Available online at www.sciencedirect.com

SciVerse ScienceDirect

Speech Communication 55 (2013) 950–963

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Rapid speaker adaptation using compressive sensing

Wen-Lin Zhang[a,*], Dan Qu[a], Wei-Qiang Zhang[b], Bi-Cheng Li[a]

[a] *Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China*

[b] *Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*

Received 29 January 2013; received in revised form 17 June 2013; accepted 20 June 2013
Available online 28 June 2013

Abstract

Speaker-space-based speaker adaptation methods can obtain good performance even if the amount of adaptation data is limited. However, it is difficult to determine the optimal dimension and basis vectors of the subspace for a particular unknown speaker. Conventional methods, such as eigenvoice (EV) and reference speaker weighting (RSW), can only obtain a sub-optimal speaker subspace. In this paper, we present a new speaker-space-based speaker adaptation framework using compressive sensing. The mean vectors of all mixture components of a conventional Gaussian-Mixture-Model-Hidden-Markov-Model (GMM-HMM)-based speech recognition system are concatenated to form a supervector. The speaker adaptation problem is viewed as recovering the speaker-dependent supervector from limited speech signal observations. A redundant speaker dictionary is constructed by a combination of all the training speaker supervectors and the supervectors derived from the EV method. Given the adaptation data, the best subspace for a particular speaker is constructed in a maximum a posterior manner by selecting a proper set of items from this dictionary. Two algorithms, i.e. matching pursuit and l_1 regularized optimization, are adapted to solve this problem. With an efficient redundant basis vector removal mechanism and an iterative updating of the speaker coordinate, the matching pursuit based speaker adaptation method is fast and efficient. The matching pursuit algorithm is greedy and sub-optimal, while direct optimization of the likelihood of the adaptation data with an explicit l_1 regularization term can obtain better approximation of the unknown speaker model. The projected gradient optimization algorithm is adopted and a few iterations of the matching pursuit algorithm can provide a good initial value. Experimental results show that matching pursuit algorithm outperforms the conventional testing methods under all testing conditions. Better performance is obtained when direct l_1 regularized optimization is applied. Both methods can select a proper mixed set of the eigenvoice and reference speaker supervectors automatically for estimation of the unknown speaker models.
© 2013 Elsevier B.V. All rights reserved.

Keywords: Speaker adaptation; Speaker subspace; Compressive sensing; Matching pursuit; l_1 regularization

1. Introduction

Model space speaker adaptation is a key component of current speech recognition systems. The basic idea is that given some adaptation data, the parameters of a speaker independent (SI) system are transformed to match the acoustic characteristics of an unknown speaker, resulting in a speaker dependent (SD) system. Many speaker adaptation methods have been proposed in the past decades (Huo and Lee, 1997; Shinoda, 2010). Among them, the speaker-space-based methods have been proven to the fastest, which can obtain good performance given only a few seconds adaptation data. A common assumption for these methods depends on that the SD acoustic models lie in a low dimensional speaker subspace. This low dimensional speaker subspace is obtained from the training data, thus accounts for the a priori knowledge of the training speakers. The basis of the speaker subspace can be obtained from the training speakers' SD models and speaker adaptation is no more than estimation of the coordinate of the new SD model in this subspace.

* Corresponding author. Tel.: +86 371 81630727; fax: +86 371 81630984.
E-mail addresses: zwlin_2004@163.com (W.-L. Zhang), qudanqudan@sina.com (D. Qu), wqzhang@tsinghua.edu.cn (W.-Q. Zhang), lbclm@163.com (B.-C. Li).

0167-6393/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.specom.2013.06.012

Once the basis is determined, the speaker-dependent coordinate vector can be estimated using a simple quadratic optimization method (Hazen and Glass, 1997; Kuhn et al., 2000). The main difference of various speaker-space-based methods lies in the construction of the basis of the speaker subspace. For example, in the well-known eigenvoice (EV) method (Kuhn et al., 2000; Kenny et al., 2004), the basis vectors which are called the eigenvoices, are obtained by performing principal component analysis (PCA) on the training speaker SD model parameters. Then the K leading eigenvectors which represent the greatest variabilities of different training speaker models are preserved as the K basis vectors. In reference speaker weighting (RSW) (Hazen and Glass, 1997; Mak et al., 2006; Teng et al., 2009), all training speaker SD models are reserved for the candidate basis vectors. During speaker adaptation, a subset of them are chosen according to some heuristic criteria to linearly represents the unknown SD model. In aspect model weighting (AMW) (Hahm et al., 2010), the basis is constructed by a set of aspect models which are the mixture model of the training speakers' SD models and are trained based on likelihood maximization with respect to the training data. In all these methods, given the speaker subspace, the corresponding coordinate of an unknown speaker is always estimated using the maximum likelihood criterion. However, there is a common difficulty in determining the dimension of the speaker subspace. When the adaptation data is limited, a small dimensional speaker subspace is preferred. As the adaptation data increases, larger speaker subspace yields better performance. Unfortunately, none of these methods can provide the best speaker subspace given varying amounts of adaptation data for a particular unknown speaker.

In this paper, we discuss the generalization of the speaker-space-based speaker adaptation methods using the compressive sensing theory (Donoho, 2006). In fact, the parameters of the SD model lie in a very high dimensional space. The core issue of the speaker adaptation problem is to estimate the high dimensional vector of model parameters from a few speech signal observations.

Breakthrough results in compressive sensing (CS) have shown that high dimensional signals (vectors) can often be accurately recovered from a relatively small number of non-adaptive linear projection observations, provided that they possess a compact representation in some basis. In machine learning, sparse representation and compressive sensing are widely employed to address the problem of data sparsity and model complexity. Recently, it finds a lot of applications in speech processing and recognition. For instance, exemplar-based sparse representations were proposed for noise robust automatic speech recognition (Gemmeke et al., 2011). l_1 regularization is used to derive sparse representations of GMM-supervectors for speaker identification (Naseem et al., 2010) and verification (Kua et al., 2011). In Boominathan and Murty (2012), the orthogonal matching pursuit algorithm is used to derive sparse representation of each feature vector using a dictionary of feature vectors belonging to many speakers for speaker identification. Recently, l_1 and l_2 regularization method is used to derive an i-vector based sparse representation classification method for speaker verification (Kua et al., 2013). For speaker adaptation of a speech recognition system, element-wise l_2 regularization is applied to the maximum likelihood linear regression (MLLR) matrix, resulting in a ridge MLLR method (Li et al., 2010), which can make a shrinkage of the adaptation parameters and give significant word error rate reduction from the errors obtained by standard MLLR in an utterance-by-utterance unsupervised adaptation scenario. By imposing sparseness constraints, sparse maximum a posteriori (MAP) adaptation is proposed in Olsen et al. (2011) and Olsen et al. (2012), which can save significantly on storage and even improve the quality of the resulting speaker-dependent model.

Actually, all the speaker-space-based methods implicitly assume a low dimensional speaker subspace, which provides a sparse representation of the SD model parameters. The main contribution of this paper lies in two aspects: firstly, we use a redundant basis dictionary to construct the speaker space. The basis vectors are a combination of all the eigenvoices and training speakers' SD models. As a benefit of PCA, the subspace spanned by the leading eigenvoices captures the most inter-speaker variability of all training speaker models. However, the intra-speaker variability is not modeled and each eigenvoice is no longer a valid training speaker model. In reference speaker weighting method, each basis vector (reference model) is constructed directly from a training speaker model. All basis vectors are equally important and the intra-speaker information is well preserved. Experimental results of Teng et al. (2007) show that the results of eigenvoices always fall between the best and the worst results of random selections of reference speaker models. The motivation of using a dictionary combining all the eigenvoices and training speaker models is that the advantage of both methods (i.e. RSW and EV) can be utilized during speaker adaptation. Secondly, given the adaptation data, algorithms from compressive sensing theory are introduced to automatically select a varying subset of the dictionary entries which can best represent the unknown SD model. Different from the RSW method, the selection process is based on direct likelihood maximization with respect to the adaptation data. Two optimization scheme, namely the matching pursuit scheme (Mallat and Zhang, 1993; Tropp and Gilbert, 2007) and the l_1 regularized optimization scheme (Tibshirani, 1996; Figueiredo et al., 2007), are derived for speaker adaptation. Matching pursuit is a greedy algorithm, which iteratively selects one basis vector for combination until some stopping condition is reached, while the l_1 regularized optimization algorithm uses an explicit l_1 norm regularization term to force some components of the speaker coordinate vector to be zero, thus selects the optimal basis vectors through the nonzero components. Although the matching pursuit algorithm is sub-optimal, but it is very fast and can provide a good starting point for the l_1 regularized optimization algorithm, which can always obtain better solution.

The rest of this paper is organized as follows. In Section 2, a brief review of the basic principal of speaker-space-based speaker adaptation methods is given. The advantages and disadvantages of different conventional methods are discussed. In Section 3, speaker adaptation using matching pursuit is derived. The sequential selection of the basis vectors is presented. A redundant basis vector removal mechanism is introduced to guarantee the numerical stability and a fast iterative updating algorithm of the speaker coordinate vector is given to improve the convergence speed. In Section 4, we apply l_1 regularized optimization to get a better estimation of the unknown speaker model. We present the experimental set-up, recognition results and discussions in Section 5. Finally we conclude this paper in Section 6.

2. Review of the speaker-space-based speaker adaptation

In this section, we give a brief review of the basic principle of speaker-space-based speaker adaptation and introduces the notations used in this paper.

Suppose there are a set of speaker independent hidden Markov models (HMMs) containing a total of M mixture components and a training speaker population comprising S speakers using D dimensional feature vector. For mixture component m, let μ_m and Σ_m denote the speaker independent mean vector and covariance matrix respectively. For speaker s and mixture component m, let $\mu_m^{(s)}$ denotes the speaker dependent mean vector. A speaker supervector denoted by $\boldsymbol{\mu}^{(s)}$ is a supervector obtained by concatenating the mean vectors $\mu_m^{(s)}, m = 1, 2, \cdots, M$, for a specific speaker s. The order of mixture component is arbitrary, but all SD models and the SI model must be ordered in the same way. Accordingly, the speaker supervector of the SI model is defined as $\mu = \begin{bmatrix} \mu_1^T & \mu_2^T & \cdots & \mu_M^T \end{bmatrix}^T$. Both $\boldsymbol{\mu}^{(s)}$ and μ lie in an $M \cdot D$ dimensional space. In this paper, we only discuss the adaptation of the mean vectors, that is, the estimation of the speaker supervector $\boldsymbol{\mu}^{(s\prime)}$ for an unknown speaker $s\prime$ using some adaptation data.

2.1. General formulation

In speaker-space-based speaker adaptation, the basic assumption is that the speaker supervector $\boldsymbol{\mu}^{(s)}$ is located in a low dimensional linear subspace $\boldsymbol{\Gamma}^K$, where K ($K \ll M \cdot D$) is the dimension of the speaker subspace. Assume the basis vectors of $\boldsymbol{\Gamma}^K$ are $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_K\}$ and the corresponding coordinate of speaker s is $\mathbf{x}_K^{(s)}$. Let the mth mixture component of the kth basis vector is $\mathbf{e}_{k,m}$ and define $\mathbf{E}_{K,m} = \begin{bmatrix} \mathbf{e}_{1,m} & \mathbf{e}_{2,m} & \cdots & \mathbf{e}_{K,m} \end{bmatrix}$, then the decomposition of the speaker dependent mean vector for component m can be denoted by

$$\boldsymbol{\mu}_m^{(s)} = \boldsymbol{\mu}_m + \mathbf{E}_{K,m}\mathbf{x}_K^{(s)}, \tag{1}$$

where the SI mean vector $\boldsymbol{\mu}_m$ can be viewed as the origin of the speaker subspace. In the following, we write $\mathbf{x}_K$ for $\mathbf{x}_K^{(s)}$ without any confusion.

Let $\mathbf{O} = \{\mathbf{o}(1), \mathbf{o}(2), \cdots, \mathbf{o}(T)\}$ denotes the sequence of feature vectors of the adaptation data, using the expectation maximization (EM) algorithm (Kuhn et al., 2000), the auxiliary function is given as follows

$$Q(\mathbf{x}_K) = -\frac{1}{2}\sum_t\sum_m \gamma_m(t)[\mathbf{o}(t) - \boldsymbol{\mu}_m - \mathbf{E}_{K,m}\mathbf{x}_K]^T \boldsymbol{\Sigma}_m^{-1}[\mathbf{o}(t) - \boldsymbol{\mu}_m - \mathbf{E}_{K,m}\mathbf{x}_K] \tag{2}$$

where $\gamma_m(t)$ is the posterior probability of being in mixture m at time t given the observation sequence $\mathbf{O}$ and current estimation of SD model.

Let $s_m^{(0)} = \sum_t \gamma_m(t)$ and $\mathbf{s}_m^{(1)} = \sum_t \gamma_m(t)[\mathbf{o}(t) - \boldsymbol{\mu}_m]$ be the zeroth-order and first-order statistics of the observations, Eq. (2) can be simplified to

$$Q(\mathbf{x}_K) = -\frac{1}{2}\mathbf{x}_K^T\mathbf{A}_K\mathbf{x}_K + \mathbf{b}_K^T\mathbf{x}_K + Const \tag{3}$$

where

$$\mathbf{A}_K = \sum_m s_m^{(0)}\mathbf{E}_{K,m}^T\boldsymbol{\Sigma}_m^{-1}\mathbf{E}_{K,m} = \begin{bmatrix} a(1,1) & a(1,2) & \cdots & a(1,K) \\ a(2,1) & a(2,2) & \cdots & a(2,K) \\ \vdots & \vdots & \ddots & \vdots \\ a(K,1) & a(K,2) & \cdots & a(K,K) \end{bmatrix} \tag{4}$$

and

$$\mathbf{b}_K = \sum_m \mathbf{E}_{K,m}^T\boldsymbol{\Sigma}_m^{-1}\mathbf{s}_m^{(1)} = \begin{bmatrix} b(1) \\ b(2) \\ \vdots \\ b(K) \end{bmatrix}, \tag{5}$$

where

$$a(i,j) = \sum_m s_m^{(0)}\mathbf{e}_{i,m}^T\boldsymbol{\Sigma}_m^{-1}\mathbf{e}_{j,m}, i, \quad j = 1, 2, \cdots, K \tag{6}$$

and

$$b(k) = \sum_m \mathbf{e}_{k,m}^T\boldsymbol{\Sigma}_m^{-1}\mathbf{s}_m^{(1)}, \quad k = 1, 2, \cdots, K. \tag{7}$$

Setting the derivative of (3) with respect to $\mathbf{x}_K$ to zero yields

$$\mathbf{x}_K = \mathbf{A}_K^{-1}\mathbf{b}_K. \tag{8}$$

The above formula is the well-known maximum likelihood eigen-decomposition (MLED) algorithm proposed in Kuhn et al. (2000).

2.2. Discussion of conventional methods

Eigenvoice (Kuhn et al., 2000) and reference speaker weighting (RSW) (Hazen and Glass, 1997) are two conventional speaker-space-based adaptation algorithms. Both methods use (8) to estimate the speaker coordinate $\mathbf{x}_K$.

The major difference lies in the construction of the basis of the speaker subspace. In eigenvoice method, the basis vectors are obtained by performing PCA to the training speaker supervectors, then the K leading eigenvectors which represent the greatest variabilities of different training speaker models are preserved as the K basis vectors. The eigenvoices are orthogonal and those corresponding to large eigenvalues are more important than those with small eigenvalues. However, the intra-speaker variability is not modeled. Each eigenvoice is no longer a valid training speaker model, which means that there does not exist a real speaker whose acoustic model corresponds to that eigenvoice. On the other hand, RSW stores all training speaker supervectors as candidate basis vectors and selects a subset of them as the "reference models" in the adaptation stage. All reference models are equally important and the intra-speaker information is well preserved. The selected subset can be viewed as a set of nonorthogonal basis vectors for the speaker subspace.

In Teng et al. (2007), an interesting phenomenon was observed that the results of eigenvoices fall between the best and the worst results of random selections of reference speakers. This fact shows that a direct combination of acoustic models could be better than a combination of a set of basis vectors produced by PCA, if the models to be combined are well selected. This phenomenon intrigues interest in the simpler RSW adaptation techniques (Mak et al., 2006; Teng et al., 2007; Teng et al., 2009).

Two different reference speaker selection strategies are proposed in recent literatures. One strategy is called the maximum-likelihood (ML) reference speaker selection (Mak et al., 2006), where the top K training speakers who have the highest likelihood of the adaptation data are taken as the reference speakers of the new speaker. The other one is called the reference model interpolation (RMI) (Teng et al., 2007), which dynamically select the reference models using the estimated interpolation weights as an indicator of the pertinence of reference models. The K reference models which have the biggest absolute weight values are selected as the pertinent reference models. However both methods are heuristic where the basis vector selection process is not directly related to the objective function of the maximum likelihood criterion (i.e. the auxiliary function (3)).

More recently, an aspect-model-based RSW method (Hahm et al., 2010) is proposed to obtain a more compact set of basis vectors, which are linear combinations of the training speaker supervectors and the combination factors are trained based on likelihood maximization with respect to the training data. As the eigenvoice method, these basis vectors are obtained and fixed prior to the speaker adaptation stage and the chosen speaker subspace is optimal in the average sense over all training speakers, but may not be optimal for a particular unknown speaker.

In this paper, we extend the speaker-space-based adaptation method using compressive sensing theory. Firstly, we use a redundant dictionary which contains all eigenvoices and training speaker supervectors, in hope that the advantages of them can be combined during speaker adaptation. Secondly, given the adaptation data, the basis vectors which best linearly represent the unknown SD model are selected through direct optimization of the auxiliary function (3). Unlike conventional methods, both the dimension (K) of the speaker subspace and the optimal basis vectors are determined automatically in maximum a posteriori manner through the optimization process.

In the following sections, matching pursuit and l_1 regularized optimization, which are two well-known algorithms for sparse signal recovery and compressive sensing, are introduced to derive efficient basis vector selection algorithms for the speaker adaptation task. Matching pursuit is an iterative greedy algorithm. Although it can only obtain a sub-optimal solution, it can be very fast and efficient using a carefully designed updating formula. Direct optimization of the objective function using an explicit l_1 norm regularization term can always obtain better sparse representation of the unknown SD model. Current optimization algorithms for the l_1 regularization problem are iterative, whose convergence speed depends on the initial value of the parameter vector. Fortunately, the matching pursuit algorithm can provide a good starting point.

3. Speaker adaptation using matching pursuit

The center of the speaker-space-based speaker adaptation is to approximate the unknown speaker's supervector by linear combination of a set of basis vectors. Because of the large dimension of the speaker supervector and the limited adaptation data available, the combination must be sparse, that is only a few basis vectors are actually used. So speaker-space-based adaptation is intrinsically a problem of sparse signal recovery from a few observations. However, computing an optimal k-term approximation of a signal (the speaker supervector $\boldsymbol{\mu}^{(s)}$) with k vectors selected in a redundant dictionary is NP-hard. In sparse signal recovery and compressive sensing, pursuit strategies construct non-optimal yet efficient approximations with computational algorithms. The well-known matching pursuits are greedy algorithms that select the dictionary vectors one by one, with applications to compression, denoising, and pattern recognition. In this section, we derive an efficient speaker adaptation algorithm using matching pursuit.

3.1. The key ideas

Matching pursuit (Mallat and Zhang, 1993) is first introduced by Mallat and Zhang which computes signal approximations from a redundant dictionary, by iteratively selecting one vector at a time. The objective of the original matching pursuit algorithm is to minimize the approximation error, while in speaker adaptation, using maximum likelihood criterion and the EM algorithm, the objective is to maximize the auxiliary function (3).

Suppose $\mathbb{D}$ is the redundant speaker supervector dictionary and let $\mathbb{D}_k$ denotes the subset of selected basis vectors at iteration k. The $(k+1)$th iteration of the matching pursuit based speaker adaptation consists of the following two steps:

1. **sequential maximum likelihood basis vector selection**: from the dictionary $\mathbb{D} \setminus \mathbb{D}_k$, find the basis vector $\mathbf{e}_{p_{(k+1)}}$ which can contribute to the largest improvement of the auxiliary function (3), and add it to the selected subset $\mathbb{D}_{k+1} = \mathbb{D}_k \bigcup \{\mathbf{e}_{p_{(k+1)}}\}$.
2. **speaker coordinate update**: estimate the coordinate $\mathbf{x}_{k+1}$ using elements of $\mathbb{D}_{k+1}$ as basis vectors of the speaker subspace.

Step 2 is corresponding to the back-projection step of the original matching pursuit algorithm (Mallat and Zhang, 1993). In speaker adaptation, it involves solving the linear Eq. (8). In order to guarantee numeric stability, matrix $\mathbf{A}_k$ must be avoided to be singular. So in addition to the above two steps, a third step is introduced to remove redundant basis vectors from the candidate set $\mathbb{D} \setminus \mathbb{D}_k$ in order to guarantee the regularity of matrix $\mathbf{A}_k$:

3. **redundant basis vector elimination**: In Step 1, remove from the candidate set $\mathbb{D} \setminus \mathbb{D}_k$ those atoms that may cause singularity of matrix $\mathbf{A}_k$.

In fact, step 3 not only guarantees the numeric stability but also keeps the approximation from over-fitting through suppression of large coefficient values. Step 1-3 are described in detail in Section 3.2, 3.4 respectively. In Section 3.5, we present two stopping criteria for the pursuit process. The whole adaptation procedure is summarized in Section 3.6.

3.2. Sequential maximum likelihood basis vector selection

In the first iteration ($k = 1$), only one basis vector which can best represent the unknown speaker is chosen from the dictionary. According to (8) the coordinate for the ith basis vector can be calculated as

$$x_i = (a(i,i))^{-1} b(i), \quad i = 1, 2, \cdots, K, \tag{9}$$

where K is the size of the dictionary.

Substituting (9) to (3), the maximum value of the auxiliary function using the chosen ith basis vector can be calculated as

$$Q^1(x_i) = \frac{1}{2}(a(i,i))^{-1} b(i)^2 + Const. \tag{10}$$

According to the maximum likelihood criterion, the first basis vector $\mathbf{e}_{p_1}$ should be chosen as

$$p_1 = \arg\max_i Q^1(x_i). \tag{11}$$

After iteration k, suppose k optimal basis vectors have been obtained as $\mathbb{D}_k = \{\mathbf{e}_{p_1}, \mathbf{e}_{p_2}, \cdots, \mathbf{e}_{p_k}\}$. In the next iteration we choose $\mathbf{e}_{p_{(k+1)}}$ such that the auxiliary function (3) is maximized. Fixing the speaker coordinate vector $\mathbf{x}_k$, we add each basis vector $\mathbf{e}_l \in \mathbb{D} \setminus \mathbb{D}_k$ one by one and estimate the corresponding new coefficient x_l. The new auxiliary function can be written as

$$\begin{aligned} Q^{k+1}(x_l) = -\frac{1}{2}\sum_t\sum_m \gamma_m(t)[\mathbf{o}'(t) - \mathbf{E}_k(m)\mathbf{x}_k \\ -x_l\mathbf{e}_l(m)]^T \boldsymbol{\Sigma}_m^{-1}[\mathbf{o}'(t) - \mathbf{E}_k(m)\mathbf{x}_k - x_l\mathbf{e}_l(m)] \end{aligned} \tag{12}$$

Setting the derivative of (12) with respect to x_l to zero yields

$$x_l = (a(l,l))^{-1}\left(b(l) - \sum_{i=1}^{k} x_i a(l,i)\right) \tag{13}$$

Substituting (13) to (3), the new maximum of the auxiliary function can be calculated as

$$\begin{aligned} Q^{k+1}(x_l) &= Q^k(\mathbf{x}_k) \\ &+ \frac{1}{2}(a(l,l))^{-1}\left(b(l) - \sum_{i=1}^{k} x_i a(l,i)\right)^2, \end{aligned} \tag{14}$$

where $Q^k(\mathbf{x}_k)$ is maximum value of the auxiliary function in subspace $\boldsymbol{\Gamma}^k = span\{\mathbf{e}_{p_1}, \mathbf{e}_{p_2}, \cdots, \mathbf{e}_{p_k}\}$.

So the increment of the auxiliary function by adding $\mathbf{e}_l$ to $\mathbb{D}_k$ is

$$\Delta Q^{k+1}(\mathbf{e}_l) = \frac{1}{2}(a(l,l))^{-1}\left(b(l) - \sum_{i=1}^{k} x_i a(l,i)\right)^2. \tag{15}$$

Then we can choose the best basis vector $\mathbf{e}_{p_{(k+1)}}$ for iteration $k+1$ such that

$$p_{(k+1)} = \arg\max_l \Delta Q^{k+1}(\mathbf{e}_l). \tag{16}$$

3.3. Speaker coefficients update

In the above sequential basis vector selection process, after iteration $k+1$ the speaker coordinate $\begin{bmatrix} \mathbf{x}_k \\ x_{p_{(k+1)}} \end{bmatrix}$ is not the best projection of the speaker supervector in subspace $\boldsymbol{\Gamma}^{k+1}$. If we exchange the $(k+1)$th and the $(p_{(k+1)})$th rows and columns of matrix $\mathbf{A}_K$ as well as the corresponding rows of vector $\mathbf{b}_K$, the speaker coordinate $\mathbf{x}_{k+1}$ has to be re-estimated as $\mathbf{x}_{k+1} = \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}$. Here the key computation involves the inversion of matrix $\mathbf{A}_{k+1}$. In this section, an efficient updating formula of $\mathbf{A}_{k+1}^{-1}$ from previous $\mathbf{A}_k^{-1}$ is derived, resulting in an efficient speaker coordinate updating formula.

Define

$$\mathbf{v}_l^k = [\,a(1,l) \quad a(2,l) \quad \cdots \quad a(k,l)\,]^T, \quad k < l \leqslant K, \tag{17}$$

then $\mathbf{A}_{k+1}$ could be written as

$$\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{v}_{k+1}^k \\ (\mathbf{v}_{k+1}^k)^T & a(k+1,k+1) \end{bmatrix}. \tag{18}$$

Using the block matrix inversion formula (Petersen and Pedersen, 2008), it can be derived that

$$\mathbf{A}_{k+1}^{-1} = \begin{bmatrix} \mathbf{A}_k^{-1} + \beta_{k+1}^k \mathbf{c}_{k+1}^k (\mathbf{c}_{k+1}^k)^T & -\beta_{k+1}^k \mathbf{c}_{k+1}^k \\ -\beta_{k+1}^k (\mathbf{c}_{k+1}^k)^T & \beta_{k+1}^k \end{bmatrix} \tag{19}$$

where

$$\beta_{k+1}^k = 1/\left(a(k+1,k+1) - (\mathbf{v}_{k+1}^k)^T \mathbf{c}_{k+1}^k\right) \tag{20}$$

and

$$\mathbf{c}_{k+1}^k = \mathbf{A}_k^{-1} \mathbf{v}_{k+1}^k. \tag{21}$$

Based on $\mathbf{A}_k^{-1}$ of the previous iteration, $\mathbf{A}_{k+1}^{-1}$ can be easily calculated using (19)–(21). Then the speaker coordinate vector can be updated by $\mathbf{x}_{k+1} = \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}$.

3.4. Redundant basis vector elimination

In the above steps, there is a potential problem when the speaker dictionary is redundant. In the $(k+1)$th iteration, if the selected basis vectors $\{\mathbf{e}_{p_i}\}_{i=1}^{k+1}$ are correlated, the inversion of matrix $\mathbf{A}_{k+1}$ is unstable. So those vectors which correlate with vectors of $\mathbb{D}_k$ should be removed from $\mathbb{D} \setminus \mathbb{D}_k$ during iteration $k+1$.

If we exchange every component of $\mathbf{e}_k$ with that of $\mathbf{e}_{p_{(k)}}$ after each iteration k, we can write $\mathbb{D}_k = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k\}$. Note that we have exchanged the $(k+1)$th and the $(p_{(k+1)})$th rows and columns of matrix $\mathbf{A}_K$ as well as the corresponding rows of vector $\mathbf{b}_K$, so the definitions of $\mathbf{A}_K, \mathbf{b}_K$ and $\{\mathbf{e}_k\}_{k=1}^K$ are consistent. Suppose $\mathbf{e}_l \in \mathbb{D} \setminus \mathbb{D}_k$ is correlated with $\{\mathbf{e}_i\}_{i=1}^k$, there exist some nonzero coefficients $\alpha_i \in \mathbb{R}, i = 1, 2, \cdots, k$ such that

$$\mathbf{e}_l = \sum_{i=1}^k \alpha_i \mathbf{e}_i. \tag{22}$$

From the definitions of $a(i,j)$ (Eq. (6)), we have

$$\begin{aligned} a(j,l) &= \sum_m s_m^{(0)} \mathbf{e}_{j,m}^T \boldsymbol{\Sigma}_m^{-1} \mathbf{e}_{l,m} = \sum_m s_m^{(0)} \mathbf{e}_{j,m}^T \boldsymbol{\Sigma}_m^{-1} \left[\sum_{i=1}^k \alpha_i \mathbf{e}_{i,m}\right] \\ &= \sum_{i=1}^k \alpha_i \left[\sum_m s_m^{(0)} \mathbf{e}_{j,m}^T \boldsymbol{\Sigma}_m^{-1} \mathbf{e}_{i,m}\right] = \sum_{i=1}^k \alpha_i a(j,i). \end{aligned} \tag{23}$$

Substituting Eq. (23) to the definition of $\mathbf{v}_l^k$ (Eq. (17)) yields

$$\begin{aligned} \mathbf{v}_l^k &= [\,a(1,l) \quad a(2,l) \quad \cdots \quad a(k,l)\,]^T \\ &= \left[\,\sum_{i=1}^k \alpha_i a(1,i) \quad \sum_{i=1}^k \alpha_i a(2,i) \quad \cdots \quad \sum_{i=1}^k \alpha_i a(k,i)\,\right]^T \\ &= \mathbf{A}_k \boldsymbol{\alpha}, \end{aligned} \tag{24}$$

where $\boldsymbol{\alpha} = [\,\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_k\,]^T$. Then it can be concluded that $\boldsymbol{\alpha} = \mathbf{A}_k^{-1}\mathbf{v}_l^k = \mathbf{c}_l^k$ (see definition of (21)). Similarly, it can be easily derived that

$$a(l,l) = \sum_{i=1}^k \alpha_i a(l,i) = (\mathbf{v}_l^k)^T \boldsymbol{\alpha} = (\mathbf{v}_l^k)^T \mathbf{c}_l^k. \tag{25}$$

If $\mathbf{e}_l$ is correlated with the selected basis vectors $\{\mathbf{e}_i\}_{i=1}^k$, the left hand side and right hand side of (25) must be equal. So $a(l,l) - (\mathbf{v}_l^k)^T \mathbf{c}_l^k$ can be used as a measure of the correlation between $\mathbf{e}_l$ and $\{\mathbf{e}_i\}_{i=1}^k$. A small value indicates close correlations, in which case $\mathbf{e}_l$ should be removed from the candidate set of the $(k+1)$th iteration.

Here an interesting fact can be observed that the difference between $a(l,l)$ and $(\mathbf{v}_l^k)^T \mathbf{c}_l^k$ is the reciprocal of β_l^k (see definition of (20)). According to Eq. (19), β_l^k is proportional to the last row and column of $\mathbf{A}_{k+1}^{-1}$ if we select $\mathbf{e}_l$ in the $(k+1)$th iteration. If the difference between $a(l,l)$ and $(\mathbf{v}_l^k)^T \mathbf{c}_l^k$ is close to zero, β_l^k will be very large, which will cause some components of new speaker coordinate vector $\mathbf{x}_{k+1}$ $(= \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1})$ being too large. So avoiding the difference being too small will prevent the resulting approximation of the speaker model from overfitting.

In our algorithm, we use $\mathbb{C}_k$ to record the these redundant set of basis vectors for iteration k, then the candidate set of basis vectors should be $\mathbb{D} \setminus (\mathbb{D}_k \bigcup \mathbb{C}_k)$.

3.5. Stopping criteria

We propose two criteria for the iterative process to stop. Firstly, we can limit the dimension of the speaker subspace according to some heuristic criteria. The following simple formula is used to determine the maximal dimension:

$$N = \min(\eta\gamma, S) \tag{26}$$

where γ is the total frame count of the adaptation speech, S is the count of training speakers and η is a constant set by hand, which determines at most how many basis vectors are added for each new frame of data. The pursuit process will stop if $k > N$. Eq. (26) is equivalent to setting the maximum number of basis vectors proportional to the adaptation data available. A similar formula is adopted by Povey and Yao (2012) for a basis method of constrained MLLR. The basic justification for this formula is that a constant number of basis vectors cannot deal with varying amount of adaptation data. The more adaptation data we have, the more free parameters we can robustly estimate, so more basis vectors should be added.

Secondly, we can set a threshold for the minimal increment of the objective function Q as new basis vector added. In iteration $k+1$, if the following condition is met, the whole iteration will be stopped:

$$\max_l \Delta Q^{k+1}(\mathbf{e}_l) < \delta, \tag{27}$$

where $\Delta Q^{k+1}(\mathbf{e}_l)$ (Eq. (15)) is an approximation of the increment of Q by selecting $\mathbf{e}_l$ at iteration $k+1$, and $\delta > 0$ is a predetermined threshold for the minimal increment of the objective function. This criterion is equivalent to adding an l_0 regularization term to the objective function, that is

$$Q'(\mathbf{x}_k) = Q(\mathbf{x}_k) - \delta\|\mathbf{x}_k\|_0. \tag{28}$$

Using l_0 regularization, we penalize each adding basis vector by a constant value δ. In iteration $k+1$, when the best basis vector is added, the l_0 norm of the speaker coordinate vector $\mathbf{x}$ will be increased by 1. If the increment of the original objective function Q is less than δ, the regularized objective function $Q\prime$ will be decreased. Then the pursuit process should be stopped and the final dimension of the speaker subspace should be k. In Eq. (27), we use $\max_l \Delta Q^{k+1}(\mathbf{e}_l)$ to approximate the maximum increment of objective function Q by adding a new basis vector. The l_0 regularization looks very similar to acoustic model selection methods based on Bayesian information criterion (BIC) (Chen and Gopinath, 1999) or minimum description length (MDL) (Shinoda and Watanabe, 2000; Cho and Kim, 2010). However, in contrast to BIC and MDL, the penalty term in l_0 regularization is data-dependent, resulting in an adaptive model selection method, which is more robust.

In our implementation, the above two criteria are simply combined to get a robust estimation of the speaker subspace dimension. Inspired by the well-known parameter selection property of l_0 regularization (Bruckstein et al., 2009), we change the threshold δ in a predefined range ($10 \sim 30$) to find its optimal value, while setting η to a relatively large value (e.g. 0.2) to put a loose constraint on the maximum number of basis vectors. When either of the two stopping conditions is reached the iteration process stops.

3.6. Procedure of the matching pursuit based speaker adaptation algorithm

The whole procedure of the proposed matching pursuit based speaker adaptation can be efficiently implemented as Algorithm 1.

Algorithm 1: Speaker adaptation based on matching pursuit.

```
1: Let
   $k = 0, \mathbb{D} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_K\}, \mathbb{D}_0 = \{\}, \mathbb{C}_0 = \mathbb{C}_1 = \{\},$
   $\mathbf{x}_0 = 0, \mathbf{E}_{0,m} = [], m = 1, 2, \cdots, M.$
   $\triangleright$ Initialization
2: Choose $\eta > 0, \delta > 0$ and $\tau > 0$.
3: Given $\mathbf{O}$, accumulate $s_m^{(0)}, \mathbf{s}_m^{(1)}$ for each mixture
   component $m$, compute $\mathbf{A}_K$ and $\mathbf{b}_K$.
4: repeat
5:     for $\mathbf{e}_l \in \mathbb{D} \setminus (\mathbb{D}_k \bigcup \mathbb{C}_k)$ do
6:         if $k == 0$ then
7:             $\Delta Q^1(\mathbf{e}_l) \leftarrow \frac{1}{2}(a(l,l))^{-1}(b(l))^2$
8:         else
9:             $\mathbf{v}_l^k \leftarrow [\, a(1,l) \quad a(2,l) \quad \cdots \quad a(k,l) \,]^T$
10:            $\mathbf{c}_l^k \leftarrow \mathbf{A}_k^{-1}\mathbf{v}_l^k$
11:            $(\beta_l^k)^{-1} \leftarrow a(l,l) - (\mathbf{v}_l^k)^T\mathbf{c}_k^k$
12:            if$|(\beta_l^k)^{-1}| < \tau$ then
13:                $\mathbb{C}_{k+1} \leftarrow \mathbb{C}_{k+1} \bigcup \{\mathbf{e}_l\}$    $\triangleright$ Record the
   redundant basis vectors in $\mathbb{C}_{k+1}$
14:                $\Delta Q^{k+1}(\mathbf{e}_l) \leftarrow -\infty$
15:            else
16:
   $\Delta Q^{k+1}(\mathbf{e}_l) \leftarrow \frac{1}{2}(a(l,l))^{-1}\left(b(l) - \sum_{i=1}^{k} x_i a(l,i)\right)^2$
17:            end if
18:        end if
19:    end for
20:    $\Delta\widehat{Q}^{k+1} \leftarrow \max_l \Delta Q^{k+1}(\mathbf{e}_l)$
21:    if$\Delta\widehat{Q}^{k+1} > \delta$ then
22:        $p_{k+1} \leftarrow \arg\max_l \Delta Q^{k+1}(\mathbf{e}_l)$    $\triangleright$ Choose the
   $(k+1)$th basis vector
23:        if$k == 0$ then
24:            $A_1^{-1} \leftarrow a(p_1, p_1)^{-1}$
25:        else
26:            $\mathbf{c} \leftarrow \mathbf{c}_{p_{(k+1)}}^k, \beta \leftarrow 1/(\beta_{p_{(k+1)}}^k)^{-1}$
27:            $\mathbf{A}_{k+1}^{-1} \leftarrow \begin{bmatrix} \mathbf{A}_k^{-1} + \beta\mathbf{c}\mathbf{c}^T & -\beta\mathbf{c} \\ -\beta\mathbf{c}^T & \beta \end{bmatrix}$
28:        end if
29:        $\mathbf{x}_{k+1} \leftarrow \mathbf{A}_{k+1}^{-1}\mathbf{b}_{p_{(k+1)}}$    $\triangleright$Update the speaker
   coordinate $\mathbf{x}_{k+1}$
30:        Exchange the $(k+1)$th and $p_{(k+1)}$th rows and
   columns of $\mathbf{A}_K$.
31:        Exchange $b_{k+1} \leftrightarrow b_{p_{(k+1)}}$.
32:        $\mathbf{E}_{k+1,m} \leftarrow [\, \mathbf{E}_{k,m} \quad \mathbf{e}_{p_{(k+1)},m} \,], m = 1, 2, \cdots, M.$
33:    end if
34:    $k \leftarrow k + 1$
35: until $k > \min(\eta\gamma, S)$ or $\Delta\widehat{Q}^{k+1} < \delta$
36: return $\boldsymbol{\mu}_m^{(s)} \leftarrow \boldsymbol{\mu}_m + \mathbf{E}_{k,m}\mathbf{x}_k, \quad m = 1, 2, \cdots, M.$
```

In Algorithm 1, Step 4 to Step 35 are the matching pursuit iterations. In each iteration k, $\mathbb{D}_k$ and $\mathbb{C}_k$ records the selected and removed redundant basis vectors respectively. During the iterations, $\mathbf{E}_k(m)$ (Step 32) keeps the matrix of the selected basis vectors for mixture m and $\mathbf{A}_k^{-1}$ holds the corresponding inverse of matrix $\mathbf{A}_k$. In Step 35, the two stopping criteria of Section 3.5 are checked. The resulting SD mean vectors are returned in Step 36.

4. Speaker adaptation based on l_1 regularization

The above matching pursuit algorithm is very fast, but it is greedy and sub-optimal. In compressive sensing, direct optimization of the objective function using an explicit l_1 norm regularization term can always obtain better solution. The l_1 norm regularization is sometimes referred to as the Lasso (least absolute shrinkage and selection operator) (Tibshirani, 1996), which can perform an element-wise shrinkage of the parameter vector towards zero, thus leading to a kind of variable selection. In fact, the l_1 norm is a convex approximation of the l_0 norm, which equals to the count of non-zero components of the parameter vector. Thus l_1 norm penalty has the effect of reducing the free parameters of the system, yielding a more robust estimation. Recently, there has also been a renewed interest in regularization approaches to address the problem of robust model and feature estimation from limited acoustic data. For instance, in Lu et al. (2011), l_1 and l_2 regularization are combined to improve estimation of the state-specific parameters in the subspace Gaussian mixture model (SGMM). In Wiesler et al. (2011), sparse features for log-linear acoustic models are selected through l_1 regularization , which can reduce training and recognition time. In Yu et al. (2012), l_1 regularization is used to reduce the non-zero connections of the deep neural network (DNN) without sacrificing speech recognition performance.

For speaker-space-based adaptation, l_1 regularization can be used to obtain sparse representation of the unknown SD models. The regularized estimation of $\mathbf{x}_K$ can be written as:

$$\hat{\mathbf{x}}_K = \arg\max_{\mathbf{x}_K} -\frac{1}{2}\mathbf{x}_K^T\mathbf{A}_K\mathbf{x} + \mathbf{b}_K^T\mathbf{x}_K - \lambda\|\mathbf{x}_K\|_1, \tag{29}$$

where $\|\mathbf{x}_K\|_1 = \sum_k |x_k|$ and $\lambda > 0$ is the weighting factor of the l_1 norm. From a Bayesian perspective, this is equivalent to placing a zero-mean Laplace prior distribution with a scale parameter of $\lambda^{-1}\mathbf{I}$ on $\mathbf{x}_K$, in which case we can interpret (29) as a MAP estimate.

Eq. (29) is a convex unconstrained optimization problem. A lot of convex optimization algorithms have been adapted to solve this problem, including interior-point methods (Candès et al., 2006), projected gradient methods (Figueiredo et al., 2007), and iterative thresholding (Daubechies et al., 2004). None of these methods give a closed form solution to (29), instead some optimum value is obtained after some iterations. A good initial value is important for these algorithms to converge fast. Fortunately, the low complexity and efficient matching pursuit algorithm presented in Section 3 can provide such a good starting point.

In our experiments, we run a predefined k (e.g. $k = 50$) iterations of the proposed matching pursuit algorithm. Suppose the indices of the selected basis vectors are $\{p_1, p_2, \cdots, p_k\}$ and the corresponding speaker coordinate vector is $\mathbf{x}_k^{MP}$, then initial value of problem (29) can be obtained by $\mathbf{x}_K^0 = [x_1^0, x_2^0, \cdots, x_K^0]^T$, where

$$x_k^0 = \begin{cases} x_{p_l}^{MP} & \text{if } k == p_l \text{ for some } 1 \leqslant l \leqslant k \\ 0 & \text{else.} \end{cases} \tag{30}$$

For the optimization algorithm, we choose the projected gradient algorithm of (Figueiredo et al., 2007).

5. Experimental evaluation

This section presents an experimental study to evaluate the performance of the proposed compressive sensing based methods described in Section 3 and 4 on an Mandarin Chinese continuous speech recognition task and the Wall Street Journal (WSJ) 20k open vocabulary continuous speech recognition task. For the Mandarin Chinese continuous speech recognition task, supervised speaker adaptation using different amount of adaptation data was evaluated. For the large-vocabulary WSJ task, single-utterance-based unsupervised speaker adaptation scenario was investigated. For both corpora, we compare the proposed compressive sensing based methods with various conventional methods. Detailed experimental setups and results are presented below for each task.

5.1. Experiments on the Mandarin Chinese task

5.1.1. Experimental setup

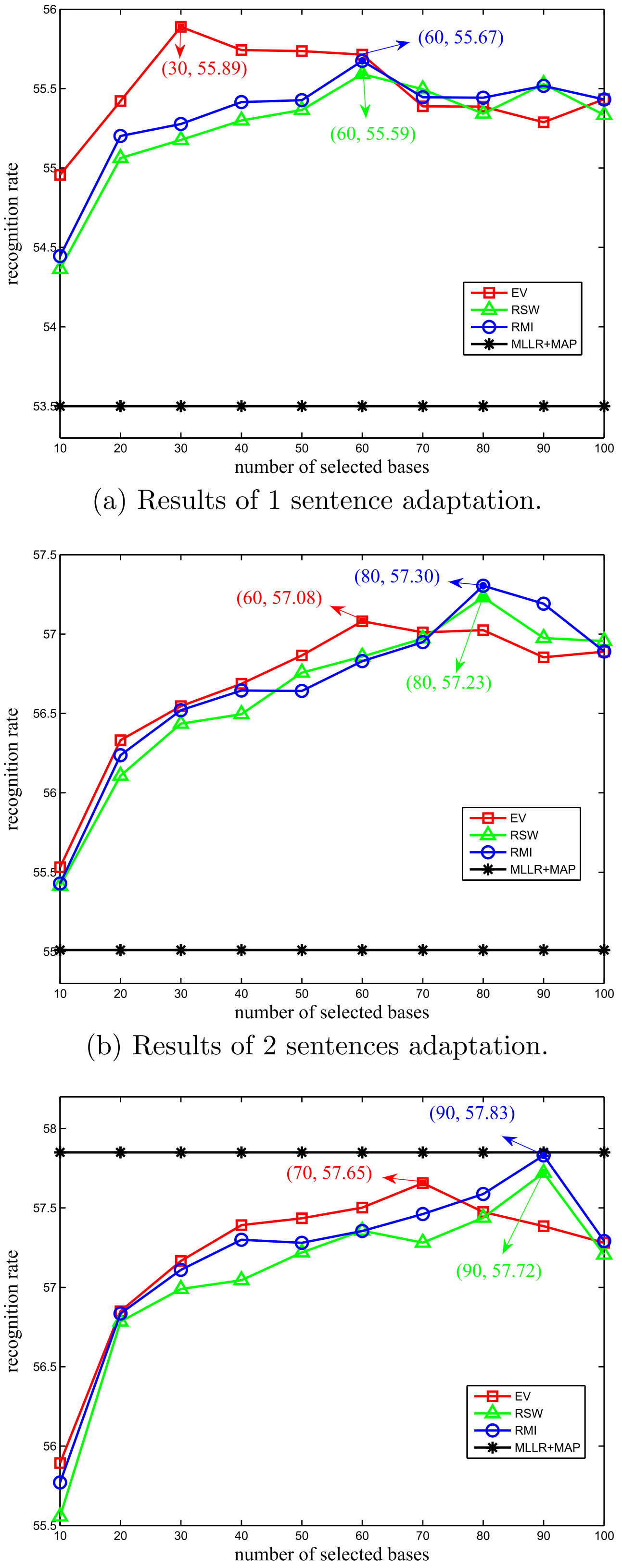
Supervised speaker adaptation were performed on the Microsoft speech database (Chang et al., 2001) with a Mandarin Chinese continuous speech recognition task. Utterances from 100 male speakers were used for training data, and those from the other 25 male speakers were used for evaluation. Each training speaker contributed approximately 200 sentences for training for a total of 19,688 sentences and 454,315 syllables (about 33 hours total). Each test speaker had 20 sentences available for testing (each testing sentence lasts for about 5 seconds). All experiments were based on the standard HTK (v 3.4.1) (Young et al., 2009) tool set. The frame length and frame step size were set as 25ms and 10ms, respectively. Acoustic features were constructed from 13 dimensional Mel-frequency cepstral coefficients and their first and second derivatives. To train the acoustic models, we use the syllable based approach. The basic units for acoustic modeling are 27 initial and 157 tonal final units of Mandarin Chinese as described in Chang et al. (2001). Each tonal syllable is consisted of zero or one initial and several tonal final units. Monophone models were first created using all 19,688 sentences. Each monophone is corresponding to one of the initial or tonal final units. Then all possible cross-syllable triphone expansions based on the full syllable dictionary were performed, resulting in a total of 295,180 triphones. Out of these triphones, 95,534 triphones actually occur in the training corpus. Each triphone was modeled by a 3-state left-to-right HMM without skips. After decision tree based state clustering, the number of unique tied states was reduced to 2,392. We then use the HTK toolkits Gaussian splitting capability to incrementally increase the number of Gaussians per state to 8 Gaussians per mixture, resulting in a total of 19,136 different Gaussian components in the SI model.

During testing, HVite was used as the decoder with a full connected tonal syllable recognition network. All 1,679 tonal syllables are listed in the network and any syllable can be followed by any other syllable, or they may be separated by short pause or silence. This recognition task puts the highest demand on the quality of the acoustic models. We drew 1, 2, 4 sentences randomly from each testing speaker for adaptation in supervised mode and tonal syllable recognition rate was measured among the remaining 16 sentences. To ensure statistical robustness of the results, each experiment was repeated 8 times using cross-validation and the recognition rates were averaged. The recognition accuracy of the SI model was 53.04% (the baseline reference result reported in Chang et al. (2001) is 51.21%).

5.1.2. Results on conventional speaker adaptation methods

For the purpose of comparison, we carried out experiments using conventional maximum likelihood linear regression followed by maximum a posterior adaptation (denoted by MLLR + MAP) (Digalakis and Neumeyer, 1996), eigenvoice and RSW-based methods. For MLLR + MAP adaptation, we experimented with different parameter settings. For MAP, the prior weighting factor of the SI model was varied between 10 and 40. For MLLR, three types of transformation matrix (a diagonal matrix, a 3-block-diagonal matrix and a full matrix) with different number of regression classes (16, 32 and 64) are evaluated. The best results were obtained at a prior weighting factor of 10 (for MAP) and 32 regression classes with a 3-block-diagonal transformation matrix (for MLLR). For eigenvoice adaptation, the dimension K of the speaker subspace was varied from 10 to 100. For the RSW-based methods, all training speaker (100) models are used for candidate reference models. Two criteria for selection of the reference models were tested with different count of selected models ranging from 10 to 100. Experimental results for 1, 2 and 4 sentences adaptation are presented in Fig. 1a-c respectively. In these figures, “RSW” denotes the maximum-likelihood reference speaker selection method (Mak et al., 2006) and “RMI” is the reference model interpolation method (Teng et al., 2007). For MLLR+MAP, only the best results are shown. For the other three speaker-space-based methods, the change of the recognition rates with different subspace dimensions (K) are plotted and the best results are marked for comparison.

From Fig. 1, it can be observed that when the adaptation data is limited to 1 or 2 sentences, the speaker-space-based methods (i.e. the EV, RSW and RMI methods) outperform the MLLR+MAP method. But when the adaptation data is increased to 4 sentences, the MLLR+MAP method can obtain best result. Among the three speaker-space-based methods, EV outperforms the two RSW-based methods when the adaptation data is 1 sentence and the two RSW-based methods perform better when more adaptation data is provided. To achieve its best results, the two RSW-based methods need more basis vectors than that of the EV method. This can be attribute to the non-orthogonality of the basis vectors used in the RSW-based methods, whereas in the EV method the basis vectors are orthogonal. The good results of the RSW-based methods show that the orthogonal property of basis vectors is not so important as one might imagine. Under all testing conditions, the reference model interpolation (RMI) method performs slightly better than the maximum-likelihood reference speaker selection (RSW) method. The reason may be that the for-



(a) Results of 1 sentence adaptation.

(b) Results of 2 sentences adaptation.

(c) Results of 4 sentences adaptation.

Fig. 1. Average tonal syllable recognition rate (%) after supervised speaker adaptation using four conventional methods.

Table 1
Average tonal syllable recognition rate (%) after supervised speaker adaptation (the average count of selected basis vectors are shown in parentheses, best results of each testing condition are marked in bold font).

Methods		Number of adaptation sentences		
		1	2	4
MLLR+MAP		53.50 (-)	55.01 (-)	57.85 (-)
Eigenvoice		55.89 (30)	57.08 (60)	57.65 (70)
RSW		55.59 (60)	57.23 (80)	57.72 (90)
RMI		55.67 (60)	57.30 (80)	57.83 (90)
MP	$\delta = 10$	55.51 (75.4)	57.12 (87.9)	57.21 (96.8)
	$\delta = 20$	56.28 (52.6)	57.21 (76.5)	57.64 (92.4)
	$\delta = 30$	**56.68** (36.5)	**57.42** (65.4)	**57.90** (83.6)
	$\delta = 40$	56.32 (33.8)	57.34 (58.6)	57.80 (79.5)

mer selects the reference speakers according a criterion that closely related to the maximization of objective function (3), whereas in the latter method the linear combination of maximum-likelihood reference speaker models could not necessarily generates good approximation of the unknown SD model.

5.1.3. Results on matching pursuit based speaker adaptation

We tested the proposed matching pursuit (MP) algorithm (Algorithm 1) using a speaker dictionary consisting of all eigenvoices and training speaker supervectors. To guarantee the numeric stability, all supervectors were normalized to have an l_2 norm of 100. The parameter δ of the second stopping criterion (see Section 3.5) was varied between 10 and 40 and the other two parameters (η and τ) were always fixed to 0.2 and 0.1 respectively in all experiments. Ideally all these parameters should be obtained on development data, independently from the test set. But we did not have separate development data. The basic idea is that if the simply chosen parameters yield better performance than other tuned methods, the new method should give even more improvement with well-tuned parameters. For each parameter setting, we calculated the average count ($\bar{k}$) of the selected basis vectors over all testing speakers. Experimental results for 1, 2 and 4 sentences adaptation are shown in Table 1, where the best results of the 4 conventional methods are also shown for comparison.

From Table 1, it can be observed that with larger δ, less basis vectors are selected for approximation of the unknown SD model. Best results are attained at $\delta = 30$ and the performance is better than the best results of the three conventional speaker-space-based methods under all testing conditions. It is worth noting that the average numbers of selected basis vectors are between that of the eigenvoice and that of the two RSW-based methods, which means that the matching pursuit algorithm selected a proper set of mixed eigenvoice and reference speaker models for better linear combinations of the unknown SD models. The advantage of the matching pursuit algorithm is more significant when less adaptation data is provided. For 1 sentence adaptation, the performance is improved by relatively 1.4% compared with the eigenvoice method. For 4 sentence adaptation, the average recognition rate is comparable to that of the MLLR + MAP method.

Because all the differences displayed in Table 1 are small, statistical significance tests were performed using the suite of significance tests implemented by NIST [1]. Three significance tests were applied, including the matched pair (MP) sentence segment (word error) test, the signed paired (SI) comparison test (speaker word accuracy rate), and the Wilcoxon (WI) signed rank test (speaker word accuracy rate). Pair-wise significance tests show that the differences between the matching pursuit method with $\delta = 30$ and the 4 conventional methods shown in Table 1 were all statistically significant according to all of the above tests at a 5% level of significance.

5.1.4. Results on l_1 regularized speaker adaptation

To obtain better linear approximations of the unknown speakers, the proposed l_1 regularized speaker adaptation method (denoted by "l_1") of Section 4 was tested using different regularization parameters. We used the same speaker dictionary as that of Section 5.1.3. The l_1 weighting factor λ was varied between 5 and 30. Before the projected gradient algorithm (Figueiredo et al., 2007) begins, we ran 50 iterations of the matching pursuit algorithm to obtain an initial value of the speaker coordinate vector using Eq. (30). For each parameter setting, we calculated the average count ($\bar{k}$) of the nonzero components of the speaker coordinate vectors ($\mathbf{x}_K$) over all testing speakers. These nonzero components correspond to the selected basis vectors through l_1 regularization. Experimental results for 1, 2 and 4 sentences adaptation are shown in Table 2, where the best results of the 4 conventional methods and the matching pursuit algorithm are also shown for comparison.

From Table 2, it can be observed that the weighting factor λ has a great impact on the count of the nonzero components of the speaker coordinate. When λ becomes larger, less basis vectors are selected through these nonzero components. Best results are obtained when $\lambda = 20$. For this parameter setting, compared with the matching pursuit algorithm, better performance is obtained under all testing conditions and for each testing condition the average count

[1] ftp://jaguar.ncsl.nist.gov/pub/sctk-2.4.0-20091110-0958.tar.bz2

Table 2
Average tonal syllable recognition rate (%) after supervised speaker adaptation (the average count of selected basis vectors are shown in parentheses, best results of each testing condition are marked in bold font).

Methods		Number of adaptation sentences		
		1	2	4
MLLR+MAP		53.50 (-)	55.01 (-)	57.85 (-)
Eigenvoice		55.89 (30)	57.08 (60)	57.65 (70)
RSW		55.59 (60)	57.23 (80)	57.72 (90)
RMI		55.67 (60)	57.30 (80)	57.83 (90)
MP ($\delta = 30$)		56.68 (36.5)	57.42 (65.4)	57.90 (83.6)
l_1	$\lambda = 5$	55.37 (73.2)	57.28 (75.6)	57.36 (86.2)
	$\lambda = 10$	56.60 (48.6)	57.40 (66.7)	57.58 (78.3)
	$\lambda = 20$	**57.22** (38.2)	**57.62** (63.8)	**57.96** (75.2)
	$\lambda = 30$	56.94 (36.8)	57.36 (59.6)	57.52 (68.9)

of selected basis vectors is comparable, which implies that the l_1 regularization method can obtain better solution for speaker adaptation using redundant speaker dictionary. The performance is improved significantly when the adaptation is limited to 1 sentence, where a relatively 2.4% improvement is achieved compared to the conventional eigenvoice method. For 4 sentences adaptation, the average recognition rate is slightly better than that of the MLLR + MAP method.

Again statistical significance tests were performed using the suite of significance tests implemented by NIST. The differences between the l_1 method with $\lambda = 20$ and all other methods shown in Table 2 were found to be statistically significant at a 5% level of significance according to the "MP", "SI" and "WI" tests.

In above experiments, matching pursuit was used to generate the initial speaker coordinate vectors for l_1 regularization. In fact, problem (29) is convex. For a convex optimization problem if a solution is local optimal it is also global optimal (Boyd and Vandenberghe, 2004). So theoretically the projected gradient algorithm (Figueiredo et al., 2007) is guaranteed to get the global optimal value of $\mathbf{x}_K$ no matter which value is used to seed the l_1 regularization. But different initial values require different numbers of iterations to find the optimal solution. This can be verified by the following experiments. We use the best solutions ($\mathbf{x}_K$) of the EV, RSW and RMI algorithms to seed the projected gradient algorithm for 1 sentence adaptation respectively. All experimental parameters are the same (we set λ to 20). Experimental results show that all methods find almost the same value of $\mathbf{x}_K$ for each speaker, resulting the same recognition rate. We calculate the average counts of iterations for the projected gradient algorithm among all testing speakers for each initialization method. The average counts corresponding to EV, RSW, RMI and MP are 1423, 2457, 2238 and 439 respectively. So using the initial value adapted from the matching pursuit algorithm yields the fastest convergence speed. The matching pursuit algorithm do supply a good start point for the l_1 regularization method.

One drawback of the l_1 regularization method is that its computation requirement seems to be much more than that of EV or RSW. The new method requires several iterations of matching pursuit and then gradient-based numerical iterations for solving the lasso problem. But computers are getting faster and faster, this should not be a problem for rapid speaker adaptation with a few adaptation data. In our experiments, we did not observe significant slowness for the l_1 regularization method compared with the conventional EV and RSW methods in all testing conditions. Besides, in occasions with high requirements on speed or limited computing resources, we could go back to the matching pursuit algorithm, which is much faster than the l_1 regularization method.

5.2. Experiments on the WSJ task

In the Mandarin Chinese continuous speech recognition task of the previous section, the HMM acoustic model typically consists of approximately 19,000 Gaussians. This section evaluates the behavior of unsupervised speaker adaptation using the new algorithms on the large-vocabulary WSJ task. A 20K word vocabulary WSJ task was evaluated with an acoustic model consisting of over 50,000 Gaussians. A single-utterance-based unsupervised adaptation scenario was investigated. This involved a two-pass decoding strategy for each utterance. A hypothesized transcription was obtained using the SI model during the first pass. Speaker adaptation was performed using the hypothesized transcriptions. The final result was obtained in a second decoding pass using the adapted model.

The baseline SI system was configured as follows. The standard SI-284 WSJ training set was used for training the SI model. It consists of 7,138 WSJ0 utterances from 83 WSJ0 speakers and 30,275 WSJ1 utterances from 200 WSJ1 speakers. So there is a total of about 70 hours of read speech in 37,413 training utterances from 283 speakers. The same acoustic feature settings as that of the Mandarin Chinese task were used. There were 22,699 cross-word triphones based on 39 base phonemes and these were tree-clustered to 3,339 tied states. At most 16 component mixture distributions were estimated for each of the tied states, resulting in a total of 53,424 Gaussians.

The standard Nov'92 20K open non-verbalized test set (333 sentences from 8 speakers) were used for evaluation using the standard 20K-vocabulary trigram language

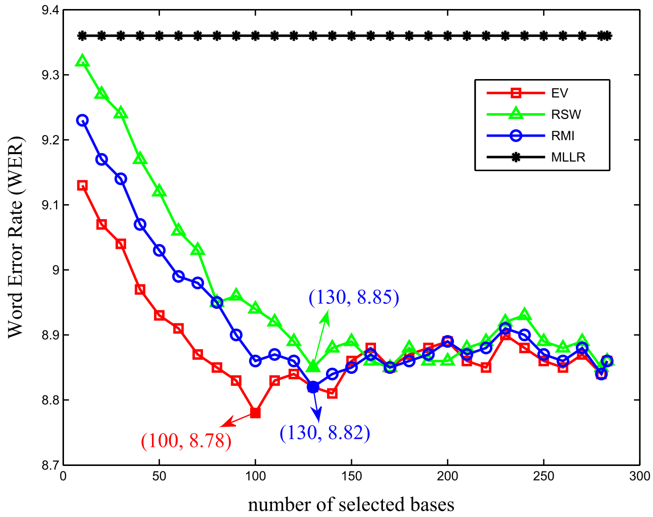


Fig. 2. WER (%) after single-utterance-based unsupervised adaptation using conventional methods.

Table 3
WER (%) after single-utterance-based unsupervised speaker adaptation for the WSJ Nov’92 test set.

Methods	WER	Average Count of Selected Basis Vectors
MLLR	9.36	–
Eigenvoice	8.78	100
RSW	8.85	130
RMI	8.82	130
MP	8.72	106.4
l_1	8.65	93.6

Table 4
Statistical significance of differences in measured WER for the new methods and 4 testing conventional methods.

Methods	MP	l_1
MLLR	MP: MP, SI: MP, WI: MP	MP: l_1, SI: l_1, WI: l_1
Eigenvoice	MP: MP, SI: MP, WI: MP	MP: l_1, SI: l_1, WI: l_1
RSW	MP: MP, SI: MP, WI: MP	MP: l_1, SI: l_1, WI: l_1
RMI	MP: MP, SI: MP, WI: MP	MP: l_1, SI: l_1, WI: l_1
MP	–	MP: l_1, SI: same, WI: same

model that came along with the WSJ corpus. We use word error rate (WER) as the metric for evaluation of the recognition results. The baseline system results in a 9.42% WER, which is comparable to the 9.46% WER reported in Woodland et al. (1994) using a similar configuration.

For the purpose of comparison, we carried out experiments using conventional eigenvoice (EV) method and the two RSW-based methods with varying amount of basis vectors. Experiments using the MLLR + MAP method showed almost no improvement over the SI model due to the limited adaptation data available in the single-utterance-based unsupervised adaptation scenario. Instead, we applied MLLR with one global 3-block-diagonal transformation matrix to get an idea of the difficulty of the task. Experimental results are shown in Fig. 2. For the three speaker-space-based methods, the change of the WERs with different subspace dimensions (K) are plotted and the best results are marked for comparison.

From Fig. 2, it can be observed that the speaker-space-based methods (i.e. the EV, RSW and RMI methods) outperform the MLLR method. Among all testing methods, best result is obtained using the EV method with 100 eigenvoices. The corresponding WER is 8.78%, which is 6.8% relative improvement over that of the SI system. For the two RSW-based methods, best results are obtained with 130 reference speakers. The RMI methods (with an 8.82% WER) performs slightly better than maximum-likelihood reference speaker selection method (denoted by “RSW” in Fig. (2), with an 8.85% WER).

We tested the proposed matching pursuit (MP) method and the l_1 regularized optimization method using a speaker dictionary consisting of all eigenvoices and training speaker supervectors (the size of the dictionary is 566). To guarantee the numeric stability, all supervectors were normalized to have an l_2 norm of 100. To evaluate the performance of the proposed methods, we used the WSJ1 Hub 1 development test data (denoted by “si_dt_20” in the WSJ1 corpus) as the development set to tune the system parameters. The WER of the SI system was 13.82% on the development set. For the matching pursuit algorithm, the parameter δ of the second stopping criterion (see Section 3.5) was varied between 10 and 40 and the other two parameters (η and τ) were fixed to 0.2 and 0.1 respectively. For the l_1 regularized optimization method, the l_1 weighting factor λ was varied between 10 and 100. The lowest WER for the matching pursuit algorithm was 13.42% with $\delta = 20$ and the lowest WER for the l_1 regularization algorithm was 13.24% with $\lambda = 60$. Then single-utterance-based unsupervised speaker adaptation on the evaluation set was performed using these parameter settings. Recognition results are summarized in Table 3. The best results for the four conventional methods are also presented for comparison.

From Table 3, it can be observed that both matching pursuit (MP) and l_1 regularized optimization (l_1) methods can yield better results than all the conventional methods. For the matching pursuit method, the WER is 8.72% with an average count of selected basis vectors of 106.4, which is between that of the eigenvoice and the two RSW-based methods. For the l_1 regularization method, the WER is 8.65%, which is 8.2% relative improvement over that of the SI system. The average count of selected basis vectors is 93.6, which is even smaller than that of the eigenvoice method, showing good basis vector selection ability of the l_1 regularization method.

Because the all the differences in measured WER for the systems displayed in Table 3 are very small, again statistical significance tests were performed. Table 4 displays the results of applying the “MP”, “SI” and “WI” tests. Each entry in the table displays results of a pair-wise significance test of two systems. If two systems are statistically different at a 5% level of significance for a given test, the system with the lower WER will appear in the table entry for that test. If the systems are not judged to be significantly different by a given test, the entry will contain "same" for that test. Table 4 shows that the WER differences between the matching pursuit method and the 4 conventional methods were all statistically significant according to all of the above tests at a 5% level of significance. Further more, the WER difference between the l_1 regularized optimization method and the matching pursuit method was found to be statistically significant in the matched pair (MP) sentence segment (word error) test, but not statistically significant in the other two tests. This implies that statistically speaking, the l_1 regularized optimization method shows little advantage over the matching pursuit method in single-utterance-based unsupervised speaker adaptation scenario.

6. Conclusion

We presented a new framework of speaker-space-based speaker adaptation method using compressive sensing theory. All the eigenvoice and training speaker supervectors are combined to construct a redundant speaker dictionary, from which an optimal subset of basis vectors are selected to recover the model parameters for a particular unknown speaker. Two optimization algorithms from compressive sensing are adapted for the selection of optimal basis vectors. The matching pursuit algorithm is greedy and very fast, and can provide a good starting point for the l_1 regularized optimization algorithm which can obtain better approximation of the unknown speaker model. Both algorithms outperforms the conventional speaker-space-based methods under all testing conditions, especially when the adaptation data is limited to 1 sentence (about 5 seconds). Although the mixed dictionary seems to combine the advantages of eigenvoice and reference speaker weighting methods, direct optimal construction of the speaker dictionary from the training data remains an unsolved problem. We will look at this direction for our further work.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 61175017 and No. 61005019) and the National High-Tech Research and Development Plan of China (No. 2012AA011603).

References

Boominathan, V., Murty, K.S.R., 2012. Speaker recognition via sparse representations using orthogonal matching pursuit. In: Proc. of ICASSP. pp. 4381–4384.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge Univ. Press.

Bruckstein, A.M., Donoho, D.L., Elad, M., 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. 51 (1), 34–81.

Candès, E., Romberg, J., Tao, T., 2006. Robust uncertainty principles: exact signal reconstruction from highly incomplete Fourier information. IEEE Trans. Inf. Theory 52 (2), 489–509.

Chang, E., Shi, Y., Zhou, J., et al., 2001. Speech lab in a box : a Mandarin speech toolbox to jumpstart speech related research. In: Proc. of Eurospeech. pp. 2799–2802.

Chen, S.S., Gopinath, R.A., 1999. Model selection in acoustic modeling. In: Proc. of Eurospeech. pp. 1087–1090.

Cho, H.-Y., Kim, S., 2010. A new distance measure for a variable-sized acoustic model based on MDL technique. ETRI J. 32 (5), 795–800.

Daubechies, I., Defrise, M., Mol, C.D., 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. 57 (11), 1413–1457.

Digalakis, V.V., Neumeyer, L.G., 1996. Speaker adaptation using combined transformation and Bayesian methods. IEEE Trans. Speech Audio Process. 4 (4), 294–300.

Donoho, D., 2006. Compressed sensing. IEEE Trans. Inf. Theory 52 (4), 1289–1306.

Figueiredo, M.A.T., Nowak, R.D., Wright, S.J., 2007. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Proces. 1 (4), 586–597.

Gemmeke, J.F., Virtanen, T., Hurmalainen, A., 2011. Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Trans. Acoust. Speech Signal Process. 19 (7), 2067–2080.

Hahm, S., Ohkawa, Y., Ito, M., Suzuki, M., Ito, A., Makino, S., 2010. Aspect-model-based reference speaker weighting. In: Proc. of ICASSP. pp. 4302–4305.

Hazen, T.J., Glass, J.R., 1997. A comparison of novel techniques for instantaneous speaker adaptation. In: Proceedings of the European Conference on Speech Communication and Technology. pp. 2047–2050.

Huo, Q., Lee, C.-H., 1997. On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate. IEEE Trans. Speech Audio Process. 5 (2), 161–172.

Kenny, P., Boulianne, G., Ouellet, P., et al., 2004. Speaker adaptation using an eigenphone basis. IEEE Trans. Speech Acoust. Process. 12 (6), 579–589.

Kua, J.M.K., Ambikairajah, E., Epps, J., Togneri, R., 2011. Speaker verification using sparse representation classification. In. Proc. of ICASSP, pp. 4548–4551.

Kua, J.M.K., Epps, J., Ambikairajah, E., 2013. I-vector with sparse representation classification for speaker verification. Speech Commun. 55 (5), 707–720.

Kuhn, R., Junqua, J.-C., Nguyen, P., Niedzielski, N., 2000. Rapid speaker adaptation in eigenvoice space. IEEE Trans. Speech Audio Process. 8 (6), 695–707.

Li, J., Tsao, Y., Lee, C.-H., 2010. Shrinkage model adaptation in automatic speech recognition. In: Proc. InterSpeech. pp. 1656–1659.

Lu, L., Ghoshal, A., Renals, S., 2011. Regularized subspace Gaussian mixture models for speech recognition. IEEE Signal Process. Lett. 18 (7), 419–422.

Mak, B., Lai, T.-C., Hsiao, R., May. 2006. Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers. In: Proc. of ICASSP, vol. 1.

Mallat, S.G., Zhang, Z., 1993. Matching pursuits with time-frequency dictionaries. IEEE Trans. Signal Process. 41 (12), 3397–3415.

Naseem, I., Togneri, R., Bennamoun, M. 2010. Sparse representation for speaker identification. In. Proc. of ICPR, pp. 4460–4463.

Olsen, P.A., Huang, J., Rennie, S.J., Goel, V. 2011. Sparse maximum a posteriori adaptation. In. Proc. of ASRU, pp. 53–58.

Olsen, P.A., Huang, J., Rennie, S.J., Goel, V. 2012. Affine invariant sparse maximum a posteriori adaptation. In. Proc. of ICASSP, pp. 4317–4320.

Petersen, K.B., Pedersen, M.S., 2008. The matrix cookbook.

Povey, D., Yao, K., 2012. A basis representation of constrained MLLR transforms for Robust adaptation. Comput. Speech Lang. 26 (1), 35–51.

Shinoda, K., 2010. Acoustic model adaptation for speech recognition. IEICE Trans. Inf. Syst. 93 (9), 2348–2362.

Shinoda, K., Watanabe, T., 2000. MDL-based context-dependent subword modeling for speech recognition. Acoust. Sci. Technol. 21 (2), 79–86.

Teng, W.X., Gravier, G. Bimbot, F., Soufflet, F. 2007. Rapid speaker adaptation by reference model interpolation. In. Proc. InterSpeech, pp. 258–261.

Teng, W.X., Gravier, G., Bimbot, F., Soufflet, F. 2009. Speaker adaptation by variable reference model subspace and application to large vocabulary speech recognition. In. Proc. of ICASSP, pp. 4381–4384.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B (Stat. Method.) 58 (1), 267–288.

Tropp, J., Gilbert, A.C., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory 53 (12), 4655–4666.

Wiesler, S., Richard, A., Kubo, Y., Schlüter, R., Ney, H. 2011. Feature selection for log-linear acoustic models. In. Proc. of ICASSP, pp. 5324–5327.

Woodland, P., Odell, J., Valthev, V., Young, S. 1994. Large vocabulary continuous speech recognition using HTK. In. Proc. of ICASSP, pp. 125–128.

Young, S., Evermann, G., Gales, M., et al., 2009. The HTK Book (for HTK Version 3.4).

Yu, D., Seide, F., Li, G., Deng, L. 2012. Exploiting sparseness in deep neural networks for large vocabulary speech recognition. In. Proc. of ICASSP, pp. 4409–4412.