A MAXIMUM LIKELIHOOD TRAINING APPROACH TO IRRELEVANT VARIABILITY COMPENSATION BASED ON PIECEWISE LINEAR TRANSFORMATIONS

Qiang Huo and Donglai Zhu

Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong
(Email: qhuo@cs.hku.hk, dzhu@i2r.a-star.edu.sg)

10.21437/Interspeech.2006-345

ABSTRACT

In our previous works, a maximum likelihood training approach was developed based on the concept of stochastic vector mapping (SVM) that performs a frame-dependent bias removal to compensate for environmental variabilities in both training and recognition stages. Its effectiveness was confirmed by evaluation experiments on Aurora2 and Aurora3 databases. In this paper, we present an extended ML formulation to entertain some new SVM functions that are piecewise linear transformations and are more flexible than the frame-dependent bias removal. Evaluation results on Finnish Aurora3 database show that in comparison with the performance of a baseline system based on ML-trained CDHMMs without feature compensation, the previous and the new SVM-based feature compensation approaches achieve a relative word error rate reduction of 15.7% and 26.1% respectively for well-matched condition.

Index Terms: robust speech recognition, feature compensation, maximum likelihood, hidden Markov model.

1. INTRODUCTION

Using feature transformation in training and/or recognition stages to compensate for possible "distortions" caused by factors irrelevant for phonetic classification has been studied in robust automatic speech recognition (ASR) area for many years. In the past several years, we've also been working on this research topic based on the concept of stochastic vector mapping (SVM) that performs a frame-dependent bias removal to compensate for "environmental" variabilities in both training and recognition stages. In the following, our past attempts are summarized first.

Let's assume that a speech utterance corrupted by some "distortions" has been transformed into a sequence of feature vectors. Given a set of training data $\mathcal{Y} = \{Y_i\}_{i=1}^{I}$, where Y_i is a sequence of feature vectors of original speech, suppose that they can be partitioned into E "environment" classes, and the D-dimensional feature vector y under an environment class e follows the distribution of a mixture of Gaussians, $p(y|e) = \sum_{k=1}^{K} p(k|e)p(y|k,e) = \sum_{k=1}^{K} p(k|e)\mathcal{N}(y;\xi_k^{(e)}, R_k^{(e)})$, where $\mathcal{N}(\cdot;\xi,R)$ is a normal distribution with mean vector ξ and diagonal covariance matrix R. Readers are referred to [14] for the approach we used for the automatic clustering of environment conditions from training data $\mathcal{Y}$, the labelling of an utterance Y to a specific environment condition, and the estimation of the above model parameters. Given the set of Gaussian mixture models (GMM) $\{p(y|e)\}$, the task of frame-dependent SVM-based compensation is to estimate the compensated feature vector $\hat{x}$ from the original feature vector y by applying the environment-dependent transformation $\mathcal{F}(y;\Theta^{(e_y)})$, where $\Theta^{(e_y)}$ represents the trainable parameters of the transformation and e_y denotes the corresponding environment class to which y belongs. However, for the simplicity of notation, we will hereinafter simply use e to denote the environment class to which y belongs, if no confusion will be caused according to the context.

Previously, we have studied two SVM functions [12, 13, 15]. The first one is borrowed from [3] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_1(y;\Theta^{(e)}) = y + \sum_{k=1}^{K} p(k|y,e) b_k^{(e)} \ , \tag{1}$$

where

$$p(k|y,e) = \frac{p(k|e)p(y|k,e)}{\sum_{j=1}^{K} p(j|e)p(y|j,e)} \ , \tag{2}$$

and $\Theta^{(e)} = \{b_k^{(e)}\}_{k=1}^{K}$. The second SVM function is borrowed from [2] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_2(y;\Theta^{(e)}) = y + b_k^{(e)} \ , \tag{3}$$

where $\Theta^{(e)} = \{b_k^{(e)}\}_{k=1}^{K}$, and for the environment class e which y belongs to,

$$k = \arg \max_{k'=1,\ldots,K} p(k'|y,e) \ . \tag{4}$$

In our first attempt as reported in [12], an environment compensated minimum classification error (MCE) training approach was proposed for the joint design of SVM function parameters and HMM parameters of a recognizer. Our approach does not rely on the availability of the stereo recordings of both clean and noisy speech for the estimation of SVM function parameters, while the SPLICE algorithm proposed in [2, 3] requires stereo data. To initialize MCE training, an environment compensated maximum likelihood (ML) training approach was also developed but described only briefly due to the lack of space in [13]. Its detailed formulation was presented later in [15].

In recognition, given an unknown utterance Y, the most similar training environment class e is identified first (e.g. [14]). Then, the corresponding GMM and the mapping function are used to derive a compensated version of $\hat{X}$ from Y. For the convenience of notation, we also use hereinafter $\mathcal{F}(Y;\Theta^{(e)})$ to denote the compensated version of the utterance Y by transforming individual feature vector y_t as defined in previous SVM functions. After feature compensation, $\hat{X}$ is finally recognized by an HMM-based recognizer trained as described in [12] or [15].

There are several interesting works from other research groups that are related to our efforts. As discussed in [4], although the

This research was supported by a grant from the RGC of the Hong Kong SAR (Project Number HKU7039/02E). Donglai Zhu is now with Institute for Infocomm Research, Singapore.

fMPE approach reported in [10] was derived with a different motivation, interestingly, its feature transformation is essentially the same as what was used in [12]. The main difference lies in the objective function used (MPE (minimum phone error) in [10] *vs* MCE in [12]) and the corresponding optimization procedures for training transformation and HMM parameters. Another work is the MMI-SPLICE approach reported in [5] in which the objective function for parameter learning is maximum mutual information (MMI). As a remark, a better name for the above approach may be "MMI-PLICE" because no stereo data is required. A third work is reported in [11] in which piecewise linear transformations, that are more flexible than the frame-dependent bias removal in [12, 10, 5], are used for ML unsupervised online feature adaptation based on seed HMMs trained without feature compensation. Most recently, an effort was also reported in [6] to extend "MMI-PLICE" for entertaining piecewise linear transformations, but only experimental results for MMI training of transformations are reported. Encouraged by the promising results reported in [11], we have extended our environment compensated ML training approach [15] to entertain some more flexible piecewise linear transformations. The main purpose of this paper is to report our study on this topic.

The rest of the paper is organized as follows. In Section 2, we describe two new SVM functions and the corresponding ML training procedures. In Section 3, we report the experimental results, and finally we conclude the paper in Section 4.

2. WHAT'S NEW

2.1. New Stochastic Vector Mapping Functions

In this paper, two new SVM functions are studied. The first one is borrowed from [8] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_3(y;\Theta^{(e)}) = A^{(e)}y + b^{(e)} \ , \tag{5}$$

where $A^{(e)}$ is a nonsingular $D\times D$ matrix, $b^{(e)}$ is a D-dimensional vector, and $\Theta^{(e)} = \{A^{(e)}, b^{(e)}\}$.

The second SVM function we used is defined as follows:

$$\hat{x} \triangleq \mathcal{F}_4(y;\Theta^{(e)}) = A^{(e)}y + \sum_{k=1}^{K} p(k|y,e)b_k^{(e)} \ , \tag{6}$$

where $\Theta^{(e)} = \{A^{(e)}; b_k^{(e)}, k = 1,\ldots,K\}$. Similar to what we did in [15], we use $\mathcal{F}_4(y;\Theta^{(e)})$ in the recognition stage only. In the training stage, the following SVM function is used instead to simplify the relevant derivation and to reduce the computational complexity:

$$\hat{x} \triangleq \mathcal{F}_5(y;\Theta^{(e)}) = A^{(e)}y + b_k^{(e)} \ , \tag{7}$$

where k is calculated by using Eq. (4).

Let's assume that each basic speech unit in our speech recognizer is modelled by a Gaussian mixture continuous density HMM (CDHMM), whose parameters are denoted as $\lambda = \{\pi_s, a_{ss'}, c_{sm}, \mu_{sm}, \Sigma_{sm}; s, s' = 1,\cdots,S; m = 1,\cdots,M\}$, where S is the number of states, M is the number of Gaussian components for each state, $\{\pi_s\}$ is the initial state distribution, $a_{ss'}$'s are state transition probabilities, c_{sm}'s are Gaussian mixture weights, $\mu_{sm} = [\mu_{sm1},\cdots,\mu_{smD}]^{Tr}$ is a D-dimensional mean vector, and $\Sigma_{sm} = diag\{\sigma^2_{sm1},\cdots,\sigma^2_{smD}\}$ is a diagonal covariance matrix. Our environment compensated ML training approach is to maximize, by adjusting SVM function parameters $\Theta = \{\Theta^{(e)}, e = 1,\cdots,E\}$ and CDHMM parameters $\Lambda = \{\lambda\}$, the following likelihood function

$$\mathcal{L}(\Theta,\Lambda) = \prod_{i=1}^{I} p(\mathcal{F}(Y_i;\Theta)|\Lambda) \tag{8}$$

defined on the training set $\mathcal{Y}$. In the following two subsections, we describe in detail two ML training procedures for SVM functions $\mathcal{F}_3(y;\Theta^{(e)})$ and $\mathcal{F}_4(y;\Theta^{(e)})$ respectively.

2.2. Joint ML Training of the Parameters of SVM Function $\mathcal{F}_3(y;\Theta^{(e)})$ and CDHMMs

When SVM function $\mathcal{F}_3(y;\Theta^{(e)})$ is used, our joint ML training procedure is as follows:

Step 1: First, a set of CDHMMs, Λ, are trained from multi-condition training data $\mathcal{Y}$ and used as the initial values of HMM parameters. The initial values of transformation matrices $A^{(e)}$ are set to be identity matrices and the initial values of bias vectors $b^{(e)}$ are set to be zero vectors.

Step 2: Second, given the HMM parameters Λ, for each environment class e, we estimate the environment dependent mapping function parameters $\bar{\Theta}^{(e)}$ by using the Constrained MLLR (CMLLR) approach described in [8] to increase the likelihood function $\mathcal{L}(\Theta,\Lambda)$. One EM iteration is performed in our experiments.

Step 3: Third, we transform each training utterance using the mapping function in Eq. (5) with parameters $\bar{\Theta}$. Using the environment compensated utterances, several (5 in our experiments) EM iterations are performed to re-estimate CDHMM parameters $\bar{\Lambda}$, with an increase of the likelihood function $\mathcal{L}(\bar{\Theta},\Lambda)$.

Step 4: Repeat **Step 2** and **Step 3** several times if necessary. In our experiments, we skipped this step.

After the above steps, we obtain the $\bar{\Theta}$ and $\bar{\Lambda}$ as an ML estimation of mapping function parameters and CDHMM parameters, which can be used in the recognition stage for feature compensation as shown in Eq. (5).

2.3. Joint ML Training of the Parameters of SVM Function $\mathcal{F}_4(y;\Theta^{(e)})$ and CDHMMs

When SVM function $\mathcal{F}_4(y;\Theta^{(e)})$ is used, our joint ML training procedure is as follows:

Step 1: *Initialization*

This step is the same as **Step 1** described in the previous subsection, except that the initial values of the bias vectors $b_k^{(e)} = [b_{k1}^{(e)},...,b_{kD}^{(e)}]^{Tr}$ are set to be zero vectors.

Step 2: *Estimating SVM Function Parameters* Θ

Given the HMM parameters Λ, for each environment class e, we estimate the environment dependent mapping function parameters $\bar{\Theta}^{(e)}$ to increase the likelihood function $\mathcal{L}(\Theta,\Lambda)$. Let's consider a particular environment class e and use I_e to denote the subset of the subscript of training utterance Y_i which belongs to the environment class e. By using the general EM algorithm and the specific

SVM function in Eq. (7) for feature compensation, the auxiliary $\mathcal{Q}$-function for $\Theta^{(e)}$ becomes

$$\mathcal{Q}_e = \sum_{i\in I_e}\sum_t\sum_s\sum_m \zeta_{it}(s,m) \log\left[\mathcal{N}(A^{(e)}y_{it}+b_{q_t}^{(e)};\mu_{sm},\Sigma_{sm})|\det(A^{(e)})|\right] . \quad (9)$$

In the above equation, $\zeta_{it}(s,m)$ is the occupation probability of Gaussian component m in state s, at time t of the current compensated observation. It can be calculated with a Forward-Backward procedure using the training utterance $\hat{X}_i$ (compensated from Y_i with the current Θ) against the current HMM parameters Λ in the E-step. y_{it} is the t-th frame feature vector of the utterance Y_i. For simplicity, the following two-stage iterative procedure is used to increase the above $\mathcal{Q}$-function: firstly, update $A^{(e)}$ while keeping $b_k^{(e)}$ fixed; secondly, update $b_k^{(e)}$ by using the feature vectors transformed by $A^{(e)}$ only.

Step 2-1: *Estimating* $A^{(e)}$

The derivation of the updating formula for $A^{(e)}$ is similar to that in CMLLR [8]. By differentiating the $\mathcal{Q}$-function with respect to the r-th row of $A^{(e)}$ (hereinafter denoted as $A_r^{(e)}$) and equating to zero, the following updating formula can be derived:

$$A_r^{(e)} = \alpha_r^{(e)}p_r^{(e)}G_r^{(e)-1} + v_r^{(e)}G_r^{(e)-1} , \quad (10)$$

where $p_r^{(e)}$ is the cofactor row vector $[c_{r1}^{(e)} \ldots c_{rD}^{(e)}]$ with $c_{rl}^{(e)} = cof(A_{rl}^{(e)})$, and

$$G_r^{(e)} = \sum_{i\in I_e}\sum_t\sum_s\sum_m \frac{1}{\sigma_{smr}^2}\zeta_{it}(s,m)y_{it}y_{it}^{Tr} , \quad (11)$$

$$v_r^{(e)} = \sum_{i\in I_e}\sum_t\sum_s\sum_m \frac{1}{\sigma_{smr}^2}\zeta_{it}(s,m)(\mu_{smr}-b_{q_tr}^{(e)})y_{it}^{Tr} , \quad (12)$$

$$\alpha_r^{(e)} = -\frac{\varepsilon_2}{2\varepsilon_1} \pm \frac{\sqrt{\varepsilon_2^2+4\beta^{(e)}\varepsilon_1}}{2\varepsilon_1} , \quad (13)$$

$$\beta^{(e)} = \sum_{i\in I_e}\sum_t\sum_s\sum_m \zeta_{it}(s,m) , \quad (14)$$

$$\varepsilon_1 = p_r^{(e)}G_r^{(e)-1}p_r^{(e)Tr} , \quad (15)$$

$$\varepsilon_2 = p_r^{(e)}G_r^{(e)-1}v_r^{(e)Tr} . \quad (16)$$

The value of $\alpha_r^{(e)}$ is selected that maximizes

$$\mathcal{Q}_e = \beta^{(e)}\log|\alpha_r^{(e)}\varepsilon_1+\varepsilon_2| - \frac{1}{2}\alpha_r^{(e)2}\varepsilon_1 . \quad (17)$$

Step 2-2: *Estimating* $b_k^{(e)}$

After the above step, we transform each training feature vector y to $\tilde{x}$ by using the updated $\{A^{(e)}\}$ as follows:

$$\tilde{x} = A^{(e)}y . \quad (18)$$

Then, as described in detail in [15], $\{b_k^{(e)}\}$ can be estimated by using the compensated feature vectors $\{\tilde{x}\}$ as follows:

$$b_{kd}^{(e)} = \frac{\sum_{i\in I_e}\sum_{t,s,m}\mathbf{1}[k,i,t]\zeta_{it}(s,m)(\mu_{smd}-\tilde{x}_{itd})/\sigma_{smd}^2}{\sum_{i\in I_e}\sum_{t,s,m}\mathbf{1}[k,i,t]\zeta_{it}(s,m)/\sigma_{smd}^2} , \quad (19)$$

where

$$\mathbf{1}[k,i,t] = \begin{cases} 1 \text{ if } k = \arg\max_{k'} p(k'|y_{it},e) \\ 0 \text{ otherwise} \end{cases} . \quad (20)$$

In the above equation, $\zeta_{it}(s,m)$ is the occupation probability of Gaussian component m in state s, at time t of the current compensated observation. It can be calculated with a Forward-Backward procedure using the training utterance $\hat{X}_i$ against the current HMM parameters Λ. Each feature vector in $\hat{X}_i$ is a result of the following feature compensation:

$$\hat{x} = \tilde{x} + b_k^{(e)} , \quad (21)$$

where k is calculated by using Eq. (4).

The above iteration of **Step 2-1** and **Step 2-2** can be repeated N_b times if necessary. In our experiments, only one iteration is performed.

Step 3: *Estimating CDHMM Parameters* Λ

After the above step, we further transform each training feature vector $\tilde{x}$ derived in Eq. (18) to $\hat{x}$ as follows:

$$\hat{x} = \tilde{x} + \sum_{k=1}^{K} p(k|y,e)b_k^{(e)} . \quad (22)$$

Using the above environment compensated training feature vectors $\{\hat{x}\}$, N_h EM iterations are performed to re-estimate CDHMM parameters $\bar{\Lambda}$, with an increase of the likelihood function $\mathcal{L}(\bar{\Theta},\Lambda)$.

Step 4: *Repeat* **Step 2** *and* **Step 3** N_e *times*.

According to our past experience as reported in [15], we also use the setting of $N_b = 1, N_h = 5, N_e = 1$ in this study. In recognition stage, the feature vectors of an unknown utterance are compensated by using Eq. (6).

3. EXPERIMENTS AND RESULTS

We use Aurora3 database to verify the effectiveness of our algorithms. Aurora3 contains utterances of connected digits in four European languages, namely Finnish, Spanish, German and Danish. All utterances were recorded by using both close-talking (CT) and hands-free (HF) microphones in cars under several driving conditions to reflect some realistic scenarios for typical in-vehicle ASR applications. There are roughly three conditions: *quiet*, *low noise*, and *high noise*. For each language, the database is divided into three subsets according to matching degree between training data and test data: Well-Matched condition (WM), Middle Mismatched condition (MM) and High Mismatched condition (HM). In the following discussions, only the results on the WM subset of the Finnish Database [1] are used, where both training and testing data include utterances recorded by both CT and HF microphones from all noise conditions.

In our experiments, the ETSI Advanced Front-End (AFE) as described in [7] is used for feature extraction from a speech utterance. A feature vector sequence is extracted from the input speech utterance via a sequence of processing modules that include noise reduction, waveform processing, cepstrum calculation, blind equalization, and "server feature processing". Each frame of feature vector has 39 features that consists of 12 MFCCs (C_1 to C_{12}),

Table 1. A comparison of word error rates (WERs in %) of different approaches and the relative word error rate reduction (in %) of several SVM-based approaches versus a baseline system.

Approaches	WER (%)	Relative Improvement (%)
Baseline	3.95	-
SVM1	3.33	15.7
SVM3	3.08	22.0
SVM4	2.92	26.1

a combined log energy and C_0 term, and their first and second order derivatives. Although all the feature vectors are computed from a given speech utterance, the feature vectors that are sent to the speech recognizer and the training module are those corresponding to speech frames, as detected by a VAD module described in Annex A of [7].

Each digit is modeled as a whole word left-to-right CDHMM with 16 emitting states, 3 Gaussian mixture components with diagonal covariance matrices per state. Besides, two pause models, "sil" and "sp", are created to model the silence before/after the digit string and the short pause between any two digits, respectively. The "sil" model is a 3-emitting state CDHMM with a flexible transition structure as that of HMM described in [9]. Each state is modeled by a mixture of 6 Gaussian components with diagonal covariance matrices. The "sp" model consists of 2 dummy states and a single emitting state which is tied with the middle state of "sil".

During recognition, an utterance can be modeled by any sequence of digits with the possibility of a "sil" model at the beginning and at the end and a "sp" model between any two digits. All of the recognition experiments are performed with the search engine of HTK3.0 toolkit.

A set of baseline CDHMMs are trained first by running the training scripts published in Aurora3 CDs, i.e., the standard ML training implemented in HTK. The Word Error Rate (WER) of this CDHMM-based baseline system is 3.95%.

As in [15], in SVM-based experiments, all the training data are clustered into 8 different environment classes (i.e. $E = 8$), of which each is modeled by a GMM consisting of 32 Gaussian components (i.e. $K = 32$) [14]. As reported in [15], an SVM-based approach using the SVM function in Eq. (1), called SVM1 in [15], achieves a WER of 3.33% under the same experimental setup as described above.

In this study, the following sets of new SVM-based experiments are conducted:

SVM3: the SVM function $\mathcal{F}_3(y;\Theta^{(e)})$ in Eq. (5) is used in recognition for feature compensation and the training procedure in Section 2.2 is used for training model parameters;

SVM4: the SVM function $\mathcal{F}_4(y;\Theta^{(e)})$ in Eq. (6) is used in recognition for feature compensation and the training procedure in Section 2.3 is used for training model parameters.

SVM3 and SVM4 approaches achieve WERs of 3.08% and 2.92% respectively. Table 1 compares word error rates of the above different approaches. So far, SVM4 approach achieves the best performance.

4. CONCLUSIONS AND DISCUSSIONS

In this paper, we have presented an ML training approach to irrelevant variability compensation based on piecewise linear transformations. In comparison with our previous approaches based on frame-dependent bias removal, the new approach achieves a much better performance because more flexible feature transformations offer a better opportunity to compensate for some more complicated distortions.

Although we have demonstrated the usefulness of the SVM-based approaches for several robust ASR applications where diversified yet representative training data are available, the performance improvement of SVM-based approaches is less significant in the case of that there is a severe mismatch between training and testing conditions. In order to improve the performance further, one possibility is to perform unsupervised online adaptation of SVM function parameters. We have conducted a study along this direction and will report its results elsewhere.

5. REFERENCES

[1] Aurora document AU/217/99, "Availability of Finnish SpeechDat-Car database for ETSI STQ WI008 front-end standardisation," Nokia, Nov 1999.

[2] L. Deng, A. Acero, M. Plumpe, and X.-D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," *Proc. ICSLP-2000*, 2000, pp.III-806-809.

[3] L. Deng, A. Acero, L. Jiang, J. Droppo, and X.-D. Huang, "High-performance robust speech recognition using stereo training data," *Proc. ICASSP-2001*, 2001, pp.I-301-304.

[4] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Processing Letters*, Vol.12, No.6, pp.477-480, 2005.

[5] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," *Proc. Eurospeech-2005*, 2005, pp.989-992.

[6] J. Droppo, M. Mahajan, A. Gunawardana, and A. Acero, "How to train a discriminative front end with stochastic gradient descent and maximum mutual information," *Proc. ASRU-2005*, 2005, pp.41-46.

[7] ETSI standard document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms", ETSI ES 202 050 v1.1.1 (2002-10), 2002.

[8] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, Vol. 12, pp.75-98, 1998.

[9] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *ISCA ITRW ASR-2000*, Paris, France, 2000.

[10] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," *Proc. ICASSP-2005*, 2005, pp.I-961-964.

[11] K. Visweswariah and P. Olsen, "Feature adaptation using projection of Gaussian posteriors," *Proc. Eurospeech-2005*, 2005, pp.1785-1788.

[12] J. Wu and Q. Huo, "An environment compensated minimum classification error training approach and its evaluation on Aurora2 database," *Proc. ICSLP-2002*, Denver, 2002, pp.I-453-456.

[13] J. Wu and Q. Huo, "Several HKU approaches for robust speech recognition and their evaluation on Aurora connected digit recognition tasks," *Proc. Eurospeech-2003*, 2003, pp.21-24.

[14] J. Wu, D. Zhu and Q. Huo, "A study of minimum classification error training for segmental switching linear Gaussian hidden Markov models," *Proc. ICSLP-2004*, 2004.

[15] J. Wu, Q. Huo and D. Zhu, "An environment compensated maximum likelihood training approach based on stochastic vector mapping," *Proc. ICASSP-2005*, 2005, pp.I-429-432.