1

A Family of Stereo-Based Stochastic Mapping Algorithms for Noisy Speech Recognition

Mohamed Afify[1], Xiaodong Cui[2] and Yuqing Gao[2]
[1]*Orange Labs, Smart Village,*
[2]*IBM T.J. Watson Research Center, Yorktown Heights,*
[1]*Cairo, Egypt*
[2]*NY, USA*

1. Introduction

The performance of speech recognition systems degrades significantly when they are operated in noisy conditions. For example, the automatic speech recognition (ASR) front-end of a speech-to-speech (S2S) translation prototype that is currently developed at IBM [11] shows noticeable increase in its word error rate (WER) when it is operated in real field noise. Thus, adding noise robustness to speech recognition systems is important, especially when they are deployed in real world conditions. Due to this practical importance noise robustness has become an active research area in speech recognition. Interesting reviews that cover a wide variety of techniques can be found in [12], [18], [19].

Noise robustness algorithms come in different flavors. Some techniques modify the features to make them more resistant to additive noise compared to traditional front-ends. These novel features include, for example, sub-band based processing [4] and time-frequency distributions [29]. Other algorithms adapt the model parameters to better match the noisy speech. These include generic adaptation algorithms like MLLR [20] or robustness techniques as model-based VTS [21] and parallel model combination (PMC) [9]. Yet other methods design transformations that map the noisy speech into a clean-like representation that is more suitable for decoding using clean speech models. These are usually referred to as feature compensation algorithms. Examples of feature compensation algorithms include general linear space transformations [5], [30], the vector Taylor series approach [26], and ALGONQUIN [8]. Also a very simple and popular technique for noise robustness is multi-style training (MST)[24]. In MST the models are trained by pooling clean data and noisy data that resembles the expected operating environment. Typically, MST improves the performance of ASR systems in noisy conditions. Even in this case, feature compensation can be applied in tandem with MST during both training and decoding. It usually results in better overall performance compared to MST alone. This combination of feature compensation and MST is often referred to as adaptive training [22].

In this chapter we introduce a family of feature compensation algorithms. The proposed transformations are built using stereo data, i.e. data that consists of simultaneous recordings of both the clean and noisy speech. The use of stereo data to build feature mappings was very popular in earlier noise robustness research. These include a family of cepstral

Source: Speech Recognition, Technologies and Applications, Book edited by: France Mihelič and Janez Žibert,
ISBN 978-953-7619-29-9, pp. 550, November 2008, I-Tech, Vienna, Austria

normalization algorithms that were proposed in [1] and extended in robustness research at CMU, a codebook based mapping algorithm [15], several linear and non-linear mapping algorithms as in [25], and probabilistic optimal filtering (POF) [27]. Interest in stereo-based methods then subsided, mainly due to the introduction of powerful linear transformation algorithms such as feature space maximum likelihood liner regression (FMLLR)[5], [30] (also widely known as CMLLR). These transformations alleviate the need for using stereo data and are thus more practical. In principle, these techniques replace the clean channel of the stereo data by the clean speech model in estimating the transformation. Recently, the introduction of SPLICE [6] renewed the interest in stereo-based techniques. This is on one hand due to its relatively rigorous formulation and on the other hand due to its excellent performance in AURORA evaluations. While it is generally difficult to obtain stereo data, it can be relatively easy to collect for certain scenarios, e.g. speech recognition in the car or speech corrupted by coding distortion. In some other situations it could be very expensive to collect field data necessary to construct appropriate transformations. In our S2S translation application, for example, all we have available is a set of noise samples of mismatch situations that will be possibly encountered in field deployment of the system. In this case stereo-data can also be easily generated by adding the example noise sources to the existing ”clean” training data. This was our basic motivation to investigate building transformations using stereo-data.

The basic idea of the proposed algorithms is to stack both the clean and noisy channels to form a large augmented space and to build statistical models in this new space. During testing, both the observed noisy features and the joint statistical model are used to predict the clean observations. One possibility is to use a Gaussian mixture model (GMM). We refer to the compensation algorithms that use a GMM as stereo-based stochastic mapping (SSM). In this case we develop two predictors, one is iterative and is based on maximum a posteriori (MAP) estimation, while the second is non-iterative and relies on minimum mean square error (MMSE) estimation. Another possibility is to train a hidden Markov model (HMM) in the augmented space, and we refer to this model and the associated algorithm as the stereo-HMM (SHMM). We limit the discussion to an MMSE predictor for the SHMM case. All the developed predictors are shown to reduce to a mixture of linear transformations weighted by the component posteriors. The parameters of the linear transformations are derived, as will be shown below, from the parameters of the joint distribution. The resulting mapping can be used on its own, as a front-end to a clean speech model, and also in conjunction with multistyle training (MST). Both scenarios will be discussed in the experiments. GMMs are used to construct mappings for different applications in speech processing. Two interesting examples are the simultaneous modeling of a bone sensor and a microphone for speech enhancement [13], and learning speaker mappings for voice morphing [32]. HMMcoupled with an N-bset formulation was recently used in speech enhancement in [34].

As mentioned above, for both the SSMand SHMM, the proposed algorithm is effectively a mixture of linear transformations weighted by component posteriors. Several recently proposed algoorithms use linear transformations weighted by posteriors computed from a Gaussian mixture model. These include the SPLICE algorithm [6] and the stochastic vector mapping (SVM)[14]. In addition to the previous explicit mixtures of linear transformations, a noise compensation algorithm in the log-spectral domain [3] shares the use of a GMM to model the joint distribution of the clean and noisy channels with SSM. Also joint uncertainty

decoding [23] employs a Gaussian model of the clean and noisy channels that is estimated using stereo data. Last but not least probabilistic optimal filtering (POF) [27] results in a mapping that resembles a special case of SSM. A discussion of the relationships between these techniques and the proposed method in the case of SSM will be given. Also the relationship in the case of an SHMMbased predictor to the work in [34] will be highlighted.
The rest of the chapter is organized as follows. We formulate the compensation algorithm in the case of a GMM and describe MAP-based and MMSE-based compensation in Section II. Section III discusses relationships between the SSM algorithm and some similar recently proposed techniques. The SHMM algorithm is then formulated in Section IV. Experimental results are given in Section V. We first test several variants of the SSM algorithm and compare it to SPLICE for digit recognition in the car environment. Then we give results when the algorithm is applied to large vocabulary English speech recognition. Finally results for the SHMM algorithm are presented for the Aurora database. A summary is given in Section VI.

2. Formulation of the SSM algorithm

This section first formulates the joint probability model of the clean and noisy channels in Section II-A, then derives two clean feature predictors; the first is based on MAP estimation in Section II-B, while the second is based on MMSE estimation in Section II-C. The relationships between the MAP and MMSE estimators are studied in Section II-D.

A. The Joint Probability Gaussian Mixture Model

Assume we have a set of stereo data $\{(x_i, y_i)\}$, where x is the clean (matched) feature representation of speech, and y is the corresponding noisy (mismatched) feature representation. Let N be the number of these feature vectors, i.e $1 \le i \le N$. The data itself is an M-dimensional vector which corresponds to any reasonable parameterization of the speech, e.g. cepstrum coefficients. In a direct extension the y can be viewed as a concatenation of several noisy vectors that are used to predict the clean observations. Define $z \equiv (x, y)$ as the concatenation of the two channels. The first step in constructing the mapping is training the joint probability model for $p(z)$. We use Gaussian mixtures for this purpose, and hence write

$$p(z) = \sum_{k=1}^{K} c_k \mathcal{N}(z; \mu_{z,k}, \Sigma_{zz,k}) \tag{1}$$

where K is the number of mixture components, c_k, $\mu_{z,k}$, and $\Sigma_{zz,k}$, are the mixture weights, means, and covariances of each component, respectively. In the most general case where L_n noisy vectors are used to predict L_c clean vectors, and the original parameter space is M-dimensional, z will be of size $M(L_c + L_n)$, and accordingly the mean μ_z will be of dimension $M(L_c + L_n)$ and the covariance Σ_{zz} will be of size $M(L_c + L_n) \times M(L_c + L_n)$. Also both the mean and covariance can be partitioned as

$$\mu_{z,k} = \begin{pmatrix} \mu_{x,k} \\ \\ \mu_{y,k} \end{pmatrix} \tag{2}$$

$$\Sigma_{zz,k} = \begin{pmatrix} \Sigma_{xx,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{yy,k} \end{pmatrix} \tag{3}$$

where subscripts x and y indicate the clean and noisy speech respectively.
The mixture model in Equation (1) can be estimated in a classical way using the expectation-maximization (EM) algorithm. Once this model is constructed it can be used during testing to estimate the clean speech features given the noisy observations. We give two formulations of the estimation process in the following subsections.

B. MAP-based Estimation

MAP-based estimation of the clean feature x given the noisy observation y can be formulated as:

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x|y) \tag{4}$$

The estimation in Equation (4) can be further decomposed as

$$\begin{aligned} \hat{x} &= \underset{x}{\operatorname{argmax}}\, p(x|y) = \underset{x}{\operatorname{argmax}} \sum_k p(x,k|y) \\ &\equiv \underset{x}{\operatorname{argmax}} \log \sum_k p(x,k|y) \end{aligned} \tag{5}$$

Now, define the log likelihood as $L(x) \equiv \log\sum_k p(x, k \mid y)$ and the auxiliary function $Q(x, \bar{x}) \equiv \sum_k p(k \mid \bar{x}, y) \log p(x, k \mid y)$. It can be shown by a straightforward application of Jensen's inequality that

$$L(x) - L(\bar{x}) \geq Q(x,\bar{x}) - Q(\bar{x},\bar{x}) \tag{6}$$

The proof is simple and is omitted for brevity. The above inequality implies that iterative optimization of the auxiliary function leads to a monotonic increase of the log likelihood. This type of iterative optimization is similar to the EM algorithm and has been used in numnerous estimation problems with missing data. Iterative optimization of the auxiliary objective function proceeds at each iteration as follows

$$\begin{aligned} \hat{x} &= \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) \log p(k|y) p(x|k,y) \\ &= \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) [\log p(k|y) + \log p(x|k,y)] \\ &\equiv \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) \log p(x|k,y) \\ &\equiv \underset{x}{\operatorname{argmax}} \frac{-1}{2} \sum_k p(k|\bar{x},y) \left[\log|\Sigma_{x|y,k}| + (x-\mu_{x|y,k})^T \Sigma_{x|y,k}^{-1} (x-\mu_{x|y,k}) \right] \end{aligned} \tag{7}$$

where $\bar{x}$ is the value of x from previous iteration, and $x|y$ is used to indicate the statistics of the conditional distribution $p(x|y)$. By differentiating Equation (7) with respect to x, setting the resulting derivative to zero, and solving for x, we arrive at the clean feature estimate given by

$$\sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\hat{x} = \sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\mu_{x|y,k} \tag{8}$$

which is basically a solution of a linear system of equations. $p(k|\bar{x}, y)$ are the usual posterior probabilities that can be calculated using the original mixture model and Bayes rule, and the conditional statistics are known to be

$$\mu_{x|y,k} = \mu_{x,k} + \Sigma_{xy,k}\Sigma_{yy,k}^{-1}(y - \mu_{y,k}) \tag{9}$$

$$\Sigma_{x|y,k} = \Sigma_{xx,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\Sigma_{yx,k} \tag{10}$$

Both can be calculated from the joint distribution $p(z)$ using the partitioning in Equations (2) and (3). A reasonable initialization is to set $\bar{x} = y$, i.e. initialize the clean observations with the noisy observations.

An interesting special case arises when x is a scalar. This could correspond to using the i^{th} noisy coefficient to predict the i^{th} clean coefficient or alternatively using a time window around the i^{th} noisy coefficient to predict the i^{th} clean coefficient. In this case, the solution of the linear system in Equation (8) reduces to the following simple calculation for every vector dimension.

$$\hat{x} = \frac{\sum_k p(k|\bar{x},y)\mu_{x|y,k}/\sigma_{x|y,k}^2}{\sum_k p(k|\bar{x},y)/\sigma_{x|y,k}^2} \tag{11}$$

where $\sigma_{x|y,k}^2$ is used instead of $\Sigma_{x|y,k}$ to indicate that it is a scalar. This simplification will be used in the experiments. It is worth clarifying how the scalar Equation (11) is used for SSM with a time-window as mentioned above. In this case, and limiting our attention to a single feature dimension, the clean speech x is 1-dimensional, while the noisy speech y has the dimension of the window say L_n, and accordingly the mean $\mu_{x|y,k}$ and the variance $\sigma_{x|y,k}^2$ will be 1-dimensional. Hence, everything falls into place in Equation (11).

The mapping in Equations (8)-(10) can be rewritten, using simple rearrangement, as a mixture of linear transformations weighted by component posteriors as follows:

$$\hat{x} = \sum_k p(k|\bar{x},y)(A_k y + b_k) \tag{12}$$

where $A_k = CD_k$, $b_k = Ce_k$, and

$$C = \left(\sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\right)^{-1} \tag{13}$$

$$e_k = \Sigma_{x|y,k}^{-1}\left(\mu_{x,k} - \Sigma_{yy,k}^{-1}\Sigma_{xy,k}\mu_{y,k}\right) \tag{14}$$

$$D_k = \Sigma_{x|y,k}^{-1}\Sigma_{yy,k}^{-1}\Sigma_{xy,k} \tag{15}$$

C. MMSE-based Estimation

The MMSE estimate of the clean speech feature x given the noisy speech feature y is known to be the mean of the conditional distribution $p(x|y)$. This can be written as:

$$\hat{x} = E[x|y] \tag{16}$$

Considering the GMM structure of the joint distribution, Equation (16) can be further decomposed as

$$\begin{aligned} \hat{x} &= \int_x p(x|y)xdx = \sum_k \int_x p(x,k|y)xdx \\ &= \sum_k p(k|y)\int_x p(x|k,y)xdx \\ &= \sum_k p(k|y)E[x|k,y] \end{aligned} \tag{17}$$

In Equation (17), the posterior probability term $p(k|y)$ can be computed as

$$p(k|y) = \frac{p(k,y)}{p(y)} = \frac{p(y|k)p(k)}{\sum_k p(y|k)p(k)} \tag{18}$$

and the expectation term $E[x|k, y]$ is given in Equation (9).

Apart from the iterative nature of the MAP-based estimate the two estimators are quite similar. The scalar special case given in Section II-B can be easily extended to the MMSE case. Also the MMSE predictor can be written as a weighted sum of linear transformations as follows:

$$\hat{x} = \sum_k p(k|y)(F_k y + g_k) \tag{19}$$

where

$$F_k = \Sigma_{xy,k}\Sigma_{yy,k}^{-1} \tag{20}$$

$$g_k = \mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k} \tag{21}$$

From the above formulation it is clear that the MMSE estimate is not performed iteratively and that no matrix inversion is required to calculate the estimate of Equation (19). More indepth study of the relationships between the MAP and the MMSE estimators will be given in Section II-D.

D. Relationships between MAP and MMSE Estimators

This section discusses some relationships between the MAP and MMSE estimators. Strictly speaking, the MMSE estimator is directly comparable to the MAP estimator only for the first iteration and when the latter is initialized from the noisy speech. However, the following discussion can be seen as a comparison of the structure of both estimators.

To highlight the iterative nature of the MAP estimator we rewrite Equation (12) by adding the iteration index as

$$\hat{x}^{(l)} = \sum_k p(k|\bar{x}^{(l-1)}, y)(A_k y + b_k) \tag{22}$$

where l stands for the iteration index. First, if we compare one iteration of Equation (22) to Equation (19) we can directly observe that the MAP estimate uses a posterior $p(k \mid \overline{x}^{(l-1)}, y)$ calculated from the joint probability distribution while the MMSE estimate employs a posterior $p(k \mid y)$ based on the marginal probability distribution. Second, if we compare the coefficients of the transformations in Equations (13)-(15) and (20)-(21) we can see that the MAP estimate has the extra term

$$\left(\sum_k p(k|\hat{x}^{(l-1)}, y) \Sigma_{x|y,k}^{-1} \right)^{-1} \tag{23}$$

which is the inversion of the weighted summation of conditional covariance matrices from each individual Gaussian component and that requires matrix inversion during run-time[1].

If we assume the conditional covariance matrix $\Sigma_{x|y,k}$ in Equation (23) is constant across k, i.e. all Gaussians in the GMM share the same conditional covariance matrix $\Sigma_{x|y}$, Equation (23) turns to

$$\begin{aligned} & \left(\sum_k p(k|\hat{x}^{(l-1)}, y) \Sigma_{x|y,k}^{-1} \right)^{-1} \\ = & \left(\Sigma_{x|y}^{-1} \sum_k p(k|\hat{x}^{(l-1)}, y) \right)^{-1} \\ = & \left(\Sigma_{x|y}^{-1} \cdot 1 \right)^{-1} = \Sigma_{x|y} \end{aligned} \tag{24}$$

and the coefficients A_k and b_k for the MAP estimate can be written as

$$\begin{aligned} A_k &= \Sigma_{x|y} \Sigma_{x|y}^{-1} \Sigma_{xy,k} \Sigma_{yy,k}^{-1} \\ &= \Sigma_{xy,k} \Sigma_{yy,k}^{-1} \end{aligned} \tag{25}$$

[1] Note that other inverses that appear in the equations can be pre-computed and stored.

$$\begin{aligned} b_k &= \Sigma_{x|y}\Sigma_{x|y}^{-1}\left(\mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k}\right) \\ &= \mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k} \end{aligned} \tag{26}$$

The coefficients in Equations (25) and (26) are exactly the same as those for the MMSE estimate that are given in Equations (20) and (21).

To summarize, the MAP and MMSE estimates use slightly different forms of posterior weighting that are based on the joint and marginal probability distributions respectively. The MAP estimate has an additional term that requires matrix inversion during run-time in the general case, but has a negligible overhead in the scalar case. Finally, one iteration of the MAP estimate reduces to the MMSE estimate if the conditional covariance matrix is tied across the mixture components. Experimental comparison between the two estimates is given in Section V.

3. Comparison between SSM and other similar techniques

As can be seen from Section II, SSM is effectively a mixture of linear transformations weighted by component posteriors. This is similar to several recently proposed algorithms. Some of these techniques are stereo-based such as SPLICE while others are derived from FMLLR. We discuss the relationships between the proposed method and both SPLICE and FMLLR-based methods in Sections III-A and III-B, respectively. Another recently proposed noise compensation method in the log-spectral domain also uses a Gaussian mixture model for the joint distribution of clean and noisy speech [3]. Joint uncertainty decoding [23] employs a joint Gaussian model for the clean and noisy channels, and probabilistic optimal filtering has a similar structure to SSMwith a time window. We finally discuss the relationship of the latter algorithms and SSM in Sections III-C,III-D, and III-E, respectively.

A. SSM and SPLICE

SPLICE is a recently proposed noise compensation algorithm that uses stereo data. In SPLICE, the estimate of the clean feature $\hat{x}$ is obtained as

$$\hat{x} = \sum_k p(k|y)(y + r_k) \tag{27}$$

where the bias term r_k of each component is estimated from stereo data (x_n, y_n) as

$$r_k = \frac{\sum_n p(k|y_n)(x_n - y_n)}{\sum_n p(k|y_n)} \tag{28}$$

and n is an index that runs over the data. The GMM used to estimate the posteriors in Equations (27) and (28) is built from noisy data. This is in contrast to SSM which employs a GMM that is built on the joint clean and noisy data.

Compared to MMSE-based SSM in Equations (19), (20) and (21), we can observe the following. First, SPLICE builds a GMM on noisy features while in this paper a GMM is built on the joint clean and noisy features (Equation (1)). Consequently, the posterior probability $p(k|y)$ in Equation (27) is computed from the noisy feature distribution while $p(k|y)$ in Equation (19) is computed from the joint distribution. Second, SPLICE is a special case of

SSM if the clean and noisy speech are assumed to be perfectly correlated. This can be seen as follows. If perfect correlation is assumed between the clean and noisy feature then $\Sigma_{xy,k} = \Sigma_{yy,k}$, and $p(k \mid x_n)=p(k \mid y_n)$. In this case, Equation (28) can be written as

$$\begin{aligned} r_k &= \frac{\sum_n p(k|y_n)(x_n - y_n)}{\sum_n p(k|y_n)} \\ &= \frac{\sum_n p(k|y_n)x_n - \sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &= \frac{\sum_n p(k|y_n)x_n}{\sum_n p(k|y_n)} - \frac{\sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &\doteq \frac{\sum_n p(k|x_n)x_n}{\sum_n p(k|x_n)} - \frac{\sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &= \mu_{x,k} - \mu_{y,k} \end{aligned} \tag{29}$$

The latter estimate will be identical to the MMSE estimate in Equations (20) and (21) when $\Sigma_{xy,k} = \Sigma_{yy,k}$.

To summarize, SPLICE and SSM have a subtle difference concerning the calculation of the weighting posteriors (noisy GMM vs. joint GMM), and SSM reduces to SPLICE if perfect correlation is assumed for the clean and noisy channels. An experimental comparison of SSM and SPLICE will be given in Section V.

B. SSM and FMLLR-based methods

There are several recently proposed techniques that use a mixture of FMLLR transforms. These can be written as

$$\hat{x} = \sum_k p(k|y)(U_k y + v_k) \tag{30}$$

where $p(k \mid y)$ is calculated using an auxiliary Gaussian mixture model that is typically trained on noisy observations, and U_k and v_k are the elements of FMLLR transformations that do not require stereo data for their estimation. These FMLLR-based methods are either applied during run-time for adaptation as in [28], [33], [16] or the transformation parameters are estimated off-line during training as in the stochastic vector mapping (SVM) [14]. Also online and offline transformations can be combined as suggested in [14]. SSM is similar in principle to training-based techniques and can be also combined with adaptation methods. This combination will be experimentally studied in Section V.

The major difference between SSM and the previous methods lies in the used GMM (again noisy channel vs. joint), and in the way the linear transformations are estimated (implicitly derived from the joint model vs. FMLLR-like). Also the current formulation of SSM allows the use of a linear projection rather than a linear transformation and most these techniques assume similar dimensions of the input and output spaces. However, their extension to a projection is fairly straightforward. In future work it will be interesting to carry out a systematic comparison between stereo and non-stereo techniques.

C. SSM and noise compensation in the log-spectral domain

A noise compensation technique in the log-spectral domain was proposed in [3]. This method, similar to SSM, uses a Gaussian mixture model for the joint distribution of clean

and noisy speech. However, the model of the noisy channel and the correlation model are not set free as in the case of SSM. They are parametrically related to the clean and noise distributions by the model of additive noise contamination in the log-spectral domain, and expressions of the noisy speech statistics and the correlation are explicitly derived. This fundamental difference results in two important practical consequences. First, in contrast to [3] SSM is not limited to additive noise compensation and can be used to correct for any type of mismatch. Second, it leads to relatively simple compensation transformations during run-time and no complicated expressions or numerical methods are needed during recognition.

D. SSM and joint uncertainty decoding

A recently proposed technique for noise compensation is joint uncertainty decoding (JUD)[23]. Apart from the fact that JUD employs the uncertainty decoding framework[7], [17], [31][2] instead of estimating the clean feature, it uses a joint model of the clean and noisy channels that is trained from stereo data. The latter model is very similar to SSM except it uses a Gaussian distribution instead of a Gaussian mixture model. On one hand, it is clear that a GMM has a better modeling capacity than a single Gaussian distribution. However, JUD also comes in a model-based formulation where the mapping is linked to the recognition model. This model-based approach has some similarity to the SHMM discussed below.

E. SSM and probabilistic optimal filtering (POF)

POF [27] is a technique for feature compensation that, similar to SSM, uses stereo data. In POF, the clean speech feature is estimated from a window of noisy features as follows:

$$\hat{x} = \sum_{i=1}^{I} p(i|z) W_i^T Y \tag{31}$$

where i is the vector quantization region index, I is the number of regions, z is a conditioning vector that is not necessarily limited to the noisy speech, Y consists of the noisy speech in a time window around the current vector, and W_i is the weight vector for region i. These weights are estimated during training from stereo data to minimize a conditional error for the region.

It is clear from the above presentation that POF bears similarities to SSM with a time window. However, some differences also exist. For example, the concept of the joint model allows the iterative refinement of the GMM parameters during training and these parameters are the equivalent to the region weights in POF. Also the use of a coherent statistical framework facilitates the use of different estimation criteria e.g. MAP and MMSE, and even the generalization of the transformation to the model space as will be discussed below. It is not clear how to perform these generalizations for POF.

4. Mathematical formulation of the stereo-HMM algorithm

In the previous sections we have shown how a GMM is built in an augmented space to model the joint distribution of the clean and noisy features, and how the resulting model is

[2] In uncertainty decoding the noisy speech pdf $p(y)$ is estimated rather than the clean speech feature.

used to construct feature compensation algorithm. In this section we extend the idea by training an HMM in the augmented space and formulate an appropriate feature compensation algorithm. We refer to the latter model as the stereo-HMM (SHMM).

Similar to the notation in Section II, denote a set of stereo features as $\{(x, y)\}$, where x is the clean speech feature vector, y is the corresponding noisy speech feature vector. In the most general case, y is L_n concatenated noisy vectors, and x is L_c concatenated clean vectors. Define $z \equiv (x, y)$ as the concatenation of the two channels. The concatenated feature vector z can be viewed as a new feature space where a Gaussian mixture HMM model can be built[3]. In the general case, when the feature space has dimension M, the new concatenated space will have a dimension $M(L_c + L_n)$. An interesting special case that greatly simplifies the problem arises when only one clean and noisy vectors are considered, and only the correlation between the same components of the clean and noisy feature vectors are taken into account. This reduces the problem to a space of dimension $2M$ with the covariance matrix of each Gaussian having the diagonal elements and the entries corresponding to the correlation between the same clean and noisy feature element, while all other covariance values are zeros.

Training of the above Gaussian mixture HMM will lead to the transition probabilities between states, the mixture weights, and the means and covariances of each Gaussian. The mean and covariance of the k^{th} component of state i can, similar to Equations (2) and (3), be partitioned as

$$\mu_{z,i,k} = \begin{pmatrix} \mu_{x,i,k} \\ \\ \mu_{y,i,k} \end{pmatrix} \tag{32}$$

$$\mu_{z,i,k} = \begin{pmatrix} \mu_{x,i,k} \\ \\ \mu_{y,i,k} \end{pmatrix} \tag{33}$$

where subscripts x and y indicate the clean and noisy speech features respectively.

For the k^{th} component of state i, given the observed noisy speech feature y, the MMSE estimate of the clean speech x is given by $E[x|y, i, k]$. Since (x, y) are jointly Gaussian, the expectation is known to be

$$\begin{aligned} & E[x|y,i,k] \\ = \; & \mu_{x|y,i,k} \\ = \; & \mu_{x,i,k} + \Sigma_{xy,i,k}\Sigma^{-1}_{yy,i,k}(y - \mu_{y,i,k}) \end{aligned} \tag{34}$$

[3] We will need the class labels in this case in contrast to the GMM.

The above expectation gives an estimate of the clean speech given the noisy speech when the state and mixture component index are known. However, this state and mixture component information is not known during decoding. In the rest of this section we show how to perform the estimation based on the N-best hypotheses in the stereo HMM framework.

Assume a transcription hypothesis of the noisy feature is *H*. Practically, this hypothesis can be obtained by decoding using the noisy marginal distribution $p(y)$ of the joint distribution $p(x, y)$. The estimate of the clean feature, $\hat{x}$, at time *t* is given as:

$$\begin{aligned}
\hat{x}_t &= E[x_t|y_1^T] \\
&= \int_{x_t} p(x_t|y_1^T) x_t dx_t \\
&= \sum_H \sum_i \sum_k \int_{x_t} p(x_t, i, k, H|y_1^T) x_t dx_t \\
&= \sum_H \sum_i \sum_k \int_{x_t} p(x_t, i, k|y_1^T, H) p(H|y_1^T) x_t dx_t \\
&= \sum_H p(H|y_1^T) \sum_i \sum_k p(i, k|y_1^T, H) \cdot \\
&\quad \int_{x_t} p(x_t|i, k, y_1^T, H) x_t dx_t
\end{aligned} \tag{35}$$

where the summation is over all the recognition hypotheses, the states, and the Gaussian components. The estimate in Equation (35) can be rewritten as:

$$\hat{x}_t = \sum_H p(H|y_1^T) \sum_i \sum_k \gamma_{ik}^H(t) E[x_t|y_t, i, k] \tag{36}$$

where $\gamma_{ik}^H(t) = p(s_t = i, \xi_t = k|y_1^T, H)$ is the posterior probability of staying at mixture component *k* of state *i* given the feature sequence y_1^T and hypothesis *H*. This posterior can be calculated by the forward-backward algorithm on the hypothesis *H*. The expectation term is calculated using Equation (34). $p(H|y_1^T)$ is the posterior probability of the hypothesis *H* and can be calculated from the N-best list as follows:

$$p(H|y_1^T) = \frac{p(y_1^T|H)^\nu p(H)^\nu}{\sum_j p(y_1^T|H_j)^\nu p(H_j)^\nu} \tag{37}$$

where the summation in the denominator is over all the hypotheses in the N-best list, and υ is a scaling factor that need to be experimentally tuned.

By comparing the estimation using the stereo HMM in Equation (36) with that using a GMM in the joint feature space as shown, for convenience, in Equation (38),

$$\hat{x}_t = \sum_k p(k|y_t) E[x_t|k, y_t] \tag{38}$$

we can find out the difference between the two estimates. In Equation (36), the estimation is carried out by weighting the MMSE estimate at different levels of granularity including Gaussians, states and hypotheses. Additionally, the whole sequence of feature vectors, y_1^T = (y_1, y_2, $\cdots$, y_T), has been exploited to denoise each individual feature vector x_t. Therefore, a better estimation of x_t is expected in Equation (36) over Equation (38).

Figure (1) illustrates the whole process of the proposed noise robust speech recognition scheme on stereo HMM. First of all, a traditional HMM is built in the joint (clean-noisy) feature space, which can be readily decomposed into a clean HMM and a noisy HMM as its marginals. For the input noisy speech signal, it is first decoded by the noisy marginal HMM to generate a word graph and also the N-best candidates. Afterwards, the MMSE estimate of the clean speech is calculated based on the generated N-best hypotheses as the conditional expectation of each frame given the whole noisy feature sequence. This estimate is a weighted average of Gaussian level MMSE predictors. Finally, the obtained clean speech estimate is re-decoded by the clean marginal HMM in a reduced searching space on the previously generated word graph.

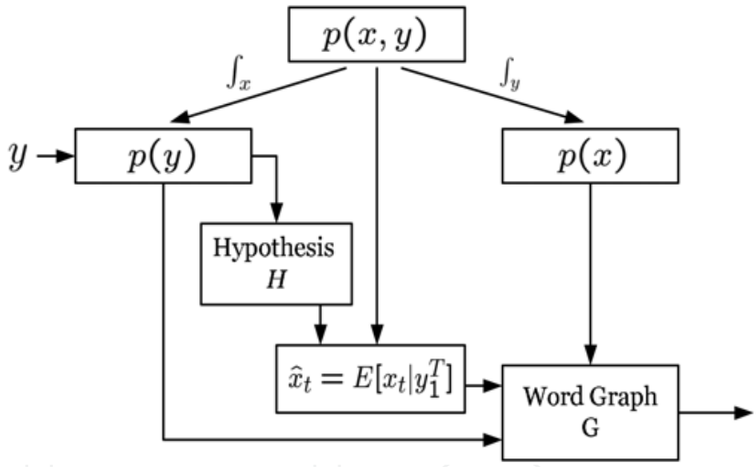

Fig. 1. Denoising scheme of N-best hypothesis based on stereo acoustic model.

A word graph based feature enhancement approach was investigated in [34] which is similar to the proposed work in the sense of two pass decoding using word graph. In [34], the word graph is generated by the clean acoustic model on enhanced noisy features using signal processing techniques and the clean speech is actually "synthesized" from the HMM Gaussian parameters using posteriori probabilities. Here, the clean speech is estimated from the noisy speech based on the joint Gaussian distributions between clean and noisy features.

5. Experimental evaluation

In the first part of this section we give results for digit recognition in the car environment and compare the SSM method to SPLICE. In the second part, we provide results when SSMis applied to large vocabulary spontaneous English speech recognition. Finally, we present SHMM results for the Aurora database.

A. SSM experiments for digit recognition in the car

The proposed algorithm is evaluated on a hands-free database (CARVUI database) recorded inside a moving car. The data was collected in Bell Labs area, under various driving conditions (highway/city roads) and noise environments (with and without radio/music in the background). About 2/3rd of the recordings contain music or babble noise in the background. Simultaneous recordings were made using a close-talking microphone and a 16-channel array of 1st order hypercardioid microphones mounted on the visor. A total of 56 speakers participated in the data collection, including many non-native speakers of American English. Evaluation is limited to the digit part of the data base. The speech material from 50 speakers is used for training, and the data from the 6 remaining speakers is used for test, leading to a total of about 6500 utterances available for training and 800 utterances for test. The test set contains about 3000 digits. The data is recorded at 24kHz sampling rate and is down-sampled to 8kHz and followed by MFCC feature extraction step for our speech recognition experiments. The feature vector consists of 39 dimensions, 13 cepstral coefficients and their first and second derivatives. Cepstral mean normalization (CMN) is applied on the utterance level. CMN is considered, to a first order approximation, as compensating for channel effects, and hence a channel parameter is not explicitly included in the compensation algorithm. The recognition task consists of simple loop grammar for the digits. In our experiments, data from 2 channels only are used. The first one is the close-talking microphone (CT), the second one is a single channel from the microphone array, referred to as Hands-Free data (HF) henceforward. 10 digit models and a silence model are built. Each model is left to right with no skipping having six states, and each state has 8 Gaussian distributions. Training and recognition is done using HTK [35]. A baseline set of results for this task are given in Table I.

Condition	WER
clean/clean	3.7
clean/noisy	14.1
noisy/noisy	6.1
clean/VTS	9.4
clean/COMP	6.9

Table I Baseline word error rate (WER) results (in %) of the close-talking (CT) microphone data and hands-free (HF) data

The first three lines refer to train/test conditions where the clean refers to the CT and noisy to the HF. The third line, in particular, refers to matched training on the HF data. The fourth and fifth lines correspond to using clean training and noisy test data that is compensated using conventional first order vector Taylor series (VTS) [26], and the compensation method in [3]. Both methods use a Gaussian mixture for the clean speech of size 64, and no explicit channel compensation is used as CMN is considered to partially account for channel effects. It can be observed from the table that the performance is clearly effected, as expected, by the addition of noise. Using noisy data for training improves the result considerably but not to the level of clean speech performance. VTS gives an improvement over the baseline, while the method in [3] shows a significant gain. More details about these compensation experiments can be found in [3] and other related publications.

The mapping is applied to the MFCC coefficients before CMN. After applying the compensation, CMN is performed followed by calculating the delta and delta-delta. Two methods were tested for constructing the mapping. In the first, a map is constructed between the same MFCC coefficient for the clean and noisy channels. In the second, a time window, including the current frame and its left and right contexts, around the i^{th} MFCC noisy coefficient is used to calculate the i^{th} clean MFCC coefficient. We tested windows of sizes three and five respectively. Thus we have mappings of dimensions 1 × 1, 3 × 1, and 5 × 1 for each cepstral dimension. These mappings are calculated according to Equation (11). In all cases, the joint Gaussian mixture model $p(z)$ is initialized by building a codebook on the stacked cepstrum vectors, i.e. by concatenation of the cepstra of the clean and noisy speech. This is followed by running three iterations of EMtraining. Similar initialization and training setup is also used for SPLICE. In this subsection only one iteration of the compensation algorithm is applied during testing. It was found in initial experiments that more iterations improve the likelihood, as measured by the mapping GMM, but slightly increase the WER. This comes in contrast to the large vocabulary results of the following section where iterations in some cases significantly improve performance. We do not have an explanation of this observation at the time of this writing.

In the first set of experiments we compare between SPLICE,MAP-SSMand MMSE-SSM, for different GMM sizes. No time window is used in these experiments. The results are shown in Table II. It can be observed that the proposed mapping outperforms SPLICE for all GMM sizes with the difference decreasing with increasing the GMM size. This makes sense because with increasing the number of Gaussian components, and accordingly the biases used in SPLICE, we can theoretically approximate any type of mismatch. Both methods are better than the VTS result in Table I, and are comparable to the method in [3]. The mapping in [3] is, however, more computationally expensive than SPLICE and SSM. Also, MAP-SSM and MMSE-SSM show very similar performance. This again comes in cotrast to what is observed in large vocabulary experiments where MMSE-SSMoutperforms MAP-SSM in some instances.

	16	64	256
SPLICE	9.0	8.6	8.3
MAP-SSM	8.3	8.3	8.0
MMSE-SSM	8.5	8.3	8.2

Table II Word error rate results (in %) of hands-free (HF) data using the proposed map-based mapping (MAP-SSM), SPLICE, and MMSE-SSM for different GMM sizes.

Finally Table III compares the MAP-SSM with and without the time window. We test windows of sizes 3 and 5. The size of the GMM used is 256. Using a time window gives an improvement over the baseline SSM with a slight cost during runtime. These results are not given for SPLICE because using biases requires that both the input and output spaces have the same dimensions, while the proposed mapping can be also viewed as a projection. The best SSM configuration, namely SSM-3, results in about 45% relative reduction in WER over the uncompensated result.

	SSM-1	SSM-3	SSM-5
WER	8.0	7.4	7.6

Table III Word error rate results (in %) of hands-free (HF) data using three different configurations of MAP-SSM for 256 GMM size and different time window size.

B. SSM experiments for large vocabulary spontaneous speech recognition

In this set of experiments the proposed technique is used for large vocabulary spontaneous English speech recognition. The mapping is applied with the clean speech models and also in conjunction with MST. The speech recognition setup and practical implementation of the mapping are first described in Section V-B.1. This is followed by compensation experiments for both the MAP and MMSE estimators in Section V-B.2.

B.1 Experimental setup

Experiments are run for a large vocabulary spontaneous speech recognition task. The original (clean) training data has about 150 hours of speech. This data is used to build the clean acoustic model. In addition to the clean model an MST model is also trained from the MST data. The MST data is formed by pooling the clean training data and noisy data. The noisy data are generated by adding humvee, tank and babble noise to the clean data at 15 dB. Different noise types are randomly added to different parts of each utterance. These three types of noise are chosen to match the military deployment environments in the DARPA Transtac Project. Thus, there are about 300 hours of training data in the MST case corresponding to the clean and 15 db SNR. When SSM is applied in conjunction with MST, the MST models are trained from SSM compensated data. This is done as follows. The SSM mapping is first trained as will be detailed below. It is then applied back to the noisy training data to yield noise-compensated features. Finally, the clean and noise compensated features are pooled and used to train the acoustic model. This is in the same spirit of using speaker-adaptive training (SAT) scheme, where some adaptation or compensation method is used in both training and decoding. The acoustic models are constructed in the same way and only differ in the type of the data used. Feature extraction, model training, mapping construction, and decoding will be outlined below.

The feature space of the acoustic models is formed as follows. First, 24 dimensional Mel-frequency cepstrum coefficients (MFCC) are calculated. The MFCC features are then mean normalized. 9 vectors, including the current vector and its left and right neighbours, are stacked leading to a 216-dimensional parameter space. The feature space is finally reduced to 40 dimensions using a combination of linear discriminant analysis (LDA) and a global semi-tied covariance (STC)matrix [10].

The acoustic model uses Gaussian mixture models associated to the leaves of a decision tree. The tree clustering is done by asking questions about quinphone context. The phoneme inventory has 54 phonemes for American English, and each phoneme is represented by a 3-state HMM. The model parameters are estimated using the forward-backward algorithm. First, a quinphone decision tree is built, an LDA matrix is computed and the model parameters are estimated using 40 EM iterations with the STC matrix updated each iteration. Upon finishing the estimation, the new model is used to re-generate the alignments based on which a new decision tree is built, the LDA matrix is re-computed and another 40 EM iterations are performed for model parameter estimation and STC matrix update. The clean model has 55K Gaussians while the MST models have 90K Gaussians. This difference is due to the difference in the amount of training data. The training and decoding are carried out on the IBM Attila toolkit.

Generally speaking SSM is SNR-specific and noise-type specific, i.e. a different mapping is built for each SNR and each noise type. However, as mentioned above we constructed only one mapping (at 15 dB) that corresponds to the mean SNR of the training data. The training of the mapping is straightforward. It amounts to the concatenation of the clean and noisy channels in the desired way and building a GMM using the EMalgorithm. All the mappings

used in the following experiments are of size 1024. It was confirmed in earlier work [2] that using larger sizes only give marginal improvements. The mapping is trained by starting from 256 random vectors, and then running one EM iteration and splitting until reaching the desired size. The final mapping is then refined by running 5 EM iterations. The mapping used in this section is scalar, i.e. it can be considered as separate mappings between the same coefficients in the clean and noisy channels. Although using different configurations can lead to better performance, as for example in Section V-A, this was done for simplicity. Given the structure of the feature vector used in our system, it is possible to build the mapping either in the 24-dimensional MFCC domain or in the 40-dimensional final feature space. It was also shown in [2] that building the mapping in the final feature space is better, and hence we restrict experiments in this work to mappings built in the 40-dimensional feature space. As discussed in Section II there are two possible estimators that can be used with SSM. Namely, the MAP and MMSE estimators. It should be noted that the training of the mapping in both cases is the same and that the only difference happens during testing, and possibly in storing some intermediate values for efficient implementation.

A Viterbi decoder that employs a finite state graph is used in this work. The graph is formed by first compiling the 32K pronunciation lexicon, the HMM topology, the decision tree, and the trigram language model into a large network. The resulting network is then optimized offline to a compact structure which supports very fast decoding. During decoding, generally speaking, the SNR must be known to be able to apply the correct mapping. Two possibilities can be considered, one is rather unrealistic and assumes that the SNR is given while the other uses an environment detector. The environment detector is another GMM that is trained to recognize different environments using the first 10 frames of the utterance. In [2], it was found that there is almost no loss in performance due to using the environment detector. In this section, however, only one mapping is trained and is used during decoding. Also as discussed in Section II the MAP estimator is iterative. Results with different number of iterations will be given in the experiments.

The experiments are carried out on two test sets both of which are collected in the DARPA Transtac project. The first test set (Set A) has 11 male speakers and 2070 utterances in total recorded in the clean condition. The utterances are spontaneous speech and are corrupted artificially by adding humvee, tank and babble noise to produce 15dB and 10dB noisy test data. The other test set (Set B) has 7 male speakers with 203 utterances from each. The utterances are recorded in a real-world environment with humvee and tank noise running in the background. This is a very noisy evaluation set and the utterances SNRs are measured around 5dB to 8dB, and we did not try to build other mappings to match these SNRs. This might also be considered as a test for the robustness of the mapping.

B.2 Experimental results

In this section SSM is evaluated for large vocabulary speech recognition. Two scenarios are considered, one with the clean speech model and the other in conjunction with MST. Also the combination of SSM with FMLLR adaptation is evaluated in both cases. For MAP-based SSM both one (MAP1) and three (MAP3) iterations are tested.

Table IV shows the results for the clean speech model. The first part of the table shows the uncompensated result, the second and third parts give the MAP-based SSM result for one and three iterations, respectively, while the final part presents MMSE-based SSM. In each part the result of combining FMLLR with SSM compensation is also given. The columns of the table correspond to the clean test data, artificially corrupted data at 15 dB, and 10 dB, and real field data. In all cases it can be seen that using FMLLR brings significant gain,

except in theMMSE-based SSMwhere it only leads to a slight improvement. MAP-based SSMshows some improvement only for test set B and using three iterations, in all other cases it does not improve on the clean result. MMSE-based SSM, on the other hand, shows excellent performance in all cases and outperforms its MAP-based counterpart. One explanation for this behavior can be attributed to the tying effect that is shown in Section II for MMSE estimation. In large vocabulary experiments a large mapping is needed to represent the new acoustic space with sufficient resolution. However, this comes at the expense of the robustness of the estimation. The implicit tying of the conditional covariances in the MMSE case can address this tradeoff and might be a reason of the improved performance in this case. Another way to address this, and that might be of benefit to the MAP-based algorithm is to construct the mapping in subspaces but this has to be experimentally confirmed. Finally, it is clear from the table that SSM does not hurt the clean speech performance. The best result for the real field data, which is for MMSE-based SSM with FMLLR, is 41% better than the baseline, and is 35% better than FMLLR alone.

	Set A			Set B
	clean	15 dB	10 dB	5-8 dB
clean model	4.84	18.40	33.66	47.72
clean model + fmllr	3.23	14.30	27.89	43.28
SSM_MAP1	4.87	18.05	33.32	48.24
SSM_MAP1 + fmllr	3.23	14.41	28.79	43.63
SSM_MAP3	4.87	18.03	33.36	46.04
SSM_MAP3 + fmllr	3.23	14.43	28.36	41.68
SSM_MMSE	4.84	13.39	25.52	28.43
SSM_MMSE + fmllr	3.26	13.23	25.12	28.25

Table IV Word error rate results (in %) of the compensation schemes against clean acoustic model

	Set A			Set B
	clean	15 dB	10 dB	5-8 dB
MST model	7.67	11.06	18.90	46.74
MST model + fmllr	3.87	7.69	14.13	25.87
SSM_MAP1	4.57	9.75	18.46	43.59
SSM_MAP1 + fmllr	2.74	6.96	14.07	23.83
SSM_MAP3	4.77	9.32	17.59	40.58
SSM_MAP3 + fmllr	2.76	6.79	13.78	22.85
SSM_MMSE	4.15	10.41	20.39	31.57
SSM_MMSE + fmllr	2.76	8.50	17.66	18.31

Table V Word error rate results (in %) of the compensation schemes against mst acoustic model

Table V displays the same results as table IV but for the MST case. The same trend as in table IV can be observed, i.e. FMLLR leads to large gains in all situations, and SSMbrings

decent improvements over FMLLR alone. In cotrast to the clean model case, MAP-based SSM and MMSE-based SSM are quite similar in most cases. This might be explained by the difference in nature in the mapping required for the clean and MST cases, and the fact that the model is trained on compensated data which in some sense reduces the effect of the robustness issue raised for the clean case above. The overall performance of the MST model is, unsurprisingly, better than the clean model. In this case the best setting for real field data, also MMSE-based SSM with FMLLR, is 60% better than the baseline and 41% better than FMLLR alone.

C. Experimental Results for Stereo-HMM

This section gives results of applying stereo-HMM compensation on the Sets A and B of the Aurora 2 database. There are four types of noise in the training set which include subway, babble, car and exhibition noise. The test set A has the same four types of noise as the training set while set B has four different types of noise, namely, restaurant, street, airport and station. For each type of noise, training data are recorded under five SNR conditions: clean, 20 dB, 15 dB, 10 dB and 5 dB while test data consist of six SNR conditions: clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB. There are 8440 utterances in total for the four types of noise contributed by 55 male speaker and 55 female speakers. For the test set, each SNR condition of each noise type consists of 1001 utterances leading to 24024 utterances in total from 52 male speakers and 52 female speakers.

Word based HMMs are used, with each model having 16 states and 10 Gaussian distributions per state. The original feature space is of dimension 39 and consists of 12 MFCC coefficients, energy, and their first and second derivatives. In the training set, clean features and their corresponding noisy features are spliced together to form the stereo features. Thus, the joint space has dimension 78. First, a clean acoustic model is trained on clean features only on top of which single-pass re-training is performed to obtain the stereo acoustic model where the correlation between the corresponding clean and noisy components is only taken into account. Also a multi-style trained (MST) model is constructed in the original space to be used as a baseline. The results are shown in Tables VI-VIII. Both the MST model and the stereo model are trained on the mix of four types of training noise.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.1	98.5	97.9	95.4	89.7	66.3
SSM(0.6)	99.1	98.6	98.0	95.6	89.7	66.6
SSM(0.3)	99.1	98.6	98.0	95.8	90.2	67.2
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.1	98.4	97.4	94.6	86.8	63.0
SSM(0.6)	99.1	98.5	97.3	94.6	86.7	63.1
SSM(0.3)	99.2	98.6	97.3	94.5	86.6	63.1

Table VI Accuracy on aurora 2 set A and set B. evaluated with $N = 5$.

A word graph, or lattice, is constructed for each utterance using the noisy marginal of the stereo HMMand converted into an N-best list. Different sizes of the list were tested and results for lists of sizes 5, 10 and 15 are shown in the tables. Hence, the summation in the

denominator of Equation (37) is performed over the list, and different values (1.0, 0.6 and 0.3) of the weighting υ are evaluated (denoted in the parentheses in the tables). The language model probability $p(H)$ is taken to be uniform for this particular task. The clean speech feature is estimated using Equation (36). After the clean feature estimation, it is rescored using the clean marginal of the stereo HMM on the word graph. The accuracies are presented as the average across the four types of noise in each individual test set.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.2	98.6	97.9	95.7	89.6	66.4
SSM(0.6)	99.2	98.6	97.9	95.7	89.8	66.7
SSM(0.3)	99.2	98.6	98.0	95.9	90.0	67.3
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.2	98.6	97.5	94.8	87.1	63.7
SSM(0.6)	99.2	98.6	97.5	94.7	87.1	63.7
SSM(0.3)	99.2	98.5	97.4	94.6	87.0	63.8

Table VII Accuracy on aurora 2 set A and set B. evaluated with N = 10.

From the tables we observe that the proposed N-best based SSMon stereo HMMperforms better than theMST model especially for unseen noise in Set B and at low SNRs. There are about 10%-20% word error rate (WER) reduction in Set B compared to the baseline MST model. It can be also seen that there is little influence for the weighting factor, this might be due to the uniform language model used in this task but might change for other scenarios. By increasing the number of N-best candidates in the estimation, the performance increases but not significantly.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.2	98.6	97.9	95.7	89.8	66.4
SSM(0.6)	99.2	98.6	97.9	95.8	89.9	66.7
SSM(0.3)	99.2	98.6	98.0	96.0	90.1	67.1
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.2	98.4	97.5	94.8	87.2	63.8
SSM(0.6)	99.2	98.5	97.5	94.8	87.2	63.9
SSM(0.3)	99.2	98.5	97.4	94.6	87.1	64.0

Table VIII Accuracy on aurora 2 set A and set B. evaluated with N = 15.

6. Summary

This chapter presents a family of feature compensation algorithms for noise robust speech recognition that use stereo data. The basic idea of the proposed algorithms is to stack the features of the clean and noisy channels to form a new augmented space, and to train

statistical models in this new space. These statistical models are then used during decoding to predict the clean features from the observed noisy features. Two types of models are studied. Gaussian mixture models which lead to the so-called stereo-based stochastic mapping (SSM)algorithm, and hidden Markov models which result in the stereo-HMM (SHMM) algorithm. Two types of predictors are examined for SSM, one is based on MAP estimation while the other is based on MMSE estimation. Only MMSE estimation is used for the SHMM, where an N-best list is used to provide the required recognition hypothesis. The algorithms are extensively evaluated in speech recognition experiments. SSM is tested for both digit recognition in the car, and a large vocabulary spontaneous speech task. SHMM is evaluated on the Aurora task. In all cases the proposed methods lead to significant gains.

7. References

A. Acero, Acoustical and environmental robustness for automatic speech recognition, Ph.D. Thesis, ECE Department, CMU, September 1990.

M. Afify, X. Cui and Y. Gao, ”Stereo-Based Stochastic Mapping for Robust Speech Recognition,” in Proc. ICASSP’07, Honolulu, HI, April 2007.

M. Afify, ”Accurate compensation in the log-spectral domain for noisy speech recognition,” in IEEE Trans. on Speech and Audio Processing, vol. 13, no. 3, May 2005.

H. Bourlard, and S. Dupont, ”Subband-based speech recognition,” in Proc. ICASSP’97, Munich, Germany, April 1997.

V. Digalakis, D. Rtischev, and L. Neumeyer, ”Speaker adaptation by constrained estimation of Gaussian mixtures,” IEEE Transactions on Speech and Audio Processing, vol. 3, no. 5, pp. 357-366, 1995.

J. Droppo, L. Deng, and A. Acero, ”Evaluation of the SPLICE Algorithm on the AURORA 2 Database,” in Proc. Eurospeech’01, Aalborg, Denmark, September, 2001.

J. Droppo, L. Deng, and A. Acero, ”Uncertainity decoding with splice for noise robust speech recognition,” in Proc. ICASSP’02, Orlando, Florida, May 2002.

B. Frey, L. Deng, A. Acero, and T. Kristjanson, ”ALGONQUIN: Iterating Laplace’s method to remove multiple types of acoustic distortion for robust speech recognition,” in Proc. Eurospeech’01, Aalborg, Denmark, September, 2001.

M. Gales, and S. Young, ”Robust continuous speech recognition using parallel model combination,” IEEE Transactions on Speech and Audio Processing, vol. 4, 1996.

M. Gales, ”Semi-tied covariance matrices for hidden Markov models,” IEEE Transactions on Speech and Audio Processing, vol. 7, pp. 272-281, 1999.

Y. Gao, B. Zhou, L. Gu, R. Sarikaya, H.-K. Kuo. A.-V.I. Rosti,M. Afify,W. Zhu, ”IBMMASTOR:Multilingual automatic speech-to-speech translator,” Proc. ICASSP’06, Tolouse, France, 2006.

Y.Gong, ‘”Speech recognition in noisy environments: A survey,” Speech Communication, Vol.16, pp.261-291, April 1995.

J. Hershey, T. Kristjansson, and Z. Zhang, ”Model-based fusion of bone and air sensors for speech enhancement and robust speech recognition,” in ISCAWorkshop on statistical and perceptual audio processing, 2004.

Q. Huo, and D. Zhu, ”A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations,” in Proc. Interspeech’06, Pittsburgh, Pennsylvania, September, 2006.

B.H. Juang, and L.R. Rabiner,”Signal restoration by spectral mapping,” in Proc. ICASSP’87, pp.2368-2372, April 1987.

S. Kozat, K. Visweswariah, and R. Gopinath, "Feature adaptation based on Gaussian posteriors," in Proc. ICASSP'06, Tolouse, France, April 2006.

T. Kristjansson, B. Frey, "Accounting for uncertainity in observations: A new paradigm for robust speech recognition," in Proc. ICASSP'02, Orlando, Florida, May 2002.

C.H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," Speech Communication, vol. 25, pp. 29-47, 1998.

C.H. Lee, and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," Proceedings of the IEEE, vol. 88, no. 88, pp. 1241-1269, August 2000.

C. Leggetter, and P. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in Proc. ARPA spoken language technology workshop, pp. 104-109, Feb 1995.

J. Li, L. Deng, Y. Gong, and A. Acero, "High performance HMMadaptation with joint compensation of additive and convolutive distortions via vector Taylor series," in Proc. ASRU 2007, Kyoto, Japan, 2007.

H. Liao, and M. Gales, "Adaptive training with joint uncertainity decoding for robust recognition of noisy data," in Proc. ICASSP'07, Honolulu, HI, April 2007.

H. Liao, and M. Gales, "Joint uncertainity decoding for noise robust speech recognition," in Proc. Eurospeech'05, Lisbone, Portugal, September 2005.

R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word recognition," Proc. of DARPA Speech Recognition Workshop, Mar. 24-26, 1987, pp. 96-99.

C. Mokbel, and G. Chollet,"Word recognition in the car:Speech enhancement/Spectral transformations," in Proc. ICASSP'91, Toronto, 1991.

P.J. Moreno, B. Raj, and R.M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in Proc. ICASSP, Atlanta, GA, May 1996, pp.733-736.

L. Neumeyer, and M. Weintraub, "Probabilistic optimal filtering for robust speech recognition," in Proc. ICASSP'94, Adelaide, Australia, April 1994.

M.K. Omar, personal communication.

A. Potamianos, and P. Maragos, "Time-frequency distributions for automatic speech recognition, "IEEE Transactions on Speech and Audio Processing, vol. 9, pp. 196-200, March 2001.

G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space projections for speaker adaptation," in Proc. ICASSP'01, Salt lake City,Utah, April, 2001.

V. Stouten, H. Van Hamme, and P. Wambacq, "Accounting for the uncertainity of speech estimates in the context of model-based feature enhancement," in Proc. ICSLP'04, Jeju, Korea, September 2004.

Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Transactions on Speech and Audio Processing, vol. 6, pp. 131-142, January 1998.

K. Visweswariah, and P. Olsen, "Feature adaptation using projection of Gaussian posteriors," in Proc. Interspeech'05, Lisbone, Portugal, September 2005.

Zh. Yan, F. Soong, and R.Wang, "Word graph based feature enhancement for noisy speech recognition," Proc. ICASSP, Honolulu, HI, April 2007.

S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P.Woodland, The HTK book (for HTK Version 3.1), December 2001.

Speech Recognition

Edited by France Mihelic and Janez Zibert

ISBN 978-953-7619-29-9

Hard cover, 550 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

Chapters in the first part of the book cover all the essential speech processing techniques for building robust, automatic speech recognition systems: the representation for speech signals and the methods for speech-features extraction, acoustic and language modeling, efficient algorithms for searching the hypothesis space, and multimodal approaches to speech recognition. The last part of the book is devoted to other speech processing applications that can use the information from automatic speech recognition for speaker identification and tracking, for prosody modeling in emotion-detection systems and in other speech processing applications that are able to operate in real-world environments, like mobile communication services and smart homes.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mohamed Afify, Xiaodong Cui and Yuqing Gao (2008). A Family of Stereo-Based Stochastic Mapping Algorithms for Noisy Speech Recognition, Speech Recognition, France Mihelic and Janez Zibert (Ed.), ISBN: 978-953-7619-29-9, InTech, Available from: http://www.intechopen.com/books/speech_recognition/a_family_of_stereo-based_stochastic_mapping_algorithms_for_noisy_speech_recognition

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen