14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

CROSSLINGUAL ADAPTATION OF SEMI-CONTINUOUS HMMS USING MAXIMUM LIKELIHOOD AND MAXIMUM A POSTERIORI CONVEX REGRESSION

Frank Diehl, Asunción Moreno, Enric Monte

TALP Research Center
Universitat Politècnica de Catalunya (UPC)
Jordi Girona 1-3, 08034 Barcelona, Spain
{frank,asuncion,enric}@gps.tsc.upc.edu

ABSTRACT

In this work we present a novel adaptation design for semicontinuous HMMs (SCHMM). The method, which is developed in the scope of a crosslingual model adaptation task, consists in adjusting the states' mixture weights associated to the prototype densities of the codebook. The mixture weights of the target language are modelled as convex combinations of prototype weights. They are defined by an acoustic regression scheme applied to the source models, followed by a refinement using probabilistic latent semantic analysis (PLSA). In order to find suitable combination weights for the convex combinations we present a maximum likelihood (ML) as well as a maximum a posteriori (MAP) estimate. Thus, we name them maximum likelihood convex regression (MLCR) and maximum a posteriori convex regression (MAPCR). Finally, a crosslingual model adaptation task transferring multilingual Spanish-English-German HMMs to Slovenian demonstrates the performance of the method.

1. INTRODUCTION

In recent years automatic speech recognition (ASR) technology has found its way into the markets of many countries. One prerequisite for this success was the effort made during the last decade for providing appropriate speech databases in many languages. However, although speech databases for a multitude of languages are available now, it often remains a challenge to develop an ASR system for a new language. Reasons are manifold. A company might not be able to buy a large speech database, the available databases do not match the desired environmental conditions, or a database for the required language may not be available.

To reduce the dependency on suitable speech databases crosslingual acoustic modelling has become an active research area [1], [2], [3]. Instead of relying on the availability of a complete speech database in the target language, a research line consists of trying to take advantage of already available acoustic models of some other languages. Usually a two step procedure using a limited amount of target data is used to transform acoustic models from a source language to the target language. The first step consists of predicting suitable target models out of a phonetic-acoustic decision tree of the source language. In a second step the predicted models are adapted to the target language using a limited amount of target data. The adaptation policy normally consists of a maximum a posteriori (MAP) approach [4], or maximum likelihood linear regression (MLLR) [5].

As known to the authors, until now almost all results reported on language adaptation of hidden Markov models (HMM) refer to the use of continuous density HMMs (CDHMM). However, semicontinuous HMMs (SCHMM) are still widely-used. Their lower complexity paired with high performance make them an attractive alternative to CDHMMs especially for small and medium scale systems. Unfortunately, adaptation techniques for SCHMMs are not as well-developed as in the case of CDHMMs. Beside MAP adaptation proposed in [6], a powerful transformation based method is still not established. Of course, MLLR can be applied to SCHMMs too. Nevertheless, its use is of limited effect due to SCHMMs relying on one common codebook. Applying regression class specific MLLR transformations for different model groups is impossible. Only common transformations over all models are feasible weakening the strength of MLLR significantly.

Recently a new transformation based adaptation approach for SCHMMs was presented. In [7] the authors suggested to model the mixture weights of the HMMs as a convex combination of a set of prototype weights. The prototype weights themselves were defined by an acoustic regression scheme deployed to the source models followed by a refinement applying probabilistic latent semantic analysis (PLSA) [8]. For fixing the combination weights of the convex combinations a ML expression was derived by an appropriate modification of the corresponding auxiliary function taken from the Baum-Welsh reestimation framework.

In this work we first reformulate the derivation given in [7] by a straightforward ML procedure, avoiding the use of Baum's auxiliary function. The resulting algorithm is called maximum likelihood convex regression (MLCR). In a second step the ML procedure is extended to a MAP solution which is named maximum a posteriori convex regression (MAPCR). Finally, we study the case of a crosslingual model adaptation task, transferring multilingual Spanish-English-German models to Slovenian. Besides comparing the case of MLCR with MAPCR, simulation results for MLLR are given too.

2. THE DATA MODEL

In a SCHMM system the output densities $p(x|s)$ of the states $s \in \{1,...,M\}$ are defined as superpositions of prototype densities $p(x|k)$ of a common codebook with $k \in \{1,...,K\}$ naming the prototypes.

$$p(x|s) = \sum_{k=1}^{K} c_{sk}\, p(x|k) \tag{1}$$

The mixture weights c_{sk} fulfil standard probabilistic constraints and can be interpreted as the conditional probabilities that prototype density k is active when being in state s, i.e.

$$c_{sk} = P(k|s) \tag{2}$$

$$= \frac{P(k,s)}{P(s)}. \tag{3}$$

This work was granted by the CICYT under contract TIC2002-04447-C02.

We define an adapted or smoothed version $\underline{\bar{c}}_s = [\bar{c}_{s1}, ..., \bar{c}_{sK}]^T$ of the mixture weights $\underline{c}_s = [c_{s1}, ..., c_{sK}]^T$ as a convex combination

$$\underline{\bar{c}}_s = \underline{U}_s \underline{\alpha}_s \tag{4}$$

of a set of L prototype weights vectors $\underline{u}_{sl}$ forming matrix $\underline{U}_s$. I.e., matrix $\underline{U}_s$ models our belief of the acoustic neighbourhood of $\underline{c}_s$.
The L-dimensional vector $\underline{\alpha}_s$ represents the combination weights which need to be estimated under standard probabilistic constraints. As indicated by the subscript s the $\underline{u}_{sl}$ and the $\underline{\alpha}_s$ depend on the current state s. We set $L << K$ to get the desired reduction in the number of free parameters.

3. MAXIMUM LIKELIHOOD CONVEX REGRESSION

In contrast to [7] where the ML solution of estimation problem (4) is obtained within the Baum-Welsh reestimation framework, here we provide a more straightforward ML formulation which avoids the use of Baum's auxiliary function.
The key for doing so is to define the observation sequence in a way different from the usual one. The most natural way for defining the observation sequence might be by the sequence of melcepstrum values or suchlike, i.e. by the x from (1). Nevertheless, for the problem on hand we do not so. In contrast, we remember that we model the sequence of x as generated by a hidden Markov process, i.e. by a sequence of joint events $o_{sk} = \{\text{the process is in state } s \text{ applying prototype density } k\}$. Though, we are not able to observe the sequence of o_{sk}, we are able to develop an expression for its likelihood and to relate it to our model (4).
We start to derive a ML expression for all prototype weights $\alpha = [\underline{\alpha}_1, ..., \underline{\alpha}_M]$ by defining the observation sequence $S = (o^n)_{1 \le n \le N}$. It is modelled as the realisation of an underlying sequence of independent identical distributed (i.i.d.) random variables $(O^n)_{1 \le n \le N}$ which are defined on a finite set of events $O = \{o_{11}, ..., o_{sk}, ..., o_{MK}\}$ with the o_{sk} as explained above.
With the i.i.d assumption of the random variables O^n the log-likelihood $\mathscr{L}(\alpha)$ of the observation sequence S is expressed as

$$\begin{aligned} \mathscr{L}(\alpha) &= \log P(S) && (5)\\ &= \log P(o^1, ..., o^N) && (6)\\ &= \log \prod_{n=1}^{N} P(o^n) && (7)\\ &= \log \prod_{s=1}^{M} \prod_{k=1}^{K} P(o_{sk})^{n_{sk}}. && (8) \end{aligned}$$

In (8) we used the empirical event counts $n_{sk} = |\{o^n : o^n = o_{sk}\}|$ which allows us to group the likelihoods of identical events o^n together. Next we introduce the event $o_s = \{\text{the process is in state } s\}$ and express, according to (3), the mixture weights c_{sk} as

$$c_{sk} = \frac{P(o_{sk})}{P(o_s)}. \tag{9}$$

After substituting c_{sk} by its model $\bar{c}_{sk}$ we get

$$P(o_{sk}) = \bar{c}_{sk} P(o_s), \tag{10}$$

and the log-likelihood of (8) can be expressed as

$$\begin{aligned} \mathscr{L}(\alpha) &= \log \prod_{s=1}^{M} \prod_{k=1}^{K} (\bar{c}_{sk} P(o_s))^{n_{sk}} && (11)\\ &= \sum_{s=1}^{M} \left(\log \prod_{k=1}^{K} \bar{c}_{sk}{}^{n_{sk}} + \log \prod_{k=1}^{K} P(o_s)^{n_{sk}} \right). && (12) \end{aligned}$$

Inspecting (12) we find that the log-likelihood of the complete observation sequence S is composed out of the sum of the individual log-likelihoods for each state s. Bearing in mind that the terms including $P(o_s)$ do not affect the maximisation of $\mathscr{L}(\alpha)$, (12) can be decomposed to the individual log-likelihoods

$$\mathscr{L}(\underline{\alpha}_s) = \log \prod_{k=1}^{K} \bar{c}_{sk}{}^{n_{sk}} \tag{13}$$

for each state $s \in \{1, ..., M\}$. That is, the joint maximisation of $\mathscr{L}(\alpha)$ respective α is broken down to the individual maximisation of the $\mathscr{L}(\underline{\alpha}_s)$ for each state s.
For the further evaluation of (13) the counts n_{sk} are needed. Unfortunately it is difficult to obtain them, if it is possible at all. Nevertheless, we can easily obtain estimates for their expectation values $\bar{n}_{sk}$. As the adaptation of HMMs implies the availability of more or less adequate source models, it is always possible to reestimate the source models by the adaptation data. The resulting reestimated mixture weights $\underline{c}_s$ constitute a first estimate of the target models but can also be seen as a measurement providing us with statistics of the adaptation data. This gets clear remembering, [9], that the individual c_{sk} can be interpreted as normalised counts

$$c_{sk} = \frac{\bar{n}_{sk}}{\bar{n}_s} \tag{14}$$

with $\bar{n}_{sk}$ denoting the expected number of times the hidden Markov process is in state s applying the kth prototype density, and $\bar{n}_s$ the expected number of times the hidden Markov process is in state s.
In light of our problem we use the $\bar{n}_{sk}$ as an estimate for the n_{sk} and reformulate (13) as

$$\begin{aligned} \mathscr{L}(\underline{\alpha}_s) &= \log \prod_{k=1}^{K} \bar{c}_{sk}{}^{\bar{n}_{sk}} && (15)\\ &= \log \prod_{k=1}^{K} \bar{c}_{sk}{}^{c_{sk}\bar{n}_s} && (16)\\ &= \bar{n}_s \sum_{k=1}^{K} c_{sk} \log(\bar{c}_{sk}). && (17) \end{aligned}$$

Using vector notation and plugging in (4) we get

$$\mathscr{L}(\underline{\alpha}_s) = \bar{n}_s \underline{c}_s^T \log(\underline{U}_s \underline{\alpha}_s) \tag{18}$$

and the final optimisation problem can be stated as

$$\arg\min_{\underline{\alpha}_s} -\underline{c}_s^T \log(\underline{U}_s \underline{\alpha}_s) \tag{19}$$

$$\begin{aligned} &\textit{subject to} && \sum_{l=1}^{L} \alpha_{sl} = 1\\ &\textit{and} && \alpha_{sl} \ge 0 \qquad \forall l \in \{1, .., L\} \end{aligned}$$

which need to be solved for each $s \in \{1, ..., M\}$. With the $\underline{\alpha}_s$ as defined by (19) the adapted mixture weights $\underline{\bar{c}}_s$ are finally given by (4).
Though we found no closed form solution for the problem, it is identified as convex and can be solved by convex optimisation [10].
A graphical interpretation of the ML solution is given in Fig. 1. Solving problem (19) consists in projecting the measurement $\underline{c}_s$ onto the probabilistic sub-simplex spanned by the convex combination of $\underline{U}_s$, minimising the distance, i.e. the cross-entropy, between the measurement $\underline{c}_s$ and the solution $\underline{\bar{c}}_s$.
As explained above, the proposed adaptation method consists of two steps. First, we retrain the source models by the adaptation

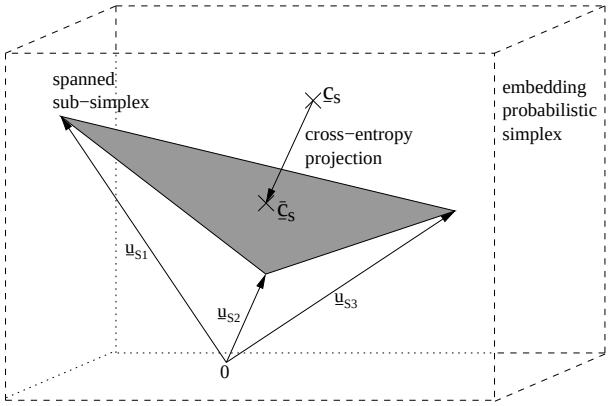

Figure 1: *Graphical interpretation of the ML solution.*

data. Due to the limited amount of target data, the resulting models are quite rough estimates of the final target models. Anyhow, they can be seen as noisy measurements of the target models and provide us with statistics about the adaptation data. Second, these noisy measurements are smoothed by projecting them onto probabilistic solution sub-simplexes constituting our expectations where to find the final solutions of our adaptation problem.

It remains to define a method to assign the source to the target models and to define appropriate solution sub-simplexes $\underline{U}_s$. For both tasks a regression scheme applying the phonetic-acoustic decision tree of the source language is applied, also explaining where the name of the method, maximum likelihood convex regression (MLCR), stems from. A detailed description of the corresponding regression scheme is given in Section 4 and 5.

4. TARGET MODEL PREDICTION AND ACOUSTIC REGRESSION CLASSES

Crosslingual model adaptation starts with the prediction of suitable target models out of the decision tree of the source language. In our system we use a phonetic-acoustic decision tree for state tying. It is constructed during the training of the source models constituting a function from a generic phonetic feature space to a state space. The input domain consists of phonetic feature vectors assigned to the central phone and the phonetic contexts of a state. The features are generic, i.e. to a large degree independent of the used language. An example might be $(plosive, bilabial, voiced)$. The output domain holds the weights vectors of the states.

With the input domain being of generic nature the tree can also be used to predict the tied states of the models of a new language. After setting up the feature vectors for the new language one calls the tree applying the features. Afterwards the predicted weights vectors are assigned to the models of the target language. This procedure effectively defines the target models $\underline{c}_s$ and initialises their training. In a further step the decision tree is also used to define the solution sub-simplexes $\underline{U}_s$ by exploiting the acoustic neighbourhood knowledge given by neighbouring leaves of the tree. Acoustic regression classes are defined by cutting the tree above its leaves constructing a set of subtrees. In Fig. 2 we illustrate this situation. It shows a fictitious decision tree which is cut by the dashed line. It results in three subtrees with 6, 3, and 2 leaves and the corresponding mixture weights of the base states $\{\underline{b}_1, \underline{b}_2, \underline{b}_3, \underline{b}_4, \underline{b}_5, \underline{b}_6\}$, $\{\underline{b}_7, \underline{b}_8, \underline{b}_9\}$ and $\{\underline{b}_{10}, \underline{b}_{11}\}$. The cut is accomplished searching the complete tree for the subtree giving minimal accumulated model entropy. The search, starting from the tree's root, stops after having reached the number of desired regression classes, i.e. nodes.

All leaves of a subtree share one or more common phonetic features

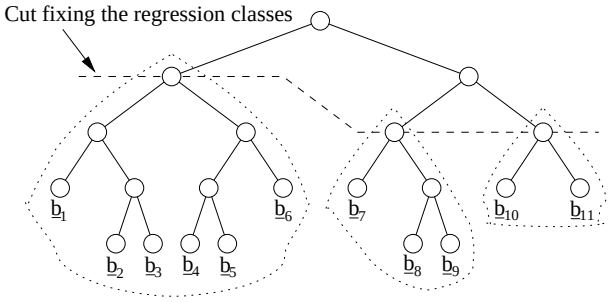

Figure 2: *Sub-simplex construction.*

or, in other words, are to some degree acoustically similar. In consequence, the mixture weights vectors $\underline{b}_i$ associated to the leaves of a subtree form a more or less consistent acoustic sub-simplex. Furthermore, for the adaptation task we assume that the base models we use for initialisation are already close to the target models. From these considerations we posit that each target model should lie within the space spanned by its initialisation state and its neighbouring states occupying the same subtree.

Although each set of base states $\underline{b}_i$ already constitutes a possible solution space for a $\bar{\underline{c}}_s$ we do not stack them directly to $\underline{U}_s$. When doing so we would actually lose, at least to some degree, our control over the amount of free parameters to estimate. As seen from Fig. 2, the generated subtrees exhibit different numbers of leaves which would result in a hardly controllable number of combination weights $\underline{\alpha}_s$ for each sub-simplex. To overcome this problem, but also to reduce stochastic dependencies within the $\underline{U}_s$, we use PLSA to derive the $\underline{U}_s$ out of the $\underline{b}_i$ assigned to a regression class.

5. PROBABILISTIC LATENT SEMANTIC ANALYSIS

The starting point for PLSA is the so called aspect model [8]. For a pair of random variables $(x,y) \in (X,Y) = \{(x_k, y_s) | 1 \leq k \leq K \wedge 1 \leq s \leq M\}$ an underlying production mechanism involving a hidden variable $z \in Z = \{z_1, .., z_L\}$ is assumed. The hidden variable is called factor or latent variable, and the production mechanism assumes conditional independence between X and Y given the latent variable Z. With this model the joint probability $P(x,y)$ is expressed as

$$P(x,y) = \sum_{z \in Z} P(x|z)P(y|z)P(z). \tag{20}$$

The parameters of (20) are estimated using the EM algorithm as described in [8]. With the terms $\underline{P} = [P(x_k, y_s)]_{k,s}$, $\underline{\tilde{U}} = [P(x_k|z_l)]_{k,l}$, $\underline{\tilde{\Sigma}} = diag[P(z_l)]_l$, and $\underline{\tilde{V}} = [P(y_s|z_l)]_{s,l}$, (20) can be arranged in matrix form as

$$\underline{P} = \underline{\tilde{U}} \underline{\tilde{\Sigma}} \underline{\tilde{V}}^T. \tag{21}$$

Equation (21) shows a formal similarity to a singular value decomposition (SVD). The most prominent one is the interpretation of the columns of matrix $\underline{\tilde{U}}$ as base vectors of a sub-simplex. On the other hand, in contrast to a SVD, PLSA constitutes a generative model. I.e. we are free to chose the model complexity and therefore the number of latent variables L to control the size of $\underline{\tilde{U}}$.

In light of our search for suitable basis vectors $\underline{u}_{sl}$, we use PLSA as follows. After having identified the neighbouring base vectors for a specific state s the vectors are stacked together forming matrix $\underline{B}_s$. E.g., for the first subtree of Fig. 2 we get $\underline{B}_s = [\underline{b}_1, \underline{b}_2, \underline{b}_3, \underline{b}_4, \underline{b}_5, \underline{b}_6]$. The components of matrix $\underline{B}_s$ stand for the conditional probabilities $P(k|s, T_s)$, i.e. the probability of mixture density k given state s and subtree T_s. To get the joint probabilities $P(k, s|T_s)$ we multiply the columns of $\underline{B}_s$ by the $P(s|T_s)$, the occupation probabilities of the states conditioned on the current subtree T_s. The $P(s|T_s)$ can be derived during the training of the base models.

Finally, after having fixed the model complexity L, the matrix of joint probabilities $P(k,s|T_s)$ is decomposed by PLSA providing $\tilde{U}$ being actually the desired base $\underline{U}_s$ in the MLCR solution (19).

6. MAXIMUM A POSTERIORI CONVEX REGRESSION

A critical point of the ML solution is the definition of the solution sub-simplex $\underline{U}_s$. The ML solution implicitly assumes that a good solution is more likely to lie close to $\underline{U}_s$ than to $\underline{c}_s$. Though this might be a reasonable assumption in case of a small amount of adaptation data, it loses its justification as more data becomes available. If we had plenty of data we would anticipate that the solution for the adaptation problem is $\underline{c}_s$ itself. A natural way to take this relationship into consideration is the introduction of prior information about our confidence in the solution sub-simplex $\underline{U}_s$, i.e. extending the ML approach to a MAP approach.
For a MAP formulation of the problem we start by including $\underline{c}_s$ into the solution sub-simplex by extending $\underline{U}_s$ by $\underline{c}_s$, demanding the extension of $\underline{\alpha}_s$ by α_{sL+1}, too. I.e. the data model changes to

$$\bar{\underline{c}}_s = [\underline{U}_s, \underline{c}_s]\,[\alpha_{s1}, ..., \alpha_{sL}, \alpha_{sL+1}]^T. \tag{22}$$

In case of the prior distribution $p(\underline{\alpha}_s)$ we are actually only interested in weighting the solutions between $\underline{U}_s$ and $\underline{c}_s$. In practise this means that we can set $p(\underline{\alpha}_s) = p(\alpha_{sL+1})$, being equivalent to a non-informative, uniform prior for the original αs.
Hence, analogous to (16), the objective function $\mathscr{M}(\underline{\alpha}_s)$ to maximise can be stated as

$$\mathscr{M}(\underline{\alpha}_s) = log\left(p(\alpha_{sL+1})\prod_{k=1}^{K}\bar{c}_{sk}{}^{c_{sk}\bar{n}_s}\right) \tag{23}$$

where $\bar{c}_{sk}$ refers to the extended data model of (22). For the prior density we assume the form of a gamma distribution

$$p(\alpha_{sL+1}) = C\alpha_{sL+1}^{\mu} exp(-\eta\,\alpha_{sL+1}) \tag{24}$$

with $\alpha_{sL+1} \in [0,1]$, $\mu, \eta \geq 0$ and C a suitable normalisation constant. Parameter μ and η serve to control the shape of the priors. As depicted in Fig 3, for a small η the prior gets uniform expressing our uncertainty respective $\underline{U}_s$. But, as η gets bigger the prior changes its shape to a peak concentrating its probability mass near zero. In this case α_{sL+1} is expected to be rather small, reflecting our suspicion regarding the measurement $\underline{c}_s$. The value of μ is of minor importance especially if η gets big. Using vector notation and definition (24), the final optimisation problem according to (23) becomes

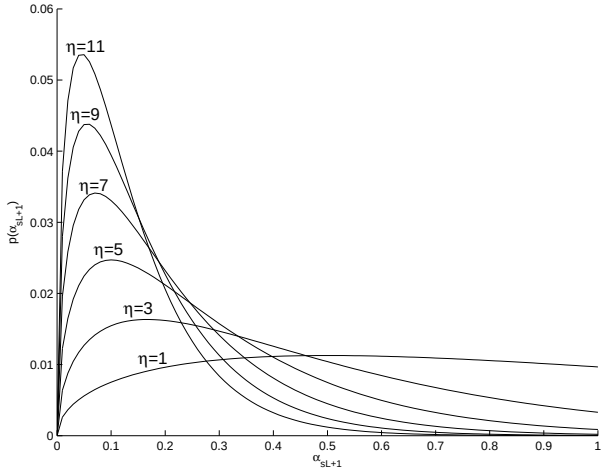

Figure 3: *Prior distributions* $p(\alpha_{sL+1})$ *for* $\mu = 0.5$.

$$\begin{aligned} \arg\min_{\underline{\alpha}_s} \quad & -\bar{n}_s\underline{c}_s^T \log\left([\underline{U}_s, \underline{c}_s]\,[\alpha_{s1}, ..., \alpha_{sL+1}]^T\right) \\ & -\mu \log \alpha_{sL+1} + \eta\,\alpha_{sL+1} \end{aligned} \tag{25}$$

$$\begin{aligned} \textit{subject to} \quad & \sum_{l=1}^{L+1}\alpha_{sl} = 1 \\ \textit{and} \quad & \alpha_{sl} \geq 0 \qquad \forall l \in \{1, .., L+1\} \end{aligned}$$

which is convex, i.e. can also be solved by convex optimisation. The $\bar{n}_s$, i.e. the expected number of times the hidden Markov process stays in state s, can easily be obtained as byproduct when reestimating the source models by the adaptation data.
A key role in the interpretation of (25) inheres the state count $\bar{n}_s$. It is the factor balancing the information reflected by the measurement and the prior. As a reliable measurement comes with a high state count, a high $\bar{n}_s$ gives emphasise to the likelihood part of (25) shifting the solution to $\underline{c}_s$. On the other hand, if $\bar{n}_s$ is small the prior terms will dominate. With η sufficiently big α_{sL+1} is forced to be close to zero leading to a solution near to the original, not extended sub-simplex.

7. SYSTEM OVERVIEW

We use a SCHMM system calculating every 10ms twelve melcepstrum coefficients (MFCC) (and the energy) using cepstral mean subtraction. First and second order differential MFCCs plus the differential energy are employed. For each stream a codebook is constructed consisting of 256 and 32 (delta energy) Gaussian mixtures, respectively. We use 3-state state-tied left-to-right demiphones. Demiphones [11] can be thought of as triphones which are cut in the middle giving a left and a right demiphone. For state tying we apply a binary decision tree to each state position but over all source phonemes resulting in six trees. Thus, beside context questions also questions respective the central phoneme of a model are asked. The questions used by the decision tree are of phonetic character and are derived from the IPA-chart.

8. THE ADAPTATION TASK

The method is tested for a crosslingual model adaptation task using SpeechDat-II fixed telephone databases. Starting point is a set of multilingual speaker independent source models trained on Spanish, English and German. Target language is Slovenian. The multilingual system is trained on phonetically rich sentences of 3000 speaker, 1000 from each language. After state tying the model set consists of 3000 tied states.
We use two adaptation sets with 10 and 20 speakers. Both sets are balanced with respect to sex, they consist of 85 and 170 sentences, respectively. The independent test set, 50 women and 50 men, consists of 614 clauses of phonetically rich words mixed with application words. The resulting grammar, just a word list, comprises 372 words.
For training a monolingual Slovenian reference system we use the phonetically rich sentences of 900 speaker of the Slovenian database.

9. TESTS AND DISCUSSION

The adaptation process starts by predicting initial target models out of the source model tree. We note that only 1832 of the 3000 leaves of the source tree were used by the target language. Next, the measurements $\underline{c}_s$ were calculated by one iteration Baum-Welsh training on the adaptation data. For initialisation we used the predicted

source models.
The number of regression classes was fixed to 100 and the number of latent variables to 30. PLSA was carried out on the $\underline{B}_s$, i.e. assuming equal probable source states.
Table 1 summarises the simulation results for both adaptation sets comprising 10 and 20 speakers, respectively. For η, the critical

Table 1: *Test results with WERs in* [%].

#Speaker	10	20
MONO	9.61	
PRED	46.09	
MEA	49.51	35.18
MLLR	48.21	34.04
MLCR	29.48	29.80
MAPCR_5	27.52	26.38
MAPCR_7	26.71	24.92
MAPCR_9	26.71	23.45
MAPCR_{11}	27.20	23.45

parameter of the prior density, a grid search was carried out. The resulting η-values are given by the subscripts of the MAPCR labels, μ was set to 0.5. The reference result for a monolingual Slovenian system, MONO, is given by 9.61% WER. Using directly the predicted, i.e. not retrained or adapted multilingual models, PRED, lead to a WER of 46.09%. The outcomes obtained by applying the measurements are called MEA. For the results of the adapted models the 95% confidence interval is given by ca. $\pm$3.5% WER.
Comparing the MEA-WERs to the PRED-WER we find that retraining the PRED models by the small adaptation set worsens the situation whereas a substantial gain of ca. 11% WER for the bigger data set is observed. Going on by inspecting the MLLR results we see that MLLR hardly helps. Though, up to 1.30% improvement is obtained it can not compete with the performance of MLCR or MAPCR. Thus, merely adapting the common codebook, as done by MLLR, is barely a good adaptation policy for SCHMMs.
Continuing by applying MLCR to the measurements the WERs drop to 29.48% and 29.80%. This result confirms our assumption that a good solution may be found in $\underline{U}_s$, the solution sub-simplex in the MLCR case. This is further reinforced by the fact that we get nearly identical results for both data sets. On the other hand, the fact that we do not achieve a better result for the bigger data set, which also performs much better in the not adapted case, also indicates that MLCR is not able to take advantage of the better starting point.
Inspecting the remaining results we see that MAPCR is able to overcome this drawback. Though we also observe a gain of $2-3$% WER for the small data set, for the bigger one the WER drops up to 6.35% resulting in the smallest WER of 23.45%. Focusing on η, controlling effectively the shape of the prior, we observe that the best results are obtained for $\eta = 9$ and $\eta = 11$. This corresponds to a quite narrow prior distribution, concentrating its probability mass near zero. In other words, also in the case of MAPCR, the best solutions tend to lie close to $\underline{U}_s$.
Comparing the best adapted result of 23.45% WER with the 9.61% WER of the monolingual reference system, one observes a performance gap of 13.84% WER. This gap is caused by several reasons. First, the amount of adaptation data might just be too small. Second, in case of crosslingual model adaptation one is confronted by a phonetic context mismatch caused by using a decision tree dedicated to another language. Polyphone decision tree specialisation as proposed in [3] might reduce this problem. Finally, we mention that neither MLCR nor MAPCR is able to predict unseen events. The method can only be applied to seen states, respective measurements $\underline{c}_s$. States which did not appear in the adaptation data were not adapted. In this case the original models predicted from the decision tree were used.

10. SUMMARY

This paper describes a novel adaptation framework for SCHMMs. It is based on the projection of a measurement vector to an expected solution space. By incorporating acoustic regression classes the method makes efficient use of prior acoustic information of the source models. Though solutions in a ML as well as in a MAP context are given, maximum a posteriori convex regression (MAPCR) has proven to outperform maximum likelihood convex regression (MLCR) clearly.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Dr. José B. Mariño for his valuable suggestions to this paper, and Dr. Albino Nogueiras Rodríguez for his collaboration in the development and set up of the tools used in this work.

REFERENCES

[1] C. Nieuwoudt and E. C. Botha, "Cross-language use of acoustic information for automatic speech recognition," *Speech Communication*, vol. 38, no. 1, pp. 101–113, 2002.

[2] A. Zgank, Z. Kacic, and B. Horvat, "Comparison of acoustic adaptation methods in multilingual speech recognition environment," *International Conference On Text, Speech and Dialogue*, vol. 2807, no. 6, pp. 245–250, 11 2003.

[3] T. Schultz and A. Waibel, "Language portability in acoustic modeling," *Workshop On Multilingual Speech Communication*, pp. 59–64, 10 2000.

[4] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *International Conference On Acoustics, Speech, and Signal Processing*, vol. 2, no. 2, pp. 291–298, 4 1994.

[5] C. J. Leggetter and P. C. Woodland, "Speaker adaptation of HMMs using linear regression," Tech. Rep., Cambridge CB2 1PZ, 6 1994.

[6] Qiang Huo, Chorkin Chan, and Chin Hui Lee, "Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition," *Transactions On Speech and Audio Processing*, vol. 3, no. 5, pp. 334–345, 9 1995.

[7] F. Diehl, A. Moreno, and E. Monte, "Crosslingual adaptation of semi-continuous HMMs using acoustic regression classes and sub-simplex projection," *COST278 and ISCA Tutorial and Research Workshop (ITRW) On Applied Spoken Language Interaction in Distributed Environments*, 11 2005.

[8] T. Hofmann, "Probabilistic latent semantic analysis," *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[9] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 2 1989.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge CB2 2RU, UK, 2004.

[11] J. B. Mariño, P. Pachès-Leal, and A. Nogueiras, "The demiphone versus the triphone in a decision-tree state-tying framework," *International Conference On Spoken Language Processing*, vol. 1, no. 5, 11 1998.