Speaker Adaptation using Relevance Vector Regression for HMM-based Expressive TTS

Doo Hwa Hong, Joun Yeop Lee, Se Young Jang, and Nam Soo Kim

Department of Electrical and Computer Engineering and INMC,
Seoul National University, Korea

{dhhong, jylee, syjang}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

The conventional maximum likelihood linear regression (MLLR)-based adaptation algorithm employed to acoustic hidden Markov models (HMMs) is too restricted in linear regression to represent the details of mapping charateristics. To overcome this problem, we propose the relevance vector regression (RVR)-based model parameter adaptation technique. In this framework, the conventional technique is extended to have much more basis functions. Also, the weights for conducting a transform matrix are obtained by sparse Bayesian learning, in which most of the weights become zero due to the definition of the prior with the precision hyper-parameters. Furthermore, by using the appropriate kernel functions, RVR can take both of the advantages of linear and nonlinear regression. In the experiments, the emotional speech database is used for adaptation to evaluate the proposed method compared with the conventional constrained MLLR. From the experimental results, we conclude that the RVR adaption method performs better than the conventional method.

Index Terms: speech synthesis, speaker adaptation, MLLR, relevance vector regression

1. Introduction

Maximum likelihood linear regression (MLLR) is one of the most popular techniques for parameter adaptation in Hidden Markov model (HMM)-based systems [1, 2]. In the MLLR approach, original parameters of the HMM-based system are mapped to their adapted values through a set of affine transformations which are estimated from a small amount of adaptation data. MLLR was first proposed for speaker adaptation in order to improve the performance of the speech recognition systems, and later a variety of extensions have been developed with applications to other areas [3–6]. Similar effect can be gained in the feature space with the use of the constrained MLLR (CMLLR) approach, in which the mean and variance transformations are required to have the equivalent form. Constrained structured maximum *a posteriori* linear regression (CSMAPLR) is also a well-known method for speaker adaptation in the speech synthesis area [7]. In this technique, robust linear transformations are obtained by employing the structured maximum *a posteriori* (SMAP) criterion rather than maximum likelihood (ML).

MLLR employs a linear mapping to transform the base model parameters to the corresponding adapted parameters. Though MLLR is simple and tractable, linear mapping used in MLLR is considered to be too restrictive to approximate a sophisticated mapping between source and target model parameters. Especially in adaptive speech synthesis, numerous models exist which may not be mapped properly onto the target speech with a small amount of adaptation data when a simple affine transform is applied. One of the promising ways to overcome this restriction is to employ nonlinear mappings.

In the area of speech recognition, the maximum penalized likelihood kernel regression (MPLKR) algorithm was proposed for fast speaker adaptation [8, 9]. In MPLKR, kernels are employed in the MLLR framework as the weights of regression vectors, and a penalization term is appended to the likelihood formulation in order to avoid overfitting. The basic idea of this technique is to map the mean vector of the base model to a high-dimensional feature space via a nonlinear mapping before performing linear regression. In our previous work, we also proposed the factored maximum penalized likelihood kernel regression (FMPLKR) technique for style-adaptive speech synthesis which conducts a nonlinear kernel regression between the mean vectors of the base model and the adaptation data obtained from different speaking styles [10, 11].

In this paper, we propose a novel speaker adaptation algorithm which uses the relevance vector regression (RVR) technique. In this work, transformed parameters are represented by the weighted sum of kernel functions centered at relevance vectors. By using nonlinear kernels as basis functions, the transformation function achieves flexibility to fit a given data set. Among the basis functions, only a few bases corresponding to relevance vectors are selected to define a transformation form and the others are dismissed by sparse Bayesian learning. Performance of the proposed technique is evaluated in a series of experiments on expressive speech synthesis, and compared with the results from the conventional MLLR-based method.

2. MLLR

In conventional MLLR adaptation, a P-dimensional mean vector $\boldsymbol{\mu}_s \in R^P$ of a particular distribution s of the HMM state is transformed to $\hat{\boldsymbol{\mu}}_s$ via

$$\hat{\boldsymbol{\mu}}_s = \boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{b} \quad (1)$$

$$= \boldsymbol{W}\boldsymbol{\xi}_s \quad (2)$$

where $\boldsymbol{W}$ is a $P \times (P+1)$ regression matrix which can be decomposed into $\boldsymbol{W} = [\boldsymbol{A} \quad \boldsymbol{b}]$ with $\boldsymbol{A}$ and $\boldsymbol{b}$ indicating the parameters of the affine transformation, and $\boldsymbol{\xi}_s$ denotes a $(P+1)$-dimensional augmented mean vector of the distribution s defined by:

$$\boldsymbol{\xi}_s = [\boldsymbol{\mu}_s^\top \quad 1]^\top \quad (3)$$

with $\top$ denoting the transpose of a matrix or a vector. In (3), it is seen that by appending a constant 1 to the mean vector $\boldsymbol{\mu}_s$, the original affine transform $\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{b}$ can be written as a linear formulation as given in (2). The output probability density

function of the distribution s is assumed to be a single Gaussian distribution with the mean vector $\boldsymbol{\mu}_s$ and covariance matrix $\boldsymbol{\Sigma}_s$. Generally, $\boldsymbol{\Sigma}_s$ is assumed to be diagonal [1].

The regression parameter $\boldsymbol{W}$ is estimated according to the ML criterion, and the expectation maximization (EM) algorithm is applied to increase the likelihood iteratively. Let $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_T\}$ be the given adaptation data vectors that are used to adapt the mean vectors of the HMM. During the *E* (expectation) step of the EM algorithm, we first compute the posterior probability of the distribution s at each time defined by:

$$\gamma_t(s) \quad = \quad Pr\left(\theta(t) = s|\boldsymbol{O},\ \lambda\right) \tag{4}$$

where $\theta(t)$ indicates the distribution index at time t and λ represents the current adaptation parameters. Then, during the *M* (maximization) step, we update the regression parameter $\boldsymbol{W}$ so as to maximize the expectation of the complete data log-likelihood in the unconstrained framework, i.e.:

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}}\Bigg[-\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{T}\gamma_t\left(s\right) \times \left(\boldsymbol{o}_t - \boldsymbol{W}\boldsymbol{\xi}_s\right)^{\top}\boldsymbol{\Sigma}_s^{-1}\left(\boldsymbol{o}_t - \boldsymbol{W}\boldsymbol{\xi}_s\right)\Bigg] \tag{5}$$

where $\hat{\boldsymbol{W}}$ is the updated parameter and S denotes the number of distribution in the same regression class. The solution to (5) is computed by differentiation with respect to each row of $\boldsymbol{W}$.

After estimating the mean transformation, the covariance transformation from $\boldsymbol{\Sigma}_s$ to $\hat{\boldsymbol{\Sigma}}_s$ is trained as defined by:

$$\hat{\boldsymbol{\Sigma}}_s = \boldsymbol{H}\boldsymbol{\Sigma}_s\boldsymbol{H}^{\top} \tag{6}$$

where $\boldsymbol{H}$ is a $P \times P$ transformation matrix. In unconstrained MLLR, $\boldsymbol{H}$ is estimated by the EM algorithm according to the ML criterion.

In CMLLR, the covariance transformation matrix $\boldsymbol{H}$ is forced to be the same as $\boldsymbol{A}$ in (1). Thus, the maximization of the expected log-likelihood in the constrained framework is given by:

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}}\Bigg[-\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{T}\gamma_t\left(s\right)\Big(\log|\boldsymbol{\Sigma}_s| - \log|\boldsymbol{A}|^2 + \left(\hat{\boldsymbol{o}}_t - \boldsymbol{\mu}_s\right)^{\top}\boldsymbol{\Sigma}_s^{-1}\left(\hat{\boldsymbol{o}}_t - \boldsymbol{\mu}_s\right)\Big)\Bigg] \tag{7}$$

where $\hat{\boldsymbol{o}}_t = \boldsymbol{A}^{-1}(\boldsymbol{o}_t + \boldsymbol{b})$. The solution to (7) is found by an iterative optimization scheme with respect to each row of $\boldsymbol{W}$. For more details, the reader is referred to [2].

3. Relevance vector regression for model adaptation

The adaptation scheme given in (2) can be rewritten as follows:

$$\hat{\boldsymbol{\mu}}_s = \sum_{j=1}^{P+1}\boldsymbol{w}_j\xi_{s,j} \tag{8}$$

where $\boldsymbol{w}_j$ and $\xi_{s,j}$ denotes the j-th column vector of $\boldsymbol{W}$ and the j-th element of $\boldsymbol{\xi}_s$, respectively. From (8), we can see that each element of $\hat{\boldsymbol{\mu}}_s$ turns out to be a linear combination of a number of variables. In this form, each $\xi_{s,j}$ acts as a variable and the (p, j)-th component w_{pj} of $\boldsymbol{W}$ is treated as a weight for the j-th variable to compose the p-th element of $\hat{\boldsymbol{\mu}}_s$.

Motivated by this viewpoint, we can extend (8) to a more generalized form as follows:

$$\hat{\boldsymbol{\mu}}_s = \sum_{j=1}^{M+1}\boldsymbol{w}_j\phi_j(\boldsymbol{\mu}_s) \tag{9}$$

$$= \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{\mu}_s) \tag{10}$$

where $\{\boldsymbol{w}_j | j = 1, 2, \cdots, M+1\}$ represents a set of weight vectors and bias, and $\phi_j(\boldsymbol{\mu}_s)$ indicates the j-th element of the $(M+1)$-dimensional feature vector $\boldsymbol{\phi}(\boldsymbol{\mu}_s)$ defined by

$$\boldsymbol{\phi}(\boldsymbol{\mu}_s) = \left[\phi_1(\boldsymbol{\mu}_s) \quad \cdots \quad \phi_M(\boldsymbol{\mu}_s) \quad 1\right]^{\top} \tag{11}$$

where $\phi_j(\boldsymbol{\mu}_s)$ is the j-th basis function. The number of basis functions M can be set greater than P, then (9) implies an over-complete representation. Another point to note in (9) is that ϕ_j is a nonlinear function which makes the regression flexible.

A promising way to define the nonlinear function $\phi_j(\cdot)$ is to apply a kernel map. Let $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_P\}$ denote a set of P vectors of dimension p. Then a kernel map is defined by

$$\phi_j(\boldsymbol{\mu}_s) = \kappa\left(\boldsymbol{\mu}_s, \boldsymbol{c}_j\right) \tag{12}$$

where $\kappa(\cdot, \cdot)$ denotes a kernel function. Combining (9) and (12), the model parameter is transformed in the following way:

$$\hat{\boldsymbol{\mu}}_s = \sum_{j=1}^{M}\boldsymbol{w}_j\kappa\left(\boldsymbol{\mu}_s, \boldsymbol{c}_j\right) + \mathbf{b} \tag{13}$$

where $\mathbf{b}$ indicates the bias term identical to $\boldsymbol{w}_{(M+1)}$.

Since (13) is a typical kernel regression form in which the mean of the output is assumed to be a weighted sum of kernel functions, we can apply the sparse Bayesian learning technique resulting in RVR [12].

3.1. Parameter estimation

Since the output is not a scalar but a vector, we apply the multivariate relevance vector machine(RVM) [13] for regression as follows:

$$\boldsymbol{o}_t \quad = \quad \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{\mu}_s) + \boldsymbol{\epsilon}_s \tag{14}$$

$$\quad = \quad \mathcal{N}(\boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{\mu}_s), \bar{\boldsymbol{\Sigma}}_s) \tag{15}$$

where $\boldsymbol{\epsilon}_s$ denotes a zero-mean additive noise with the covariance matrix $\bar{\boldsymbol{\Sigma}}_s$.

The optimal parameters of RVR are found by EM algorithm. During the E step, the posterior probability $\gamma_t(s)$ in (4) of the distribution s at each time t is computed. During the M step, the most probable weight matrix $\boldsymbol{W}$ is obtained by sparse Bayesian learning. The data set $\mathcal{D}$ for RVM training is conducted by collecting the input $\boldsymbol{\mu}'_s$ and output $\boldsymbol{o}'_t$ pair which are multiplied by the square root of non-zero $\gamma_t(s)$:

$$\mathcal{D} = \{(\boldsymbol{\mu}'_t(s), \boldsymbol{o}'_t(s)) | \gamma_t(s) \neq 0\} \tag{16}$$

where

$$\boldsymbol{\phi}'_t(\boldsymbol{\mu}_s) \quad = \quad \sqrt{\gamma_t(s)}\boldsymbol{\phi}(\boldsymbol{\mu}_s), \tag{17}$$

$$\boldsymbol{o}'_t(s) \quad = \quad \sqrt{\gamma_t(s)}\boldsymbol{o}_t. \tag{18}$$

We assume that the precision hyper-parameter α_m is fixed across every element of $\boldsymbol{w}_m$. Therefore, the prior on $\boldsymbol{w}_m$ is defined by:

$$p(\boldsymbol{w}_m|\alpha_m) = \prod_{p=1}^{P} \mathcal{N}(0, \alpha_m^{-1}). \tag{19}$$

Then, the terms of the marginal log-likelihood function which depend on α_m are given by

$$\mathcal{L}(\alpha_m) = \frac{1}{2}\sum_{p=1}^{P}\left(\log\alpha_m - \log(\alpha_m + s_{mp}) + \frac{q_{mp}^2}{\alpha_m + s_{mp}}\right) \tag{20}$$

where

$$s_{mp} = \boldsymbol{\phi}_m^\top \boldsymbol{C}_{-mp}^{-1}\boldsymbol{\phi}_m \tag{21}$$

$$q_{mp} = \boldsymbol{\phi}_m^\top \boldsymbol{C}_{-mp}^{-1}\boldsymbol{o}_p \tag{22}$$

with

$$\boldsymbol{\phi}_m = [\phi_m(\boldsymbol{\mu}_1), \phi_m(\boldsymbol{\mu}_2), \cdots, \phi_m(\boldsymbol{\mu}_N)]^\top \tag{23}$$

$$\boldsymbol{o}_p = [o_{1p}, o_{2p}, \cdots, o_{Np}]^\top \tag{24}$$

$$\boldsymbol{C}_{-mp} = \bar{\sigma}_{pp}^2\boldsymbol{I} + \sum_{i\neq m}\alpha_i\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\top \tag{25}$$

where N is the number of pairs in data set $\mathcal{D}$, $\boldsymbol{I}$ indicates the identity matrix, and $\bar{\sigma}_{pp}$ is the (p,p)-th entry of $\bar{\boldsymbol{\Sigma}}$. The optimal value of the hyper-parameter α_m that maximizes (19) can be obtained by solving the polynomial optimization problem.

There are several alternative ways to define the noise covariance matrix $\bar{\boldsymbol{\Sigma}}_s$. It can be assumed to be identical for all s, which means that it takes account of a regression error but variance is ignored in RVM training. On the other hand, it can be considered that the transformed covariance matrix is given by

$$\bar{\boldsymbol{\Sigma}}_s = \hat{\boldsymbol{\Sigma}}_s = \boldsymbol{H}\boldsymbol{\Sigma}_s\boldsymbol{H}^\top, \tag{26}$$

which means that the covariance transformation is also obtained during RVM training. Note that if $\bar{\boldsymbol{\Sigma}}_s$ is not diagonal, the RVM training becomes too complicated. If the covariance matrix is assumed to be diagonal, then $\boldsymbol{H}$ in (26) is constrained to be diagonal and can be obtained through noise estimation in sparse Bayesian learning [12].

3.2. Construction of kernel mapping

A simple way to define $\boldsymbol{\phi}$ is employing the radial basis function (RBF) kernel by which the local characteristics are considered:

$$\kappa_G(\boldsymbol{\mu}_s, \boldsymbol{c}_j) = \exp\left(-\frac{\|\boldsymbol{\mu}_s - \boldsymbol{c}_j\|^2}{2\sigma_\kappa^2}\right). \tag{27}$$

The center of RBF kernel $\boldsymbol{c}_j$ is determined by each original mean vector of the regression class. Thus, the number of basis functions M is the same to that of distributions S in a regression class.

Using affine transformation, the MLLR-based method could model the global characteristics of mapping between the original model and the target model but it yields too generalized results. In contrast, RVR with the RBF kernel could represent the details of the local characteristics of transformation but suffers from overfitting due to a near point and uncertainty due to a far point from the relevance vectors. To overcome the disadvantages of each technique by compensating them with the desirable qualities of one another, we also propose RVR to include not only RBF kernels but also linear kernels as follows:

$$\boldsymbol{o}_n = \mathcal{N}(\hat{\boldsymbol{\mu}}_s, \hat{\boldsymbol{\Sigma}}_s) \tag{28}$$

$$= \mathcal{N}(\boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{\mu}_s), \hat{\boldsymbol{\Sigma}}_s) \tag{29}$$

$$= \mathcal{N}(\boldsymbol{W}_L\boldsymbol{\phi}_L + \boldsymbol{W}_G\boldsymbol{\phi}_G(\boldsymbol{\mu}_s), \hat{\boldsymbol{\Sigma}}_s) \tag{30}$$

where $\boldsymbol{W} = [\boldsymbol{W}_L \quad \boldsymbol{W}_G]$ and $\boldsymbol{\phi}(\boldsymbol{\mu}_s) = [\boldsymbol{\phi}_L^\top \quad \boldsymbol{\phi}_G^\top(\boldsymbol{\mu}_s)]^\top$ with linear and nonlinear regression matrices $\boldsymbol{W}_L$ and $\boldsymbol{W}_G$, respectively, and linear and nonlinear feature $\boldsymbol{\phi}_L$ and $\boldsymbol{\phi}_G$ defined by RBF kernels, respectively. The linear feature $\boldsymbol{\phi}_L$ is obtained by linear kernel functions such that it is equal to $\boldsymbol{\mu}_s$ as given by

$$\phi_{L,j}(\boldsymbol{\mu}_s) = \kappa_L(\boldsymbol{\mu}_s, \boldsymbol{e}_j) \tag{31}$$

$$= \boldsymbol{\mu}_s^\top\boldsymbol{e}_j = \mu_{s,j} \tag{32}$$

where $\boldsymbol{e}_j$ indicates the j-th linear basis vector in the observation space. In this form, both local and global regression characteristics are captured in a unified framework.

4. Experiments

In order to evaluate the performance of the proposed technique when applied to speech synthesis, we conducted several experiments on objective measurement and subjective listening tests. All of the speech data collected for speech synthesis were Korean spoken language.

For the construction of the baseline speech synthesizer, a Korean speech database spoken by a male (HNC) and a female (YMK) speaker was applied. Each speaker provided 4,000 utterances of narrative speech data amounting to 525 and 507 minutes, respectively. A baseline narrative speech synthesizer was trained for each gender separately. We also collected the expressive speech data for adaptation from the utterances of two other speakers, JEK (male) and SKJ (female), with four different emotions: neutral, angry, joyful, and sad. The adaptation data consisted of 55 utterances amounting to 5 minutes on average for each speaker. Among the 55 utterances, we used 5 utterances for training the regression matrices and the remaining 50 utterances for evaluating the performance for each gender.

Each utterance was sampled at 16 kHz and a 20 ms Hamming window was applied with 5 ms frame shift for speech feature extraction. The acoustic features were obtained by STRAIGHT analysis [14]. As for the spectrum feature, a 25th-order mel-scaled cepstrum vector was extracted from each frame. By attaching the Δ- and $\Delta\Delta$-cepstra derived from the extracted mel-scaled cepstrum sequence, the spectrum feature can be represented by a 75-dimensional vector at each frame. We also extracted the fundamental frequency and 5-dimensional band aperiodicity from each frame for the excitation feature. As the basic unit of speech synthesis, we applied quinphones followed by context-dependent reading style text analysis described in [15]. Each quinphone was modeled by a 5-state left-to-right structured HMM where the observation distribution in each state was given by a single Gaussian with a diagonal covariance matrix.

In this experiment, we aimed at adapting the HMM parameters of the baseline narrative speech synthesizer to the given emotional speech data only for spectrum feature. The transform matrices were computed for static, Δ, and $\Delta\Delta$ spectrum features separately. The baseline models of speakers HNC and YMK were trained based on the decision tree-based clustering technique in which tied states share their observations and distributions. The numbers of clusters for each speaker are as follows: 7,295 and 5,001 clusters for the spectrum, 16,510 and

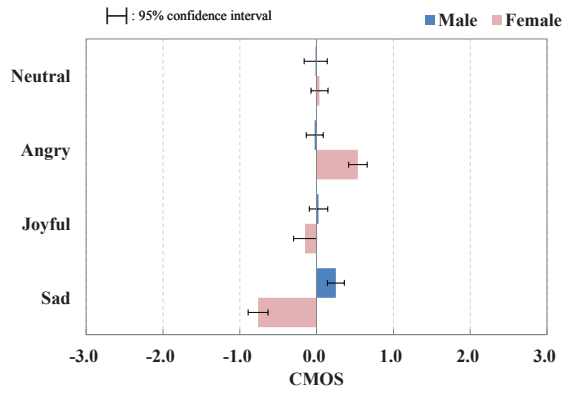

Figure 1: Results of CMOS test between CMLLR and RVR_G. Note that the lines on the top of the bars indicate the 95% confidence intervals.

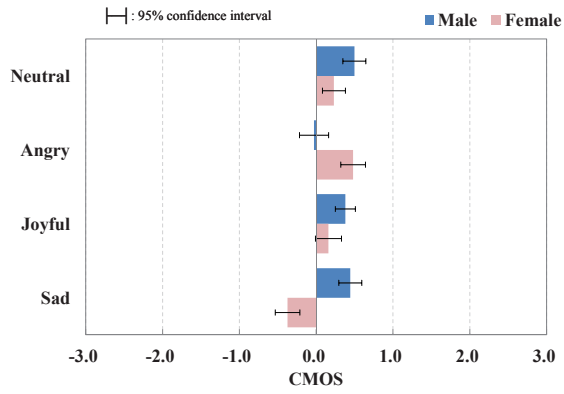

Figure 2: Results of CMOS test between CMLLR and RVR_J. Note that the lines on the top of the bars indicate the 95% confidence intervals.

16,943 clusters for the pitch, 4,585 and 3559 for the aperiodicity, and 2,134 and 1,585 clusters for the duration, respectively.

4.1. Objective performance evaluation

We compared the proposed algorithms, RVR using only RBF kernels (RVR_G) and RVR using both linear and RBF kernels jointly (RVR_J), with the conventional CMLLR. Based on previous experiment, we set σ_κ in (27) to not be fixed but to be determined dynamically by 10 times the sum of the standard deviations of $\boldsymbol{\mu_s}$. For the covariance transformation, H in (26) is forced to be diagonal.

Table 1: *Average mel-cepstral distance between the original and synthesized speech of speaker JEK (male).*

	Neutral	Angry	Joyful	Sad
CMLLR	5.196	7.040	6.013	6.383
RVR_G	5.265	6.957	6.095	7.086
RVR_J	**5.168**	**6.941**	**5.817**	**6.235**

Table 2: *Average mel-cepstral distance between the original and synthesized speech of speaker SKJ (female).*

	Neutral	Angry	Joyful	Sad
CMLLR	5.875	6.732	6.150	5.969
RVR_G	5.847	6.723	6.127	**5.962**
RVR_J	**5.827**	**6.653**	**5.885**	6.033

The results of objective performance test are shown in Tables 1 and 2, where we evaluated the average mel-cepstral distance in dB scale. The average mel-cepstral distance is obtained by the squared Euclidean norm of the difference between the original and synthesized speech mel-cepstra. From the results, we can find that the RVR_J approach is more effective in reducing the mel-cepstral distance than the other two algorithms, and the performance of RVR_G is unstable due to overfitting.

4.2. Subjective performance evaluation

Next, we performed a subjective listening test to compare the proposed algorithms with the conventional technique, in which 14 listeners participated and 10 sentences were used. After applying different numbers of regression classes, the best performed model with respect to objective measurement are chosen for each listening test. In the test, each listener was provided with speech samples synthesized through different methods, and the speech quality was measured in terms of the comparative mean opinion score (CMOS) [16]: for each test a pair of two speech files were given and each subject provided his/her preference score in speech quality in the range of [-3, 3] with a positive value indicating that the former shows a better quality than the latter, and negative value indicating the opposite. The results are shown in Figs. 1 and 2 from which we can find that the proposed approach using both linear and nonlinear kernels jointly produced a better speech quality than the conventional CMLLR method.

5. Conclusions

In this paper, we have proposed the RVR-based adaptation algorithms for HMM-based speech synthesis. The proposed approach provides nonlinear regression between the mean vector of the base model and the corresponding mean vectors of adaptation data with the use of kernel methods. By sparse Bayesian learning, only a few number of relevant bases and their weights are adopted and the rest are discarded. Additionally, using more than one kernel for basis functions makes the system have both advantages of linear and nonlinear regression: generality and locality. From the experimental results, it has been found that the proposed algorithm outperformed the conventional MLLR-based method in terms of the objective measure as well as the subjective listening quality.

6. Acknowledgements

This research was supported by the Mobile communication division, Samsung Electronics, co. Ltd. and the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP(Institute for Information & communications Technology Promotion)

7. References

[1] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[2] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 2, pp. 75–98, Apr. 1998.

[3] K. Visweswariah, V. Goel, and R. Gopinath, "Structuring linear transforms for adaptation using training time information," in *Proc. ICASSP*, vol. 1, 2002, pp. 585–588.

[4] B.-W. Mak and R.-H. Hsiao, "Kernel eigenspace-based MLLR adaptation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 3, pp. 784–795, Mar. 2007.

[5] Z. N. Karam and W. M. Campbell, "A multi-class MLLR kernel for SVM speaker recognition," in *Proc. ICASSP*, 2008, pp. 4117–4120.

[6] Y.-H. Sung, C. Boulis, and D. Jurafsky, "Maximum conditional likelihood linear regression and maximum a posteriori for hidden conditional random fields speaker adaptation," in *Proc. ICASSP*, 2008, pp. 4293–4296.

[7] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[8] I. W. Tsang, J. T. Kwok, B. Mak, K. Zhang, and J. Pan, "Fast speaker adaption via maximum penalized likelihood kernel regression," in *Proc. ICASSP*, vol. 1, 2006, pp. 997–1000.

[9] B.-W. Mak, T.-C. Lai, I. W. Tsang, and J.-Y. Kwok, "Maximum penalized likelihood kernel regression for fast adaptation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 7, pp. 1372–1381, Sep. 2009.

[10] J. S. Sung, D. H. Hong, H. W. Koo, and N. S. Kim, "Factored maximum likelihood kernelized regression for HMM-based singing voice synthesis." in *Proc. Interspeech*, 2013, pp. 359–363.

[11] J. S. Sung, D. H. Hong, and N. S. Kim, "Factored maximum penalized likelihood kernel regression for HMM-based style-adaptive speech synthesis." *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 2, pp. 251–261, Apr. 2014.

[12] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, Jun. 2001.

[13] A. Thayananthan, R. Navaratnam, B. Stenger, P. H. Torr, and R. Cipolla, "Multivariate relevance vector machines for tracking," in *Proc. European Conference on Computer Vision (ECCV)*, 2006, pp. 383–390.

[14] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *Proc. ICASSP*, vol. 2, 1997, pp. 1303–1306.

[15] J. S. Sung, D. H. Hong, K. H. Oh, and N. S. Kim, "Excitation modeling based on waveform interpolation for HMM-based speech synthesis." in *Proc. Interspeech*, 2010, pp. 813–816.

[16] V. Grancharov and W. B. Kleijn, "Speech quality assessment," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 83–100.