STATISTICAL PARAMETRIC SPEECH SYNTHESIS BASED ON PRODUCT OF EXPERTS

Heiga Zen[†‡], *Mark J. F. Gales*[†], *Yoshihiko Nankaku*[‡], *Keiichi Tokuda*[‡]

† Toshiba Research Europe Ltd. Cambridge Research Laboratory, Cambridge, UK
‡ Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan

heiga.zen@crl.toshiba.co.uk

ABSTRACT

Multiple-level acoustic models (AMs) are often combined in statistical parametric speech synthesis. Both linear and non-linear functions of the observation sequence are used as features in these AMs. This combination of multiple-level AMs can be expressed as a product of experts (PoE); the likelihoods from the AMs are scaled, multiplied together and then normalized. Currently these multiple-level AMs are individually trained and only combined at the synthesis stage. This paper discusses a more consistent PoE framework where the AMs are jointly trained. A generalization of trajectory HMM training can be used for multiple-level Gaussian AMs based on linear functions. However for the non-linear case this is not possible, so a scheme based on contrastive divergence learning is described. Experimental results show that the proposed technique provides both a mathematically elegant way to train multiple-level AMs and statistically significant improvements in the quality of synthesized speech.

***Index Terms*—** Statistical parametric speech synthesis, trajectory HMM, product of experts.

1. INTRODUCTION

Statistical parametric speech synthesis based on hidden Markov models (HMMs) [1] has grown in popularity in recent years [2]. It has various advantages against unit-selection synthesis, such as flexibility to change its voice characteristics, small footprint, and robustness. However a major limitation is the quality of synthesized speech. Zen *et al.* [2] pointed out three major factors that degrade the quality of the synthesized speech; vocoder, accuracy of acoustic models (AMs), and over-smoothing. This paper addresses two of these: the accuracy of AMs; and over-smoothing.

One way to improve the accuracy of the AMs is to use trajectory HMMs [3]. This model was derived from the HMM by imposing explicit relationships between static and dynamic features during training. It thus addresses an inconsistency between training and synthesis that exists with HMMs trained in the standard fashion, and has been found to improve the quality of the synthesized speech. To reduce the over-smoothing effect, the combination of multiple-level AMs has recently been proposed [4, 5, 6, 7, 8, 9], in which acoustic features of the training data at various levels (*e.g.*, phoneme, syllable, word, phrase, utterance) are extracted and modelled *individually*. At the synthesis stage, speech parameters that *jointly* maximize the output probabilities from these multiple-level AMs are generated. Additionally, the output probabilities from the AMs are weighted to control the balance between the AMs. The weights are tuned manually or optimized using held-out data.

This paper proposes a technique to *jointly* estimate these multiple-level AMs within the product of experts (PoE) framework [10]. The output probabilities from the individual models are multiplied together, effectively forming an intersection of the distributions. This is an efficient way to model high-dimensional data which simultaneously satisfies many different low-dimensional constraints. Each expert is able to focus on satisfying just one of these low-dimensional constraints. The use of the PoE framework allows general multiple-level AMs to be trained cooperatively, removing the need to estimate weights.

The rest of this paper is organized as follows. Section 2 discusses the relationship between the trajectory HMM and PoE. Section 3 shows how to combine multiple-level AMs as a PoE and how to estimate parameters of PoEs. Experimental results are given in Section 4. Concluding remarks are presented in the final section.

2. TRAJECTORY HMM AS PRODUCT OF EXPERTS

2.1. Definition of Trajectory HMM

The trajectory HMM [3], which is an underlying generative model of HMM-based speech synthesis [1], is defined as

$$p\left(\boldsymbol{c} \mid \lambda\right) = \sum_{\forall \boldsymbol{q}} p\left(\boldsymbol{c} \mid \boldsymbol{q}, \lambda\right) P\left(\boldsymbol{q} \mid \lambda\right), \tag{1}$$

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \lambda\right) = \mathcal{N}\left(\boldsymbol{c}; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{2}$$

where λ denotes a set of parameters, $\boldsymbol{c} = [c_1, \dots, c_T]^\top$ is a static feature sequence for an utterance, c_t is the static feature at time t,[1] $\boldsymbol{q} = \{q_1, q_2, \dots, q_T\}$ is a state sequence, $q_t \in \{1, \dots, N\}$ is the state at t, N is the number of state-output distributions, T is the number of frames in $\boldsymbol{c}$, and $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ correspond to the $T \times 1$ mean vector and the $T \times T$ covariance matrix for $\boldsymbol{q}$. They are given by

$$\boldsymbol{R}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{r}_{\boldsymbol{q}}, \tag{3}$$

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}, \qquad \boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}, \tag{4}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top, \dots, \boldsymbol{\mu}_{q_T}^\top\right]^\top, \qquad \boldsymbol{\mu}_{q_t} = \left[\mu_{q_t}^{(0)}, \mu_{q_t}^{(1)}, \mu_{q_t}^{(2)}\right]^\top, \tag{5}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1}, \dots, \boldsymbol{\Sigma}_{q_T}\right], \quad \boldsymbol{\Sigma}_{q_t} = \operatorname{diag}\left[\Sigma_{q_t}^{(0)}, \Sigma_{q_t}^{(1)}, \Sigma_{q_t}^{(2)}\right], \tag{6}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ correspond to the 3×1 mean vector and the 3×3 covariance matrix associated with the i-th state. In Eq. (4), $\boldsymbol{W}$ is typically a $3T \times T$ window matrix appending the first and second-order dynamic features to $\boldsymbol{c}$. If zeroth- (static), first-, and second-order dynamic features are calculated as

$$f_t^{(0)}(\boldsymbol{c}) = c_t, \tag{7}$$

$$f_t^{(1)}(\boldsymbol{c}) = \left(c_{t+1} - c_{t-1}\right)/2, \tag{8}$$

$$f_t^{(2)}(\boldsymbol{c}) = c_{t-1} - 2c_t + c_{t+1}, \tag{9}$$

[1] For notational simplicity, static features are assumed to be scalar values. Extension for vectors is straightforward.

then $\boldsymbol{W}$ becomes

$$
\left[\begin{array}{c} \vdots \\ \hline f_{t-1}^{(0)}(\boldsymbol{c}) \\ f_{t-1}^{(1)}(\boldsymbol{c}) \\ f_{t-1}^{(2)}(\boldsymbol{c}) \\ \hline f_{t}^{(0)}(\boldsymbol{c}) \\ f_{t}^{(1)}(\boldsymbol{c}) \\ f_{t}^{(2)}(\boldsymbol{c}) \\ \hline f_{t+1}^{(0)}(\boldsymbol{c}) \\ f_{t+1}^{(1)}(\boldsymbol{c}) \\ f_{t+1}^{(2)}(\boldsymbol{c}) \\ \hline \vdots \end{array}\right]
=
\overset{\boldsymbol{W}}{\left[\begin{array}{cccccc} \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \hline \cdots & 0 & 1 & 0 & 0 & \cdots \\ \cdots & -0.5 & 0 & 0.5 & 0 & \cdots \\ \cdots & 1 & -2 & 1 & 0 & \cdots \\ \hline \cdots & 0 & 0 & 1 & 0 & \cdots \\ \cdots & 0 & -0.5 & 0 & 0.5 & \cdots \\ \cdots & 0 & 1 & -2 & 1 & \cdots \\ \hline \cdots & 0 & 0 & 0 & 1 & \cdots \\ \cdots & 0 & 0 & -0.5 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & -2 & \cdots \\ \hline \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right]}
\overset{\boldsymbol{c}}{\left[\begin{array}{c} \vdots \\ \hline c_{t-2} \\ \hline c_{t-1} \\ \hline c_{t} \\ \hline c_{t+1} \\ \hline \vdots \end{array}\right]}
$$

where $f_t^{(d)}(\boldsymbol{c})$ is a function to calculate the d-th-order dynamic feature at time t for given $\boldsymbol{c}$. ML estimation techniques of trajectory HMMs based on the Viterbi (most likely path) [3] and Markov chain Monte Carlo (MCMC) [11] approximations have been derived.

2.2. PoE Interpretation of Trajectory HMM

Equation (2) can be reformulated as

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \lambda\right) = \frac{1}{Z_{\boldsymbol{q}}^{(\mathrm{trj})}} \Psi^{(\mathrm{trj})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda\right), \tag{10}$$

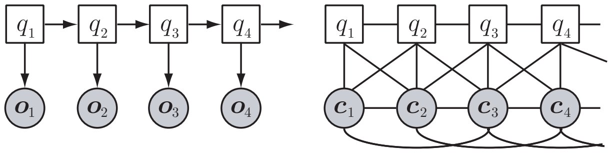
$$\Psi^{(\mathrm{trj})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda\right) = \prod_{t=1}^{T} \prod_{d=0}^{2} \mathcal{N}\left(f_t^{(d)}(\boldsymbol{c}); \mu_{q_t}^{(d)}, \Sigma_{q_t}^{(d)}\right), \tag{11}$$

where $Z_{\boldsymbol{q}}^{(\mathrm{trj})}$ is a normalization constant (also known as partition function) which ensures that the resulting distribution is a valid *probability* distribution. It is given as

$$\begin{aligned} Z_{\boldsymbol{q}}^{(\mathrm{trj})} &= \int_{\boldsymbol{c}} \Psi^{(\mathrm{trj})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda\right) d\boldsymbol{c} = \int_{\boldsymbol{c}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) d\boldsymbol{c} \qquad (12) \\ &= \frac{\sqrt{(2\pi)^{MT} \left|\boldsymbol{P}_{\boldsymbol{q}}\right|}}{\sqrt{(2\pi)^{3MT} \left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^{\top} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}\right)\right\}. \end{aligned}$$

As discussed in [3, 12], Eq. (10) can be viewed as a PoE [10]; local constraints are modeled by Gaussian experts then they are multiplied over time and normalized to yield a valid probability distribution.[2] Here, the number of experts is larger than the number of input dimensions (3 × over-complete).

It is interesting to note that Eq. (10) can also be viewed as a Markov random field (MRF) whose cliques are defined by Eqs. (7)–(9) and clique potential functions are given by Gaussian distributions. Because there is a latent variable (state sequence) and potential functions are Gaussian distributions, a trajectory HMM can be viewed as a hidden Gaussian Markov random field (HGMRF). It is known that PoEs and MRFs can be represented as undirected graphs. The graphical model representations of HMM and trajectory HMM whose window matrix is specified by Eqs. (7)–(9) are shown in Fig. 1. Note that edges in Fig. 1(b) depends on cliques specified by $f_t^{(d)}(\boldsymbol{c})$. Therefore, if different $f_t^{(d)}(\boldsymbol{c})$ are used, the graphical model structure of trajectory HMM also differs.



(a) HMM (b) trajectory HMM

Fig. 1. Graphical model representation of (a) HMM and (b) trajectory HMM whose window matrix is specified by Eqs. (7)–(9).

3. COMBINATION OF MULTIPLE-LEVEL ACOUSTIC MODELS AS PRODUCT OF EXPERTS

To reduce the over-smoothing effect, techniques based on combining multiple-level AMs have been proposed [4, 5, 6, 8, 9]. These approaches generate speech parameters using

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \prod_{i=1}^{M} \left\{p(f_i(\boldsymbol{c}) \mid \lambda_i)\right\}^{\alpha_i} = \arg\max_{\boldsymbol{c}} \sum_{i=1}^{M} \alpha_i \log p(f_i(\boldsymbol{c}) \mid \lambda_i) \tag{13}$$

where M is the number of AMs and α_i, λ_i, and $f_i(\boldsymbol{c})$ correspond to a weight, the set of parameters, and an arbitrary function to extract features from $\boldsymbol{c}$ for the i-th AM. Considering each AM as an "expert," Eq. (13) can be interpreted as generating $\boldsymbol{c}$ from a PoE. The standard approach is to train each AM (expert) independently, then tune the weights to obtain the best performance. This section shows how to estimate these multiple-level AMs simultaneously based on the PoE framework. This removes the need to use (or train) expert weights as the individual expert variances will subsume their role.

3.1. Linear and Gaussian Case

Most of techniques for combining multiple-level AMs adopt linear functions (*e.g.*, DCT coefficients [6, 8], averages [9] and summations [5, 7]) for $\{f_i(\boldsymbol{c})\}_{i=1}^{M}$ and use Gaussian distributions to model $\{p(f_i(\boldsymbol{c}) \mid \lambda_i)\}_{i=1}^{M}$. In this case, PoEs obtained by combining these multiple-level AMs may be viewed as trajectory HMMs with different structures for $\boldsymbol{W}$, $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$. Therefore, parameter estimation techniques derived for trajectory HMMs [3, 11] can directly be applied to estimate these multiple-level AMs simultaneously.

3.2. Non-Linear and Non-Gaussian Cases

One very successful combination of multiple-level AMs in HMM-based speech synthesis is the speech parameter generation algorithm considering global variance (GV) [4]. A GV is defined as an intra-utterance variance of a speech parameter trajectory and typically modeled by a Gaussian distribution. A PoE consisting of a trajectory HMM and a GV Gaussian distribution can be written as

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) = \frac{1}{Z_{\boldsymbol{q}}^{(\mathrm{trj\cdot GV})}} \Psi^{(\mathrm{trj\cdot GV})}\left(\boldsymbol{c}; \boldsymbol{q}, \Lambda\right), \tag{14}$$

$$Z_{\boldsymbol{q}}^{(\mathrm{trj\cdot GV})} = \int_{\boldsymbol{c}} \Psi^{(\mathrm{trj\cdot GV})}\left(\boldsymbol{c}; \boldsymbol{q}, \Lambda\right) d\boldsymbol{c}, \tag{15}$$

$$\Psi^{(\mathrm{trj\cdot GV})}\left(\boldsymbol{c}; \boldsymbol{q}, \Lambda\right) = \left\{\Psi^{(\mathrm{trj})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda\right)\right\}^{\alpha} \cdot \Psi^{(\mathrm{GV})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda_v\right), \tag{16}$$

$$\Psi^{(\mathrm{GV})}\left(\boldsymbol{c}; \boldsymbol{q}, \lambda_v\right) = \mathcal{N}\left(f_v(\boldsymbol{c}); \mu_v, \Sigma_v\right), \tag{17}$$

where $\Lambda = \{\lambda, \lambda_v\}$, λ_v is the set of parameters for modeling GVs, $Z_{\boldsymbol{q}}^{(\mathrm{trj\cdot GV})}$ is a normalization constant, α is an utterance-length adaptive

[2] From this point of view, the HMM whose state-output vector includes static and dynamic features is unnormalized PoE.

weight (typically $\alpha = 1/3T$), μ_v and Σ_v correspond to the mean and variance of the GV Gaussian distribution, and $f_v(\boldsymbol{c})$ is a function to compute GV of $\boldsymbol{c}$ defined as

$$f_v(\boldsymbol{c}) = \frac{1}{T}\sum_{t=1}^{T}(c_t - \bar{c})^2, \quad \bar{c} = \frac{1}{T}\sum_{t=1}^{T} c_t. \tag{18}$$

Equation (18) shows that $f_v(\boldsymbol{c})$ is a non-linear (quadratic) function of $\boldsymbol{c}$. Unlike the linear and Gaussian case described in Section 3.1, it is unable to apply the training algorithm for trajectory HMMs directly to estimate this PoE. Furthermore, there is no closed-form solution to calculate the normalization constant $Z_{\boldsymbol{q}}^{(\mathrm{trj}\cdot\mathrm{GV})}$.

However, the parameters of this PoE can be estimated using *contrastive divergence* learning [13]. This is a gradient-based learning technique and well-approximates ML estimation of parameters. An important of contrastive divergence learning is that computation of the normalization constant is not required. The contrastive divergence parameter update is written as

$$\Lambda' = \Lambda - \eta \cdot \delta\Lambda, \tag{19}$$

$$\delta\Lambda = \left\langle \frac{\partial \Psi_{\mathrm{trj}\cdot\mathrm{GV}}(\boldsymbol{c};\boldsymbol{q},\Lambda)}{\partial \Lambda} \right\rangle_{p^0} - \left\langle \frac{\partial \Psi_{\mathrm{trj}\cdot\mathrm{GV}}(\boldsymbol{c};\boldsymbol{q},\Lambda)}{\partial \Lambda} \right\rangle_{p^\infty} \tag{20}$$

$$\approx \left\langle \frac{\partial \Psi_{\mathrm{trj}\cdot\mathrm{GV}}(\boldsymbol{c};\boldsymbol{q},\Lambda)}{\partial \Lambda} \right\rangle_{p^0} - \left\langle \frac{\partial \Psi_{\mathrm{trj}\cdot\mathrm{GV}}(\boldsymbol{c};\boldsymbol{q},\Lambda)}{\partial \Lambda} \right\rangle_{p^j}, \tag{21}$$

where Λ' is the set of parameters after update, η is a user-defined learning rate, $\langle\cdot\rangle_{p^0}$ denotes the expectation over the empirical (data) distribution, and $\langle\cdot\rangle_{p^j}$ denotes the expectation over the model distribution after j MCMC sampling iterations. Refer to [13] for further details about the contrastive divergence learning.

Note that multiple-level AMs with non-Gaussian distributions [7, 8] can also be estimated simultaneously using the contrastive divergence learning because it can be applied to PoEs with non-Gaussian experts with non-linear feature functions.

4. EXPERIMENTS

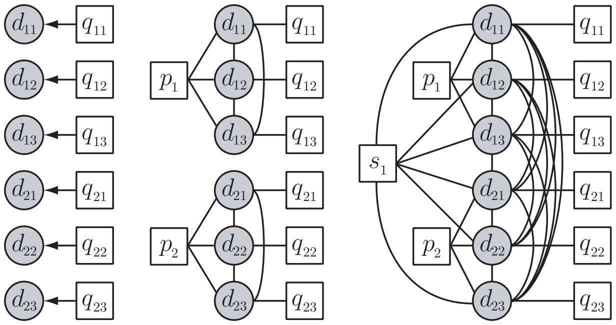
4.1. Experimental Conditions

About 2,500 English sentences uttered by a female professional speaker were used for training. The speech analysis conditions and model topologies of Nitech-HTS 2005 [14] were used. Note that phoneme segmentations, prosodic labels, and $\log F_0$ values were manually annotated. After training the baseline system, PoEs were estimated using the baseline system as their initial models.

4.2. Multiple-level Duration Models as PoE

The first experiment investigated the effect of joint estimation for multiple-level duration models (state and phoneme [5] and state, phoneme, and syllable [7]). Phoneme and syllable duration models were built using manually annotated segmentations, *i.e.*, they were not integrated into the Baum-Welch re-estimation process. They were clustered by decision trees based on the MDL criterion [15] in the same way as state duration models [1]. Then, the proposed technique was applied to estimate these state, phoneme, and syllable duration models. The feature functions to extract phoneme and syllable durations from a sequence of state durations were defined as

$$f_i^{(\mathrm{ph})}(\boldsymbol{d}) = \sum_{j=1}^{N_i} d_{ij}, \qquad f_k^{(\mathrm{syl})}(\boldsymbol{d}) = \sum_{i\in \mathrm{syl}(k)} \sum_{j=1}^{N_i} d_{ij}, \tag{22}$$



(a) baseline (st) (b) PoE (st*ph) (c) PoE (st*ph*syl)

Fig. 2. Graphical model representation of (a) baseline, (b) PoE (state*phoneme), and (c) PoE (state*phoneme*syllable) duration models. In this figure, d_{ij}, q_{ij}, p_i, and s_k correspond to state duration of state i of phoneme j, state-duration distribution of state i of phoneme j, phoneme-duration distribution of phoneme j, and syllable-duration distribution of syllable k.

where $\boldsymbol{d} = \left[d_{11}, \ldots, d_{1N_1}, \ldots, d_{P1}, \ldots, d_{PN_P}\right]^\top$ is a sequence of state durations, d_{ij} denotes the state duration of state i of phoneme j, P denotes the number of phonemes in an utterance, N_i is the number of states in phoneme i, and syl(k) denotes the set of phonemes associated to syllable k. Because Eq. (22) is linear and state, phoneme, and syllable durations were modeled by Gaussian distributions, the estimation technique described in Section 3.1 can be applied. The graphical model representations of the baseline state duration model and multiple-level state duration models, whose window matrices are specified by Eq. (22), are illustrated in Fig. 2. It can be seen from the figure that the multiple-level duration models have a more complex dependency structure than the baseline system.

Table 1. Root mean square errors (RMSEs) of duration prediction by baseline and proposed duration models. In this table, `st*ph` and `st*ph*syl` correspond to products of state and phoneme duration models and product of state, phoneme, and syllable duration models, and "uPoE" denotes unnormalized PoE (conventional techniques).

Duration models	RMSE in frame (rel. imp. in %)		
	phoneme	syllable	pause
baseline (`st`)	5.08	8.98	35.0
uPoE (`st*ph`) [5]	4.62 (9.1)	8.13 (9.5)	**31.8** (9.1)
uPoE (`st*ph*syl`) [7]	4.62 (9.1)	8.11 (9.7)	**31.8** (9.1)
PoE (`st*ph`)	4.60 (9.4)	8.04 (10.5)	31.9 (8.9)
PoE (`st*ph*syl`)	**4.57** (10.0)	**8.02** (10.7)	31.9 (8.9)

Table 1 shows the duration prediction results. The duration prediction accuracy was evaluated on an evaluation set (137 sentences) not included in the training set. The "uPoE" systems use the standard independent training of the "experts" with the weights optimized to minimize RMSEs (of phoneme) over the development set (137 sentences) not included in the training and test sets by grid search. The weight of the phoneme duration models for uPoE (`st*ph`) was 1.9, and the weights of phoneme and syllable duration models for uPoE (`st*ph*syl`) were 2.3 and 0.7, respectively. It can be seen from the table that PoE achieved significant error reduction over the baseline system and comparable performance to uPoE without requiring the use of the development set for weight tuning.

4.3. GV as PoE

The second experiment investigated the effect of joint estimation of trajectory HMMs and GV Gaussian distributions. Instead of using the entire database at each iteration of the contrastive divergence learning, the data was split into "mini batches" of about 250 utterances each, and used only the data from one batch at each iteration. In this experiment, 10,000 stochastic gradient iterations was used, each performing a contrastive divergence learning step ($j = 10$). The learning rate was started from $\eta = 0.01$ and annealed (halved) at every 2,000 iterations. To improve the learning speed, the momentum method was used [16]; the parameter updates were supplemented by adding 0.9 times the previous update. Initializing the MCMC sampler at the data point, which is a typical setting used in the contrastive divergence learning, did not work well for training GV-PoE because GV-PoE is highly non-linear and its model distribution has multiple modes.[3] To address this problem, instead of initializing the MCMC sampler at the data point, it was initialized at the speech parameter trajectory generated by the speech parameter generation algorithm including the GV "expert". This point should be in the mode of interest. To draw samples from $\Psi_{\mathrm{trj\cdot GV}}(\boldsymbol{c}; \boldsymbol{q}, \Lambda)$, hybrid (Hamiltonian) Monte Carlo sampling [18] with 20 leap-frog steps was used. The leap step was fixed to 0.001. The context-dependent logarithmic GV without silence [19] rather than standard GV [4] was used in this experiment.

To evaluate the performance of PoE between trajectory HMM and GV, a paired-comparison preference listening test was conducted. This test compared the baseline using independent training of the experts and weights (unnormalized PoE), and PoE systems, over 70 sentences from the evaluation set. The subjects were seven speech researchers (two native US English speakers, four native UK English speakers, and one native South African English speaker). Thirty sentences were randomly chosen from the evaluation set for each subject. Samples were presented in a random order for each test sentence. Before starting the test, the subjects listened to speech samples of one sentence to become familiar with the task. This sentence was randomly chosen for each subject and excluded from the actual test. After listening to each test sample, the subjects were asked to choose their preferred one. Note that the subjects could select "No preference" if they had no preference.

Table 2. Preference scores (%) between the baseline, unnormalized PoE (`trj*GV`) and proposed, normalized PoE (`trj*GV`) systems.

uPoE (`trj*GV`)	PoE (`trj*GV`)	No preference
17.1	32.4	50.5

Table 2 shows the preference test result. It can be seen from the table that the proposed PoE system achieved a better score than the baseline one. The difference between uPoE and PoE was statistically significant at the $p < 0.05$ level. The speech parameter trajectories generated from uPoE had much smaller variations in GV than natural speech, *i.e.*, the GVs of the generated speech parameter trajectories were almost the same as the mean of GV Gaussian distribution. On the other hand, the speech parameter trajectories generated from PoE had much larger variations in GV than those from uPoE and close to the natural speech.

5. CONCLUSIONS

This paper reformulated techniques of combining multiple-level AMs in statistical parametric speech synthesis as PoEs and showed estimation techniques of these PoEs. Experimental results showed that the proposed technique achieved significant improvements in the quality of synthesized speech over the conventional ones.

Future work includes investigation of other feature functions and/or distributions for experts in the proposed framework.

6. REFERENCES

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[2] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2007.

[4] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[5] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method," in *Proc. Blizzard Challenge Workshop*, 2006.

[6] J. Latorre and M. Akamine, "Multilevel parametric-base F0 model for speech synthesis," in *Proc. Interspeech*, 2008, pp. 2274–2277.

[7] B.-H. Gao, Y. Qian, Z.-Z. Wu, and F.-K. Soong, "Duration refinement by jointly optimizing state and longer unit likelihood," in *Proc. Interspeech*, 2008, pp. 2266–2269.

[8] Y. Qian, Z.-Z. Wu, and F.-K. Soong, "Improved prosody generation by maximizing joint likelihood of state and longer units," in *Proc. ICASSP*, 2009, pp. 3781–3784.

[9] C.-C. Wang, Z.-H. Ling, B.-F. Zhang, and L.-R. Dai, "Multi-layer F0 modeling for HMM-based speech synthesis," in *Proc. ISCSLP*, 2008, pp. 129–132.

[10] G. Hinton, "Product of experts," in *Proc. ICANN*, vol. 1, 1999, pp. 1–6.

[11] H. Zen, K. Tokuda, and T. Kitamura, "Estimating trajectory HMM parameters by Monte Carlo EM with Gibbs sampler," in *Proc. ICASSP*, 2006, pp. 1173–1176.

[12] C. Williams, "How to pretend that correlated variables are independent by using difference observations," *Neural Computation*, vol. 17, no. 1, pp. 1–6, 2005.

[13] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[14] H. Zen, T. Toda, M. Nakamura, and T. Tokuda, "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 325–333, 2007.

[15] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. Eurospeech*, 1997, pp. 99–102.

[16] D. Rumelhart and J. McCelland, *Parallel distributed processing*. MIT Press, 1986.

[17] G. Hinton, M. Welling, and A. Mnih, "Wormholes improve contrastive divergence," in *Proc. NIPS*, 2003, pp. 417–424.

[18] R. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," University of Tronto, Tech. Rep. CRG-TR-93-1, 1993.

[19] J. Yamagishi, H. Zen, Y.-J. Wu, T. Toda, and K. Tokuda, "The HTS-2008 system: Yet another evaluation of the speaker-adaptive HMM-based speech synthesis system in the 2008 Blizzard Challenge," in *Proc. Blizzard Challenge Workshop*, 2008.

[3] It is known that the contrastive divergence learning does not work well if the model distribution has multiple modes and these modes are separated by low-probability regions [17].