Rapid Feature Space MLLR Speaker Adaptation with Bilinear Models

Shilei Zhang, Peder A. Olsen, Yong Qin
IBM China Research Lab

Outline

- **Introduction**
- **Standard FMLLR**
- **FMLLR Using Bilinear Models**
- **Experimental Results**
- **Conclusion**

Introduction

- **Mismatch between the training and testing conditions leads to loss of some performance based on well-trained models.**
- **In model-based adaptation, three basic categories of speaker adaptation methods**
  - Speaker-clustering based methods (including eigenspace-based methods)
  - Bayesian-based method, such as Maximum a Posteriori (MAP)
  - Transformation based method, Maximum-Likelihood Linear Regression (MLLR)
- **Some adaptation techniques provide limited improvements if any with small amounts of test data;**

- **For small amount data, the following adaptation give better performance;**
  - Speaker-clustering or eigenspace-based adaptation method
    - eigenvoice, eigen-MLLR,cluster weighting, or reference speaker weighting
    - Kernel-based adaptation, such as kernel eigenvoice (KEV); kernel eigenspace-based MLLR (KEMLLR) ; MPLKR
  - Imposing various constraint on MLLR: a) block-diagonal or diagonal MLLR; b) MAPLR/FMAPLR, c) DLLR
- **General speaking, the key point is introducing a prior knowledge analysis on the training speakers, and incorporating it to decoding process.**
  - eigen FMLLR
  - fMAPLR
  - **Bilinear models**

FMLLR

- **FMLLR has proved to be highly effective as a method for unsupervised adaptation to a new speaker or environment**
- **It requires only a single transform matrix and bias vector to be estimated**

$$\hat{O}(\tau) = AO(\tau) + b = W\xi(\tau)$$

- **The EM algorithm gives an auxiliary function that can be maximized with respect to W yield an increase in the likelihood:**

$$\theta(\Theta, \hat{\Theta}) = \beta \log(p_i^T w_i) - \frac{1}{2}\sum_{i=1}^{N}[w_i^T G^{(i)} w_i - 2 w_i^T k^{(i)}]$$

Bilinear Models

- **Data contains two components: style and content Want to represent them separately**

Symmetric Bilinear Model:

y : observed data

a : style vector

b : content vector

I, j : components of style and content

W : matrix of basis vectors

$$\mathbf{y} = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{ij} a_i b_j$$

$$= \mathbf{a}^{\mathrm{T}}\mathbf{W}\mathbf{b}$$

Asymmetric Bilinear Model:

A : matrix of style-specific basis vectors

$$= \mathbf{A}\mathbf{b}$$

Y : (SK) x C

A : (SK) x J

b : J x C

More flexible model

Easier to deal with

- **Model building: Fit asymmetric model (find A and B for known styles and contents) using SVD**

$(SD) \times C$ matrix:

$$\overline{\mathbf{M}} = \begin{bmatrix} \mathbf{m}^{11} & \cdots & \mathbf{m}^{1C} \\ \vdots & \ddots & \vdots \\ \mathbf{m}^{S1} & \cdots & \mathbf{m}^{SC} \end{bmatrix} = \mathbf{U}\dot{\mathbf{S}}\mathbf{V}^{T} = \mathbf{A}\mathbf{B}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{1} \\ \vdots \\ \mathbf{A}^{s} \\ \vdots \\ \mathbf{A}^{S} \end{bmatrix} \qquad \mathbf{B} = \left[\mathbf{b}^{1} \cdots \mathbf{b}^{C}\right]$$

- **Adaptation process: Find style matrix that best explains data for incomplete style**

Bilinear model building for fMLLR

- **The observation fMLLR matrix: 'style' is defined as speaker; 'content' is defined as the columns.**

$$W = \begin{bmatrix} b_1 & a_{11} & \cdots & a_{1N} \\ b_2 & a_{21} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ b_N & a_{N1} & \cdots & a_{NN} \end{bmatrix} \in R^{DxD+1}$$

- **The bilinear model for observation matrix was computed based on the stacked fMLLR transforms from the training speakers. The observation matrix is arranged as a (SD)x(D+1) matrix**

$$\bar{M}_A = \begin{bmatrix} W^1 - W^0 \\ \vdots \\ W^s - W^0 \\ \vdots \\ W^S - W^0 \end{bmatrix}, 1 \le s \le S \qquad W^0 = \sum_{i=1}^{S} W^i$$

- **Then matrix $\bar{M}_A$ can be decomposed and expressed for asymmetric bilinear model as** $\bar{M}_A = AB \quad A \in R^{(SD)xJ} \quad B \in R^{Jx(D+1)}$
  - $\bar{M}_A$ is decomposed as USV^T by SVD.
  - the style parameter A is defined as the first J columns of US and content basic vector B is defined as the first J rows of V^T

Adaptation process

- **We can get the adaptation process based on maximum likelihood criterion** $\hat{O}(\tau) = W\xi(\tau) = (W_0 + AB)\xi(\tau)$

- **Auxiliary function is**

$$\begin{cases} \theta(M,\hat{M}) = \beta \log(p_i w_i^T) - 1/2 \sum_{i=1}^{n} [(w_{0i} + A_i B) G^{(i)} (w_{0i} + A_i B)^T - 2(w_{0i} + A_i B) k^{(i)^T}] \\ G^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau)\xi(\tau)\xi(\tau)^T \\ k^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \mu_i^{(m)} \sum_{\tau=1}^{T} \gamma_m(\tau)\xi(\tau)^T \end{cases}$$

- **Then**

$$\theta(M,\hat{M}) = \beta \log(p_i w_i^T) - 1/2 \sum_{i=1}^{n} [(w_{0i} + A_i B) G^{(i)} (w_{0i} + A_i B)^T - 2(w_{0i} + A_i B) k^{(i)^T}]$$

$$= \beta \log(p_i w_i^T) - 1/2 \sum_{i=1}^{n} [w_{0i} G^{(i)} w_{0i}^T + (A_i B) G^{(i)} (A_i B)^T + 2 w_{0i} G^{(i)} (A_i B)^T - 2(w_{0i} + A_i B) k^{(i)^T}]$$

- **Ignoring all terms independent of** A_i

$$\theta(M,\hat{M}) = \beta \log(p_i w_i^T) - 1/2 \sum_{i=1}^{n} [A_i \hat{G}^{(i)} A_i^T - 2 A_i \hat{k}^{(i)T}]$$

- **Where**

$$\begin{cases} \hat{G}^{(i)} = (B G^{(i)} B^T) \\ \hat{k}^{(i)T} = k^{(i)} B^T - w_{0i} G^{(i)} B^T \end{cases}$$

- **Differentiating with respect to** A_i **yields**

$$\frac{\partial \theta(M,\hat{M})}{\partial A_i} = \beta \frac{p_i B^T}{p_i w_i^T} - A_i \hat{G}^{(i)} + \hat{k}^{(i)}$$

- **The optimization is on a row by row basis, assuming that the above equation is equating to zero for row** i **, then**

$$\beta \frac{p_i B^T}{p_i w_i^T} = A_i \hat{G}^{(i)} - \hat{k}^{(i)}$$

$$p_i w_i^T \hat{k}^{(i)} \hat{G}^{(i)-1} + \beta p_i B^T \hat{G}^{(i)-1} = p_i w_i^T A_i \qquad (1)$$

- **Then** $$A_i = \hat{k}^{(i)} \hat{G}^{(i)^{-1}} + \frac{\beta}{p_i w_i^T} p_i B^T \hat{G}^{(i)^{-1}}$$
$$= \alpha(p_i B^T + \lambda \hat{k}^{(i)}) \hat{G}^{(i)^{-1}}$$
$$= (\alpha p_i B^T + \hat{k}^{(i)}) \hat{G}^{(i)^{-1}}$$

- **To find α, λ, substituting this expression for A_i in equation 1),**

$$\beta - \alpha^2 p_i B^T \hat{G}^{(i)^{-1}} (BP_i^T + \lambda \hat{k}^{(i)^T}) - \alpha p_i w_{0i}{}^T = 0$$

$$\alpha^2 p_i B^T G^{(i)^{-1}} (p_i B^T)^T + \alpha(p_i B^T \hat{G}^{(i)^{-1}} \hat{k}^{(i)^T} + p_i w_{0i}{}^T) - \beta = 0$$

- **There will be two possible solutions in α. The value will be selected that maximizes auxiliary function.**

- **selection J**
  - The amount of adaptation data;
  - Singular values;
  - Develop set to decide J;
  - Pre-selected J based on the ML objective function automatically.

Experiment Results

- **Connected Digits Experiments**
  - *eT0:* about 580 utterances ranging in length from 2 to 5 digits (18s per speaker)
  - *eT1:* about 580 utterances ranging in length from 6-15 digits (38s per speaker)
  - Var1/2/3: parked, low speed, high speed

Table 1. Performance comparison in sentence error rate (SER)

SER(%)	*eT2.var1*	*eT2.var2*	*eT1.var1*	*eT1.var2*	*eT0.var1*	*eT0.var2*
SI	12.7	31.1	12.1	28.6	11.3	23.5
+FMLLR	10.3	27.5	11.3	24.8	10.2	20.0
+Bilinear	10.1	27.0	11.8	24.3	9.7	19.9

- **Mandarin Voice Search Database**
  - The length of each utterance is about 6 seconds on average

Table 2. Performance comparison in character error rate (CER)

CER	*Voice Search database*
SI +SA w/ FMLLR	15.20%
SI+ SA w/ Bilinear model	13.75%

Conclusions and discussion

- **Bilinear models can effectively incorporate prior information and reduce the number of free parameters.**
- **Future works**
  - J selection with variant amount of adaptation data
  - class-dependent bilinear model adaptation by training the different bilinear model based on class information
  - efficiently control the speaker number of training dataset to further improve the robustness and performance
- **Any questions, pls contact with ShiLei Zhang (slzhang@cn.ibm.com)**