ACOUSTIC MODELING IN STATISTICAL PARAMETRIC SPEECH SYNTHESIS – FROM HMM TO LSTM-RNN

Heiga Zen

Google

ABSTRACT

Statistical parametric speech synthesis (SPSS) combines an acoustic model and a vocoder to render speech given a text. Typically decision tree-clustered context-dependent hidden Markov models (HMMs) are employed as the acoustic model, which represent a relationship between linguistic and acoustic features. Recently, artificial neural network-based acoustic models, such as deep neural networks, mixture density networks, and long short-term memory recurrent neural networks (LSTM-RNNs), showed significant improvements over the HMM-based approach. This paper reviews the progress of acoustic modeling in SPSS from the HMM to the LSTM-RNN.

***Index Terms*—** Statistical parametric speech synthesis; artificial neural networks; hidden Markov models; long short-term memory;

1. INTRODUCTION

The goal of text-to-speech (TTS) synthesis is to render a naturally sounding speech waveform given a text to be synthesized. Figure 1 outlines a human speech production process. A text (or concept) is first translated into movements of articulators and organs. Using air-flow from a lung, vocal source excitation signals containing periodic (by vocal cord vibration) and aperiodic (by turbulent noise) components are generated. By filtering the source signals by time-varying vocal tract transfer functions controlled by the articulators, their frequency characteristics are modulated. Finally, the filtered source signals are emitted. The aim of TTS is to mimic this process by computers in some way.

Text-to-speech can be viewed as a sequence-to-sequence mapping problem; from a sequence of discrete symbols (text) to a real-valued time series (waveform). Typical TTS systems consist of text analysis and speech synthesis parts. The text analysis part includes a number of natural language processing (NLP) steps, such as word segmentation, text normalization, part-of-speech (POS) tagging, and grapheme-to-phoneme (G2P) conversion. This part performs a mapping from a sequence of discrete symbols to another sequence of discrete symbols (*e.g.*, sequence of characters to sequence of words). The speech synthesis part performs mapping from a sequence of discrete symbols to real-valued time series. It includes prosody prediction and speech waveform generation. The former and latter parts are often called "front-end" and "back-end" in TTS, respectively. Although both of them are important to achieve high-quality TTS systems, this paper focuses on the latter one.

Statistical parametric speech synthesis (SPSS) [1] is one of the major approaches in the back-end part. This approach uses an acoustic model to represent the relationship between linguistic and acoustic features and a vocoder to render a speech waveform given acoustic features. This approach offers various advantages over concatenative speech synthesis [2], which is another major approach in the back-end part of TTS systems, such as small footprint [3,4] and flexibility to change its voice characteristics [5–8]. However, the naturalness of the synthesized speech from SPSS is not as good as that of the best samples from concatenative speech synthesizers. Zen *et al.* reported three major factors that can degrade the naturalness [1]; quality of vocoder, accuracy of acoustic model, and effect of over-smoothing. This paper addresses the accuracy of acoustic model.

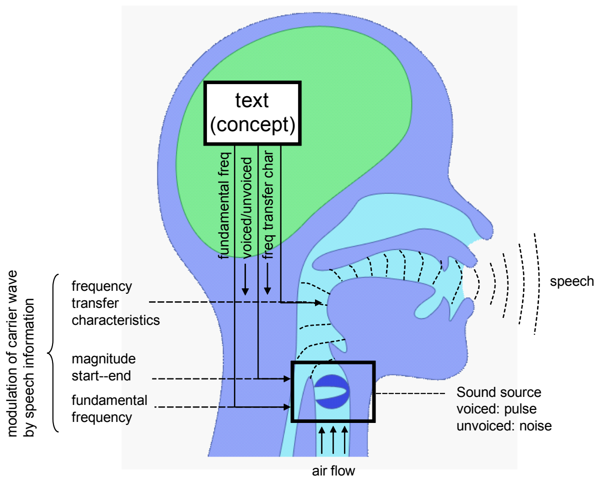
text (concept)
fundamental freq
voiced/unvoiced
freq transfer char
speech
modulation of carrier wave by speech information
frequency transfer characteristics
magnitude start--end
fundamental frequency
Sound source voiced: pulse unvoiced: noise
air flow

Fig. 1. Outline of speech production process.

Although there have been many attempts to develop a more accurate acoustic model for SPSS [9–19], the hidden Markov model (HMM) [20] is the most popular one. Statistical parametric speech synthesis with HMMs is known as HMM-based speech synthesis [9].

Inspired from the success in machine learning and automatic speech recognition, 5 different types of artificial neural network-based acoustic models were proposed in 2013 [21–25]. Zen *et al.* proposed an approach which uses a multi-layer artificial neural network to represent a mapping function from linguistic features to acoustic features [21]. It had significant impact to the research community and opened a research direction. Although this approach is relatively new, it has already exploded in popularity, *e.g.*, [21, 25–39]. Here, an artificial neural network is trained to learn a mapping function from linguistic features (input) to acoustic features (output) [21]. Artificial neural network-based acoustic models offer an efficient and distributed representation [40] of complex dependencies between linguistic and acoustic features [41] and have shown the potential to produce natural sounding synthesized speech [21, 26]. It was further extended to predict full conditional multimodal probability distribution of acoustic features rather than predicting only conditional single expected values [29]. Another significant extension is the use of recurrent neural networks (RNNs) [42],

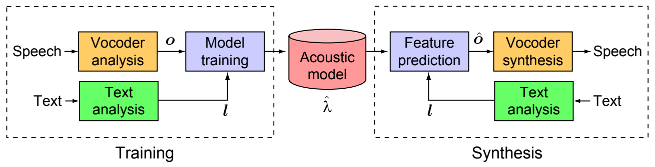

Fig. 2. Flowchart of a typical SPSS system.

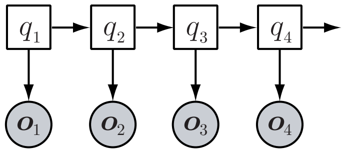

Fig. 3. Graphical model for an HMM. Circles and squares denote real-valued and discrete random variables, respectively. Likewise, clear means hidden variables, whereas shaded means observed ones.

especially long short-term memory RNNs (LSTM-RNNs) [43], to model the sequential nature of speech, which have correlations between consecutive frames. The state-of-the-art LSTM-RNN-based SPSS achieved significantly better subjective mean opinion scores (MOSs) than the HMM and feed-forward deep neural network-based approaches [38,44].

Note that use of neural networks in speech synthesis is not a new idea; in 1990s, there were a few papers about applications of neural networks to speech synthesis [45,46]. The main difference between the current and previous generations of neural network-based TTS systems include increase in the amount of data, having more layers, improvements of training algorithms, and existence of various algorithms used in HMM-based speech synthesis, such as high-quality vocoders (*e.g.*, STRAIGHT [47], Vocaine [48]) and oversmoothing compensation techniques (*e.g.*, global variance [49], modulation spectrum [50]).

This paper aims to provide a review of the progress of acoustic modeling in SPSS, starting from the HMM to the recent LSTM-RNN. The rest of this paper is organized as follows. Section 2 revisits HMM-based speech synthesis. Sections 3 and 4 summarize alternative acoustic models and context dependency modeling techniques, respectively. Section 5 describes neural network-based acoustic models. Concluding remarks are shown in the final section.

2. HMM-BASED SPEECH SYNTHESIS

Figure 2 illustrates a typical SPSS system which consists of training and synthesis stages. At the training stage, acoustic (real-valued) and linguistic (discrete) feature sequences are extracted from a speech waveform and its transcription, respectively. Then an acoustic model is trained to model the conditional distribution of an acoustic feature sequence given a linguistic feature sequence as

$$\hat{\Lambda} = \arg\max_{\Lambda} p(\boldsymbol{o} \mid \boldsymbol{l}, \Lambda), \tag{1}$$

where $\boldsymbol{o}$ is the acoustic feature sequence, $\boldsymbol{l}$ is the linguistic feature sequence, and Λ denotes the acoustic model.

At the synthesis stage, a text to be synthesized is first converted to the corresponding linguistic feature sequence. Then the most probable acoustic feature sequence for the linguistic feature sequence is predicted from the trained acoustic model as[1]

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \boldsymbol{l}, \hat{\Lambda}). \tag{2}$$

Finally, a speech waveform is rendered from the predicted acoustic feature sequence using a vocoder.

This section reviews acoustic modeling and acoustic feature prediction in HMM-based speech synthesis.

[1] Although random sampling from the trained acoustic model is also possible, samples from current acoustic models sound noisy and unnatural [39,51].

2.1. Modeling

The HMM [20] with single Gaussian state-output distributions uses an acoustic model of the form

$$p(\boldsymbol{o} \mid \boldsymbol{l}, \Lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) P(\boldsymbol{q} \mid \boldsymbol{l}, \Lambda) \tag{3}$$

$$= \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid q_t, \Lambda) P(q_t \mid q_{t-1}, \boldsymbol{l}, \Lambda) \tag{4}$$

$$= \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}) a_{q_t q_{t-1}} \tag{5}$$

where $\boldsymbol{o}_t$ is an acoustic feature vector at frame t, T is the number of frames, $\boldsymbol{q} = \{q_t, \ldots, q_T\}$ is a sequence of hidden discrete states, q_t is a hidden state at frame t, $\boldsymbol{\mu}_{q_t}$ and $\boldsymbol{\Sigma}_{q_t}$ correspond to the mean vector and covariance matrix associated with the state-output distribution at q_t, a_{ij} is the transition probability from state i to j, $a_{q_1 q_0}$ is the initial state probability of state q_1, $\boldsymbol{l} = \{l_1, \ldots, l_P\}$ is a sequence of linguistic features associated with $\boldsymbol{o}$, l_p is a linguistic feature vector associated with p-th phoneme, and Λ denotes a set of context-dependent phoneme HMMs. The parameters of HMMs can be estimated based on the maximum likelihood (ML) criterion by the expectation-maximization (EM) algorithm [20]. A graphical model [52] for the HMM is shown in Fig. 3. It can be seen from the figure that $\boldsymbol{o}_t$ depends only on q_t; statistics remain unchanged if the associated discrete state is the same.

It is well known that the acoustic features of a particular phone in human speech are not only determined by the individual phonetic content but also affected by various background events associated with the phone. The background events which can affect the acoustic realization of a phone are referred to as its *contexts*. There are normally around 50 different types of contexts used in SPSS [53]. The standard approach to handling contexts in HMM-based acoustic modeling is to use a distinct HMM for each individual combination of contexts, referred to as a context-dependent HMM. The amount of available training data is normally not sufficient for robustly estimating all context-dependent HMMs since there is rarely sufficient data to cover all of the context combinations required. To address these problems, top-down decision tree based context clustering [54] is widely used. Figure 4 illustrates this approach. The state-output distributions of the context-dependent HMMs are grouped into "clusters" and the distribution parameters within each cluster are shared. The assignment of HMMs to clusters is performed by examining the context combination of each HMM through a binary decision tree, where one context-related binary question is associated with each non-terminal node. The decision tree is constructed by sequentially selecting the questions which yield the largest log likelihood gain of the training data. With the use of context-related questions and state parameter sharing, the unseen contexts and data sparsity prob-

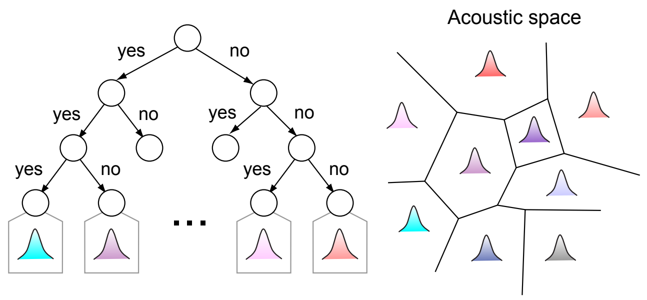

Fig. 4. Top-down decision tree-based clustering of HMM states.

lems are effectively addressed. As the approach has been successfully used in speech recognition, HMM-based statistical parametric speech synthesis naturally employs a similar approach to model very rich contexts.

2.2. Synthesis

The synthesis stage aims to find the most probable acoustic feature sequence $\hat{\boldsymbol{o}}$ given a linguistic feature sequence $\boldsymbol{l}$ and a set of trained context-dependent HMMs $\hat{\Lambda}$. Equation 2 can be approximated as

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \boldsymbol{l}, \hat{\Lambda}) \tag{6}$$

$$= \arg\max_{\boldsymbol{o}} \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \tag{7}$$

$$\approx \arg\max_{\boldsymbol{o},\boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \tag{8}$$

$$= \arg\max_{\boldsymbol{o},\boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \boldsymbol{l}, \hat{\Lambda}) P(\boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \tag{9}$$

$$\approx \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\Lambda}) \tag{10}$$

where $\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda})$.[2] If each HMM has the left-to-right topology and single Gaussian state-output distributions, the solution of Eq. (10) becomes as follows

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid \hat{q}_t, \hat{\Lambda}) \tag{11}$$

$$= \arg\max_{\boldsymbol{o}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{\hat{q}_t}, \boldsymbol{\Sigma}_{\hat{q}_t}) \tag{12}$$

$$= \arg\max_{\boldsymbol{o}} \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}) \tag{13}$$

$$= \boldsymbol{\mu}_{\hat{\boldsymbol{q}}} \tag{14}$$

where $\boldsymbol{\mu}_{\hat{q}_t}$ and $\boldsymbol{\Sigma}_{\hat{q}_t}$ are the mean vector and covariance matrix associated with $\hat{q}_t$, and $\boldsymbol{\mu}_{\hat{\boldsymbol{q}}} = [\boldsymbol{\mu}_{\hat{q}_1}^\top, \ldots, \boldsymbol{\mu}_{\hat{q}_T}^\top]^\top$ and $\boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}} = \mathrm{diag}[\boldsymbol{\Sigma}_{\hat{q}_1}, \ldots, \boldsymbol{\Sigma}_{\hat{q}_T}]$ are the mean vector and the covariance matrix over the entire utterance given $\boldsymbol{q}$. Figure 5 illustrates statistics of a series of left-to-right HMMs with single Gaussian state-output distributions given a state sequence. It can be seen from the figure that $\boldsymbol{\mu}_{\hat{\boldsymbol{q}}}$ becomes a step-wise sequence due to the use of discrete states and conditional independence assumptions. It is known that speech reconstructed from $\boldsymbol{\mu}_{\hat{\boldsymbol{q}}}$ has audible discontinuity at state boundaries [55].

[2] $\hat{\boldsymbol{q}}$ is typically determined by a set of state duration models.

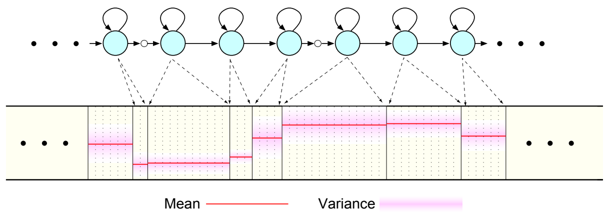

Fig. 5. Illustration of statistics of a series of left-to-right HMMs with single Gaussian state-output distribution given a state sequence.

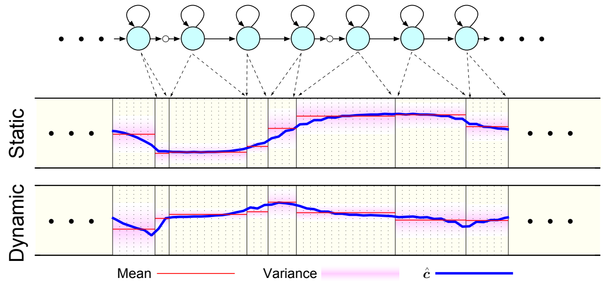

Fig. 6. Illustration of statistics of static and dynamic features of a series of left-to-right HMMs with single Gaussian state-output distribution given a state sequence.

To address this problem, Tokuda *et al.* introduced dynamic features [56, 57] as constraints and proposed the speech parameter generation algorithm [58]. Typically an observation vector $\boldsymbol{o}_t$ consists of static acoustic features $\boldsymbol{c}_t$ (*e.g.*, cepstrum) and dynamic acoustic features $\Delta\boldsymbol{c}_t$ (*e.g.*, delta cepstrum). The dynamic acoustic features are often computed as a linear combination of its neighbouring static acoustic features as

$$\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top]^\top, \tag{15}$$

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L}^{L} w_\tau \boldsymbol{c}_t, \tag{16}$$

where L is a window length and w_τ is a window coefficient. In this case, the relationship between observation vector sequence $\boldsymbol{o} = [\boldsymbol{o}_1^\top, \ldots, \boldsymbol{o}_T^\top]^\top$ and static acoustic feature vector sequence $\boldsymbol{c} = [\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_T^\top]^\top$ can be expressed in a matrix form as

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{17}$$

where $\boldsymbol{W}$ is a sparse window coefficient matrix that appends dynamic acoustic features to $\boldsymbol{c}$. The speech parameter generation algorithm introduces Eq. (17) as a constraint of Eq. (10) as

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}) \quad s.t. \quad \boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}. \tag{18}$$

This is equivalent to maximize w.r.t. $\boldsymbol{c}$ rather than $\boldsymbol{o}$ as

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}) \tag{19}$$

$$= \arg\max_{\boldsymbol{o}} \log \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}). \tag{20}$$

The partial derivative of the log output probability part in Eq.(20) w.r.t. $\boldsymbol{c}$ yields

$$\frac{\partial \log \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}})}{\partial \boldsymbol{c}} \propto \frac{\partial}{\partial \boldsymbol{c}} (\boldsymbol{W}\boldsymbol{c} - \boldsymbol{\mu}_{\hat{\boldsymbol{q}}})^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} (\boldsymbol{W}\boldsymbol{c} - \boldsymbol{\mu}_{\hat{\boldsymbol{q}}})$$
$$= \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{W}\boldsymbol{c} - \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{\mu}_{\hat{\boldsymbol{q}}} \tag{21}$$

By equating Eq. (21) to $\mathbf{0}$, a set of linear equations to determine $\hat{\boldsymbol{c}}$ is derived as

$$\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{W} \boldsymbol{c} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}. \tag{22}$$

Although the dimensionality of Eq. (22) can be tens of thousands, it can be solved efficiently with $\mathcal{O}(LT)$ operations by utilizing Cholesky decomposition and the sparse structure in $\boldsymbol{W}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ [58].

Figure 6 plots statistics of static and dynamic acoustic features and the most probable static acoustic features $\boldsymbol{c}$. It can be seen from the figure that $\hat{\boldsymbol{c}}$ becomes smooth and satisfies the statistics of both static and dynamic acoustic features. The determined static acoustic features are used with a vocoder to reconstruct a speech waveform given a text.

2.3. Characteristics of HMM-based acoustic modeling

The HMM has the following characteristics;

- Inconsistent; dynamic feature constraints are *not used* at the training stage, *i.e.*, Eq. (1) is used, whereas they are *used* at the synthesis stage [51], *i.e.*, Eq. (18) rather than Eq. (2) is used.
- Efficient clustering; efficient algorithm for decision tree-based clustering exists.
- Tractable training; efficient algorithm to marginalize over hidden variables exists thus training by the EM algorithm is possible.
- High latency; latency is $\mathcal{O}(T)$ as finding $\hat{\boldsymbol{c}}_1$ requires statistics at all frames in a utterance. With the time-recursive version of the speech parameter generation algorithm [59, 60], it reduces to $\mathcal{O}(D)$ where D is the number of frame lookahead (typically 5 for vocal tract acoustic features and 20 for vocal source acoustic features).
- Fast synthesis; computational cost to synthesize entire speech is relatively low.
- Hard to debug; In general, it is hard to find causes of problematic synthesis by SPSS. With the HMM-based acoustic model, first a problematic leaf node needs to be figured out, then statistics and data associated with the leaf node needs to be checked. On the other hand, concatenative TTS is easier to debug as it uses natural segments rather than statistics.

The code to test HMM-based SPSS is available online [61].

3. ALTERNATIVE ACOUSTIC MODELS

There have been a number of attempts to replace the HMM by an alternative acoustic model, such as trended HMMs [10], buried Markov models [11], trajectory HMMs [12], polynomial segment models [13], linear dynamical models (LDMs) [14, 18], product of experts (PoEs) [16], autoregressive HMMs (ARHMM) [15], Gaussian process regression [17], and hidden trajectory model [19]. Some of them can produce smoothly varying acoustic features without using dynamic feature constraints. Here a few of them are picked up and their details including definition, graphical models, and characteristics are described.

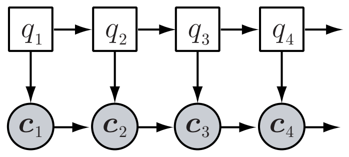

Fig. 7. Graphical model for a first-order ARHMM.

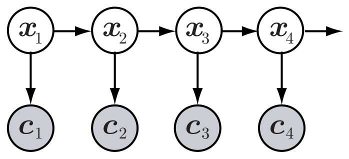

Fig. 8. Graphical model for a LDM.

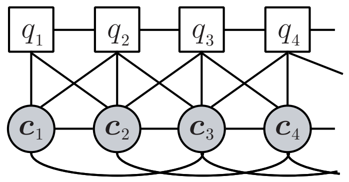

Fig. 9. Graphical model for a trajectory HMM with dynamic features computed from the current and ± 1 frames [16].

3.1. Autoregressive HMM

The autoregressive HMM [15] uses an acoustic model of the form

$$p(\boldsymbol{c} \mid \boldsymbol{l}, \Lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) P(\boldsymbol{q} \mid \boldsymbol{l}, \Lambda) \tag{23}$$

$$= \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} p(\boldsymbol{c}_t \mid \boldsymbol{c}_{t-1:t-K}, q_t, \Lambda) P(q_t \mid q_{t-1}, \boldsymbol{l}, \Lambda) \tag{24}$$

$$= \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{c}_t; \sum_{k=1}^{K} \boldsymbol{A}_{q_t}^{(k)} \boldsymbol{c}_{t-k} + \boldsymbol{b}_{q_t}, \boldsymbol{\Sigma}_{q_t}\right) a_{q_t q_{t-1}} \tag{25}$$

where $\boldsymbol{A}_{q_t}^{(k)}$ and $\boldsymbol{b}_{q_t}$ are the k-th order autoregressive coefficient matrix and a bias vector associated with q_t, respectively. A graphical model for the case $K = 1$ is shown in Fig. 7. It can be seen from the figure that although the ARHMM uses the discrete hidden states like the HMM it has additional dependency between adjacent frames. This dependency gives the ability to generating a smoothly varying acoustic feature sequence. The ARHMM has the following characteristics;

- Consistent; dynamic features are *not used* at both training and synthesis stages [51].
- Efficient clustering; efficient algorithm for decision tree-based clustering exists [62].
- Tractable training; efficient algorithm to marginalize over hidden states exists thus training by the EM algorithm is possible.
- Low latency; latency is $\mathcal{O}(1)$ as finding $\hat{\boldsymbol{c}}_1$ requires the statistics at the first frame only [15].

- Fast synthesis; computational cost to synthesize entire speech is lower than the HMM.
- Similar naturalness; subjective score was similar to the HMM [15].
- Hard to debug; In general, it is hard to find causes of problematic synthesis by SPSS. First a problematic leaf node needs to be figured out, then statistics of the ARHMM and data associated with the leaf node needs to be checked.

The code to test ARHMM-based SPSS is available online [63].

3.2. Linear Dynamical Model

The linear dynamical model [14, 18], which is also known as linear-Gaussian state-space model, uses an acoustic model of the form

$$p(\boldsymbol{c} \mid \boldsymbol{l}, \Lambda) = \int_{\boldsymbol{x}} p(\boldsymbol{c} \mid \boldsymbol{x}, \Lambda) p(\boldsymbol{x} \mid \boldsymbol{l}, \Lambda) d\boldsymbol{x} \tag{26}$$

$$= \int_{\boldsymbol{x}} \prod_{t=1}^{T} p(\boldsymbol{c}_t \mid \boldsymbol{x}_t, \Lambda) p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{l}, \Lambda) d\boldsymbol{x} \tag{27}$$

$$= \int_{\boldsymbol{x}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{c}_t; \boldsymbol{B}\boldsymbol{x}_t, \boldsymbol{Q}) \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{C}\boldsymbol{x}_{t-1}, \boldsymbol{R}) d\boldsymbol{x} \tag{28}$$

where $\boldsymbol{x}_t$ is a continuous hidden state vector, $\boldsymbol{B}$ and $\boldsymbol{C}$ are observation and evolution matrices, respectively, and $\boldsymbol{R}$ and $\boldsymbol{Q}$ are diagonal covariance matrices for $\boldsymbol{x}_t$ and $\boldsymbol{c}_t$, respectively. A graphical model is shown in Fig. 8. Unlike the HMM and ARHMM, it uses continuous hidden states rather than discrete ones. The use of continuous states gives the ability to generate smoothly varying acoustic features to the LDM. To have context-dependent modeling capability and finer time resolution, Tsiaras trained a LDM at each HMM state; state-level forced alignment was first performed with a set of trained HMMs then a LDM is trained at each HMM state given alignments.[3]

The LDM has the following characteristics;

- Consistent; dynamic features are *not used* at both training and synthesis stages.
- Clustering; although an algorithm for decision tree-based clustering exists, it requires running the EM algorithm at each split to evaluate the log likelihood [65].
- Tractable training; efficient algorithm to marginalize over hidden states exists thus training by the EM algorithm is possible [66].[4]
- Low latency; latency is $\mathcal{O}(1)$ as finding $\hat{c}_1$ requires the statistics at the first frame only [18].
- Fast synthesis; computational cost to synthesize entire speech is lower than the HMM.
- Similar naturalness; subjective score was similar to the HMM [65].
- Hard to debug; In general, it is hard to find causes of problematic synthesis by SPSS. First a problematic leaf node needs to be figured out, then statistics of the LDM and data associated with the leaf node needs to be checked.

[3] This can be viewed as a special case of the switching linear dynamical system [64] where state boundaries are known and fixed.

[4] Marginalizing over hidden variables becomes intractable once state boundaries are also treated as hidden variables [64].

3.3. Trajectory HMM

The trajectory HMM [12] was derived from the HMM by imposing explicit relationships between static and dynamic features. It uses an acoustic model of the form

$$p(\boldsymbol{c} \mid \boldsymbol{l}, \Lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) P(\boldsymbol{q} \mid \boldsymbol{l}, \Lambda) \tag{29}$$

$$= \sum_{\forall \boldsymbol{q}} \mathcal{N}(\boldsymbol{c}; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}) \prod_{t=1}^{T} a_{q_t q_{t-1}} \tag{30}$$

$$= \sum_{\forall \boldsymbol{q}} \frac{1}{Z_{\boldsymbol{q}}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}) a_{q_t q_{t-1}} \tag{31}$$

where $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ are the mean vector and the covariance matrix for $\boldsymbol{q}$, respectively. They are given as

$$\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \left(\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}, \tag{32}$$

$$\boldsymbol{P}_{\boldsymbol{q}} = \left(\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}\right)^{-1}. \tag{33}$$

Note that $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is equivalent to $\hat{\boldsymbol{c}}$ determined by solving Eq. (22), and the inverse of $\boldsymbol{P}_{\boldsymbol{q}}$ is the same as the coefficient matrix of Eq. (22). A graphical model is shown in Fig. 9. Unlike the HMM, ARHMM, and LDM, the trajectory HMM is represented as a undirected graphical model. The HMM, ARHMM, and LDM are locally normalized model, *i.e.*, the overall distribution $p(\boldsymbol{c} \mid \boldsymbol{q}, \lambda)$ is the product of the individual factors for each time t, each of which is normalized to be a valid probability distribution. On the other hand, the trajectory HMM is a globally normalized model, *i.e.*, first take the product of the unnormalized individual factors for each time t, then normalize [51]. Zen *et al.* also showed that the trajectory HMM could be formulated as a Gaussian Markov random field (GMRF) and a PoE [16] for sequences. The trajectory HMM has the following characteristics;

- Consistent; dynamic features are *used* at both the training and synthesis stages.
- Intractable clustering; no efficient decision tree-based clustering algorithm exists. Decision trees built for HMMs are often employed.
- Intractable training; no efficient algorithm to marginalize over hidden variables. A single state sequence [67] or Monte Carlo [68] approximation is typically used.
- High latency; latency is $\mathcal{O}(T)$ as finding the first frame of $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ requires statistics at all frames in a utterance.
- Fast synthesis; computational cost to synthesize entire speech is the same as the HMM.
- Better naturalness; better naturalness than the HMM and ARHMM [15].
- Hard to debug; In general, it is hard to find causes of problematic synthesis by SPSS. First a problematic leaf node needs to be figured out, then statistics of the trajectory HMM and data associated with the leaf node needs to be checked.

Note that minimum generation error (MGE) training of HMMs with squared loss [69] is equivalent to minimum mean squared error training of trajectory HMM.

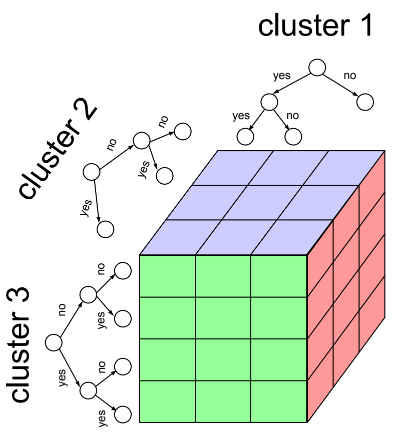

Fig. 10. Illustration of intersection three cluster-dependent decision trees [76, 78]. Here 36 distinct distributions can be composed from 10 leaf nodes.

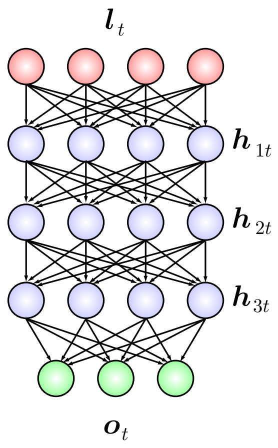

Fig. 11. A 3-layer DNN-based acoustic model. $\boldsymbol{h}_{ij}$ denotes activation at i-th layer at j-th frame.

4. ADVANCED CONTEXT DEPENDENCY MODELING

The acoustic models described above utilize top-down decision trees to capture context dependency. Although many different types of acoustic models are used, they are actually better interpreted as large *regression trees* which map a linguistic feature vector to the statistics (*e.g.*, mean and variance) of acoustic features [70].

As decision trees are the main regression model in these SPSS approaches, improving the performance of decision trees themselves is also important. There have also been attempts to improve the decision trees themselves, such as cross validation [71, 72], outlier detection [73], boosting [74], and tree intersection [75–79].

One example of tree intersection is cluster adaptive training (CAT) [80] with cluster-dependent decision trees [76, 78]. Unlike the standard CAT setup, where all clusters are assumed to share the same decision trees, this approach allows cluster-dependent decision trees. The mean vector of a single Gaussian state-output distribution associated with state i of a context-dependent HMM is represented an interpolation among its bases as

$$\boldsymbol{\mu}_i = \sum_{p=1}^{P} \lambda_p \boldsymbol{\xi}_{c(i,p)} \tag{34}$$

where P is the number of clusters, λ_p is a CAT interpolation weight for cluster p, $c(i, p)$ is a function returns the leaf node of a decision tree given cluster p and state i, and $\boldsymbol{\xi}_j$ is a cluster mean vector associated with leaf node j. Figure 10 illustrates intersection of three cluster-dependent decision trees. This framework allows the model to have the large number of distinct distributions from the small number of leaf nodes. This approach is computationally much more expensive than the standard approach due to the introduction of dependencies among trees. Iterative [75,78] and joint [76,79] tree building algorithms have been proposed.

5. ARTIFICIAL NEURAL NETWORK-BASED SPEECH SYNTHESIS

As mentioned in the previous section, the decision tree-clustered context-dependent acoustic models can be interpreted as large regression trees that map linguistic features to statistics of acoustic features. Zen *et al.* proposed an alternative scheme [21] that is based on a *deep* architecture [81]; the regression tree is replaced by a multi-layer artificial neural network.

The properties of the artificial neural network are contrasted with those of the decision tree as follows;

- Decision trees are inefficient to express complicated functions of input features, such as XOR, d-bit parity function, or multiplex problems [82]. To represent such cases, decision trees will be prohibitively large. On the other hand, they can be compactly represented by an artificial neural network [81].
- Decision trees rely on a partition of the input space and using a separate set of parameters for each region associated with a terminal node (a.k.a. local representation). This results in reduction of the amount of the data per region and poor generalization. Artificial neural networks offers distributed representation [40], which is more efficient than local one to model data with componential structure and provides better generalization as weights are trained from all training data. They also offer incorporation of high-dimensional, disparate features as inputs.
- The human speech production system is believed to have layered hierarchical structures in transforming the information from the linguistic level to the acoustic level [83]. The layered hierarchical structure in an artificial neural network will offer a better representation than models based on the shallow architectures (*e.g.*, regression trees).

5.1. DNN

Figure 11 illustrates feed-forward, multi-layer artificial deep neural network (DNN)-based acoustic modeling for SPSS [21], that directly converts an input linguistic feature vector to an output acoustic feature vector. In this approach, frame-level input linguistic features $\boldsymbol{l}_t$ rather than phoneme-level ones are used. They include binary answers to questions about linguistic contexts (*e.g.*, is-current-phoneme-vowel?), phoneme-level numeric values (*e.g.*, the number of words in the phrase, duration of the current phoneme), and frame-level features (*e.g.*, the relative position of the current frame in the current phoneme). The target acoustic feature vector $\boldsymbol{o}_t$ includes vocal tract and source parameters and their dynamic features. The

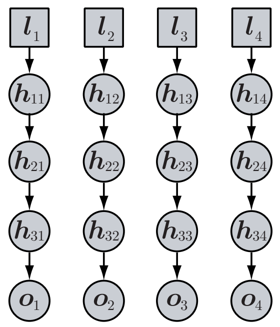

Fig. 12. Dependency graph of a 3-layer DNN.

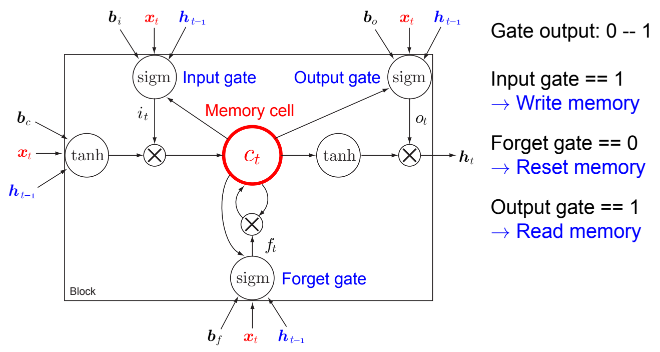

Fig. 13. Overview of a memory block in a LSTM-RNN.

weights of the DNN are trained using pairs of input and target features at each frame by back propagation. A dependency graph[5] of 3-layer DNN is shown in Fig. 12. It can be seen from the figure that there is no dependency between adjacent frames. Lack of the dependency results in discontinuity between adjacent frames. To address this problem, Zen *et al.* added dynamic acoustic features to outputs then used the speech parameter generation algorithm to generate the final smoothly varying static acoustic features [21]. Like the acoustic models in the previous section, the DNN has the following characteristics;

- Inconsistent; dynamic features are not used at the training stage but used at the synthesis stages.[6]
- No clustering; Mapping from a linguistic feature vector to an acoustic feature vector is embedded to network weights rather than a tree.
- Efficient training; can be trained by back propagation and stochastic gradient descent (SGD). Like the LDM and trajectory HMM, phoneme- or state-level boundaries are provided by a set of HMMs. These alignments are fixed while training the DNN.
- High latency; latency is $\mathcal{O}(T)$ as it uses the speech parameter generation algorithm like HMM-based approach.
- Slow synthesis; synthesizing entire utterance is computationally much more expensive than the HMM. With the HMM, finding statistics of acoustic features is done by traversing decision trees, whereas the DNN requires forward propagation, which includes matrix multiplication operations. Furthermore, this process needs to run at every *frame*, while it is required at every *state* with the HMM.
- Better naturalness; subjective score was better than the normal HMM [21].
- Weights in a DNN are harder to interpret than decision trees. However, visualizing the data and weights will be helpful.

The feed-forward DNN-based acoustic model was further extended to predict full conditional multimodal distribution of acoustic features rather than predicting only conditional single expected values using a mixture density output layer [29].

5.2. RNN

One limitation of the feed-forward DNN-based acoustic modeling is that the sequential nature of speech is ignored. Although certainly there are correlations between consecutive frames in speech data, the DNN-based approach assumes that each frame is independent. It is desirable to incorporate the sequential nature of speech data to the acoustic model itself. Recurrent neural networks [42] provide an elegant way to model speech-like sequential data that embodies correlations between neighboring frames. It can use all the available input features to predict output features at each frame [85]. Tuerk and Robinson [45] and Karaani *et al.* [46] applied simple RNNs to speech synthesis, whereas LSTM-RNN [43], which can capture long-term dependencies, were recently applied to acoustic modeling for SPSS [38, 44, 86].

The LSTM-RNN architecture is designed to model temporal sequences and their long-term dependencies [43]. It has special units called *memory blocks*. Figure 13 illustrates a memory block in a LSTM-RNN. A memory block contains a memory cell with self-connections storing the temporal state of the network in addition to 3 special multiplicative units called *gates* to control the flow of information. These gates act as a differentiable random access memory (RAM); accessing memory cell is guarded by "input", "output", and "forget" gates. This architecture allows LSTM-RNNs to keep information in their memory cells much longer than the simple RNNs.

Typically, feedback loops at hidden layers of an RNN are unidirectional; the input is processed from left to right, *i.e.*, the flow of the information is only forward direction. To use both past and future inputs for prediction, Schuster proposed the bidirectional RNN architecture [85]. It has forward and backward feedback loops that flow the information in both directions. This architecture enables the network to predict outputs using inputs of entire sequence. The bidirectional LSTM-RNNs (BLSTM-RNN) were also proposed [87]. Fan *et al.* and Fernandez *et al.* applied deep bidirectional LSTM-RNNs, which can access input features at both past and future frames, to acoustic modeling for SPSS and reported improved naturalness [44, 86]. Zen *et al.* applied the unidirectional LSTM-RNN (ULSTM-RNN), which can access input features up to the current frame, to achieve low-latency speech synthesis [38].

A dependency graph of 3-layer ULSTM-RNN with a recurrent output layer is shown in Fig. 14. It can be seen from the figure that it

[5] Noted that although this representation is similar to graphical models, a graphical model represents the conditional dependencies while the neural network representation shows the computational structure [84].

[6] Incorporating dynamic feature constraints into the training stage of DNN makes these stages consistent [35].

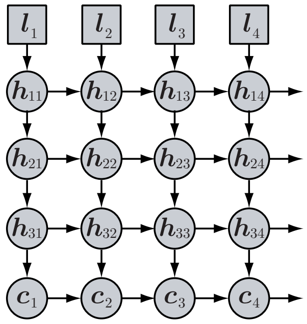
$$\begin{tikzcd}
l_1 \arrow[d] & l_2 \arrow[d] & l_3 \arrow[d] & l_4 \arrow[d] & \\
h_{11} \arrow[r] \arrow[d] & h_{12} \arrow[r] \arrow[d] & h_{13} \arrow[r] \arrow[d] & h_{14} \arrow[r] \arrow[d] & {} \\
h_{21} \arrow[r] \arrow[d] & h_{22} \arrow[r] \arrow[d] & h_{23} \arrow[r] \arrow[d] & h_{24} \arrow[r] \arrow[d] & {} \\
h_{31} \arrow[r] \arrow[d] & h_{32} \arrow[r] \arrow[d] & h_{33} \arrow[r] \arrow[d] & h_{34} \arrow[r] \arrow[d] & {} \\
c_1 \arrow[r] & c_2 \arrow[r] & c_3 \arrow[r] & c_4 \arrow[r] & {}
\end{tikzcd}$$

Fig. 14. Dependency graph of a 3-layer unidirectional LSTM-RNN with recurrent output layer.

has dependency between adjacent frames at both hidden and output layer levels. Zen *et al.* reported that having a recurrent output layer helped to produce more smoothly-varying acoustic features and improved the naturalness. The LSTM-RNN has the following characteristics;

- Consistent; dynamic features are *not used* at both training and synthesis stages.
- No clustering; Mapping from a linguistic feature vector *sequence* to an acoustic feature vector *sequence* is embedded to network weights rather than a tree.
- Efficient training; can be trained by back propagation through time [88] and SGD. Phoneme-level alignments provided by the HMM are fixed while training the LSTM-RNN.
- Low latency; an acoustic feature sequence predicted from an LSTM-RNN varies smoothly [38, 44]. Unidirectional LSTM-RNNs offer frame-synchronous generation without look-ahead.
- Slow synthesis; synthesizing entire utterance is computationally more expensive than the HMM but less than the DNN, as the network size can be smaller than the DNN and the smoothing step is not required.
- Better naturalness; subjective score was better than DNN [38].
- Weights in a LSTM-RNN are even harder to interpret than decision trees and a DNN. Visualizing the data and network weights is also harder due to its dynamic nature.

Table 1 summarizes the acoustic models for SPSS discussed in this paper. The ULSTM-RNN has several good properties as an acoustic model for SPSS; consistency between training and synthesis, compact model by distributed representation, low latency, and better naturalness. Although it uses fixed phoneme alignments at training, this limitation can be alleviated [89]. Some modern mobiles devices are fast enough to run the inference with the ULSTM-RNN. Further progress in visualization of neural networks will be helpful to debug problematic synthesis cases. Unidirectional LSTM-RNN-based SPSS has been used in several services from Google.

6. CONCLUSIONS

Statistical parametric speech synthesis combines vocoder and acoustic models to render a speech waveform given a text. Although SPSS offers various advantages over concatenative speech synthesis, such as flexibility to change its voice characteristics and small footprint, the naturalness of synthesized speech from SPSS is still not as good as the best samples from concatenative one. The accuracy of acoustic models is one of the factors that degrade the naturalness.

This paper reviewed the progress of acoustic models in SPSS from the acoustic trajectory and context modeling point of views. Although a number of different types of acoustic models have been applied to SPSS, the HMM has been the most popular one for the last two decades. However, recently proposed artificial neural network-based acoustic models look promising and have started replacing HMMs in SPSS.

One major reason why the HMM has been a dominant acoustic model in SPSS is the existence of open-source software to build end-to-end systems [61]. As there are a number of open-source software for deep learning [90–93], the author expects that artificial neural networks will be the next dominant acoustic model in the very near future.

7. REFERENCES

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

[2] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, pp. 373–376.

[3] Y. Morioka, S. Kataoka, H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura, "Minituarization of HMM-based speech synthesis," in *Proc. Autumn Meeting of ASJ*, 2004, pp. 325–326, (in Japanese).

[4] S.-J. Kim, J.-J. Kim, and M.-S. Hahn, "HMM-based Korean speech synthesis system for hand-held devices," *IEEE Trans. Consum. Electron.*, vol. 52, no. 4, pp. 1384–1390, 2006.

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Proc. Eurospeech*, 1997, pp. 2523–2526.

[6] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proc. ICASSP*, 2001, pp. 805–808.

[7] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. ICSLP*, 2002, pp. 1269–1272.

[8] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 9, pp. 1406–1413, 2007.

[9] 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.

[10] J. Dines and S. Sridharan, "Trainable speech synthesis with trended hidden Markov models," in *Proc. ICASSP*, 2001, pp. 833–837.

[11] I. Bulyko, M. Ostendorf, and J. Bilmes, "Robust splicing costs and efficient search with BMM models for concatenative speech synthesis," in *Proc. ICASSP*, 2002, pp. 461–464.

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

[13] J.W. Sun, F. Ding, and Y.H. Wu, "Polynomial segment model based statistical parametric speech synthesis system," in *Proc. ICASSP*, 2009, pp. 4021–4024.

[14] C. Quillen, "Kalman filter based speech synthesis," in *Proc. ICASSP*, 2010, pp. 4618–4621.

[15] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statistical parametric speech synthesis," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 21, no. 3, pp. 587–597, 2013.

Table 1. Summary of various acoustic models for SPSS.

	Consistent	Representation	Training	Latency	Speed	Quality	Debug
HMM [9]	No	Local	**EM**	High	**Fast**	Baseline	Hard
ARHMM [15]	**Yes**	Local	**EM**	**Low**	**Fast**	Similar	Hard
LDM [65]	**Yes**	Local	**EM**	**Low**	**Fast**	Similar	Hard
Trajectory HMM [12]	**Yes**	Local	Viterbi	High	**Fast**	**Better**	Hard
DNN [21]	No	**Distributed**	BP	High	Slow	**Better**	Harder
ULSTM-RNN [38]	**Yes**	**Distributed**	BPTT	**Low**	Slow	**Better**	Harder
BLSTM-RNN [44]	**Yes**	**Distributed**	BPTT	High	Slow	**Better**	Harder

[16] H. Zen, M. Gales, Y. Nankaku, and K. Tokuda, "Product of experts for statistical parametric speech synthesis," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 3, pp. 794–805, 2012.

[17] T. Koriyama, T. Nose, and T. Kobayashi, "Statistical parametric speech synthesis based on Gaussian process regression," *IEEE Journal of Selected Topics in Signal Process.*, vol. 8, no. 2, pp. 173–183, 2014.

[18] V. Tsiaras, R. Maia, V. Diakoloukas, Y. Stylianou, and V. Digalakis, "Linear dynamical models in speech synthesis," in *Proc. ICASSP*, 2014, pp. 300–304.

[19] M.-Q. Cai, Z.-H. Ling, and L.-R. Dai, "Statistical parametric speech synthesis using a hidden trajectory model," *Speech Commun.*, vol. 72, pp. 149–159, 2015.

[20] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[21] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[22] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 21, no. 10, pp. 2129–2139, 2013.

[23] S.-Y. Kang, X.-J. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. ICASSP*, 2013, pp. 8012–8016.

[24] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "f_0 contour prediction with a deep belief network-Gaussian process hybrid model," in *Proc. ICASSP*, 2013, pp. 6885–6889.

[25] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in *Proc. ISCA SSW8*, 2013, pp. 281–285.

[26] Y. Qian, Y. Fan, W. Hu, and F. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *Proc. ICASSP*, 2014, pp. 3857–3861.

[27] T. Raitio, H. Lu, J. Kane, A. Suni, M. Vainio, S. King, and P. Alku, "Voice source modelling using deep neural networks for statistical parametric speech synthesis," in *Proc. EUSIPCO*, 2014, pp. 2290–2294.

[28] X. Yin, M. Lei, Y. Qian, F. Soong, L. He, Z.-H. Ling, and L.-R. Dai, "Modeling dct parameterized f0 trajectory at intonation phrase level with dnn or decision tree," in *Proc. Interspeech*, 2014, pp. 2273–2277.

[29] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, 2014, pp. 3872–3876.

[30] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia, "Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning," in *Proc. Interspeech*, 2015.

[31] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," in *Proc. Interspeech*, 2015.

[32] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Proc. Interspeech*, 2015.

[33] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, 2015, pp. 4460–4464.

[34] Y. Fan, Y. Qian, F. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, 2015, pp. 4475–4479.

[35] F.-L. Xie, Y. Qian, Y. Fan, F. Soong, and H. Li, "Sequence error (SE) minimization training of neural network for voice conversion," in *Proc. Interspeech*, 2014, pp. 2283–2287.

[36] Z. Chen and K. Yu, "An investigation of implementation and performance analysis of DNN based speech synthesis system," in *Proc. ICSP*, 2014, pp. 577–582.

[37] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, 2015, pp. 4455–4459.

[38] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, 2015, pp. 4470–4474.

[39] B. Uria, I. Murray, S. Renals, and C. Valentini-Botinhao, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-RNADE," 2015.

[40] G. Hinton, J. McClelland, and D. Rumelhart, "Distributed representation," in *Parallel distributed processing: Explorations in the microstructure of cognition*, D. Rumelhart, J. McClelland, and the PDP Research Group, Eds. MIT Press, 1986.

[41] H. Zen, "Deep learning in speech synthesis," in *Keynote speech given at ISCA SSW8*, 2013, `http://research.google.com/pubs/archive/41539.pdf`.

[42] A. Robinson and F. Fallside, "Static and dynamic error propagation networks with application to speech coding," in *Proc. NIPS*, 1988, pp. 632–641.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] Y. Fan, Y. Qian, and F. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[45] C. Tuerk and T. Robinson, "Speech synthesis using artificial neural networks trained on cepstral coefficients," in *Proc. Eurospeech*, 1993, pp. 1713–1716.

[46] O. Karaali, G. Corrigan, I. Gerson, and N. Massey, "Text-to-speech conversion with neural networks: A recurrent TDNN approach," in *Proc. Eurospeech*, 1997, pp. 561–564.

[47] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: possible role of a repetitive structure in sounds," *Speech Commn.*, vol. 27, pp. 187–207, 1999.

[48] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications is speech synthesis," in *Proc. ICASSP*, 2015, pp. 4230–4234.

[49] 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.

[50] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A postfilter to modify the modulation spectrum in HMM-based speech synthesis," in *Proc. ICASSP*, 2014, pp. 290–294.

[51] M. Shannon, H. Zen, and W. Byrne, "The effect of using normalized models in statistical parametric speech synthesis," in *Proc. Interspeech*, 2011, pp. 121–124.

[52] F. Koller, *Probabilistic Graphical Models*, MIT Press, 2009.

[53] K. Tokuda, H. Zen, and A. Black, "An HMM-based speech synthesis system applied to English," in *Proc. IEEE Speech Synthesis Workshop*, 2002, CD-ROM Proceeding.

[54] J. Odell, *The use of context in large vocabulary speech recognition*, Ph.D. thesis, Cambridge University, 1995.

[55] K. Tokuda, T. Masuko, Y. Yamada, T. Kobayashi, and S. Imai, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," in *Proc. Eurospeech*, 1995, pp. 757–760.

[56] S. Sagayama and F. Itakura, "On individuality in a dynamic measure of speech," in *Proc. of Spring Meeting of ASJ*, 1979, pp. 589–590, (in Japanese).

[57] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, pp. 52–59, 1986.

[58] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[59] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of dynamic features," in *Proc. ICSP*, 1997, pp. 247–252.

[60] T. Muramatsu, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory," in *Proc. Interspeech*, 2008, pp. 1076–1079.

[61] K. Tokuda, K. Oura, K. Hashimoto, S. Shiota, H. Zen, J. Yamagishi, T. Toda, T. Nose, S. Sako, and A. Black, "The HMM-based speech synthesis software toolkit," `http://hts.sp.nitech.ac.jp/`, As of 20th August 2015.

[62] M. Shannon and W. Byrne, "Autoregressive clustering for HMM speech synthesis," in *Proc. Interspeech*, 2010, pp. 829–832.

[63] M. Shannon, "armspeech," `https://github.com/MattShannon/armspeech`.

[64] A. Rosti and M. Gales, "Switching linear dynamical systems for speech recognition," Tech. Rep. CUED/F-INFENG/TR.461, Cambridge University, 2003.

[65] V. Tsiaras, R. Maia, V. Diakoloukas, Y. Stylianou, and V. Digalakis, "Towards a linear dynamical model based speech synthesizer," in *Proc. Interspeech*, 2015.

[66] S. Roweis and Z Ghahramani, "Learning nonlinear dynamical systems using the expectation–maximization algorithm," *Kalman filtering and neural networks*, p. 175, 2001.

[67] H. Zen, K. Tokuda, and T. Kitamura, "A Viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features," in *Proc. ICASSP*, 2004, pp. 837–840.

[68] 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.

[69] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. ICASSP*, 2006, pp. 89–92.

[70] S. King, "A reading list of recent advances in speech synthesis," in *Proc. ICPhS*, 2015.

[71] Y. Zhang, Z.-J. Yan, and F. Soong, "Cross-validation based decision tree clustering for HMM-based TTS," in *Proc. ICASSP*, 2010, pp. 4602–4605.

[72] H. Zen and M. Gales, "Decision tree-based context clustering based on cross validation and hierarchical priors," in *Proc. ICASSP*, 2011, pp. 4560–4563.

[73] K.-H. Oh, J.-S. Sung, D.-H. Hong, and N.-S. Kim, "Decision tree-based clustering with outlier detection for HMM-based speech synthesis," in *Interspeech*, 2011, pp. 101–104.

[74] Y. Qian, H. Liang, and F. Soong, "Generating natural F0 trajectory with additive trees," in *Proc. Interspeech*, 2008, pp. 2126–2129.

[75] K. Saino, *A clustering technique for factor analysis-based eigenvoice models*, Master thesis, Nagoya Institute of Technology, 2008, (in Japanese).

[76] H. Zen and N. Braunschweiler, "Context-dependent additive log F0 model for HMM-based speech synthesis," in *Interspeech*, 2009, pp. 2091–2094.

[77] K. Yu, H. Zen, F. Mairesse, and S. Young, "Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis," *Speech Commn.*, vol. 53, no. 6, pp. 914–923, 2011.

[78] H. Zen, N. Braunschweiler, S. Buchholz, M. Gales, K. Knill, S. Krstulović, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1713–1724, 2012.

[79] S. Takaki, Y. Nankaku, and K. Tokuda, "Contextual additive structure for hmm-based speech synthesis," *IEEE Journal of Selected Topics in Signal Process.*, vol. 8, no. 2, pp. 229–238, 2014.

[80] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, pp. 417–428, 2000.

[81] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[82] S. Esmeir and S. Markovitch, "Anytime learning of decision trees," *J. Mach. Learn. Res.*, vol. 8, pp. 891–933, 2007.

[83] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Process. Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[84] H. Valpola, *Bayesian Ensemble Learning for Nonlinear Factor Analysis*, Ph.D. thesis, Helsinki University of Technology, 2000.

[85] M. Schuster, *On supervised learning from sequential data with applications for speech recognition*, Ph.D. thesis, Nara Institute of Science and Technology, 1999.

[86] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks," in *Proc. Interspeech*, 2014.

[87] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. ICDAR*, 2007, pp. 367–371.

[88] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[89] A. Graves, "Generating sequences with recurrent neural networks," *arXiv:1308.0850*, 2014.

[90] "Theano," `http://deeplearning.net/software/theano/`.

[91] "Torch," `http://torch.ch/`.

[92] "Caffe," `http://caffe.berkeleyvision.org/`.

[93] "Chainer," `http://chainer.org/`.