doi:10.11959/j.issn.1000-436x.2015241

基于稀疏组 LASSO 约束的本征音子说话人自适应

屈丹，张文林
（信息工程大学 信息系统工程学院，河南 郑州 450000）

摘 要：本征音子说话人自适应方法在自适应数据量不足时会出现严重的过拟合现象，提出了一种基于稀疏组 LASSO 约束的本征音子说话人自适应算法。首先给出隐马尔可夫—高斯混合模型下本征音子说话人自适应的基本原理；然后将稀疏组 LASSO 正则化引入到本征音子说话人自适应，通过调整权重因子控制模型的复杂度，并通过一种加速近点梯度的数学优化算法来实现；最后将稀疏组 LASSO 约束的自适应算法与当前多种正则化约束的自适应方法进行比较。汉语连续语音识别的说话人自适应实验表明，引入稀疏组 LASSO 约束后，本征音子说话人自适应方法的性能得到了明显提高，且稀疏组 LASSO 约束方法优于 l_1、l_2 和弹性网正则化方法。
关键词：说话人自适应；本征音子；组稀疏约束；稀疏组 LASSO 约束；近点梯度法
中图分类号：TN912.34 **文献标识码**：A

Sparse group LASSO constraint eigenphone speaker adaptation method for speech recognition

QU Dan, ZHANG Wen-lin
(Institute of Information System Engineering, PLA Information Engineering University, Zhengzhou 450000, China)

Abstract: Original eigenphone speaker adaptation method performed well when the amount of adaptation data was sufficient. However, it suffered from server overfitting when insufficient amount of adaptation data was provided. A sparse group LASSO(SGL) constraint eigenphone speaker adaptation method was proposed. Firstly, the principle of eigenphone speaker adaptation was introduced in case of hidden Markov model-Gaussian mixture model (HMM-GMM) based speech recognition system. Then, a sparse group LASSO was applied to estimation of the eigenphone matrix. The weight of the SGL norm was adjusted to control the complexity of the adaptation model. Finally, an accelerated proximal gradient method was adopted to solve the mathematic optimization. The method was compared with up-to-date norm algorithms. Experiments on an mandarin Chinese continuous speech recognition task show that, the performance of the SGL constraint eigenphone method can improve remarkably the performance of the system than original eigenphone method, and is also superior to l_1-norm, l_2-norm and elastic net constraint methods.
Key words: speaker adaptation; eigenphone; group sparse constraint; sparse group LASSO constraint; proximal gradient method

1 引言

连续语音识别系统中训练数据与测试数据的不匹配会造成系统性能的急剧下降。声学模型自适应技术就是根据少量的测试数据对声学模型进行调整，增加其与测试数据的匹配程度，从而提高系统的识别性能。造成训练与测试数据不匹配的因素包括说话人、传输信道或说话噪声环境等，相应的自适应技术分别称为“说话人自适应[1]”、“信道自适应[2]”或“环境自适应[3]”。说话人自适应技术的方法也可以应用于信道自适应或环境自适应。说话人自适应通常包括特征层自适应[4,5]和声学模型自适应，因此，声学模型的说话人自适应[1]是当前语音识别系统一个必不可少的重要组成部分。

收稿日期：2014-10-08；**修回日期**：2015-04-01
基金项目：国家自然科学基金资助项目（61175017, 61302107, 61403415）
Foundation Item: The National Natural Science Foundation of China (61175017, 61302107, 61403415)

声学模型的说话人自适应就是利用少量的未知说话人语料(自适应语料)，在最大似然或最大后验准则下，将说话人无关(SI, speaker independent)声学模型调整至说话人相关(SD, speaker- dependent)声学模型，使语音识别系统更具说话人针对性，从而提高系统的识别率。在隐马尔可夫模型的连续语音识别系统框架下，主流的说话人自适应技术可分为 3 大类[1]：基于最大后验概率、基于变换和基于说话人子空间的自适应方法，分别以最大后验（MAP, maximum a posteriori）自适应、最大似然线性回归（MLLR, maximum likelihood linear regression）及本征音(EV, eigenvoice)方法及其相应的拓展算法为代表。2004 年，Kenny 等[6]通过对 SD 声学模型中各高斯混元均值矢量相对于 SI 声学模型的变化量进行子空间分析，得到一种新的子空间分析方法。该方法与说话人子空间中的“本征音”类似，因此称该子空间的基矢量为“本征音子（EP, eigenphone）”，该空间为“音子变化子空间”，但该方法采用“多说话人”声学建模技术，只能得到训练集中说话人相关的声学模型，对于测试集中的未知说话人没有给出其声学模型的自适应方法。2011 年，文献[7]提出了一种基于本征音子的说话人自适应方法，克服了 Kenny 等方法的不足，能够对测试集未知说话人进行自适应。但该方法在自适应阶段需要估计一个高维的扩展本征音子矩阵，故其待估参数数量多于传统说话人自适应方法，因此在自适应数据量充足时，可以得到更好的自适应性能。然而，当自适应数据量不足时，即使采用说话人自适应训练（SAT, speaker adaptation training）等技术，仍会出现严重的过拟合现象。

正则化方法是目前很多领域的一种非常有效的提高模型参数稳健性的方法，在连续语音识别系统说话人自适应中也逐步应用。例如，文献[8]将 l_2 正则化方法应用于 MLLR 自适应方法的变换矩阵估计，得到一种正则化的 MLLR 说话人自适应方法，并在单句话的无监督说话人自适应中取得了良好的效果；文献[9,10]提出稀疏最大后验（SMAP, sparse maximum a posteriori）自适应方法，该方法可以在减少模型存储量的同时提高 MAP 自适应的效果，随后文献[11]又采用 l_1 正则化进行改进。文献[12]将 l_1 正则化、l_2 正则化和弹性网正则化方法应用于本征音说话人自适应，识别率得到进一步提升。

为此，本文提出了基于稀疏组 LASSO 约束的本征音子说话人自适应方法。新方法本质上是以本征音子作为字典项；在模型域寻求说话人相关模型参数的稳健性稀疏表达；对自适应问题的目标函数引入稀疏组 LASSO 正则项, 在自适应阶段通过优化过程自动选择说话人相关音子子空间基矢量及其组合系数。文中给出了一般正则化本征音自适应原理框架，并讨论了组稀疏正则化方法和稀疏组 LASSO 正则化，分别给出了其数学优化算法。

2 本征音子说话人自适应

2.1 音子变化子空间及本征音子

本文仅讨论基于隐马尔可夫模型的连续语音识别系统的说话人自适应。假设在 SI 声学模型中，共有 M 个高斯混元，特征矢量维数为 D，训练集合中共有 S 个说话人。令 n_m 和 $\boldsymbol{\mu}_m^{(s)}$ 分别为 SI 模型和第 s 个说话人 SD 模型中第 m 个高斯混元的均值矢量，定义音子变化矢量 $\boldsymbol{u}_m^{(s)}$ 为 $\boldsymbol{u}_m^{(s)}=\boldsymbol{\mu}_m^{(s)}-\boldsymbol{\mu}_m$。在本征音子说话人自适应中，对于第 s 个说话人，假设 $\{\boldsymbol{u}_m^{(s)}\}_{m=1}^{M}$ 位于一个说话人相关的 N $(N\ll M)$ 维子空间 $\boldsymbol{\Pi}^{(s)}$ 中，称 $\boldsymbol{\Pi}^{(s)}$ 为说话人相关的“音子变化子空间”。设 $\boldsymbol{\Pi}^{(s)}$ 的原点为 $\boldsymbol{v}_0^{(s)}$，基矢量为 $\{\boldsymbol{v}_n^{(s)}\}_{n=1}^{N}$，称 $\{\boldsymbol{v}_n^{(s)}\}_{n=1}^{N}$ 为第 s 个说话人的本征音子。令第 m 个高斯混元对应的坐标矢量为 $\boldsymbol{y}_m=[y_{m1}\ y_{m2}\cdots y_{mN}]^{\mathrm{T}}$，则 $\boldsymbol{u}_m^{(s)}$ 在音子变化子空间中可分解为

$$\boldsymbol{u}_m^{(s)}=\boldsymbol{v}_0^{(s)}+\sum_{n=1}^{N}y_{mn}\boldsymbol{v}_n^{(s)}=\boldsymbol{v}_0^{(s)}+\boldsymbol{V}^{(s)}\boldsymbol{y}_m=\tilde{\boldsymbol{V}}^{(s)}\tilde{\boldsymbol{y}}_m \quad (1)$$

其中，$\boldsymbol{V}^{(s)}=\left[\boldsymbol{v}_1^{(s)}\ \ \boldsymbol{v}_2^{(s)}\ \ \cdots\ \ \boldsymbol{v}_N^{(s)}\right]$ 和 $\tilde{\boldsymbol{V}}^{(s)}=\left[\boldsymbol{v}_0^{(s)}\ \ \boldsymbol{V}^{(s)}\right]$ 分别为第 s 个说话人的本征音子矩阵和扩展本征音子矩阵，其维数分别为 $D\times N$ 和 $D\times(N+1)$；$\boldsymbol{y}_m$ 和 $\tilde{\boldsymbol{y}}_m=\left[1\ \ \boldsymbol{y}_m^{\mathrm{T}}\right]^{\mathrm{T}}$ 为高斯混元坐标矢量和扩展高斯混元坐标矢量，其维数分别为 N 和 $N+1$。由于 $\boldsymbol{y}_m$ 和 $\tilde{\boldsymbol{y}}_m$ 是说话人无关的，可以通过对所有训练说话人的音子变化矢量 $\{\boldsymbol{u}_m^{(s)},s=1,2,\cdots,S,\ m=1,2,\cdots,M\}$ 进行主分量分析得到 $\boldsymbol{y}_m$ 的估计[7]。

2.2 本征音子的最大似然估计算法

假设自适应数据的特征矢量序列为 $\boldsymbol{O}=\{\boldsymbol{o}(t)\}_{t=1}^{\mathrm{T}}$，根据最大似然准则，采用期望最大化（EM, expec-

tation maximization）算法估计说话人相关本征音子矩阵 $\boldsymbol{V}^{(s)}$，其优化的目标函数为

$$Q(\boldsymbol{V}(s)) = -\frac{1}{2}\sum_{t}\sum_{m}\gamma_m(t)\left[\boldsymbol{o}(\mathrm{t}) - \boldsymbol{\mu}_m - \boldsymbol{u}_m^{(s)}\right]^{\mathrm{T}} \cdot \boldsymbol{\Sigma}_m^{-1}\left[\boldsymbol{o}(\mathrm{t}) - \boldsymbol{\mu}_m - \boldsymbol{u}_m^{(s)}\right] \tag{2}$$

其中，$\gamma_m(t)$ 表示第 t 帧特征矢量属于 SI 模型中第 m 个高斯混元的后验概率，给定自适应数据的标注，则 $\gamma_m(t)$ 可以通过 Baum-Welch 前后向算法[13]计算得到；$\boldsymbol{\Sigma}_m$ 表示第 m 个高斯混元的协方差矩阵。将式(1)代入式(2)，并令其对 $\tilde{\boldsymbol{V}}^{(s)}$ 的导数为 0，可以得到 $\tilde{\boldsymbol{V}}^{(s)}$ 的求解公式[7]。然而文献[7]给出的求解公式中涉及 $(N+1)D\times(N+1)D$ 维矩阵的逆，对于一个典型的连续语音识别系统，当音子变化子空间 N 较大时（≥100）时，存储及求逆计算都非常消耗内存和计算时间。但传统 HMM-GMM 的声学模型中，$\boldsymbol{\Sigma}_m$ 是通常是一个对角阵，令其第 d 个对角线元素为 $\sigma_{m,d}$，则目标函数式(2)可以简化为

$$Q(\tilde{\boldsymbol{V}}^{(s)}) = -\frac{1}{2}\sum_{d}\sum_{t}\sum_{m}\gamma_m(t)\sigma_{m,d}^{-1}\left[o_d(t) - \mu_{m,d} - \tilde{\boldsymbol{v}}_d^{(s)\mathrm{T}}\tilde{\boldsymbol{y}}_m\right]^2 \tag{3}$$

其中，$o_d(t)$ 及 $\mu_{m,d}$ 分别为特征矢量 $\boldsymbol{o}(t)$ 及均值矢量 $\boldsymbol{\mu}_m$ 的第 d 维元素，$\tilde{\boldsymbol{v}}_d^{(s)\mathrm{T}}$ 表示本征音子矩阵 $\tilde{\boldsymbol{V}}^{(s)}$ 的第 d 行。对式(3)进行整理可得

$$Q(\tilde{\boldsymbol{V}}^{(s)}) = -\frac{1}{2}\sum_{d}\left[\tilde{\boldsymbol{v}}_d^{(s)\mathrm{T}}\boldsymbol{A}_d\tilde{\boldsymbol{v}}_d^{(s)} - \boldsymbol{b}_d^{\mathrm{T}}\tilde{\boldsymbol{v}}_d^{(s)}\right] + C \tag{4}$$

其中，$\boldsymbol{A}_d = \sum_{t}\sum_{m}\gamma_m(t)\sigma_{m,d}^{-1}\tilde{\boldsymbol{y}}_m\tilde{\boldsymbol{y}}_m^{\mathrm{T}}$，$\boldsymbol{b}_d = \sum_{t}\sum_{m}\gamma_m(t)\sigma_{m,d}^{-1}\left[o_d(t) - \mu_{m,d}\right]\tilde{\boldsymbol{y}}_m$，$C$ 为一个常数。对式(4)求关于 $\tilde{\boldsymbol{v}}_d^{(s)}$ 的导数，并令导数为 0 可得其最优值 $\left(\tilde{\boldsymbol{v}}_d^{(s)}\right)_{\mathrm{ML}} = \boldsymbol{A}_d^{-1}\boldsymbol{b}_d$。由于各行之间的计算相互独立，因此实际计算中，可以对 $\tilde{\boldsymbol{V}}^{(s)}$ 的 D 行进行并行求解，因此求解时间很快。

3　基于稀疏组 LASSO 约束的本征音子说话人自适应

本征音子说话人自适应方法在自适应阶段需要估计一个 $D\times(N+1)$ 维的扩展本征音子矩阵，其待估参数数量多于传统说话人自适应方法，因此在自适应数据量充足时，可以得到更好的自适应性能。然而，当自适应数据量不足时，即使采用说话人自适应训练等技术，仍会出现严重的过拟合现象。文献[14]分别通过引入先验分布和对本征音子矩阵引入低秩约束来解决这一问题，但提升的性能有限，因此可以考虑更好的约束方法来解决这一问题。

扩展本征音子矩阵的最大似然估计问题，引入正则化方法后，说话人自适应目标函数变为

$$Q'(\tilde{\boldsymbol{V}}) = Q(\tilde{\boldsymbol{V}}) - J_\lambda(\tilde{\boldsymbol{V}}) \tag{5}$$

其中，$Q(\tilde{\boldsymbol{V}})$ 为原始式(4)所示最大似然估计的目标函数，$J_\lambda(\tilde{\boldsymbol{V}})$ 是一个正则化函数，其参数为正则化权重矢量 λ（$\lambda>0$）。特别地，当 $J_\lambda(\tilde{\boldsymbol{V}}) = \lambda_1\|\tilde{\boldsymbol{V}}\|_1$ 时，为 l_1 正则化；当 $J_\lambda(\tilde{\boldsymbol{V}}) = \lambda_2\|\tilde{\boldsymbol{V}}\|_2^2$ 时，为 l_2 正则化；当 $J_\lambda(\tilde{\boldsymbol{V}}) = \lambda_1\|\tilde{\boldsymbol{V}}\|_1 + \lambda_2\|\tilde{\boldsymbol{V}}\|_2^2$ 时，则为弹性网正则化(ENR, elastic net regularization）。

3.1　组稀疏正则化方法

组稀疏（GS, group sparse）正则化方法也称为“组 LASSO(group LASSO)[15]”，其正则化函数 $J_\lambda(\tilde{\boldsymbol{V}})$ 为

$$J_\lambda(\tilde{\boldsymbol{V}}) = \lambda_3\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_2 \tag{6}$$

值得提出的是，式(6)与 l_2 正则化函数 $J_\lambda(\tilde{\boldsymbol{V}}) = \lambda_2\|\tilde{\boldsymbol{V}}\|_2^2$ 不同，这里 $\boldsymbol{v}_n$ 的 l_2 范数没有进行平方运算。可以证明，由于 l_2 范数在零点是不可导的，组稀疏正则化可以使属于同一组内的参数同时向零点靠拢[15,16]。式(6)将待估矩阵 $\tilde{\boldsymbol{V}}$ 的每一列 $\boldsymbol{v}_n$ 视为一组，相当于对矩阵 $\tilde{\boldsymbol{V}}$ 施加了一个列稀疏性（column sparsity）约束，使估计得到的矩阵 $\tilde{\boldsymbol{V}}$ 中某些列同时为 0。对目标函数式(4)引入组稀疏正则项，其优化目标函数为

$$\begin{aligned}\hat{\boldsymbol{V}}(s) &= \arg\max_{\tilde{\boldsymbol{V}}(\mathrm{s})}\left[Q'(\tilde{\boldsymbol{V}}(s))\right] \\ &= \arg\max_{\tilde{\boldsymbol{V}}(s)}\left[Q(\tilde{\boldsymbol{V}}(s)) - \lambda_3\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_2\right]\end{aligned} \tag{7}$$

其中，$\lambda_3>0$ 为组稀疏正则项权重，λ_3 越大所得到的矩阵 $\hat{\boldsymbol{V}}(s)$ 的平均列稀疏度越大。

3.2　稀疏组 LASSO 正则化方法

组稀疏正则化方法使估计结果中的非零组尽量少，然而却无法保证组内参数的稀疏性。对于扩展的本征音子矩阵估计问题，组稀疏正则化可以使估计得到的矩阵 $\tilde{\boldsymbol{V}}$ 的某些列同时为 0，然而不为 0 的那些列却往往不是稀疏的。事实上 l_1 正则化可以控制矩阵 $\tilde{\boldsymbol{V}}$ 列内参数的稀疏性，因此可以将 l_1 正则

化与组稀疏正则化相结合，得到更为稳健的估计，称为“稀疏组 LASSO（SGL, sparse-group LASSO）”正则化方法[17]，其正则化函数 $J_\lambda(\tilde{\boldsymbol{V}})$ 为

$$J_\lambda(\tilde{\boldsymbol{V}})=\lambda_1\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_1+\lambda_3\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_2 \tag{8}$$

这意味着首先通过组稀疏正则化方法选择不为零的那些参数组，然后通过 l_1 正则化方法选择组内的非零参数。对于扩展的本征音子矩阵估计问题，相当于对待估矩阵同时施加列间和列内稀疏性约束，从而得到结构化的稀疏解。

对目标函数式(4)引入式(8)稀疏组 LASSO 正则项，新的优化目标函数为

$$\begin{aligned}\hat{\boldsymbol{V}}(s)&=\arg\max_{\tilde{\boldsymbol{V}}(s)}\left[Q'\left(\tilde{\boldsymbol{V}}(s)\right)\right]\\&=\arg\max_{\tilde{\boldsymbol{V}}(s)}\left[Q\left(\tilde{\boldsymbol{V}}(s)\right)-\lambda_1\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_1-\lambda_3\sum_{n=0}^{N}\|\boldsymbol{v}_n\|_2\right]\end{aligned} \tag{9}$$

式(8)与弹性网正则化函数很相似，然而这里的 l_2 范数没有平方运算，可以证明在每一个不为 0 的组（本征音子 $\boldsymbol{v}_n$）内，稀疏组 LASSO 正则化方法相当于一种特殊的弹性网正则化方法[17]。

3.3 稀疏组 LASSO 约束的本征音子自适应优化算法

对于组稀疏正则化与稀疏组 LASSO 正则化问题，常用的解法有快速迭代收缩域值算法(FISTA, fast iterative shrinkage-thresholding algorithm)[19]、加速的广义梯度下降算法[17]等，文献[20]也给出了多种正则化函数适用的一种通用数学优化方法——递增近点梯度(IPG, incremental proximal gradient)算法。由于本文的优化问题包含一个可导的正则项（l_2 正则项）和多个不可导的正则项（l_1 正则项和组稀疏正则项），对于这种问题，递增近点梯度法是一种通用的、行之有效的迭代算法；而 FISTA 算法中的动量法及其选择的参数（$t^{(k)}$ 的更新公式）可以对迭代过程进行加速。为此本文在递增近点梯度算法中引入动量法（momentum method）[19]加速其收敛过程，得到一种“加速递增近点梯度（AIPG, accelerated incremental proximal gradient）算法”。

针对式(7)和式(9)的优化问题 $\arg\max_{\tilde{\boldsymbol{V}}(s)}\left[Q'\left(\tilde{\boldsymbol{V}}(s)\right)\right]$，为了便于优化，令 $\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k+1)}\right)=-Q'\left(\tilde{\boldsymbol{V}}(s)\right)$，则优化问题变为 $\arg\min_{\tilde{\boldsymbol{V}}(s)}\left[\tilde{Q}\left(\tilde{\boldsymbol{V}}(s)\right)\right]$，采用的加速递增近点梯度算法流程如算法 1 所示。

算法 1 加速递增近点梯度算法流程

① 初始化 $k=0$，$t^{(0)}=t^{(-1)}=1$，$\tilde{\boldsymbol{V}}^{(0)}=\tilde{\boldsymbol{V}}^{(-1)}=0$，$\eta^{(0)}=1.0$，计算 $\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(0)}\right)$；

② 设置 $\boldsymbol{Y}^{(k)}=\tilde{\boldsymbol{V}}^{(k)}+\frac{t^{(k-1)}-1}{t^{(k)}}\left(\tilde{\boldsymbol{V}}^{(k)}-\tilde{\boldsymbol{V}}^{(k-1)}\right)$；

③ 计算 $\tilde{\boldsymbol{V}}^{(k+1)}=\mathrm{prox}_{J_3}\{\mathrm{prox}_{J_2}\{\mathrm{prox}_{J_1}[\boldsymbol{Y}^{(k)}+\eta^{(k)}\nabla\tilde{Q}(\boldsymbol{Y}^{(k)})]\}\}$；

④ 若 $\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k+1)}\right)>\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k)}\right)$，则设置 $\eta^{(k)}\leftarrow 0.8\eta^{(k)}$，转至③；

⑤ 若 $\frac{\left|\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k+1)}\right)-\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k)}\right)\right|}{\tilde{Q}\left(\tilde{\boldsymbol{V}}^{(k)}\right)}<10^{-5}$，则停止迭代过程，输出估计结果 $\hat{\tilde{\boldsymbol{V}}}=\tilde{\boldsymbol{V}}^{(k+1)}$；

否则，设置 $t^{(k+1)}=\frac{1+\sqrt{1+4\left(t^{(k)}\right)^2}}{2}$，$\eta^{(k+1)}=\eta^{(k)}$，$k\leftarrow k+1$，转至②。

在算法 1 中，第②步采用动量法[14]来加快其迭代收敛过程；第③步为原始递增近点梯度算法的迭代公式，其中，$\mathrm{prox}_{J_1}(\bullet)$、$\mathrm{prox}_{J_2}(\bullet)$ 和 $\mathrm{prox}_{J_3}(\bullet)$ 分别为 l_1 正则函数、l_2 正则函数和组稀疏正则函数的近点映射算子[21]，$\eta^{(k)}$ 是第 k 步迭代的步长；为进一步加快收敛速度，本文对 $\eta^{(k)}$ 进行线性搜索，即在第④步当检测到迭代后的目标函数值变大时，按 0.8 的加权系数减小步长 $\eta^{(k)}$，重新回到第③步；最后，检查本次迭代前后 $\tilde{Q}$ 的相对减少量是否小于门限 10^{-5}，若是则停止迭代，否则回到步骤②重新进行迭代。

4 实验结果及分析

为了验证本文算法的性能，采用微软中文语料库[18]进行连续语音识别的说话人自适应实验。训练集包括 100 个男性说话人，每人约 200 句话，共有 19 688 句话，每句话时长大约 5 s，总时长为 33 h。测试集中共有 25 个说话人，每人 20 句话，每句话时长也约为 5 s。

声学特征矢量采用 13 维的 MFCC 参数及其一阶、二阶差分，总特征维数为 39 维。帧长和帧移分别为 25 ms 和 10 ms。实验中，借助语音开源工具箱 HTK（hidden Markov toolkit）（版本 3.4.1）[13]训练得到 SI 基线系统。首先训练单音子声学模型，其中每个单音子对应一个汉语有调音节。根据发音

字典，对单音子进行上下文扩展，得到 295 180 个跨词的三音子有调音节，其中 95 534 个三音子在训练语料中得到覆盖。每个三音子用一个包含 3 个发射状态的、自左向右无跨越的隐马尔可夫模型进行建模。采用基于决策树的三音子状态聚类后，系统中共有 2 392 个不同的上下文相关状态。最终训练得到的说话人无关（SI）声学模型中每个状态含有 8 个高斯混元，因此声学模型中的总高斯混元数为 19 136 个。

在测试阶段，采用音节全连接的解码网络，不采用任何语法模型。采用这种解码网络的语音识别系统对声学模型的要求最高，可以充分展示声学模型的识别性能。在原始测试集上，SI 基线系统的平均有调音节正确识别率为 53.04%（文献[18]中结果为 51.21%)。

为了便于比较本文算法的性能，本文针对下列说话人自适应算法进行对比实验。

1) EP_{New}：采用最大似然估计的本征音子自适应，且进行说话人自适应训练得到的方法，简称 EP_{New} 方法。首先采用主分量分析得到本征音子矩阵和高斯混合坐标矢量；其次利用训练数据重新 SAT 后的模型；然后采用最大似然准则估计本征音子矩阵，采用 l_1 约束的最大似然准则估计高斯混合坐标矢量；不断迭代得到最终的 SAT 模型和各高斯混合坐标矢量。由于该算法具有较好的性能，因此作为后续算法的基线系统。

2) EP_{New}-L_1：基于 l_1 约束的 EP_{New} 自适应算法，l_1 范数权重 λ_1 从 10 调整到 40。

3) EP_{New}- L_2：基于 l_2 约束的 EP_{New} 自适应算法，l_2 范数权重 λ_2 从 10 调整到 2 000。

4) EP_{New}-L_1-L_2：基于弹性网正则化约束的 EP_{New} 自适应算法，其中 λ_1 从 10 到 20，λ_2 从 10 调整到 100。

5) EP_{New}-GS：基于组稀疏正则化约束的 EP_{New} 自适应算法，组稀疏权重 λ_3 从 60 调整到 150。

6) EP_{New}- SGL:基于稀疏组 LASSO 约束的 EP_{New} 自适应算法，其中 λ_1 从 10 到 20，λ_2 从 10 调整到 40。

为了比较各种方法在不同自适应数据量下的自适应效果，对每个测试说话人分别随机抽取 1 句、2 句、4 句、6 句、8 句和 10 句话作为自适应数据，从剩下语句中随机抽取 10 句话作为测试数据，重复该过程 8 次，得到 8 组实验数据，将 8 组数据的平均结果作为系统性能指标。表 1、表 2 中黑体字所示为每种自适应数据量条件下的最好实验结果，斜体字所示为引入正则化约束后平均正确识别率下降的实验结果。

4.1　经典正则化约束的本征音子自适应实验

适当引入约束条件可以提升系统性能，为了便于比较本文算法的性能，以 EP_{New} 为基线系统，首先将 l_1 正则化、l_2 正则化和弹性网正则化 3 种经典正则化方法引入到基线系统中来。

表 1 给出了本征音子算法 EP_{New} 在 3 种经典正则化方法下的实验结果，括号内数字表示所有测试说话人扩展本征音子矩阵稀疏度的平均值 $\bar{\rho}$。

表 1 结果表明，引入 l_1 正则化方法之后，自适应性能得到提高，特别是在自适应数据量不足时（少于 4 句话时），性能的提升尤为明显，过拟合现象得到有效缓解。对于某一个固定的正则化因子 λ_1（对应表 1 中 EP_{new}-L_1 方法中的某一行），随着自适应数据量的增加，平均稀疏度逐渐减小，表明扩展本征音子矩阵中的非零元素数量逐渐增加，更多的自适应参数得到估计，因此 l_1 正则化方法具有良好的参数选择功能，它可以使自适应参数数量随着数据量的增加而不断增多。

在各种自适应数据量下，随着正则化因子 λ_1 的增大（对应 EP_{new}-L_1 算法中的某一列），扩展本征音子矩阵的平均稀疏度也不断增大，而平均正确识别率先增后减。当自适应数据量为 1、2、4、6 句话时，自适应方法在 $\lambda_1 = 20$ 时取得最好的效果，而当自适应数据量更为充足时（8 句话和 10 句话时），$\lambda_1 = 10$ 可以取得更好的结果。

引入 l_2 正则化后，当自适应数据量很少时（1 或 2 句话时），系统的性能有了明显提高，且 λ_2 越大性能提高越明显；而当自适应数据量较为充足时（多于 4 句话时），随着 λ_2 的增大，平均正识率先增后减，且 λ_2 越大，系统性能的下降越明显（如表 1 中斜体字所示部分）。因此随着自适应数据量的增加，应逐渐减小 λ_2 的值以放松约束，从而获得更好的自适应效果。

从表 1 中方法的对比结果来看，总体来讲，l_2 正则化的效果不如 l_1 正则化。相关研究表明两者具有一定的互补性，因此本文也对弹性网正则化方法进行测试，它是 l_1 和 l_2 2 种正则化方法的一种线性组合。实验中，将 l_1 正则化因子 λ_1 分别固定为 10 或 20，将 l_2 正则化因子 λ_2 从 10 调整至 100。在引

表 1　经典正则化自适应算法的实验结果（正识率）（%）（括号内数字表示平均稀疏度 $\bar{\rho}$ ）

自适应方法	参数设置		自适应数据量					
			1 句	2 句	4 句	6 句	8 句	10 句
EP_{new}			42.35	51.52	58.22	59.32	60.12	60.85
EP_{new}-L_1	$\lambda_1 = 10$		52.25	56.04	58.32	59.36	**60.32**	**61.32**
			(0.61)	(0.43)	(0.23)	(0.16)	(0.12)	(0.04)
	$\lambda_1 = 20$		**53.88**	**56.55**	**58.54**	**59.54**	60.24	61.12
			(0.83)	(0.63)	(0.42)	(0.33)	(0.26)	(0.23)
	$\lambda_1 = 30$		53.63	55.96	*57.70*	*59.31*	*60.05*	*60.92*
			(0.91)	(0.74)	(*0.54*)	(*0.44*)	(*0.37*)	(*0.34*)
	$\lambda_1 = 40$		53.82	55.18	*57.30*	*59.19*	*59.89*	*60.60*
			(0.95)	(0.82)	(*0.65*)	(*0.61*)	(*0.49*)	(*0.42*)
EP_{new} -L_2	$\lambda_2 = 10$		43.52	52.64	58.26	**59.42**	**60.22**	**60.93**
	$\lambda_2 = 100$		43.95	53.25	**58.42**	*59.21*	*60.05*	*60.82*
	$\lambda_2 = 1\,000$		46.32	53.92	58.35	*59.15*	*59.27*	*59.65*
	$\lambda_2 = 2\,000$		**48.65**	**54.26**	*58.21*	*58.65*	*58.83*	*59.32*
EP_{new}-L_1-L_2	$\lambda_1 = 10$	$\lambda_2 = 0$	52.25	56.04	58.32	59.36	60.32	**61.32**
		$\lambda_2 = 10$	52.50	56.12	**58.45**	**59.42**	**60.32**	*61.26*
		$\lambda_2 = 50$	**52.56**	**56.35**	*58.12*	*59.08*	*60.18*	*61.10*
		$\lambda_2 = 100$	*52.12*	*55.94*	*57.86*	*58.45*	*59.50*	*60.59*
	$\lambda_1 = 20$	$\lambda_2 = 0$	53.88	56.55	58.54	59.54	**60.24**	**61.12**
		$\lambda_2 = 10$	53.92	**56.60**	**58.62**	**59.65**	*60.21*	*61.10*
		$\lambda_2 = 50$	**53.96**	56.58	58.56	*59.12*	*59.95*	*60.86*
		$\lambda_2 = 100$	*53.40*	*56.34*	*57.51*	*58.42*	*59.32*	*60.60*

入 l_2 正则化方法后，与原始的 l_1 正则化方法相比（$\lambda_1 > 0, \lambda_2 = 0$ 时），弹性网正则化方法的平均正识率略有所提升。且随着自适应数据量的增加，l_2 正则化因子 λ_2 应逐渐减小；当 l_2 正则化因子取得过大时，平均正识率反而会下降。

4.2　稀疏组 LASSO 约束的本征音子自适应实验

本节针对组稀疏正则化和稀疏组 LASSO 正则化方法进行自适应实验。由上面分析可知，利用式(6)给出的组稀疏正则化函数，使估计得到的扩展本征音子矩阵 $\tilde{\boldsymbol{V}}$ 出现许多元素全为 0 的列。为了了解正则化因子 λ_3 对矩阵 $\tilde{\boldsymbol{V}}$ 的列稀疏性影响，定义“列稀疏度”θ 为矩阵 $\tilde{\boldsymbol{V}}$ 中全为 0 的列数占总列数的比例。实验中将组稀疏正则化因子 λ_3 从 60 调整到 150。更重要一点，本节将通过实验验证组稀疏正则化与 l_1 正则化的互补性，将两者进行线性组合，得到稀疏组 LASSO 正则化方法。实验中，将 l_1 正则化因子 λ_1 分别固定为 10 和 20，改变组稀疏正则化因子 λ_3 的值进行实验。

表 2 给出了不同自适应数据量下的典型实验结果，表中括号内单个数字为所有测试说话人扩展本征音子矩阵的平均列稀疏度 $\bar{\theta}$，以 2 个数字 $(\bar{\rho}, \bar{\theta})$ 的形式分别表示扩展本征音子矩阵的“平均稀疏度 $\bar{\rho}$ ”与“平均列稀疏度 $\bar{\theta}$ ”。

由表 2 可见，在自适应数据量较少时，引入组稀疏正则化后，系统识别性能得到显著提高；随自适应数据量的增大，应逐渐减少正则化因子 λ_3 以获得更好的自适应效果。在相同的自适应数据量下（列纵向比较），随着 λ_3 的增大，平均列稀疏度也逐渐增大，而平均正识率却先增后减。正则化因子对平均列稀疏度的影响在自适应数据量少时（如 1 句话时）更为明显，而当自适应数据量超过 4 句话时，平均列稀疏度始终接近于 0，这是由于正则化函数 $J_3(\tilde{\boldsymbol{V}})$ 的近点映射算子[21]本质上是一个乘性收缩算子，因此迭代若干次后，会使矩阵某些列的元素值变小，却难以完全等于 0。对比表 2 和表 1 结果可见，组稀疏正则化方法优于 l_2 正则化方法，由 2

表 2　组稀疏和稀疏组正则化自适应算法的实验结果（正识率）（%）（括号内单个数字表示平均稀疏度 $\bar{\rho}$，2 个数字表示 $(\bar{\rho},\bar{\theta})$）

自适应方法	参数设置		自适应数据量					
			1 句	2 句	4 句	6 句	8 句	10 句
EP_{new}			42.35	51.52	58.22	59.32	60.12	60.85
$EP_{new}GS$	$\lambda_3=60$		51.56 (0.09)	53.10 (0.02)	56.52 (0.01)	**59.36** (0.01)	**60.22** (0.0)	**61.08** (0.0)
	$\lambda_3=90$		52.75 (0.38)	53.45 (0.06)	58.34 (0.02)	59.32 (0.01)	60.16 (0.0)	60.90 (0.0)
	$\lambda_3=120$		53.05 (0.62)	**54.86** (0.15)	**58.36** (0.02)	*59.18* (*0.02*)	*59.85* (*0.0*)	*60.35* (*0.0*)
	$\lambda_3=150$		**53.56** (0.78)	54.52 (0.26)	*57.96* (*0.06*)	*58.92* (*0.02*)	*59.56* (*0.0*)	*60.01* (*0.0*)
EP_{new}-SPL	$\lambda_1=10$	$\lambda_3=0$	52.25 (0.61, 0.0)	56.04 (0.43, 0.0)	58.32 (0.23, 0.0)	59.36 (0.16, 0.0)	60.32 (0.12, 0.0)	61.32 (0.04, 0.0)
		$\lambda_3=10$	53.78 (0.61,0.01)	56.65 (0.47, 0.0)	58.45 (0.32, 0.0)	59.42 (0.22, 0.0)	**60.40** **(0.13, 0.0)**	**61.35** **(0.04, 0.0)**
		$\lambda_3=20$	54.55 (0.62, 0.01)	56.72 (0.47, 0.01)	**58.62** (0.33, 0.01)	**59.55** (0.23, 0.0)	*60.22* (*0.13*, *0.0*)	61.25 (0.04, 0.0)
		$\lambda_3=30$	**54.76** (0.62, 0.01)	**56.78** (0.47, 0.01)	58.45 (0.33, 0.01)	59.34 (0.23, 0.01)	*60.18* (*0.13*, *0.0*)	61.25 (0.04, 0.0)
		$\lambda_3=40$	54.49 (0.62, 0.02)	56.12 (0.49, 0.02)	58.34 (0.34, 0.01)	*59.25* (*0.23*, *0.01*)	*60.01* (*0.13*, *0.01*)	*60.89* (*0.04*, *0.01*)
	$\lambda_1=20$	$\lambda_3=0$	53.88 (0.83, 0.0)	56.55 (0.63, 0.0)	58.54 (0.42, 0.0)	59.54 (0.33, 0.0)	60.24 (0.26, 0.0)	61.12 (0.23, 0.0)
		$\lambda_3=10$	54.42 (0.85, 0.01)	**56.82** (0.64, 0.01)	**58.65** (0.45, 0.01)	**59.58** (0.36, 0.0)	**60.32** (0.26, 0.0)	**61.13** (0.23, 0.0)
		$\lambda_3=20$	**54.75** (0.86, 0.01)	56.65 (0.64, 0.01)	*58.42* (*0.46*, *0.01*)	*59.52* (*0.36*, *0.0*)	*60.20* (*0.26*, *0.0*)	*60.92* (*0.23*, *0.0*)
		$\lambda_3=30$	54.21 (0.86, 0.02)	*56.42* (*0.65*, *0.01*)	*58.38* (*0.46*, *0.01*)	*59.32* (*0.36*, *0.0*)	*60.22* (*0.26*, *0.0*)	*60.89* (*0.23*, *0.0*)
		$\lambda_3=40$	53.95 （0.86,0.02）	*56.21* (*0.65*, *0.02*)	*58.38* (*0.46*, *0.01*)	*59.25* (*0.36*, *0.0*)	*60.12* (*0.26*, *0.0*)	*60.89* (*0.23*, *0.0*)

种方法的近点映射算子的比较可知，组稀疏正则化方法相当于一种自适应的 l_2 正则化方法[21]，本文实验结果也验证了组稀疏正则化方法这一优势。此外对比表 2 和表 1 的结果，总体而言，在各种自适应数据量下，组稀疏正则化方法仍不及 l_1 正则化方法。

由于组稀疏正则化与 l_1 正则化具有互补性，表 2 给出了稀疏组 LASSO 约束的结果。结果表明，在 l_1 正则化基础上引入组稀疏正则化后，自适应性能得到进一步提高，特别是当自适应数据量较少时（1 或 2 句话），性能的提高尤为明显。例如，当 $\lambda_1=10,\lambda_3=30$ 时，相比于 $\lambda_1=10$ 时的 l_1 正则化方法，在 1 句话和 2 句话下，正识率分别相对提高了 4.8%和 1.3%。在正则化因子 λ_1 固定的条件下，随着自适应数据量的增加，应减少正则化因子 λ_3 以获得更好的识别效果。

从“平均稀疏度 $\bar{\rho}$”与“平均列稀疏度 $\bar{\theta}$”上看，引入组稀疏正则化后，平均稀疏度相对于仅采用 l_1 正则化时的值几乎没有变化，而平均列稀疏度都基本接近于零，这说明最终估计得到的扩展本征音子矩阵并没有呈现出明显的列稀疏性。对比表 2 中的实验设置，可以看出由于组稀疏正则化因子 λ_3 相对较小，而其对应的近点映射算子为一种乘性收缩算子，因此只能使某些列的值相对缩小，却难以将其缩小到 0。

对比表 2 和表 1 中实验结果可见，稀疏组 LASSO 正则化方法明显优于弹性网正则化方法，其原因在于组稀疏正则化方法相当于一种自适应的 l_2 正则化方法，因此其与 l_1 正则化的线性组合（即稀疏组 LASSO 正则化方法）相当于一种自适应的弹性网正则化方法。

5　结束语

本文提出了一种基于稀疏组 LASSO 约束的本征音子说话人自适应方法。新方法对自适应问题的目标函数引入稀疏组 LASSO 正则项，相当于对待估本征音子矩阵同时施加列间稀疏性约束与列内稀疏性约束，得到结构化的模型稀疏解。通过该约

束可以对自适应模型的复杂度进行有效控制，在数据量少时得到低维音子变化子空间，在数据量充足时得到高维音子变化子空间。实验证明，新算法在各种自适应数据量下均优于经典的l_1正则化、l_2正则化和弹性网正则化方法。

参考文献：

[1] ZHANG W L, ZHANG W Q, LI B C, *et al*. Bayesian speaker adaptation based on a new hierarchical probabilistic model[J]. IEEE Transactions on Audio, Speech and Language Processing[J]. 2012, 20(7): 2002-2015.

[2] SOLOMONOFF A, CAMPBELL W M, BOARDMAN I. Advances in channel compensation[A]. for SVM speaker recognition. Proceedings of International Conference on Acoustics, Speech, and Signal Processing(ICASSP)[C]. Philadelphia, USA, 2005. 629-632.

[3] PAVAN KUMAR D S, PRASAD N V, JOSHI V, *et al*. Modified splice and its extension to non-stereo data for noise robust speech recognition[A]. Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)[C]. Olomouc, Czech Republic, 2013. 174-179.

[4] HAMIDI S G, RICHARD C R. Two-stage speaker adaptation in subspace gaussian mixture models[A]. Proceedings of International Conference on Acoustics, Speech and Signal Processing(ICASSP)[C]. Florence, Italy, 2014. 6374-6378.

[5] WANG Y Q, GALE M J F. Tandem system adaptation using multiple linear feature transforms[A]. Proceedings of International Conference on Acoustics, Speech and Signal Processing(ICASSP)[C]. Vancouver, Canada, 2013. 7932-7936.

[6] KENNY P, BOULIANNE G, OUELLETET P, *et al*. Speaker adaptation using an eigenphone basis[J]. IEEE Transaction on Audio, Speech and Language Processing, 2004, 12(6):579-589.

[7] ZHANG W L, ZHANG W Q, LI B C. Speaker adaptation based on speaker-dependent eigenphone estimation[A]. Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)[C]. Hawaii, USA, 2011. 48-52.

[8] LI J, TSAO Y, LEE, C H. Shrinkage model adaptation in automatic speech recognition[A]. Proceedings of Annual Conference on International Speech Communication Association(INTERSPEECH)[C]. Makuhari, Chiba, Japan, 2010. 1656-1659.

[9] OLSEN P A, HUANG J, RENNIE S J, *et al.* Sparse maximum a posteriori adaptation[A]. Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)[C]. Hawaii, USA, 2011. 53-58.

[10] OLSEN P A, HUANG J, RENNIE S J, *et al*. Affine invariant sparse maximum a posteriori adaptation[A]. Proceedings of International Conference on Audio, Speech and Signal Processing(ICASSP)[C]. Kyoto, Japan, 2012. 4317-4320.

[11] KIM Y G, KIM H. Constrained mle-based speaker adaptation with l_1 regularization[A]. Proceedings of International Conference on Audio, Speech and Signal Processing(ICASSP)[C]. Florence, Italy, 2014. 6419-6422.

[12] 张文林, 张连海, 牛铜, 等. 基于正则化的本征音说话人自适应方法[J]. 自动化学报, 2012, 38(12):1950-1957.
ZHANG W L, ZHANG L H, NIU T, *et al*. Regularization based eigenvoice speaker adaptation method[J]. ACTA Automatica Sinica, 2012, 38 (12):1950-1957.

[13] YOUNG S, EVERMANN G, GALES M, *et al*. The HTK book (for HTK version 3.4)[EB/OL]. http://htk.eng.cam.ac.uk/docs/docs.shtml. 2009.

[14] 张文林, 张连海, 陈琦, 等. 语音识别中基于低秩约束的本征音子说话人自适应方法[J]. 电子与信息学报, 2014, 36(4):981-987.
ZHANG W L, ZHANG L H, CHEN Q, *et al*. Low-rank constraint eigenphone speaker adaptation method for speech recognition[J]. Journal of Electronics & Information Technology, 2014, 36(4):981-987.

[15] YUAN M, LIN Y. Model selection and estimation in regression with grouped variables[J]. Journal of the Royal Statistical Society(Series B), 2007, 68(1): 49-67.

[16] TAN Q F, NARAYANAN S S. Novel variations of group sparse regularization techniques with applications to noise robust automatic speech recognition[J]. IEEE Transaction on Acoustic, Speech and Signal Processing, 2012, 20(4):1337-1346.

[17] SIMON N, FRIEDMAN J, HASTIE T, *et al*. A sparse-group LASSO[J]. Journal of Computational and Graphical Statistics, 2013, 22 (2):231-245.

[18] CHANG E, SHI Y, ZHOU J, *et al*. Speech lab in a box: a Mandarin speech toolbox to jumpstart speech related research[A]. Proceedings of 7th European Conference on Speech Communication and Technology(EUROSPEECH) [C]. Aalborg, Denmark, 2001. 2799-2802.

[19] BECK A, TEBOULLE M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM Journal on Imaging Sciences, 2009, 2(1):183-202.

[20] BERTSEKAS D P. Incremental proximal methods for large scale convex optimization[J]. Mathematical Programming, 2011, 129(2): 163-195.

[21] PARIKH N, BOYD S. Proximal Algorithms. Foundations and Trends in Optimization[M]. 2013.

作者简介：

屈丹（1974-），女，吉林长春人，博士，信息工程大学副教授、硕士生导师，主要研究方向为语音处理与识别、机器学习、自然语言处理。

张文林（1982-），男，湖北蕲春人，博士，信息工程大学讲师，主要研究方向为语音处理与识别、机器学习、自然语言处理。