

# MIXTURE OF PROBABILISTIC LINEAR REGRESSIONS: A UNIFIED VIEW OF GMM-BASED MAPPING TECHNIQUES

Yu Qiao, Nobuaki Minematsu

Graduate School of Engineering, The University of Tokyo, Tokyo, Japan  
{qiao, mine}@gavo.t.u-tokyo.ac.jp

## ABSTRACT

This paper introduces a model of Mixture of Probabilistic Linear Regressions (MPLR) to learn a mapping function between two feature spaces. The MPLR consists of weighted combination of several probabilistic linear regressions, whose parameters are estimated by using matrix calculation. The mixture nature of MPLR allows it to model nonlinear transformation. The formulation of MPLR is general and independent of the types of the density models used. Two well-known GMM-based mapping methods for voice conversion [1, 2] can be regarded as special cases of MPLR. This unified view not only provides insights to the GMM-based mapping techniques, but also indicates methods to improve them. Compared to [1], our formulation of MPLR avoids solving complex linear equations and yields a faster estimation of the transform parameters. As for [2], the MPLR estimation provides a modified mapping function which overcomes an implicit problem in [2]'s mapping function. We carried out experiments to compare the MPLR-based methods with the traditional GMM-based methods [1, 2] on a voice conversion task. The experimental results show that the MPLR-based methods always have better performance in various parameter setups.

**Index Terms**— Space mapping, mixture model, linear regression, non-linear transform, voice conversion

## 1. INTRODUCTION

To find a mapping function between two feature spaces is a fundamental problem in many signal processing and pattern recognition problems. In speech engineering, a mapping function from the cepstrum feature of a source speaker to that of a target speaker can be used for voice conversion. In this paper, we propose a model called Mixture of Probabilistic Linear Regressions (MPLR) for learning the mapping function. The MPLR is made up of several Probabilistic Linear Regressions (PLR), whose parameters can be optimized through matrix calculation. The mapping function of MPLR is a weighted summation of the PLRs, where the weights depend on input samples. The MPLR is based on a similar linear calculation to PLR, however, it can deal with nonlinear transformations due to its mixture nature. Moreover, MPLR has such a flexible form that it doesn't need to specify the form of the density function.

We find that both of the two GMM-based mapping techniques [1, 2], which have been widely used for voice conversion, can be related to our MPLR. The difference between them comes from the density model and the hidden parameters used for parameter estimation. This unified view (MPLR) not only yields insights to the GMM-based mapping methods but also provides methods to improve them. As for [1], the formulation of MPLR yields a faster and more direct calculation of the mapping parameters without solving complex linear equations. We find the method of [2] includes an

implicit problem in its mapping function, and introduce a modified method to estimate the mapping parameters based on MPLR. We conduct comparison experiments between the MPLR-based methods and the traditional GMM-based methods on a voice conversion task. The results show that our MPLR-based methods have the least cepstrum distortion in various conditions. Although similar in names, our method is different from the Mixture of Linear Regressions (MLR) proposed in the context of statistics [3]. Different from our method, MLR doesn't make any use of the density of the source data. MPLR is also different from Maximum-likelihood stochastic transformation (MLST) [4] for speaker adaption of HMM. Unlike MPLR, the prior probability of MLST is calculated as the joint probability of mixture index and LR index.

## 2. LINEAR REGRESSION

Generally speaking, estimation of a mapping function can be seen as a regression problem [5, 6] from a source space to a target space. Let  $x$  denote a source vector with dimensionality  $n$ , and  $y$  denote a target vector with dimensionality  $m$ . The objective of regression is to estimate a mapping function  $y = f(x)$ . Assume we have a set of training samples  $\{x_i, y_i\}_{i=1}^I$ . Let  $X = [x_1, x_2, \dots, x_I]$  and  $Y = [y_1, y_2, \dots, y_I]$ . By minimizing the least squared error, the optimal mapping function can be estimated by  $\arg \min_f \sum_i |y_i - f(x_i)|^2$ . Assume  $f$  has a linear form. The problem reduces to a linear regression (LR) [5],

$$y'_i = Bx_i + b. \quad (1)$$

In this paper, we only consider unbiased linear regression, that is,  $E[y] = BE[x] + b$ . With argument vector  $\hat{x}_i = [x_i^T, 1]^T$ , Eq. 1 can be simplified to  $y'_i = A\hat{x}_i$ . Minimizing the summation of squared error (MSE), we have

$$\arg \min_A \sum_i |y_i - A\hat{x}_i|^2. \quad (2)$$

If we set  $p(y_i|\hat{x}_i, A) = (2\pi\sigma^2)^{-m/2} \exp(-\frac{1}{2\sigma^2}|y_i - A\hat{x}_i|^2)$ , the above Eq. 2 is essentially the same as the following maximum likelihood estimation  $\arg \max_A \prod_i p(y_i|\hat{x}_i, A)$ .

Let  $\hat{X} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_I]$ . The optimal  $A$  for Eq. 2 can be calculated by using matrix calculation,

$$A = Y\hat{X}^T(\hat{X}\hat{X}^T)^{-1}. \quad (3)$$

MSE of LR is an unbiased estimator. And according to Gauss-Markov theorem [6], among all the unbiased linear transformations, the MSE transformation of Eq. 2 has the minimum variance. For this reason, it is sometimes called the best linear unbiased estimator (BLUE).

The MSE objective function Eq. 2 of LR treats each training sample equally. In Probabilistic Linear Regression (PLR), we consider weight  $p_i$  for  $x_i$ . Note in this paper,  $p_i$  has not to be always a probability of  $x_i$ , and  $p_i$  can be a conditional probability of transform  $A$  given  $x_i$ . The optimal objective of PLR estimation is formulated as,

$$\arg \min_A \sum_i p_i |y_i - A\hat{x}_i|^2. \quad (4)$$

Define a diagonal matrix  $P$ , whose diagonal is  $[p_1, p_2, \dots, p_I]$ . The optimal  $A$  for PLR can be calculated by,

$$A = Y P \hat{X}^T (\hat{X} P \hat{X}^T)^{-1}. \quad (5)$$

### 3. MIXTURE OF PROBABILISTIC LINEAR REGRESSIONS

Although LR is simple, many real problems include nonlinear transformations which cannot be approximated well by a linear one. Perhaps the simplest idea to deal with nonlinear transformation is to divide the feature space  $S$  into several blocks and calculate a linear transformation for every block. According to Taylor theorem, there must exist a good linear approximation for each block if the division is fine enough. Statistically speaking, division of  $S$  can reduce bias of the estimated mapping. However, the hard and deterministic division of the feature space into blocks can be difficult. Especially, when the feature space has high dimensions and the number of training samples is limited, it is usually difficult to obtain enough training samples for each block. For this reason, instead of hard division of the feature space, we consider a probabilistic and soft division, which leads to the following Mixture of Probabilistic Linear Regressions (MPLR).

#### 3.1. Formulation of MPLR

This section describes a short introduction on the formulation of MPLR. Let us consider  $K$  ‘virtual spaces’  $\{S_k\}_{k=1}^K$  (Fig. 1), each of which has the same size as the source feature space. We use  $p(x|k)$  to represent the density of  $x$  in  $S_k$ . The densities  $\{p(x|k)\}$  yield information for soft division. Then we estimate a PLR  $y = A_k \hat{x}$  ( $A_k$  denotes the transformation matrix) for  $S_k$ . The final regression is a weighted combination of all PLRs, where the weights are given by posterior probability  $p(k|x)$ .  $p(k|x)$  represents a conditional probability of  $S_k$  given  $x$ . Formally, we have

$$y' = F_{\text{MPLR}}(x) = \sum_{k=1}^K p(k|x) A_k \hat{x}. \quad (6)$$

Given density  $p(x|k)$ , posterior  $p(k|x)$  can be calculated by using the Bayes’ theorem,

$$p(k|x) = \frac{w_k p(x|k)}{\sum_j w_j p(x|j)}, \quad (7)$$

where  $w_k = p(k)$  denotes a prior probability of the  $k$ -th PLR or  $S_k$ , and  $\sum_k w_k = 1$ .

The diagram of MPLR is depicted in Fig. 1. MPLR avoids the hard division of feature space, and makes effective use of all training data for estimating the transformation parameters of each PLR. It is noted that MPLR doesn’t make any special assumption on the form of  $p(x|k)$ , it can be Gaussian, Gaussian mixture, uniform, Gamma etc.. And it doesn’t include any specification on how  $p(x|k)$  can be

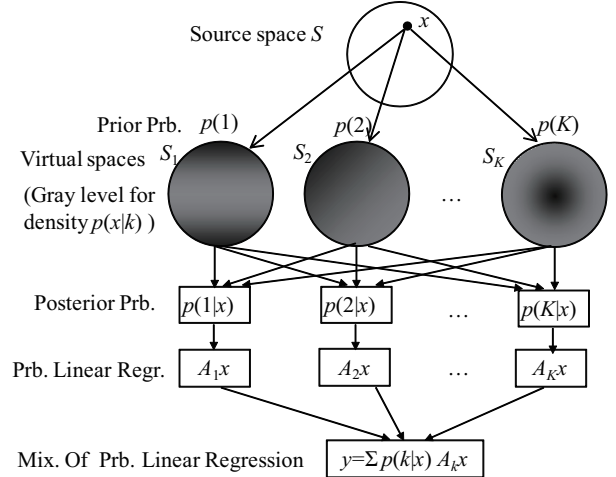


Fig. 1. Diagram of mixture of probabilistic linear regression

estimated. Just take two examples. We can estimate  $p(x|k)$  from  $x$  only using certain mixture models. Or we can calculate the joint probability  $p(x, y|k)$  at first and then estimate  $p(x|k)$  as a marginal probability  $p(x|k) = \int p(x, y|k) dy$ . Note these two estimations will practically lead to different  $p(x|k)$  as the second density estimation accounts for the joint relation between  $x$  and  $y$ . This flexibility allows us to design a specific form of  $p(x|k)$  for a certain problem.

Generally, the calculation of  $p(x|k)$  and  $p(k)$  is a density estimation problem, which has been widely addressed in statistics and pattern recognition. As we consider mixture model here, EM algorithm provides an effective tool for estimation [7]. So in the next, we assume that  $p(x|k)$  and  $p(k)$  are given, and our problem reduces to estimate the transformation matrix  $A_k$  of PLR in Eq. 6 from training data set  $\{x_i, y_i\}_{i=1}^I$ . For convenience, let  $p_{i,k} = p(x_i|k)$  and  $r_{i,k} = p(k|x_i) = \frac{w_k p_{i,k}}{\sum_j w_j p_{j,k}}$ . Define matrix  $R_k$  with diagonal  $\text{diag}(R_k) = [r_{1,k}, r_{2,k}, \dots, r_{I,k}]$ . The MSE estimation Eq. 6 is defined as,

$$\begin{aligned} & \arg \min_{\{A_k\}} \sum_i |y_i - F_{\text{MPLR}}(x_i)|^2 \\ &= \sum_i |y_i - \sum_k r_{i,k} A_k \hat{x}_i|^2 = \sum_i \left| \sum_k r_{i,k} (y_i - A_k \hat{x}_i) \right|^2. \end{aligned} \quad (8)$$

This is a linear optimization problem which can be solved directly. Let  $\hat{X}_k = [r_{1,k} \hat{x}_1, r_{2,k} \hat{x}_2, \dots, r_{I,k} \hat{x}_I]$  and  $\hat{\mathbf{X}} = [\hat{X}_1^T, \hat{X}_2^T, \dots, \hat{X}_K^T]^T$ . The optimal transform matrices  $\{A_k^*\}$  for Eq. 8 are given by

$$[A_1^*, A_2^*, \dots, A_K^*] = Y \hat{\mathbf{X}}^T (\hat{\mathbf{X}} \hat{\mathbf{X}}^T)^{-1}. \quad (9)$$

However, this is computationally expensive and can lead to biased PLR. Here we consider a fast and approximate calculation. Take  $r_{i,k}$  as a weight for  $x_i$  in  $S_k$ . Assume each PLR is unbiased  $\sum_i r_{i,k} y_i = \sum_i r_{i,k} A_k \hat{x}_i$ . We approximate Eq. 8 as

$$\arg \min_{\{A_k\}} \sum_k \sum_i r_{i,k} |y_i - A_k \hat{x}_i|^2. \quad (10)$$

According to Jensen’s inequality,  $\sum_k (w_k t_k)^2 \leq \sum_k w_k t_k^2$  ( $\sum_k w_k = 1$  and  $1 \geq w_k \geq 0$ ). The Eq. 10 is an upper bound of Eq. 8. Recalling the optimal solutions of PLR Eq. 4 5, we calculate optimal  $A_k$  for Eq. 10 as,

$$A_k = Y R_k \hat{X}_k^T (\hat{X}_k R_k \hat{X}_k^T)^{-1}. \quad (11)$$

The optimal  $A_k$  depends on training data  $\{x_i, y_i\}$  and  $r_{i,k}$ , whose values can be estimated from the density models  $p(x|k)$  and  $p(k)$ . Although Eq. 11 is only an approximately optimal answer for Eq. 8, we surprisingly found that the approximate Eq. 11 has a bit better performance on testing data than Eq. 9 in our experiments. We think this is because Eq. 9, which directly optimizes Eq. 8, may overfit the training data, and can lead to biased PLRs.

In the next, we will decompose the transformation  $A_k \hat{x}$  (Eq. 11) into another familiar form by using the conditional means and covariance matrices. Firstly, define the conditional means of  $x$  and  $y$  on  $S_k$  as,

$$\bar{x}_{R_k} = E_{p(k|x)}[x] = \frac{1}{N_k} \sum_i r_{i,k} x_i, \quad (12)$$

$$\bar{y}_{R_k} = E_{p(k|x)}[y] = \frac{1}{N_k} \sum_i r_{i,k} y_i, \quad (13)$$

where  $N_k = \sum_i r_{i,k}$  is used for normalization. Since our PLR is unbiased, we have  $\bar{y}_{R_k} = A_k \bar{x}_{R_k}$ . Then define the conditional covariance matrix and correlation matrix of  $x$  and  $y$  on  $S_k$  as,

$$\begin{aligned} V_{R_k}^{xx} &= E_{p(k|x)}[(x - E_{p(k|x)}[x])(x - E_{p(k|x)}[x])^T] \\ &= \frac{1}{N_k} \sum_i r_{i,k} (x_i - \bar{x}_{R_k})(x_i - \bar{x}_{R_k})^T, \end{aligned} \quad (14)$$

$$\begin{aligned} V_{R_k}^{yx} &= E_{p(k|x)}[(y - E_{p(k|x)}[y])(x - E_{p(k|x)}[x])^T] \\ &= \frac{1}{N_k} \sum_i r_{i,k} (y_i - \bar{y}_{R_k})(x_i - \bar{x}_{R_k})^T. \end{aligned} \quad (15)$$

Using Eq. 12, Eq. 13, Eq. 14, and Eq. 15, we obtain

$$A_k \hat{x} = \bar{y}_{R_k} + V_{R_k}^{yx} (V_{R_k}^{xx})^{-1} (x - \bar{x}_{R_k}). \quad (16)$$

In the remainder of this section, we show both of the two well-known voice conversion methods: GMM-based mapping [1] and GMM of joint vector density based mapping method [2] can be regarded as special cases of MPLR, which differ from each other in how to model densities and how to calculate  $\{r_{i,k}\}$ . Note that this unified view by MPLR not only provides insightful understanding of the methods, but also leads to techniques to improve them, for which we give the details later.

### 3.2. Connection to GMM based mapping methods [1, 2]

The GMM-based mapping (voice conversion) method was firstly introduced for voice conversion by Stylianou et al. [1]. This method makes use of GMM to model the density of source vector  $x$  as,

$$p_{\text{GMM}}(x) = \sum_{k=1}^K \alpha_k N(x|\mu_k, \Sigma_k), \quad (17)$$

where  $N(x|\mu_k, \Sigma_k)$  denotes a Gaussian distribution with mean  $\mu_k$  and covariance matrix  $\Sigma_k$ , and  $\{\alpha_k\}$  are the weights.

The authors [1] assumed that the mapping function has a form,

$$y' = F_{\text{GMM}}(x) = \sum_k p_{\text{GMM}}(k|x) (\nu_k + \Gamma_k \Sigma_k^{-1} (x - \mu_k)), \quad (18)$$

where  $p_{\text{GMM}}(k|x) = \frac{w_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^K w_j N(x|\mu_j, \Sigma_j)}$ . It is easy to see that GMM-based mapping function Eq. 18 reduces to MPLR mapping function Eq. 6, when we set  $p(x|k) = N(x|\mu_k, \Sigma_k)$ ,  $w_k = \alpha_k$ ,  $\bar{x}_{R_k} = \nu_k$  and  $V_{R_k}^{yx} = \Gamma_k$ . After the GMM is trained,  $\mu_k$  and  $\Sigma_k$

are known. And the unknown parameters  $\nu_k$  and  $\Gamma_k$  of Eq. 18 are obtained by solving the optimal problem [1],

$$\min_{\{\nu_k, \Gamma_k\}} \sum_i |y_i - \sum_k p_{\text{GMM}}(k|x_i) (\nu_k + \Gamma_k \Sigma_k^{-1} (x_i - \mu_k))|^2. \quad (19)$$

This is essentially the same as Eq. 8, and is computationally expensive, since  $\{\nu_k\}$  and  $\{\Gamma_k\}$  totally include  $Km + Knm$  variables. As discussed in [1], the optimization of Eq. 19 includes a heavy matrix-inverse step which requires  $O((Kn)^3)$  multiplications.

In our formulation of MPLR, the transformation parameters can be calculated directly by using Eq. 11, where the inverse only requires about  $O(Kn^3)$  multiplications. We call this new calculation (Eq. 11) as MPLR modified GMM-mapping, or MPLR-GMM for short. MPLR-GMM leads to faster computation with less memory cost than [1]. Note the mapping function of MPLR-GMM is not identical to that of GMM mapping [1] (Eq. 18), since MPLR-GMM doesn't optimize Eq. 19. We will compare them in experiments.

The GMM-based mapping [1] only performs density estimation on the source vectors  $\{x_i\}$ , and assumes that the target vectors  $\{y_i\}$  have the same clustering structure as the source one. To overcome this limitation, Kain et al. [2] used GMM to model the density of joint vector  $z_i = [x_i^T, y_i^T]^T$ ,

$$p_{\text{GMM-J}}(z) = \sum_{k=1}^K \alpha_k^z N(z|\mu_k^z, \Sigma_k^z), \quad (20)$$

where  $\lambda^z = \{\mu_k^z, \Sigma_k^z\}$ . We call this model 'GMM-J' for short. The mean vector  $\mu_k^z$  and  $\Sigma_k^z$  can be decomposed by,

$$\mu_k^z = \begin{bmatrix} \mu_k^x \\ \mu_k^y \end{bmatrix}, \Sigma_k^z = \begin{bmatrix} \Sigma_k^{xx} & \Sigma_k^{xy} \\ \Sigma_k^{yx} & \Sigma_k^{yy} \end{bmatrix}. \quad (21)$$

Then the transformation function [2, 8] is given by

$$\begin{aligned} y' &= F_{\text{GMM-J}}(x) \\ &= \sum_k \underbrace{\frac{\alpha_k^z N(x|\mu_k^x, \Sigma_k^{xx})}{\sum_j \alpha_j^z N(x|\mu_j^x, \Sigma_j^{xx})}}_{\text{Weight } p(k|x)} \underbrace{(\mu_k^y + \Sigma_k^{yx} \Sigma_k^{xx}{}^{-1} (x - \mu_k^x))}_{\text{Optimized for } p(k|x_i)}. \end{aligned} \quad (22)$$

If we set  $w_k = \alpha_k^z$ ,  $p(x|k) = N(x|\mu_k^x, \Sigma_k^{xx})$ ,  $\bar{x}_{R_k} = \mu_k^x$ ,  $\bar{y}_{R_k} = \mu_k^y$ ,  $V_{R_k}^{xx} = \Sigma_k^{xx}$  and  $V_{R_k}^{yx} = \Sigma_k^{yx}$ , Eq. 22 will be the same as the mapping function Eq. 6 of MPLR (remind Eq. 16).

There is an implicit problem of the GMM-J mapping function Eq. 22. Recall in the EM training of GMM, parameters  $\alpha_k^z$ ,  $\mu_k^x$ ,  $\mu_k^y$ ,  $\Sigma_k^{xx}$  and  $\Sigma_k^{yx}$  are calculated with the posterior probability  $p(k|z_i)$  of joint vector  $z$ . However, these transformation parameters should be calculated by using the posterior probability  $p(k|x_i)$  of source vector  $x$ . This is because, in the testing phase, only source vector  $x$  is given and we don't have complete information on  $z$ . In other words, in the GMM-J mapping function Eq. 22, while the weights  $\{p(k|x)\}$  are calculated from the posterior probability of source vector  $x$ , the transformation parameters  $\{\mu_k^x, \mu_k^y, \Sigma_k^{xx} \text{ and } \Sigma_k^{yx}\}$  are optimized for the posterior probability  $p(k|z)$  of joint vector  $z$ . This fact affects its performance. We use MPLR to overcome this problem. After GMM training of the joint vectors, we calculate the marginal probability of  $x$  as  $p(x|k) = \int p(z|k) dy$ , which is actually  $N(x|\mu_k^x, \Sigma_k^{xx})$ . In the next, we can calculate  $r_{i,k} = p(k|x_i)$  with Eq. 7 and the weight of  $p(k|x)$  as  $\alpha_k^x = \frac{\sum_j r_{i,j}}{\sum_j \sum_i r_{i,j}}$ . Then we use Eq. 11 to estimate the optimal transformation parameters  $A_k$ . We call this method the MPLR-modified GMM mapping with joint density estimation, or 'MPLR-GMM-J' for short.

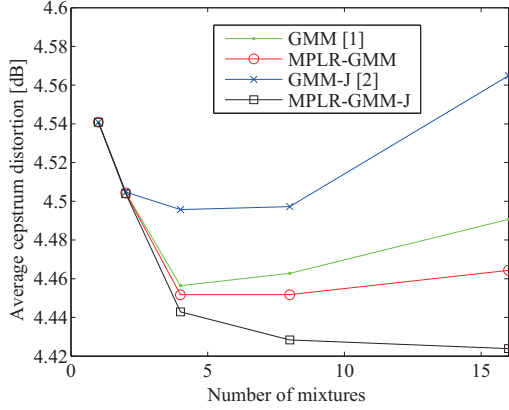


Fig. 2. Cepstrum distortion vs. the mixture number.

#### 4. EXPERIMENTS

We experimentally compared the proposed MPLR-GMM and MPLR-GMM-J with traditional GMM and GMM-J based mapping methods on a voice conversion task. The ATR-503 phoneme balanced corpus pronounced by a male speaker and a female speaker is used for evaluation. The sampling frequency of utterances is 16kHz. We converted the male voice to the female voice by using 20 dimension cepstrum features. The training data is aligned by DP matching. The cepstrum distortion [1, 8] between the target cepstrum vector  $[y_t^1, \dots, y_t^{20}]$  and the converted cepstrum vector  $[y_c^1, \dots, y_c^{20}]$  is defined as  $CD[dB](y_c, y_t) = 10/\ln 10 \sqrt{2 \sum_{d=1}^{20} (y_t^d - y_c^d)^2}$ . We calculate the average cepstrum distortion (ACD) as evaluation criteria.

We conducted two experiments for evaluation. In the first experiment, we randomly selected 40 sentences for training and used another 100 sentences for testing. We gradually changed mixture number  $K$  as 1,2,4,8,16. Note when  $K = 1$ , all the methods reduce to the classical linear regression. In the 2nd experiment, we fix the mixture number as 5 and change the number of training utterances  $M$  as 10,20,...,100. The number of utterances for testing is also set as 100. The results are summarized in Fig. 2 and Fig. 3. As one can see, the proposed MPLR-GMM-J always achieves the least ACD than among all the methods compared in various parameter setups. The performance of MPLR-GMM is a bit better than that of GMM. As discussed in Section 3.2, MPLR-GMM requires less time and memory cost than GMM.

It can be observed from Fig. 2 that the ACD difference between MPLR-GMM-J and GMM-J increases as the mixture number  $K$  increases. This is because, the transformation parameters of GMM-J depends on  $p(k|z)$  while the parameters of MPLR-GMM-J depends on  $p(k|x)$  (refer to Section 3.2 for details). Generally,  $p(k|z)$  becomes more unlike  $p(k|x)$  as mixture number  $K$  increases. It is expected that MPLR-GMM-J may have much better performance than GMM-J when  $K$  is large.

#### 5. CONCLUSIONS

This paper proposes the mixture of probabilistic linear regressions (MPLR) to learn the mapping function from a source feature space to a target one. The mapping parameters of MPLR can be estimated directly from matrix calculation, and the mixture nature of MPLR al-

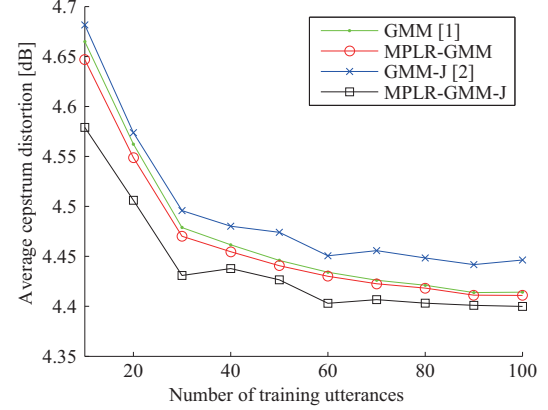


Fig. 3. Cepstrum distortion vs. the numbers of training utterances.

lows it to deal with the nonlinear mapping. Moreover, MPLR doesn't depend on a specific density model, which enables it to be suitable for various applications. We show the two famous GMM based mapping methods [1, 2] can be regarded as special cases of MPLR. And we find that the formulation of MPLR indicates methods to improve them. Compared with [1], MPLR provides faster calculation of mapping parameters and a bit better performance. The formulation of MPLR leads to a modified mapping function which can overcome an implicit conflict problem in [2]. We compared MPLR-based methods with the two traditional GMM-based methods in voice conversion. The experimental results show that our MPLR-based methods have less cepstrum distortions.

It is noted that it is not our objective in this paper to develop a high quality voice conversion (VC) system. In fact, one can combine our MPLR based methods with other VC techniques such as [8, 9] in developing practical systems. We are going to examine the proposed methods in larger database with subjective evaluations in the future. Finally, the formulation of MPLR is general and can have applications in other fields.

#### 6. REFERENCES

- [1] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. on SAP*, vol. 6, no. 2, pp. 131–142, 1998.
- [2] A. Kain and MW Macon, "Spectral voice conversion for text-to-speech synthesis," *Proc. ICASSP*, vol. 1, 1998.
- [3] R.D. De Veaux, "Mixtures of linear regressions," *Computational Statistics & Data Analysis*, vol. 8, no. 3, pp. 227–245, 1989.
- [4] VD Diakouloukas and VV Digalakis, "Maximum-likelihood stochastic-transformation adaptation of hidden Markov models," *IEEE Trans. on SAP*, vol. 7, no. 2, pp. 177–187, 1999.
- [5] D.C. Montgomery and E.A. Peck, *Introduction to linear regression analysis*, Wiley Series in Probability and Mathematical Statistics, 1982.
- [6] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*, Springer New York, 2001.
- [7] R.A. Redner and H.F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review*, vol. 26, pp. 195, 1984.
- [8] T. Toda, A.W. Black, and K. Tokuda, "Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory," *IEEE Trans. on ASLP*, vol. 15, no. 8, pp. 2222–2235, 2007.
- [9] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice Conversion with Smoothed GMM and MAP Adaptation," *Proc. Eurospeech*, 2003.