Linear transformation of Hidden Markov Models based on linear regression

Melin, H.

journal: STL-QPSR
volume: 35
number: 1
year: 1994
pages: 025-046

http://www.speech.kth.se/qpsr
Linear transformation of Hidden Markov Models based on linear regression

Håkan Melin

Abstract

This paper treats a linear transformation of word templates in a word recognition system. The object of the transformation, called LMR-transform, is to adapt the recogniser to a new acoustical environment. The transform is derived by linear regression on pairs of word utterances from two different acoustical environments. The use of the transform has been evaluated for recognition accuracy, speaker-independence and training data requirements in three different fields of application. The main field is noise adaptation in a car-environment. Other fields involve microphone and speaker adaptation. A database recorded in a car incorporating six speakers using five different microphones has been utilised for evaluation. The transform increased recognition accuracy with all of the evaluated fields of application.

Introduction

The ultimate speech recogniser would have human capabilities. It would have to be almost insensitive to noise, to who the speaker is, or to the way the speaker talks. Being able to transform models of speech to fit any noise environment, any speaker, and any speaker style, would be one step towards this ultimate speech recogniser.

Training a speech recogniser is an unwanted procedure. Training in some environments is even uncomfortable. For instance, training a recogniser for use by the driver of a car would be a danger to traffic unless someone else were driving the car. The amount of training needed to get an operational recogniser should be minimised generally, and for some environments particularly, such as in cars.

There is a basic condition that must be met for a transform approach to be usable. The number of recorded utterances from the new environment needed to learn the transformation has to be less than the number of utterances needed to do a re-training of the speech models. If this condition is not met, it would be better to do a re-training unless the transform approach gives a higher recognition accuracy, which is not likely (at this point). Ideally, no new recordings of spoken utterances would be needed. The transform could be derived from knowledge about the training and the recognition environments. This knowledge could be based on noise measurements, microphone characteristics, or knowledge of the speech production mechanisms.

This paper treats a simple implementation of the transform approach. The transform is called LMR-transform, and defines a linear transformation of the speech models in an HMM-based recogniser. The transform is generally learned from two recordings of the same spoken utterance in two different environments, a source environment and a target environment. 'Environment' refers to an acoustical environment with at least three dimensions: the category and level of noise, the type of microphone being used to pick up the speech signal, and the characteristics of the speaker.
The LMR-transform has been described by Mokbel and Chollet (1991, 1992). Some modifications to the original transform and some further developments are presented herein. Fields of application for the LMR-transform, where speech models are adapted to environmental changes in each of the three dimensions of the acoustical environment, are evaluated.

The use of the LMR-transform is evaluated for recognition accuracy, speaker-independence, and training data requirements, in three fields of application: noise, microphone, and speaker adaptation. A multi-speaker, multi-microphone database is used for evaluation.

This paper is a shorter, revised version of a Master's Thesis (Melin, 1994). The thesis includes a literature study on speech recognition robust to changes in the acoustical environment. The reference list from the literature study has been included in this paper as a bibliography.

The work reported on in this paper was done within a project called VOICE (Granström et al. 1993), in which an experimental voice-based traffic information provider for car drivers has been developed.

Some terminology

The LMR-transform is defined as a means to transform Hidden Markov Models between acoustical environments, a source and a target environment. The acoustical environment is represented mathematically by a vector subspace spanned by a large set of sample feature vectors. This vector subspace is called an acoustical subspace, \( AS \). The source environment \( E_S \) is represented by \( AS_S \) and the target environment \( E_T \) is represented by \( AS_T \). Basically, the transformation described is based on a linear mapping between \( AS_S \) and \( AS_T \) derived by linear regression. The term acoustical subspace and the basic transformation have been presented by Mokbel and Chollet.

A notation for environment configuration in recognition tests is defined as

\[
E_R: E_S \rightarrow E_T
\]

where HMMs are trained in \( E_S \), transformation is computed with \( E_S \) as source and \( E_T \) as target environment, and recognition is performed in \( E_R \). Speaker and microphone labels, defined in the Database-section, are used in defining the environments.

Example: 2akgv:3sva→3svaw denotes: HMMs are trained on speaker 3 with the sva-microphone in a clean environment, and models are transformed with a transformation computed from the clean to the noisy environment using the same speaker and microphone. Recognition is performed on speaker 2 using the akg-microphone in a noisy environment.

Two different kinds of "training" will be treated in this work: training of HMMs and training of an LMR-transform, where training means that a mathematical model (HMM or LMR-transform in this case) is estimated out of a set of data. To avoid misunderstandings, the word learning will be used in connection with the LMR-transform, and the word training in connection with HMMs.
Method

The LMR-transform

\[ T(A, M_s, M_r) : \ Y = A(X - M_s) + M_r \]  

is defined on acoustical subspaces. A source-AS, \( AS_s \), with vectors \( X \) is transformed into a target-AS, \( AS_t \), with vectors \( Y \). The transform is defined to map \( AS_s \) onto \( AS_t \) in a minimum-squared-error sense. \( M_s \) is the average of all vectors in \( AS_s \) and \( M_r \) is the average of all vectors in \( AS_t \).

The LMR-transform is applied to mean vectors of HMM:s. Let \( X_{HMM} \) be the mean vector for one state in an \( E_s \) word model and \( Y_{HMM} \) the mean vector of the corresponding state in the \( E_t \) word model. Let \( Y'_{HMM} \) be the HMM resulting from the transformation of \( X_{HMM} \)

\[ Y'_{HMM} = A(X_{HMM} - M_s) + M_r. \]  

The purpose of the transformation is to produce a vector \( Y'_{HMM} \) that approximates the vector \( Y_{HMM} \), that is, to transform an HMM modelling an item in the \( E_s \) into an HMM modelling the same item in the \( E_t \).

Thus far, the transformation described herein has been described before by Mokbel and Chollet. They also applied the transformation to the variance vector. In Melin (1994), the operation of the variance transformation degraded recognition performance. Hence, variance transformation is not used in the experiments presented in this paper.

Adaptation

In Melin (1994), the operation of \( M_s \) and \( M_r \) in equation (3) was shown to correspond to a normalisation of the linear domain filter bank coefficients. This led to an improvement of the transformation: \( M_r \) was replaced by \( M_R \) in equation (3), where \( M_R \) is the time average of the actual recognition sequence over an interval of \( w_m \) words. The new transformation has a static part and a dynamic part,

\[ Y_{stat}' = A(X_{HMM} - M_s) \]  

\[ Y'_{HMM} = Y_{stat}' + M_R. \]  

The static part (4) can be performed once and for all before the recognition phase starts, and the dynamic part (5) may be performed repeatedly during recognition.

\( M_R \) in equation (5) is adaptive and is called the running mean. The running mean is an estimation of the time-average vector over the vectors belonging to the \( w_m \) latest recognised words. The running mean is recomputed right before every new word, when a push-to-talk button (described in the Recogniser-section) is pressed. It is held constant during the recognition of the word.

When the push-to-talk button is pressed for the \( i \)th word to be recognised, a local mean vector \( L_i \) is computed. This local mean is based solely on the vectors belonging to the most recently spoken word. The running mean is then computed as
\[ M_{R,j} = \frac{M_{R,j-1}(w_m-1)+L_j}{w_m} \]

Different values for \( w_m \) were evaluated by Melin (1994). The running mean was based on the average of the 1, 2, or 5 latest spoken words, with a slight improvement in recognition accuracy with an increasing number of words. A running mean based on 5 words has been used in the experiments presented in this paper.

**Aligning utterances**

The utilised speech database contains two parts as described in the *Database*-section: a loudspeaker-part and a live-part. The loudspeaker-part contains the same instance of speech replayed through a loudspeaker in different acoustical environments. This simplifies the procedure for aligning speech sequences prior to the computation of a transform. Only the difference in starting-time between a transcribed reference speech sequence and a test speech sequence has to be found for one to be able to generate a transcription for the test sequence. The time difference is called lag and is measured in frames. Once the lag is known, an entire sequence can be aligned. This is called hard alignment.

When aligning utterances that are not recorded from the same instance of speech, it is not possible to do hard alignment. This is the case if one wants to align either utterances from two different speakers or a noise-free and a noisy recording from the live-part of the database. In these cases, a dynamic programming method is used to perform soft alignment of speech vectors.

The transform computation is done in the filter bank domain. Hence, for simplicity reasons, the alignment methods also operate on filter bank vectors. The alignment methods are further described in Melin (1994).

**Special cases: the LR-transform and the EYE-transform**

If the effects of noise in different filter bank channels are considered decorrelated, a simple transform for each component in the filter bank vector can be computed. The transform is learned by linear regression on each filter bank channel independently. The A-matrix of equation (2) will be diagonal. This transform is called an LR-transform, where LR is an abbreviation for Linear Regression. It is a special case of the general LMR-transform learned by Linear Multiple Regression. If the decorrelation criterion is not used, the A-matrix will be full. Each component in the vector resulting from transformation will then depend on all the components of the source vector.

In order to evaluate the effect of the A-matrix at transformation, comparisons will be made with a transform where \( A = 1 \) (unity matrix). This transform is considered a (trivial) special case of the LMR-transform, and will be called the EYE-transform. The EYE-transform will be used with an adaptive running mean to isolate the effect of the A-matrix of an LR/LMR-transform.

**Recogniser**

The basic tools used for HMM training, recognition and result evaluation are part of a toolkit called Hidden Markov Model Toolkit, HTK, ver 1.4 (Young and Woodland,
HTK was made for building subword-based continuous speech recognisers. In this application, it is used in recognition of isolated words. The syntax is thus defined as

\{silence\} word \{silence\},

where curly brackets denote zero or more instances. The original recognition tool, HVite, has been modified to simulate the use of a push-to-talk button. The push-to-talk (PTT) recognition solves the problem of detecting what words are actually intended for the recogniser, as a button is pressed for each of these words.

A 53-Swedish-word vocabulary, designed for controlling a traffic information system in a car, is used. Each word is modelled by a single HMM. The number of states for each model is manually chosen based upon the number of phonetical segments in the corresponding word. The number of emitting states vary from 8 to 24 within the set of HMMs. All models have a strict left-to-right topology with no skips. The models are continuous density HMMs with single-mixture modelling. Each state is modelled by a mean vector and a variance vector (a diagonal covariance matrix). A fixed variance scheme is used, where all states of an HMM share the same variance vector. In Melin (1994), some experiments were conducted to compare recognition performances when using either fixed variances or ordinarily trained variances.

Two categories of feature vectors are used, filter bank and cepstrum vectors. The speech signal is sampled at 16 kHz. The waveform samples are processed by a digital filter bank with Bark-scale frequency spaced channels, producing logarithmic-energy filter bank vectors with 16 elements. A 25.6 ms Hamming window is used. The frame interval is 10 ms. 16-element cepstrum vectors are computed from the filter bank vectors using a cosine transform. The zero'th cepstrum coefficient (energy) is not used.

Recognition is performed in the cepstrum domain and the transformation of HMM mean vectors is performed in the logarithmic filter bank domain. There are two reasons for this. Filter bank channel independences are not preserved in the cepstrum domain, as each of the cepstrum coefficients depends on every filter bank channel. Also, the filter bank vectors are more intuitive and, thus, it is easier to understand and to visualise the transform if the filter bank domain is used for transformation. The training, transformation, and recognition procedures are further described in Melin (1994).

**Database**

The speech database used for the evaluation of different methods, is thoroughly described by Roxström (1992). The database contains material from six different speakers (S1, ..., S6) reading 10 lists of words, each containing the 53 words of the vocabulary. Three of the speakers are male speakers (S2, S3, and S4), and the other three are female speakers (S1, S5, and S6). There are two different kinds of recordings, loudspeaker and live recordings. Both kinds include recordings made in clean (when the car was in a garage) and noisy environments (when the car was driven with a speed of nominally 90 km/h), using a number of different microphones.
Most of the speech in the database was recorded on tape in a studio. The tape was then replayed through a loudspeaker, mounted on the head restraint of the passenger seat in the car, while all the microphones were mounted on the dashboard, registering the speech in parallel with each other. Consequently, the recordings with the different microphones all contain identical speech. This part of the database will be called the loudspeaker-part. The other part of the database will be called the live-part. It contains live-recordings made directly in the car. The live-part includes speakers S2 and S3 and a subset of the microphones.

Three different kinds of microphone elements were used when recording the database: a high-quality studio microphone, a goose-neck microphone, and a set of eight miniature microphone elements. The eight miniature elements were used in an adaptive array microphone (Nordholm et. al., 1993). Two different signals were extracted from the array microphone. Firstly, the spatially filtered signal - the actual output of the array microphone. This signal is bandwidth-limited to 340 Hz to 3400 Hz (the bandwidth of a telephone line). Secondly, element no. 3 was recorded. This signal is not limited in bandwidth. It is called the central microphone. Some aspects of using the array microphone are described by Melin (1994).

Out of the microphone elements, five logical microphones have been defined according to table 1. The name of a logical microphone alone denotes a recording made in a clean environment. One or both of the letters v and l may be appended: v to symbolise a recording made in a noisy environment, and l to symbolise a live recording.

Example: 'akgv1' will denote a live recording in a noisy environment with the AKG-microphone.

<table>
<thead>
<tr>
<th>Label</th>
<th>microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>akg</td>
<td>the high-quality studio microphone</td>
</tr>
<tr>
<td>sva</td>
<td>the goose-neck microphone</td>
</tr>
<tr>
<td>mit</td>
<td>the central miniature microphone</td>
</tr>
<tr>
<td>eps</td>
<td>the array microphone with filtering algorithm no. 1</td>
</tr>
<tr>
<td>e3p</td>
<td>the array microphone with filtering algorithm no. 3</td>
</tr>
</tbody>
</table>

Table 1. Logical microphones, or microphone labels, used in the database.

The database has formerly been used by Roxström (1992) and Rydin and Udin (1993) for evaluating recogniser performance in a car environment.

**Database usage**

The HMM:s of the recogniser are trained using the five odd-numbered lists for the appropriate speaker and microphone from the training (source) environment. The LR/LMR-transform is computed on the words of the odd-numbered pairs of lists from the source and the target environments. Thus, the same number of uttered words is used for transform-learning as for HMM-training (in fact, twice as many words as pairs of lists are used for transform-learning). The reason for this seemingly over-rated size
of the learning data is to eliminate recognition errors originating from a possibly under-determined transform. Consequently, these results shall be considered maximum results with regards to the size of the learning data. Recognition is done on the five even-numbered lists from the recognition (target) environment.

Tests and results

This section will present three fields of application for the LR/LMR-transform and test results for each of these fields. These fields include noise adaptation, microphone adaptation, and speaker transformation. They each represent transformations for changes in one of the dimensions of the acoustical environment.

Evaluation methods

Performance of the recogniser and the LR/LMR-transform are evaluated for recognition correctness, speaker-independence, and learning data requirements. The evaluation methods will now be described.

Results

As push-to-talk recognition is used, and the recogniser is forced to choose exactly one word from the vocabulary, neither insertions nor deletions will be present in the recognised sequence. Hence, the term recognition correctness, \( C \), will be used for describing the result of a recognition simulation. \( C \) is evaluated as

\[
C = \frac{\text{number of words correctly recognised}}{\text{total number of words}}
\]  

(7)

Speaker-independence

In the noise adaptation and microphone adaptation sections, it is important to evaluate the speaker-independence of the transform. Speaker-independence in this case means that a transform learned from utterances spoken by one speaker can be successfully used in recognition of speech from any other speaker.

To evaluate speaker-independence of a transform-type, a series of recognition simulations are carried out, one per each of the 36 squares in figure 1. The speaker for which recognition and HMM-training is done, is called the recognition-speaker. The speaker used for learning the transform is called the transform-speaker. The result for any combination of recognition-speaker \( r \) and transform-speaker \( x \) is denoted \( C_{r,x} \), and corresponds to the square with row-number \( r \) and column-number \( x \) in figure 1.

The diagonal of the matrix in figure 1 contains results from recognition experiments where the recognition-speaker and the transform-speaker are the same person. The average of the diagonal elements

\[
\bar{C}_{r,x} = \frac{1}{N_S} \sum_{s=1}^{N_s} C_{r,s}
\]  

(8)

is the average recognition result with speaker-dependent transform, where \( N_S \) is the number of speakers. The average recognition result with speaker-independent transform is
\[ \overline{C}_{rs} = \frac{1}{N^2 - N^2_{rs}} \sum C_{rs} \] \hfill (9)

This is the most interesting result. It is equivalent to the average of the results from all unshaded squares in figure 1, where the recognition-speaker and the transform-speaker are different people. The difference

\[ SI = \overline{C}_{rs} - \overline{C}_{rsx} \] \hfill (10)

will be used as an overall measure of the speaker-independence of a transform for a given application.

The average of all results in one transform-speaker column, the diagonal element excluded, is the recognition average for transform-speakers with speaker-independent transform, \( \overline{C}_{rs} \). This measure will be compared to the average recognition result with speaker-dependent transform, \( \overline{C}_{rsx} \), to indicate the speaker-independence of transforms learned from different speakers.

**Fig. 1. Transform/recognition speaker matrix. This matrix defines the results and the averages used in the Tests and Results section to evaluate speaker-independence of the LR/LMR-transform.**

**Learning-data requirements**

How much data is needed for learning the LR/LMR-transform will be investigated for each field of application. Figures on results for recognition correctness versus the number of word pairs, \( W \), used for learning a transform, will be shown (figures 5, 9, and 11). For \( W = 2, 5, 10, 20 \) or 50, a total of 18 recognition simulations have been carried out, three different simulations per speaker. These three simulations used LR/LMR-transforms learned from the first \( W \) words in each of three of the odd-numbered lists, respectively. The word-order in these lists vary. Hence, different sets of words were used for each of the three simulations (\( W \leq 20 \)). The figures present one mean value and one standard-deviation interval per \( W \). The mean value is the average of the set of 18 results, and the width of the interval is twice the standard-deviation in the set of 18 results.

\( W = 0 \) denotes an EYE-transform, where the \( A \)-matrix is a unity-matrix. \( W = 280 \) denotes the use of all five pairs of lists, as is the case in all simulations in the other parts of the Tests and Results-section. Each of the data-points for \( W = 0 \) and \( W = 280 \) are based on only one simulation per speaker.

A curve drawn between the mean values in each data point in figures 5, 9, and 11 seems to approach an asymptotic line that represents the maximum average performance of the transform. When choosing a proper \( W \) for an application, one has to
balance the demand for a high recognition accuracy with the demand for a small number of words to be uttered.

**Noise adaptation**

Noise adaptation has been tested in a car environment with a car moving at a varying speed, nominally 90 km/h. Word models were trained while the car was in a garage. The noise transformation was learned from recordings made in the garage and in the moving car. Tests were made for all speakers and microphones in the database.

![Fig. 2. Noise adaptation results using live-recordings (a) in the database. LMR-transforms are learned from the respective speaker using live-recordings. (b) shows the corresponding results for the loudspeaker-part of the database (same results as in figure 3). Note that soft alignment is used in (a) while hard alignment is used in (b).](image)

**Results**

The results are shown in figure 3, sorted by microphone (3a) or speaker (3b). It is obvious from figure 3, that the goose-neck microphone (sva) is the best one. The results for speakers 5 and 6 are worse with all the other microphones than with the goose-neck microphone. The average result for recognition using the goose-neck microphone is 97%. The corresponding result when using no transform at all is 78%.

Figure 2 shows results using the live-part of the database, compared to the corresponding results for the loudspeaker-part. In all but one (akv, speaker 2) of the cases in figure 2, the loudspeaker-results are better than the live-results. The differences are pretty small, except for speaker 3 with mit-microphone.

**Speaker-independence**

To investigate speaker-independence of the LR/LMR-transform, a series of recognition simulations according to the matrix in figure 1 has been done for three different microphones.
Figure 3. Results from recognition in noise for all speakers and microphones in the database. LMR-transforms are learned from the respective speaker. (a) and (b) contain the same results, sorted by microphone and speaker, respectively.

Figure 4a shows the results for sva. The figure shows that the LR/LMR-transform is speaker-independent for the sva-microphone. The trend is that the LR-transform performs slightly better than the LMR-transform. The probable cause for this is that the extra elements of the A-matrix in the LMR-transform catch details specific to the speaker or to the training-material from which the transform is learned. Hence, the LMR-transform is more speaker-dependent than the LR-transform is. The speaker-independent average result for the LR-transform is \( \bar{C}_{res} = 96\% \) with SI = 1.4\%-units for sva. The speaker-independence measure was defined in equation (10).

Figure 4b shows the results from recognition with a transform mit\( \rightarrow \)e3pv. This transformation has different speaker-dependence characteristics compared to sva\( \rightarrow \)svav. The transformation mit\( \rightarrow \)e3pv is more speaker-dependent. This is not surprising, as e3pv is adaptive. Because of the adaptivity, the microphone will have different settings for different speakers. The LMR-transform performs much better.
than the LR-transform. The average speaker-independent result for the LMR-transform is \( C_{rs} = 85\% \) with SI = 5\%-units.

Fig. 4. Results from experiments on the speaker-dependence of the noise-transformation. The two charts show the average recognition results, \( C_{rs} \), for each of the transform-speakers. They are compared to \( C_{rs} \) (LMR). (a) sva→svav, (b) mit→e3pv. Note that e3pv is an adaptive microphone.

Size of learning data

This section investigates how much data is needed for learning the LR/LMR-transform in a noise adaptation application. The method was defined in the Learning-data requirement-section. Figure 5 shows results for recognition correctness versus the number of word pairs, \( W \), used for learning a transform. Figures 5a and 5b compare the data requirement of the LR-transform versus the requirement of the LMR-transform, when using a goose-neck microphone. Figure 5c shows the corresponding data for the e3pv e3p transform.

Fig. 5. Recognition correctness vs. the number of word pairs, \( W \), used for learning a transform. (a) sva:sva→svav using LR-transform (b) sva:sva→svav using LMR-transform (c) e3pv:e3p→e3pv using LMR-transform. Speaker 6 has been excluded from the e3p-experiments.
LMR-transform and the array microphone. Speaker 6 has been excluded from the data in figure 5c to keep the deviation interval comparable to those in figures 5a and 5b. Reasonable choices for $W$ are 10 in figure 5a and 5b, and 20 in 5c.

**Microphone adaptation**

Microphone adaptation can be used if one would like to use different microphones at training and recognition. It has been tested with three microphone-pairs in a noise-free environment. The frequency characteristics of the microphones (except the central microphone, mit) are shown in figure 6. The characteristics for the AKG and goose-neck microphones have been extracted from the manufacturer's data sheets. Characteristics for the array microphone are taken from the paper by Nordholm et al. (1993).

![Frequency characteristics for AKG, SVA, and EPS microphones.](image)

*Fig. 6. Frequency characteristics for AKG, SVA, and EPS microphones. The curves for EPS and EPSV have to be considered examples of characteristics for the microphones, as the microphones are adaptive.*

**Results**

Figure 7 presents recognition results using two pairs of microphones. Each figure contains results from three series of recognition experiments: (1) Recognition with EYE-transformation. Here, the A-matrix is a unity matrix. The mean normalisation (addition of $M_r$) is used as usual. (2) Recognition with LR-transformation of models. (3) Recognition with LMR-transformation of models.

The difference between results for the EYE-transform and an LR/LMR-transform demonstrates the effect of the A-matrix rather than the effect of applying a transform to a recogniser. This is because the EYE-transform includes a running mean adaptation.

The high quality studio microphone (AKG) is a condenser-type microphone. The goose-neck microphone (SVA) is an electret-type microphone, which is a special form of the condenser type. Both microphones are high-quality. The frequency characteristics are shown in figure 6. Figure 7a shows that recognition performance is good with a simple EYE-transform. Thus, the differences between the AKG and the SVA microphones are taken care of by the running mean normalisation. The only significant improvement observed in the figure is for speaker 6 where recognition accuracy is improved by about 2%-units.
Fig. 7. Results from (a) recognition with sva and training on akg, (b) recognition with e3p and training on mit. All charts show results with EYE-, LR- and LMR-transformation. Each data point is the result of one recognition experiment on 280 words. Note that (a) and (b) have different scales on their vertical axes.

Figure 7b shows recognition results for recognition with the e3p-microphone using HMM:s trained with the mit-microphone. The average increase in performance of the LMR-transform is 10%-units for this configuration. The LMR-transform performs better than the LR-transform.

**Speaker-independence**

To investigate speaker-independence of the LR/LMR-transform in the microphone adaptation application, a series of recognition simulations according to the matrix in figure 1 has been done for two different pairs of microphones.

Figure 8 shows the results for sva: akg→sva. The chart shows that the LR/LMR-transform is rather speaker-independent in this case. The LMR-transform gives a better result than the LR-transform. The total average result with speaker-independent LMR-
transform is $\bar{C}_{res} = 98\%$ with SI = 1.1\%-units. The speaker-independence measure was defined in the Speaker-independence-section.

Figure 8b shows the results for e3p:mit→e3p. The figure shows that the LR/LMR-transform for this application is not speaker-independent. The speaker-dependency is mainly due to the fact that the e3p-microphone is adapted to each speaker before the filter coefficients are locked (e3p does not adapt during recording). Thus, the e3p-microphone will have different settings for different speakers. Hence, a transform learned for one speaker is not likely to be applicable to another speaker. The LMR-transform performs much better than the LR-transform for this application. The total average result with speaker-independent LMR-transform is $\bar{C}_{res} = 87\%$ with SI = 7\%-units.

Size of learning data

This section investigates how much data is needed for learning the LMR-transform in the e3p:mit→e3p case. The method was defined in the Learning-data requirement-section. Figure 9 shows results for recognition correctness versus the number of word pairs, $W$, used for learning the LMR-transform. The figure shows that $W=10$ words could be chosen for the mit→e3p transformation.

Speaker adaptation

It would be beneficial if one could transform speech models trained on one speaker so they could fit another speaker. Ideally, a recogniser could be trained on a reference speaker and the speech models transformed to fit any other speaker. The transform could be derived from only a small number of carefully chosen utterances if there were an efficient way to extract features describing the differences of the vocal mechanism of a speaker. Even if the speaker transform were not good enough to produce speech models that were usable at once, or even if the new speaker would have to say more than a "small" number of utterances to learn the transform, the transformed speech models might be useful as a starting-point for speaker-adaptive training.

This section will present results from recognition simulations when using different speakers for training and recognition. An LMR-transform is used for transforming trained HMM:s to fit the recognition speaker. The goose-neck microphone is used, as it was found to be a good microphone for all speakers.

Results

In spite of the fact that the similarities and differences between any two speakers constitute an intricate relation, the LMR-transform seems to be successful in
performing a speaker transformation. The results are shown in figure 10. For each of the six train-speakers, recognition has been performed on each of the other five speakers. Averages for male and female speakers has been separated, as transforming speakers within the same sex is more successful than transforming speakers between sexes. The figures show results from recognition (1) in a clean environment with EYE-transformation (Clean/EYE), (2) in a clean environment with LMR-transformation (Clean/LMR, two cases: $W=20$ and $W=280$), and (3) in a noisy car-environment with LMR-transform (Noisy/LMR). The average improvement made by the LMR-transform for the male speakers in the clean environment is 35%-units (from 54% to 88%). For the female speakers, the improvement is 28%-units (from 50% to 78%).

![Average over male rec-speakers](image1)

![Average over female rec-speakers](image2)

**Fig. 10.** The effect of speaker transformation. For each train-speaker, the results from recognition on the three male (a) or female (b) speakers has been averaged for four different cases: recognition in a noise-free environment with an EYE-transformation (Clean/EYE), recognition in a noise-free environment with an LMR-transformation to each of the speakers (Clean/LMR, two cases: $W=20$ and $W=280$), and ditto in a noisy environment (Noisy/LMR, $W=280$). $W$ is the number of word pairs used for computing the transform. Note: in order to group male and female speakers, speaker 1 is drawn to the right in the figure.

**Size of learning data**

This section investigates how much data is needed for learning the LMR-transform in a speaker transformation application. The method was defined in the Learning-data requirement-section. Figure 11 shows some results for recognition correctness versus the number of word-pairs, $W$, used for learning the LMR-transform. The set of 18 recognition simulations per datapoint is chosen differently than in the other sections. For each $W$ in the figure, except $W=0$ and $W=280$, three different simulations per combination of train-speaker $S_t$ and recognition-speaker $S_r$, where $S_t \neq S_r$. Male and female speakers are separated in figures 11a and b. Consequently, there are six such combinations of $S_t$ and $S_r$.

The figure shows that $W$ could be 20 or 50 words depending on the demands on performance versus the demands on a small size of learning data. Results for $W=20$ has been included in figure 10. The deviation intervals are wider for female speaker. This is because the results for train-speaker 1 are better than those for train-speakers 5 and 6. This is clear from figure 10.
Fig. 11. Recognition correctness vs. the number of word pairs, $W$, used for learning the LMR-transform. The sva-microphone was used, and both training and recognition were done in a noise-free environment. (a) Male speakers (S2, S3, S4), (b) Female speakers (S1, S5, S6)

Discussion and conclusions

The LR/LMR-transform introduced by Mokbel and Chollet (1991, 1992) has been evaluated for use in three different fields of application. The major field of application is noise adaptation of HMMs in a car-environment. The other two fields are adaptations of HMMs to different microphones, and to different speakers. The transform method has been improved by making it adaptive to the mean energy level of spoken words.

Noise adaptation

In the field of noise adaptation, the LR-transform was found to be the most suitable transform when using standard microphones like the AKG, goose-neck, or central microphones. The LMR-transform yielded better results when used in a speaker-dependent manner, while the LR-transform was shown to be almost speaker-independent, and yielded almost as good results as the LMR-transform with less learning data. A learning material of 5 or 10 word pairs for the LR-transform seemed to be sufficient to gain near maximum performance. For example, when using the goose-neck microphone, the LR-transform improved the speaker-average recognition result from 78%, with no transform at all, to 95%. The individual results varied from 90% to 100% in the last case. For the adaptive microphone array, the LR-transform was found to be insufficient, while the LMR-transform improved recognition accuracy considerably.

The LMR-transform was found capable of performing speaker transformation. This means that it might be able to adapt HMMs to Lombard speech as well.

In Melin (1994), the operation of the transform has been investigated for some intuitive explanation. It was found that the running mean adaptation corresponds to some kind of energy-normalisation. It was also found that the A-matrix and the running mean co-operate in adapting the HMMs to a noisy environment. There is no separation of effects from linear distortion and additive noise, as could be the case if the transform were operating on linear-energy filter bank vectors. Instead, the
operation of the LR/LMR-transform in noise adaptation resembles the operation of Wiener filtering: the trained value for a frequency channel in an HMM is damped by the $A$-matrix if there is a lot of noise in that channel. Still, the most fruitful point of view of the operation of the LR/LMR-transform, in a noise adaptation field of application, is that of a linear mapping of acoustical subspaces.

The LR/LMR-transform method may suffer from a major drawback. The dynamic properties of the used adaptation scheme has not been tested very well. The running mean was based on the average of the 1, 2, or 5 latest spoken words, with a slight improvement in recognition accuracy with an increasing number of words. The database was recorded in a way that variations in the acoustical environment between two successive words are small. The car was driven mostly on a high-way with slowly varying speed, although some amount of city-driving and acceleration is included. Also, words are spoken with an interval of only a few seconds. Hence, what happens if words are spoken to the recogniser only now and then, and the noise environment changes in between? This is a more realistic situation for the traffic information services application. The employed sequential use of the database does not really permit testing of recogniser performance during these conditions. It would be better evaluated with real-life testing or by using the spoken words in the database in random order.

If the LR/LMR-transform could operate in a linear-energy filter bank domain instead of a logarithmic one, we might get a separation of the effects of noise and linear filtering. This is still to be investigated.

**Microphone adaptation**

The LMR-transform improved recognition accuracy considerably when HMMs trained with one microphone were transformed for recognition with a different microphone. The LR-transform, however, did not seem to be complex enough.

Tests with transformations from the central microphone ($mit$) to the array microphone ($e3p$), indicated that the LMR-transform is heavily speaker-dependent. This is due to the adaptivity of the array microphone. At the computation of a transform, a set of 10 word pairs seemed to be enough to achieve maximum performance. Average recognition accuracy with the EYE-transform was 84%. With the LMR-transform trained on a set of 10 word pairs, the speaker-average recognition accuracy was increased to 94%.

**Speaker transformation**

The LMR-transform was able to transform a set of HMMs trained on one speaker to fit another speaker of the same sex. The average accuracy for training on speech from one male speaker and recognising speech from another, was increased from 56% to 84% when learning an LMR-transform from a set of 20 word-pairs. When learning the transform from the whole HMM-training material (280 words), an average recognition accuracy of 89% was achieved. These results are encouraging.

The $A$-matrix of the LMR-transform is capable of reweighting channel energy, and smoothing and shifting energy between channels (Melin, 1994). Hence, it seems that
the operation of the A-matrix is to do some kind of average amplitude and formant shifting.

The area of use for this kind of speaker transformation of HMMs may be as an initial step in speaker adaptive training of large vocabularies. If a large vocabulary is trained from the very beginning by speaker A, speaker B could start out by learning an A→B transform from a set of 20-50 spoken words. The HMMs from speaker A could then be transformed to fit speaker B. Now, if speaker B would acquire the 84% recognition accuracy of the experiments above only by uttering those 20-50 words, he would have a far better starting point than speaker A had.

The LMR-transform might be successively employed in a continuous speech recogniser where the transform could operate on phoneme models. In that case, one could have phoneme or phoneme-class specific transforms that could better model phonetic relations and differences between different speakers. As the transform operates on the HMMs, and not on the signal, it is not a problem that the sequence of phonemes actually spoken is unknown. However, to learn phoneme-specific transforms, one would need speech data labelled on the phoneme level.

Possible improvements

Two varieties of the transform method has been evaluated: the LMR-transform with a full A-matrix, and the LR-transform with a diagonal A-matrix. The LR-transform is actually a special case of the LMR-transform. It seems that in some fields of application, for instance in noise adaptation with the microphone array or in speaker adaptation, the LR-transform is not complex enough, while the LMR-transform becomes too training-material specific. The elements furthermost away from the diagonal of the A-matrix in an LMR-transform cannot possibly have any physical relevance, as they model dependencies between distant filter bank channels. Hence, a more general form of the transform with an n-diagonal A-matrix should be used. The transform could be called an LnR-transform.

From the examination (Melin, 1994) of the A-matrices in speaker adapting LMR-transforms, it seems that either an L5R-transform or an L7R-transform would be appropriate for speaker adaptation.

Another possible improvement to the computation of a transform would be to use a running mean counterpart on the source and target data instead of the global averages used in this work.

Final conclusion

The LR/LMR-transform has been found useful for noise, microphone, and speaker adaptation when evaluated on the utilised database. However, it is doubted that the noise adaptation would be usable in a real-life application because of poor dynamic qualities.

The major advantage of the transform method is its versatility. It has been found capable of adapting HMMs to most kinds of changes in the acoustical environment. One major disadvantage is the need for spoken utterances from two acoustical environments. In the introduction, it was stated that one basic condition that has to be
met, is that the need for training data is less than that of a retraining of the HMMs. This condition was met in all of the evaluated fields of application.

Acknowledgement

I would like to thank Anders Roxström, Mats Blomberg, Kjell Elenius, Roger Lindell, and Charlotte Rider for great help during my work. This work was done within a project called VOICE. This project is a part of the area Travel & Transport Information Services, within the Swedish RTI-program '91 - '94. Main sponsors of this program are The Swedish National Board for Industrial and Technical Development, AB Volvo, Saab-Scania AB, Swedish Telecom and The Swedish National Road Administration.

Reference list


Bibliography


