a unique transformation with each mixture m. R_m is the precision matrix and C represents all constant terms.

Before computing the gradient of (8) with respect to D_n and b_n, we have to specify the parametric form of the prior distributions $P(D_n, b_n|\Phi)$. As we have no reason to assume any correlation between D_n and b_n for any n, we will split the prior $P(D_n, b_n|\Phi)$ into two priors, $P(D_n|\Phi_D)$ and $P(b_n|\Phi_b)$. Furthermore, we assume that the elements of D_n and b_n will have bounded variance. As no other assumptions can be made at this stage we employ the principle of maximum entropy[8]. As is well known from information theory, when the only condition on a probability density function (pdf) is bounded variance, a Gaussian pdf will maximize the entropy[9]. This results in the following two priors:

$$\begin{aligned}D_n &\sim \mathcal{N}(d_n; \delta, \alpha I),\ \alpha > 0\\ b_n &\sim \mathcal{N}(b_n; \nu, \beta I),\ \beta > 0\end{aligned}$$

Here d_n is a vector consisting of the diagonal of D_n. The mean values of the priors will be estimated using the same approach as in SMAPLR, and the two scaling factors α and β are parameters that reflect the bounded variation and has to be specified.

Calculating the gradient of (8) with respect to the diagonal elements of D_n and the vector b_n, then equating the results to zero gives us the following two sets of linear equations:

$$\begin{aligned}\textstyle\sum_{\forall m, f(m)=n} \bar{\gamma}_m R_m \text{diag}(\mu_m)(D_n\mu_m - b_n) + \alpha d_n =\\ \textstyle\sum_{\forall m, f(m)=n} R_m \text{diag}(\mu_m)\bar{o}_m + \alpha\delta_n\end{aligned} \tag{9}$$

and

$$\begin{aligned}\textstyle\sum_{\forall m, f(m)=n} \bar{\gamma}_m R_m(D_n\mu_m - b_n) + \beta d_n =\\ \textstyle\sum_{\forall m, f(m)=n} R_m \bar{o}_m + \beta\delta_n\end{aligned} \tag{10}$$

where diag(μ_m) refers to a matrix with the elements of the vector μ_m on its diagonal, and

$$\begin{aligned}\bar{\gamma}_m &= \sum_{t=1}^{T} \gamma_m(t)\\ \bar{o}_m &= \sum_{t=1}^{T} \gamma_m(t) o_t\end{aligned}$$

The sums are over all mixtures associated with transform n. In the next subsection we will see how we can solve the equations above in an effective manner.

2.2. A fast method for solving the linear equations

The two systems of linear equations can be written as the following matrix equation:

$$M \begin{bmatrix} d_n \\ b_n \end{bmatrix} = z \tag{11}$$

Studying the two systems of linear equations and assuming that the precision matrix R_m is diagonal, we can observe that the matrix M will have only two non-zero elements per row. It is now easy to show that (11) can be written as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} d_n \\ b_n \end{bmatrix} = z$$

where A,B,C and D all are diagonal matrices. We can now use this fact to express a closed form solution for d_n and b_n which will be very simple to evaluate. The block matrix equations can be written as follows

$$\begin{aligned}Ad_n + Bb_n &= z_u\\ Cd_n + Db_n &= z_l,\end{aligned}$$

where z_u and z_l are the upper and lower half of the vector z respectively. Using simple linear algebra we arrive at the following solution:

$$\begin{aligned}d_n &= (C - DB^{-1}A)^{-1}(z_l - DB^{-1}z_u)\\ b_n &= B^{-1}(z_u - Ad_n)\end{aligned} \tag{12}$$

As all the matrices involved in the above solutions are diagonal calculating the two vectors will be straight-forward. In the next section we will discuss some practical issues regarding computational complexity and memory overhead of this approach compared to MLLR and SMAPLR.

2.3. Computational complexity and memory use

Trying to compare the computational complexity and memory use of MLLR/SMAPLR directly to our approach on a transform by transform basis is clearly meaningless as the first two methods will need to operate on fewer clusters due to the richer set of mappings they can produce. However, the complexity and memory overhead of MLLR/SMAPLR is so much higher that even if we use a significantly higher number of transformations with the constrained version it will still be faster and use less memory. During our analysis we assume a mean vector dimension of p. We also count addition, subtraction, multiplication and division as equivalent floating point operations with regard to the computational cost.

The computational cost associated with $\{\hat{\gamma}_m\}$ and $\hat{o}_m$ is the same for all the methods and will be disregarded here. Building the systems of linear equations have very different costs however. Inspecting (9) and (10) we find that the straight-forward use of these equations gives a computational cost for our method in the order of $O(p)$ floating point operations per mixture. MLLR and SMAPLR makes use of outer products between vectors and thus yields in the order of $O(p^2)$ floating point operations per mixture.

Inspecting the equations in (12) we find the number of floating point operations needed to get a solution is in the order of $O(p)$. For MLLR/SMAPLR this will really depend on the algorithm used to solve the equations. Assuming that simple LU factorization will be used the number of floating point operations will be in the order of $O(p^4)$ as the LU factorization algorithm is $O(p^3)$ and we have p sets of equations to be solved.

The memory use for our method is assumed to be the space needed to hold our system of linear equations, as well as out transform. Inspecting (12) we see that the memory need is in the order of $O(p)$. For MLLR/SMAPLR we need p matrices of dimension $(p+1) \times (p+1)$ to estimate every row of the transform matrix, resulting in a memory use in the order of $O(p^3)$.

	Fast method	MLLR/SMAPLR
Complexity	$O(p)$	$O(p^4)$
Memory	$O(p)$	$O(p^3)$

Table 1: A comparison of memory use and computational complexity

The computational complexity and memory use for the different methods are all shown in table 1. As we see from this table we would have to use a number $\propto p^2$ more transformations than MLLR/SMAPLR to achieve the same memory use. For our experimental setup, which is presented in the next section and uses 39 dimensional feature vectors, this is a factor of 1521. To reach the same computational complexity we would have to use a number $\propto p^3$ times the number of transformations. This indicates that for all practical means we can use the number of transforms required to match the performance of MLLR/SMAPLR. It is important however, that proper care should be taken when increasing the number of transforms to avoid poor estimates due to lack of data. In the next subsection we will briefly show how robustness can be achieved using hierarchical priors.

2.4. Hierarchical priors

As in [4] we will arrange our clusters of mean vectors in a tree where every cluster associated with a node is the union of all the clusters associated with its child nodes. As well as clusters we associate a prior with every node in the tree, and each such prior is a hyperprior for the next level in the tree. This concept is visualized in fig. 1, where $W = [D\, b]$ is a compact representation of our transformation.

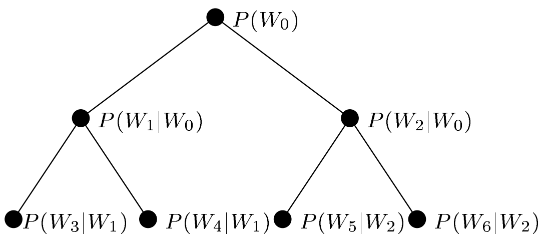

Figure 1: A hierarchical tree structure of priors

Every prior is chosen to be an approximation to the posterior pdf of the transform given the adaptation data, $P(W|\mathbf{O}, \Lambda)$. In this way we can estimate our priors in a top-down fashion as described in [4]. As for SMAPLR this gives us high quality priors that alleviates the estimation problems we otherwise might have.

In the same way as for MLLR/SMAPLR we use a threshold on the observation count for the mixtures in a specific cluster. Clusters that fall below this threshold will not have any transformations associated with them. To choose the best transform for a given mixture we traverse the tree from the leafs at the bottom and upwards, until we find a feasible transform. This process is illustrated in fig. 2.

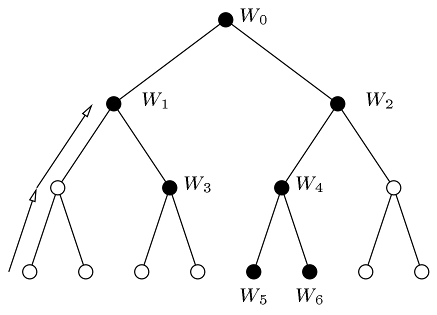

Figure 2: Adapting the mean vectors by choosing the transformation closest to the bottom of the tree.

3. Experiments and results

3.1. Database and baseline system

Our experiments was performed on the SI Recognition Outliers task (also referred to as spoke 3) from the Wall Street Journal (WSJ) corpus. The task consists of 40 adaptation utterances and 40 test utterances from each of the 10 non-native speakers. We also investigated the use of subsets consisting of 5 and 20 adaptation utterances.

Our initial speaker independent (SI) HMM was build using the SI84 training data which consists of $\sim$ 7200 utterances from 84 speakers. We used 12 Mel frequency cepstral coefficients (MFCC) along with the normalized log energy and first and second derivatives. Zero mean subtraction and mean cepstral normalization was performed as well.

Finally, we used cross word triphones that was state clustered using a phonetic decision tree, giving us a total of ~ 30000 mixtures. The pronunciation lexicon was based on the CMU dictionary[10], and the grammar was the back-off bigram from MIT Lincoln Laboratories as supplied with the WSJ corpus.

Two trees of clustered means was build for MLLR and the constrained method respectively. Both trees had a maximum of 10 children per node, but whereas the clusters for MLLR had to contain a minimum of $p + 1$ (40) mean vectors, the clusters for the constrained method was allowed to contain as little as 2.

We also had to define thresholds on the occupation count. From preliminary experiments on a single speaker from the development set we chose to use a threshold on the occupation count of 1000.0 for 5 adaptation utterances and 500.0 for 20 and 40 utterances when using MLLR, and 5.0 when using our constrained method. Note that the development set consists of speakers not in our test data.

3.2. Experimental results

The main objective with the experiments performed was to show that the constrained transformation approach perform as well as the more complex MLLR method. No formal measurements were made with respect to computational complexity as we use iterative techniques used to solve the MLLR equations, but an estimate of the memory overhead is found by counting the number of transforms used and multiplying by the number of bytes needed to hold the system of linear equations.

The experimental results are presented in figure 3. The SI model is adapted to each of the 10 non-native speakers using 5, 20 and 40 adaptation utterances. The results presented here are the averages for all the speakers. As we can see our constrained method performs comparably to MLLR.

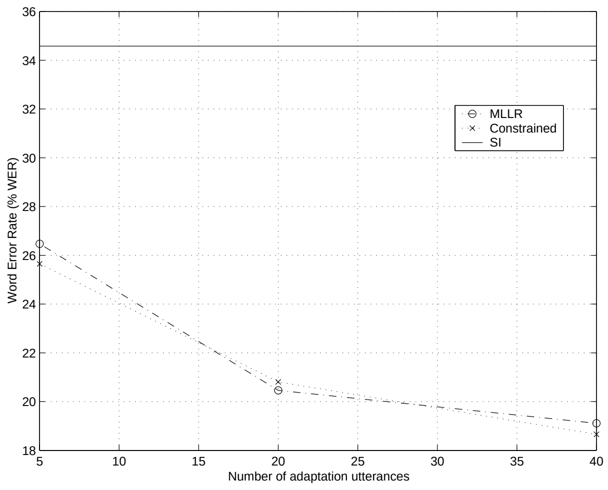

Figure 3: Word error rate (%) for supervised adaptation experiments.

In table 3.2 we have compared the memory use of MLLR and our constrained method when adapting the SI model to a single speaker using 40 utterances. As we see the amont of memory used by the latter approach is significantly smaller despite the high number of transformations used.

Method	MLLR	Constrained
# Transformations	21	2636
Approx. memory used	10.5 Mb	3.2 Mb

Table 2: Approximate memory use of MLLR and the constrained transform method.

4. Conclusions

We have presented an adaptation algorithm that uses constrained transformations, allowing us to solve the estimation problem in a computationally effective manner. The method also gives us a significant reduction in memory overhead.

As the constrained transformations are much less flexible than the full transformations used in MLLR we need to increase the number of such transformations. To avoid poor estimates due to scarcity of the adaptation data, we make use of hierarchical priors arranged in a tree structure to find high quality priors for our MAP estimates.

In our experiment we adapted a speaker independent model to handle speech from non-native speakers and showed that our method results in a performance comparable to MLLR.

5. References

[1] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, April 1994.

[2] C. J. Legetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of the parameters of continuous density hidden markov models," *Computer Speech and Language*, no. 9, pp. 171–185, 1995.

[3] O. Siohan, C. Chesta, and C.-H. Lee, "Hidden markov model adaptation using maximum a posteriori linear regression," in *Workshop on Robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finland*, 1999.

[4] O. Siohan, T. A. Myrvoll, and C.-H. Lee, "Structural maximum a posteriori linear regression for fast hmm adaptation," in *ISCA Tutorial and Research Workshop (ASR 2000)*, 2000.

[5] T. A. Myrvoll, O. Siohan, C.-H. Lee, and W. Chou, "Structural maximum a posteriori linear regression for unsupervised speaker adaptation," in *International Conference on Speech and Language Processing*, (Beijing, China), 2000.

[6] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipies in C. The Art of Scientific Computing*. Cambrigde University Press, 1988.

[7] A. J. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, 1977.

[8] E. T. Jaynes, "On the rational of maximum-entropy methods," *Proceedings of the IEEE*, vol. 70, September 1982.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[10] "The CMU Pronouncing Dictionary (version 0.6d)." http://www.speech.cs.cmu.edu/cgi-bin/cmudict.