FAST SPEAKER ADAPTATION USING NON-NEGATIVE MATRIX FACTORIZATION

Jacques Duchateau, Tobias Leroy, Kris Demuynck and Hugo Van hamme

Katholieke Universiteit Leuven, ESAT
Kasteelpark Arenberg 10, 3001 Leuven, Belgium
e-mail: Jacques.Duchateau@esat.kuleuven.be

ABSTRACT

This paper describes a new method for fast speaker adaptation in large vocabulary recognition systems. As in most HMM-based recognizers, the observation densities are modeled as a weighted sum of Gaussian densities. Instead of adapting the means of the Gaussian densities, which is typically done, the weights for the Gaussian densities in the states are adapted. By applying non-negative matrix factorization (NMF) in the proposed method, very fast adaptation was achieved. Experiments on the Wall Street Journal benchmark recognition task show relative improvements between 5% and 15%, while the adaptation converges within 0.2 seconds. Analysis of the latent speakers found by NMF learns that these latent speakers reflect the gender of the speaker most prominently, even when vocal tract length normalization is used, and that they reflect the speaker's age more clearly than the speaker's regional influences or dialect.

Index Terms— Speech recognition, adaptive systems, speaker adaptation, matrix decomposition, non-negative matrix factorization.

1. INTRODUCTION

In acoustic modeling, the term *adaptation* (of the acoustic model to the incoming speech to be recognized) often refers to two distinct phenomena. In the first type of adaptation, the model *specializes* for some specific *situation* it was trained on: depending on the training database, this *situation* may refer to a speaker, speaker dialect, speaking style, recording environment (channel, noise), etc. The second type of adaptation refers to adapting to a new situation the model was not trained for: the distributions in the model should *move* from the situation trained on, to the situation at hand.

In this paper, the experiments are based on the speaker independent Wall Street Journal (WSJ) benchmark recognition task. Therefore the acoustic model training data contains many speakers but only one environmental condition. Therefore we'll call adaptation that has to specialize *speaker adaptation*, and adaptation that has to move distributions *environment adaptation*.

The top of figure 1 depicts two typical cases (in two dimensions). Starting from the original distribution of the training data for some HMM state on the left, the middle shows the case in which the incoming speech matches the training data so only speaker adaptation is needed. To the right, there is no match with the training data and therefore both speaker and environment adaptation are needed.

In HMM-based acoustic modeling in which states are modeled as a weighted sum of Gaussian probability density functions (pdfs), adaptation techniques that change mean (and possibly covariance) of the Gaussians pdfs are plain text book material (see e.g. chapter 9.6 in [1]). What happens is shown in the middle of figure 1 for both cases (without and with environment adaptation). The crosses represent the means of the Gaussian pdfs, and if adaptation works well, they move for both cases to the distribution that is to be modeled.

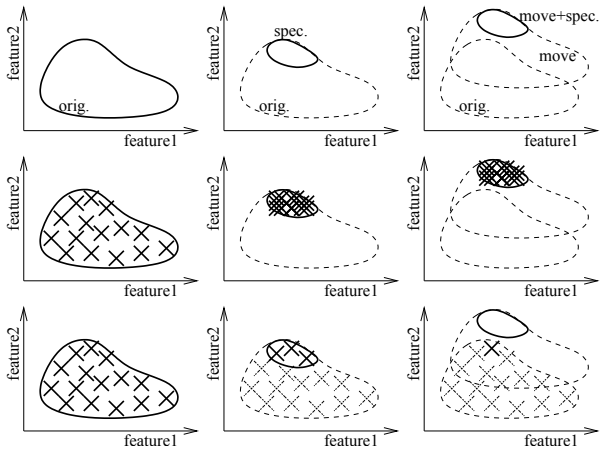

Fig. 1. Types of adaption in acoustic modeling

Adaptation techniques that change the weights in the weighted sum of Gaussians pdfs that models the HMM state, got only little attention in literature. One reason probably is that only recent, detailed acoustic models, with many Gaussians pdfs per state, can benefit from weight adaptation. Recently some papers appeared that base weight adaptation on non-negative matrix factorization (NMF), a technique suitable for matrices with (non-negative) weights.

In [2], the matrix of weights that is decomposed by NMF, is organized so that adaptation in the phonetic space results. The proposed method basically provides an elegant re-estimation of a large, detailed model based on a task specific database of any size, as long as no (severe) environment adaptation is needed.

In [3], the weights (or more precisely directly related values) in the matrix are organized per speaker, so only speaker adaptation is possible. In fact, no real model adaptation is used: information gained from the matrix decomposition is applied directly for rescoring hypotheses during search. The viewpoint of the paper is to provide a solution to speech trajectory folding in HMM-based systems (due to statistical modeling per state): it is argued that automatically generated trajectories mainly correlate to speaker variation.

While the work in this paper is closely related to the work in [3] (so the interpretation as a solution to trajectory folding still holds), we aim at pure speaker adaptation based on the weights in the acoustic models. The effect of the adaptation is shown at the bottom of figure 1. In this case, the means of the Gaussian pdfs can't move. However weights may become (almost) zero, as indicated by gray,

The research in this paper was supported by the IWT project SPACE (sbo/040102): SPeech Algorithms for Clinical and Educational applications, home page: http://www.esat.kuleuven.be/psi/spraak/projects/SPACE.

dashed crosses in the figure. It is clear that speaker adaptation is possible with this method, however environment adaptation is not (or at least not in all cases).

Even though adaptation of the means of the Gaussian pdfs is theoretically more powerful (allowing both speaker and environment adaptation) than weight adaptation (allowing only speaker adaptation), it also has some drawbacks. First, the extra power may lead to adaptation that deteriorates the modeling, e.g. in unsupervised mode. Second, mean adaptation results in one, total transform for both speaker adaptation and environment adaptation. When a meeting is to be transcribed, and a new speaker comes in, it's useful to initiate a new model that is adapted to the environment only. Furthermore, since the proposed weight adaptation is very fast (as will be shown), it may be used as a first step, based on which slower adaptation methods can be applied.

This paper is organized as follows: in sequence we describe the proposed system (section 2), the setup (section 3), the results (section 4) and analyses (section 5) of the experiments, and finally conclusions and future research (section 6).

2. SYSTEM DESCRIPTION

2.1. Overview

The general idea of the proposed method is to model a test speaker as the optimal linear combination of a number (e.g. 10) *latent* (or *base*) speakers given the incoming speech. These latent speakers result from a matrix decomposition on a matrix V containing the *reference* speakers, these are the (e.g. 100) speakers in the acoustic model training database.

This matrix V is constructed so that every column of the matrix contains a model for one of the NR reference speakers in the training database. A speaker model consists of a concatenation of all weights $v(k)$ for all NS states s and for all NC$_s$ components in the weighted sum of Gaussian pdfs that models state s, with $1 \leqslant k \leqslant$ TNC. TNC is the total number of weights for all states: TNC $= \sum_{s=1}^{\text{NS}} \text{NC}_s$, this ranges typically from 10^5 to 10^6. The speaker dependent weights for a reference speaker result from a re-estimation of the speaker independent weights based on a forced alignment (using the speaker independent model) of the training data for that speaker. Note that it doesn't matter if some (or all) Gaussians are tied over different (or all) states: when a Gaussian pdf is re-used in different states, it's weight in those states is not tied.

Then NMF is used to decompose the TNC $\times$ NR matrix V in the product of a TNC $\times$ NL matrix W and a NL $\times$ NR matrix H. Each column of W is one of the NL latent speakers, NL is a system parameter. Each column of H contains the weights in the linear combination of the latent speakers that approximates the corresponding column in V. Note that only approximates of the reference speaker models are found as NL is (normally) chosen smaller than NR so the decomposition is not exact.

Now the idea is to model the test speaker as a linear combination of the latent speakers. The same idea is used in eigenvoice-based adaptation of the means of the Gaussian pdfs [1], however the restrictions on weights (being positive and summing up to 1) lead to a constrained, convex optimization problem hence to different algorithms to solve it. Based on the incoming data, the NL weights in the linear combination should be estimated, then the adapted weights for the test speaker are a weighted sum of the weights for the latent speakers. As only few numbers have to be estimated for adaptation, we may expect the system to be very fast. In the experiments in this paper, for the weights the Maximum Likelihood (ML) estimate is used, which is calculated iteratively.

The remainder of this section details the matrix decomposition and the weight estimation.

2.2. Matrix decomposition

Depending on the chosen distance measure between V and its approximation $\widetilde{V} = W \cdot H$, and on the chosen iterative algorithm to minimize this distance, different NMF algorithms are possible. We used three methods described in literature. In [4] and [5], the Euclidian distance

$$d(V, \widetilde{V}) = \| V - \widetilde{V} \|^2 = \sum_{k=1}^{\text{TNC}} \sum_{r=1}^{\text{NR}} (V(k,r) - \widetilde{V}(k,r))^2 \qquad (1)$$

is minimized, this is the Mean Square Error (MSE) criterion. We'll call the iterative algorithm in [4], based on multiplicative update formula's, the MSE1 algorithm, and the algorithm in [5], using additive updates based on the steepest descent method with projected gradient, MSE2. In [4], also the generalized Kullback-Leibler divergence is proposed as distance measure:

$$d(V, \widetilde{V}) = \sum_{k=1}^{\text{TNC}} \sum_{r=1}^{\text{NR}} V(k,r) \log \frac{V(k,r)}{\widetilde{V}(k,r)} - V(k,r) + \widetilde{V}(k,r) \qquad (2)$$

The NMF based on this divergence, using multiplicative updates, will be called DIV. In [6] it is shown that using this divergence, NMF is equivalent to PLSA (Probabilistic Latent Semantic Analysis).

2.3. Adaptation algorithm

Let
$$\widehat{h_a} = \arg\max_{h_a} P(O|h_a) \qquad (3)$$

be the ML estimate for observation O of the NL weights h_a that define the adapted model v_a. The EM-algorithm then states that $Q(h_a, \widehat{h_a})$ needs to be maximized iteratively. Corresponding to text book formula's for Baum-Welch, for weight re-estimation Q-function

$$Q(h_a, \widehat{h_a}) = \sum_{t=1}^{\text{T}} \sum_{k=1}^{\text{TNC}} \gamma(k,t) \log(v_a(k)) \qquad (4)$$

is to be maximized, with T the number of frames in the observation and $\gamma(k,t)$ the posterior probability for weight k at time t.

Given the relation between h_a and v_a, equation 4 becomes

$$Q(h_a, \widehat{h_a}) = \sum_{t=1}^{\text{T}} \sum_{k=1}^{\text{TNC}} \gamma(k,t) \log\Big[\sum_{l=1}^{\text{NL}} W(k,l) h_a(l)\Big] \qquad (5)$$

with restriction $\sum_{l=1}^{\text{NL}} h_a(l) = 1$. Applying Lagrange multipliers (with parameter α) to this restricted optimization problem, and after partial derivation of this Q' to the unknowns, we get

$$\begin{cases} \frac{\partial Q'}{\partial h_a(l)} & = 0 = -\alpha + \sum_{t=1}^{\text{T}} \sum_{k=1}^{\text{TNC}} \frac{\gamma(k,t) W(k,l)}{\sum_{j=1}^{\text{NL}} W(k,j) h_a(j)} \\ & \qquad\qquad (l = 1, \ldots, \text{NL}) \\ \frac{\partial Q'}{\partial \alpha} & = 0 = \sum_{l=1}^{\text{NL}} h_a(l) - 1 \end{cases} \qquad (6)$$

which is a set of NL$+1$ non-linear equations for which no analytical solution could be found. Therefore an iterative optimization scheme was adopted, using initial values for $h_a(l)^{(0)}$ equal to $1/\text{NL}$, and with a multiplicative update formula for the i-th iteration as follows:

$$h_a(l)^{(i+1)} = \sum_{t=1}^{\text{T}} \sum_{k=1}^{\text{TNC}} \frac{\gamma(k,t) W(k,l)}{\sum_{j=1}^{\text{NL}} W(k,j) h_a(j)} \times h_a(l)^{(i)} \qquad (7)$$
$$(l = 1, \ldots, \text{NL})$$

NL	MSE1	MSE2	DIV	NL	MSE1	MSE2	DIV
2	5.10%	5.07%	5.12%	8	4.99%	5.17%	5.05%
3	5.07%	4.90%	4.82%	12	5.20%	5.04%	4.95%
4	5.05%	5.07%	5.07%	50	4.99%	5.00%	5.05%
6	4.97%	4.97%	4.92%	84	5.06%	NA	NA

Table 1. WER for different NMF algorithms and choices of NL

3. EXPERIMENTAL SETUP

The proposed adaptation method was investigated on the WSJ benchmark large vocabulary recognition task: 5k word closed vocabulary (so no out-of-vocabulary words), standard bigram language model, November 92 evaluation test set. The feature extraction is Mel-spectrum based, including mean subtraction on the log spectrum, and a discriminant analysis resulting in 39 features. The baseline model doesn't incorporate Vocal Tract Length Normalization (VTLN).

Training of the acoustic models is based on the standard SI-84 WSJ0 database which contains about 15 hours of speech in total from 84 speakers. Speaker independent, cross word context and position dependent acoustic models were generated with 1938 tied HMM states, defined by an automatic phonetic decision tree for 45 phones. In total 17209 tied Gaussian pdfs (with diagonal covariance) were estimated. We use a flexible tying system for the Gaussian pdfs, with on average 88.7 components in the weighted sum for the tied states, so TNC equals 171933.

To define the adaptation data, the data for each of the 8 test speakers in the November 92 evaluation test set was split: the first 10 sentences (containing about 70 seconds per speaker) were selected for adaptation, the remaining (about 30) sentences for the speaker, which we will call the test set in the following, were used to evaluate the adapted models. For the speaker independent (SI) models, the Word Error Rate (WER) on this test set equals 5.71%.

The adaptation in the experiments in this paper is both supervised (the orthographic transcript of the adaptation data is known) and off-line (first for each speaker the NL weights for the latent speakers are estimated based on the adaptation data, then the speaker adapted model is constructed and evaluated on the remaining sentences for that speaker in the test data).

4. EXPERIMENTAL RESULTS

4.1. Baseline results

Table 1 presents the results when using all available adaptation data for different choices of the number NL of latent speakers and for the three investigated NMF algorithms (see section 2.2). The table shows significant relative improvements of about 15%, while the method is robust to both the choice of NL and NMF algorithm.

4.2. Adaptation speed

In order to assess the speed of adaptation, we did the following experiment. Instead of using all the adaptation data for a speaker, segments of only 10 msec (1 frame) up to 1 second were used. In order to get an indication of the variability on the WER results, we selected 20 different segments of adaptation data randomly for each segment length (but assuring they contain speech, adaptation should not be based on silence). The circles in figure 2 show the average WER over the 20 measurements (for the 20 different segments) depending on the amount of adaptation data. The triangles indicate the best and worst result out of the 20 measurements. For segments of 50 msec or longer, the system always improved the WER over the SI model.

We also see that the adaptation converges after about 0.2 seconds, which is very fast. Even for 2 or 1 frames of adaptation data, the result is never terribly bad: it can be anticipated that a low latency implementation of (e.g. frame by frame) weight adaptation will not diverge from the optimal solution. It should be noted that in order to use this very fast weight adaptation, unsupervised adaptation is needed rather than the supervised version investigated in this paper. This also means that the posterior probabilities for the components of the tied state models cannot be calculated using the Baum-Welch algorithm as explained in section 2.3, they should be estimated from the hypotheses in the search directly.

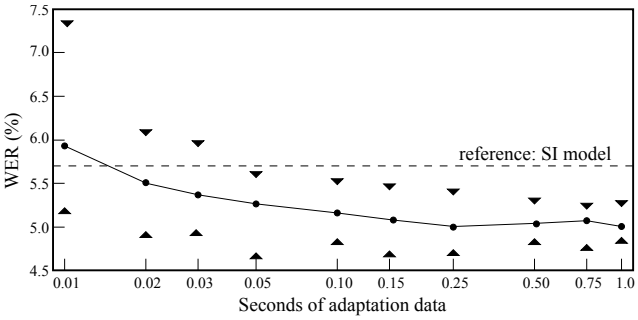

Fig. 2. WER versus amount of adaptation data

Model	measure1		measure2	
SI model	0.52		49.2	
reference speakers	0.79		7.4	
	MSE1	DIV	MSE1	DIV
2 latent speakers	0.62	0.62	33.1	33.4
8 latent speakers	0.68	0.66	19.0	21.6
20 latent speakers	0.74	0.70	12.5	16.5
test speakers, NL = 2	0.59	0.59	38.5	38.2
test speakers, NL = 8	0.61	0.61	34.5	32.1
test speakers, NL = 20	0.62	0.62	31.7	30.2

Table 2. Sparseness measures for different models

5. FURTHER ANALYSES

In this section, we analyze sparseness of the models, and interpret the latent speakers as clusters of speakers with similar properties.

5.1. Sparseness of the adapted weights

When speaker adaptation is obtained by adapting the weights, we expect both high weights (for components suitable for the test speaker) and low weights in the adapted model. The more a weight vector is *sparse*, the better the model is specialized for a specific situation. We investigated two measures for the *sparseness* of a model v.

The first is the normalized ratio of the ℓ_1-norm over the ℓ_2-norm:

$$\frac{\sqrt{\text{TNC}} - (\sum_{k=1}^{\text{TNC}} |v(k)|)/\sqrt{\sum_{k=1}^{\text{TNC}} v(k)^2}}{\sqrt{\text{TNC}} - 1} \tag{8}$$

as defined in [4]. This measure ranges from 0 for equal weights to 1 if only one weight differs from zero.

For the second measure, we sort weights in every state from large to small, count the number of components needed to cover 90% of the state's density, and average this number over all states. The lower this average, the sparser the model.

Table 2 presents both measures for the SI model, and (on average) for reference, latent and test speakers. We can see that the sparseness of test speakers is closer to that of the SI model than to that of reference speakers. Sparseness is lost in two steps: first due to decomposition, especially for a low number of latent speakers, second by combining latent speakers, especially for high NL. This is

	no VTLN		with VTLN	
	male	female	male	female
weight latent speaker 1	0.94	0.08	0.75	0.17
weight latent speaker 2	0.06	0.92	0.25	0.83

Table 3. Average latent speaker weight for gender classes

unfortunate as sparseness reflects the level of *specialization* towards the current speaker. So we tried to increase sparseness in both steps, as this may enhance the resulting model's performance.

The sparseness of the latent speakers was increased using an algorithm, described in [4], that allows to control the first sparseness measure (equation 8) during decomposition. In case of 8 latent speakers, sparseness measure 1 could be increased from 0.68 to 0.75 without a loss in WER, this results in a drop from 34.5 to 27.0 for the second measure for the test speakers.

We tried to reduce the loss of sparseness due to combining of latent speakers by putting weights for latent speakers that are below a threshold to zero after every iteration in the multiplicative weight update formula (see equation 7). Without loss in recognition performance, the number of selected latent speakers (out of 8) could be reduced to 2 on average, resulting in a drop from 34.5 to 25.9 for measure 2 (test speakers).

Unfortunately neither method to increase sparseness (nor their combination) improves recognition. Still there may be an advantage: the decrease of the second measure indicates that we can generate more compact (thus faster) speaker dependent models by removing components for a state if their weight is below some threshold.

5.2. Interpretation of the latent speaker bases

In this section, we try to understand what speaker variation is modeled by the latent speakers. One obvious origin of variation is the gender of the speaker. Therefore we did an experiment on weight adaptation for acoustic models in which VTLN is included in the feature extraction. This VTLN, described in [7], allows on-line adaptation with no latency, so that it can be used when the proposed speaker adaptation is implemented in a low latency, unsupervised setup. The VTLN is supposed to remove (most of) the variation by gender. We found a 5.18% WER for the baseline SI model, and a 4% relative improvement to (on average for different NL) 5.00% for the weight adapted models. This shows that part of the improvement found by weight adaptation is due to adaptation to gender, but part of the improvement is complementary to VTLN.

This is also shown by investigating the weights in matrix H. Each column in H contains the weights for the latent speakers in W to approximate the reference speaker in the same column in V. If we have a class of speakers in the training data, we can find the latent speaker(s) by which this class is mainly modeled by averaging the weights in H for all speakers in the class. If we do so for the classes male and female in the SI-84 WSJ0 training database, using 2 latent speakers, we find the average weights in table 3. We can see that, without VTLN, the male and female speakers are modeled almost perfectly with the first and second latent speaker respectively. With VTLN in the feature extraction of the acoustic models, this effect is smaller, but the gender is still clearly reflected in the latent speakers.

We also investigated to what extent the latent speakers reflect speaker age and speaker dialect (as specified by the region where the speaker went to primary school). Therefore we used the SI-284 WSJ1 training database, containing an additional 200 speakers, as only for those speakers information about age and region is available. For this training database, using the same adaptation and test data and the same recognition task as described before, the SI models (without VTLN) result in a 4.78% WER. This is improved by 9% relative using the weight adaptation: a WER (on average for different NL) of 4.40% is found. In the below experiment, we wanted to avoid the effect of the gender (as this is dominant for the WSJ database), so a matrix decomposition was done on the female speakers only (the same effect was seen for the male speakers). The 2 classes for age were *below 25 years* and *above 50 years*, the 2 classes for region were *east coast* and *west coast* (in both cases excluding part of the speakers). The results are given in table 4, the main conclusion is that the weights are clearly more polarized towards age. An overall conclusion is that the weight adaptation is capable of modeling things other than gender differences.

	below 25	over 50	west	east
weight latent speaker 1	0.67	0.19	0.52	0.67
weight latent speaker 2	0.33	0.81	0.48	0.33

Table 4. Average latent speaker weight for age and region classes

6. CONCLUSIONS AND FUTURE RESEARCH

In this paper, a new method was proposed for fast speaker adaptation in large vocabulary recognition systems: the weights for the Gaussians pdfs in the HMM states are adapted. By applying NMF, relative improvements of about 15% could be achieved while the adaptation converges within 0.2 seconds. However in acoustic models that include VTLN, the additional improvement of the weight adaptation drops to about 4% relative.

As future research, we intend to make the speaker adaptation more fine grained using phoneme dependent weighting of latent speakers as a speaker may show properties of one latent speaker for some phoneme cluster but properties of an other latent speaker for other phonemes. It's also interesting to know if the proposed method can adapt to the environment when a multi-condition training database is provided as in the Aurora-4 benchmark. Furthermore the complementarity can be investigated of the proposed *fast*, pure speaker adaptation technique, and the more common adaption methods that work on the means of the Gaussians pdfs, especially in a framework in which low latency, unsupervised adaptation is necessary.

7. REFERENCES

[1] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing. A Guide to Theory, Algorithm, and System Development*, Prentice Hall, New Jersey, U.S.A., 2001.

[2] Antoine Raux and Rita Singh, "Maximum-likelihood adaptation of semi-continuous HMMs by latent variable decomposition of state distributions," in *Proc. ICSLP*, Jeju Island, Korea, Oct. 2004.

[3] Dan Su, Xihong Wu, and Huisheng Chi, "Probabilistic latent speaker analysis for large vocabulary speech recognition," in *Proc. EUROSPEECH*, Antwerp, Belgium, Aug. 2007, pp. 1162–1165.

[4] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[5] Chih-Jen Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[6] Eric Gaussier and Cyril Goutte, "Relation between PLSA and NMF and implications," in *Proc. SIGIR conference on Research and Development in Information Retrieval*, Salvador, Brazil, Aug. 2005, pp. 601–602.

[7] Jacques Duchateau, Mari Wigham, Kris Demuynck, and Hugo Van hamme, "A flexible recogniser architecture in a reading tutor for children," in *Proc. ITRW on Speech Recognition and Intrinsic Variation*, Toulouse, France, May 2006, pp. 59–64.