Principles for learning controllable TTS from annotated and latent variation

Citation for published version:
Henter, G, Lorenzo-Trueba, J, Wang, X & Yamagishi, J 2017, Principles for learning controllable TTS from annotated and latent variation. in Proceedings Interspeech 2017. pp. 3956-3960, Interspeech 2017, Stockholm, Sweden, 20/08/17. https://doi.org/10.21437/Interspeech.2017-171

Digital Object Identifier (DOI):
10.21437/Interspeech.2017-171

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings Interspeech 2017

Principles for learning controllable TTS from annotated and latent variation

Gustav Eje Henter[1], *Jaime Lorenzo-Trueba*[1], *Xin Wang*[1], *Junichi Yamagishi*[1,2]

[1]Digital Content and Media Sciences Research Division, National Institute of Informatics, Japan
[2]The Centre for Speech Technology Research, the University of Edinburgh, UK
{gustav,jaime,wangxin,jyamagis}@nii.ac.jp

Abstract

For building flexible and appealing high-quality speech synthesisers, it is desirable to be able to accommodate and reproduce fine variations in vocal expression present in natural speech. Synthesisers can enable control over such output properties by adding adjustable control parameters in parallel to their text input. If not annotated in training data, the values of these control inputs can be optimised jointly with the model parameters. We describe how this established method can be seen as approximate maximum likelihood and MAP inference in a latent variable model. This puts previous ideas of (learned) synthesiser inputs such as sentence-level control vectors on a more solid theoretical footing. We furthermore extend the method by restricting the latent variables to orthogonal subspaces via a sparse prior. This enables us to learn dimensions of variation present also within classes in coarsely annotated speech. As an example, we train an LSTM-based TTS system to learn nuances in emotional expression from a speech database annotated with seven different acted emotions. Listening tests show that our proposal successfully can synthesise speech with discernible differences in expression within each emotion, without compromising the recognisability of synthesised emotions compared to an identical system without learned nuances.

Index Terms: text-to-speech, latent variables, paralinguistics

1. Introduction

Natural human speech contains vast amounts of acoustic variation that cannot be predicted from the spoken text alone, cf. [1]. Sources of meaningful variability include speaker identity and speaker state (emotion etc.), adjustments to enhance communication with the listener (speaking style, prosody, emphasis, entrainment etc.), as well as the circumstances under which the communication takes place (channel properties such as ambient noise). It has been hypothesised that an ability to replicate natural speech variability is a requirement for speech synthesisers that are more pleasant to listen to and interact with, cf. [2].

Unfortunately, the majority of text-to-speech systems do not treat acoustic variation in a structured manner, and instead model any deviations from the average behaviour as uncorrelated noise. As a result, output speech is inappropriately averaged and oversmooth, while sampled speech sounds bubbly or noisy (cf. [1, 3]). To minimise the negative effects of untreated acoustic diversity, most work focusses on training synthesisers only on consistently-read speech from a single speaker in a quiet studio environment. While it might seem that the best approach would be to annotate and learn from meaningful variation in speech databases, such annotation is usually too costly to consider, especially for subtle speech properties such as the strength of emotional expression. Nonetheless, several recent studies involving multi-speaker synthesis [4, 5, 6] have shown that acoustic diversity (at least with proper labelling) can be turned from a burden into a benefit, producing synthesisers that are both controllable *and* yield greater output quality than systems trained only on single-speaker subsets of the data.

This paper proposes a method for learning nuances in speech expression also from controlled, studio-recorded data where variation within expressions is neither annotated nor deliberately included, extending several prior approaches [7, 8, 6] for learning unannotated variation. Our main contributions are:

1. A reinterpretation of previously published methods as approximate maximum likelihood combined with MAP estimation of latent variables. (Sec. 3)
2. A straightforward method for learning unannotated nuances in different output expressions, based on constraining the latent variables using a sparse prior. (Sec. 4.3)

Our experiments in Sec. 5 train state-of-the-art RNN-based statistical parametric speech synthesisers with and without our proposed enhancement on a database with seven classes of acted emotion. Listening test results show that our proposal learns to control nuances within each annotated emotional class, without compromising the recognisability of the synthesised emotions.

2. Relation to prior work

Some of the first examples of general-purpose output control in statistical speech synthesis were based on so-called multiple-regression HMMs (MR-HMMs) [9] in classical HMM-based speech synthesis with regression trees [10, 11], with applications to controlling synthesiser speaking style [12] and manipulating its articulation [13]. It is more or less trivial to extend the idea of MR-HMMs also to speech synthesis based on deep neural networks (DNNs). This has enabled easy multi-speaker synthesis with one-hot or i-vector inputs (speaker codes) [14, 6, 5, 15, 16], as well as manipulation of speech aspects such as speaker age and gender [6].

If the training corpus has unannotated variation, a simple idea is to try to learn the values of control parameters along with the synthesiser itself, looking for the input values that most improve the predictive accuracy of speech acoustics and/or durations. Joint optimisation of network weights and unknown control inputs is easy to do in deep learning using backpropagation, and the basic principle has been rediscovered several times: Under the name *discriminant condition codes* (DCC), it has been applied to adapt both speech recognition [7] and speech synthesis [6] to new speakers. In Watts et al. [8], the same mathematical setup is referred to as *learned control vectors*.

Among previous publications, Watts et al. is perhaps the most similar to our current work. They used an intentionally diverse and expressive speech corpus (children's audiobooks) to learn a controllable speech synthesiser. We herein show that this feat is possible using only natural variation present in carefully acted speech, and that it can be combined with a coarse,

The work presented in this paper was partially supported by the TOYOTA motor cooperation.

annotated variation (per-sentence acted emotion class).

Equally important, we show that the methods in [7, 8, 6] have substantial connections to a statistical concept known as latent variables. Latent variables are common ingredient in statistical models for speech and beyond, with the states (subphones) of hidden Markov models (HMMs) [17] being one prominent example of a successful previous application of latent variables in speech technology. Latent variables are a good match for our situation, since they explicitly address the issue of handling unobserved, structured variation in data. Our insight thus not only grounds and interprets established heuristics through probability theory, but also unlocks generalisations and improvements of the basic method, as illustrated in Sec. 4.

3. Theory

This section presents a latent-variable interpretation of several learned-input methods in speech technology. To begin with, let $\boldsymbol{X}$ be a random variable whose distribution $f_{\underline{\boldsymbol{X}}\,|\,\boldsymbol{Z}}$ depends on another random variable $\boldsymbol{Z}$. In our case, the observed output(s) $\boldsymbol{x}$ would be the speech produced by a synthesiser in response to a control signal $\boldsymbol{z}$. When the input value $\boldsymbol{z}$ is unobserved or unannotated we call $\boldsymbol{Z}$ a *latent variable*.

Let us model the joint behaviour of the two random variables $\boldsymbol{X}$ and $\boldsymbol{Z}$ with a parametric family of distributions

$$f_{\boldsymbol{X},\,\boldsymbol{Z}}(\boldsymbol{x},\,\boldsymbol{z};\,\boldsymbol{\theta}) = f_{\boldsymbol{X}\,|\,\boldsymbol{Z}}\left(\boldsymbol{x}\,|\,\boldsymbol{z};\,\boldsymbol{\theta}\right) f_{\boldsymbol{Z}}(\boldsymbol{z};\,\boldsymbol{\theta}), \tag{1}$$

with parameter(s) $\boldsymbol{\theta}$. A common approach in speech technology, e.g., [7, 8, 6], is to jointly estimate the parameters (neural network weights) $\boldsymbol{\theta}$ and the unknown control-vector inputs $\boldsymbol{z}$ by maximising the log-probability of the observed data $\boldsymbol{x}$, as

$$\left(\widehat{\boldsymbol{\theta}}(\boldsymbol{x}),\,\widehat{\boldsymbol{z}}(\boldsymbol{x})\right) = \underset{(\boldsymbol{\theta},\,\boldsymbol{z})}{\operatorname{argmax}} \ln f_{\boldsymbol{X}\,|\,\boldsymbol{Z}}\left(\boldsymbol{x}\,|\,\boldsymbol{z};\,\boldsymbol{\theta}\right). \tag{2}$$

We will now show that Eq. (2) can be seen as approximate maximum-likelihood (ML) estimation of $\boldsymbol{\theta}$ under a flat prior on $\boldsymbol{z}$, with the $\boldsymbol{z}$-values interpretable as maximum a-posteriori (MAP) point estimates. As always, the log-likelihood $\mathcal{L}$ and the maximum-likelihood parameter estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ are given by

$$\mathcal{L}\left(\boldsymbol{\theta}\,|\,\boldsymbol{x}\right) = \ln f_{\boldsymbol{X}}(\boldsymbol{x};\,\boldsymbol{\theta}) \tag{3}$$

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(\boldsymbol{x}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\,\mathcal{L}\left(\boldsymbol{\theta}\,|\,\boldsymbol{x}\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln f_{\boldsymbol{X}}(\boldsymbol{x};\,\boldsymbol{\theta}) \tag{4}$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln \int f_{\boldsymbol{X},\,\boldsymbol{Z}}(\boldsymbol{x},\,\boldsymbol{z};\,\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{z}. \tag{5}$$

We also define the MAP point estimate $\widehat{\boldsymbol{z}}_{\mathrm{MAP}}$ of the latent variable $\boldsymbol{Z}$ given $\boldsymbol{\theta}$ through

$$\widehat{\boldsymbol{z}}_{\mathrm{MAP}}(\boldsymbol{x};\,\boldsymbol{\theta}) = \underset{\boldsymbol{z}}{\operatorname{argmax}} \ln f_{\boldsymbol{Z}\,|\,\boldsymbol{X}}\left(\boldsymbol{z}\,|\,\boldsymbol{x};\,\boldsymbol{\theta}\right) \tag{6}$$

$$= \underset{\boldsymbol{z}}{\operatorname{argmax}} \ln \left(f_{\boldsymbol{X}\,|\,\boldsymbol{Z}}\left(\boldsymbol{x}\,|\,\boldsymbol{z};\,\boldsymbol{\theta}\right) f_{\boldsymbol{Z}}(\boldsymbol{z};\,\boldsymbol{\theta})\right). \tag{7}$$

Now define the *auxiliary function*

$$Q\left(\boldsymbol{\theta};\,\boldsymbol{\theta}_0\right) = \mathbb{E}_{\boldsymbol{Z}}\left(\ln f_{\boldsymbol{X},\,\boldsymbol{Z}}(\boldsymbol{x},\,\boldsymbol{Z};\,\boldsymbol{\theta})\,\middle|\,\boldsymbol{X} = \boldsymbol{x};\,\boldsymbol{\theta}_0\right) \tag{8}$$

$$= \int f_{\boldsymbol{Z}\,|\,\boldsymbol{X}}\left(\boldsymbol{z}\,|\,\boldsymbol{x};\,\boldsymbol{\theta}_0\right) \ln f_{\boldsymbol{X},\,\boldsymbol{Z}}(\boldsymbol{x},\,\boldsymbol{z};\,\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{z} \tag{9}$$

$$\approx \ln f_{\boldsymbol{X},\,\boldsymbol{Z}}(\boldsymbol{x},\,\widehat{\boldsymbol{z}}_{\mathrm{MAP}}(\boldsymbol{x};\,\boldsymbol{\theta}_0);\,\boldsymbol{\theta}), \tag{10}$$

where the approximation follows from assuming that the posterior distribution $f_{\boldsymbol{Z}\,|\,\boldsymbol{X}}$ is sharply peaked, forming a spike around its most likely value $\widehat{\boldsymbol{z}}_{\mathrm{MAP}}$. Using Jensen's inequality, it is easy to prove (cf. [18]) that any parameter value $\boldsymbol{\theta}$ that increases the value of Q above $Q\left(\boldsymbol{\theta}_0;\,\boldsymbol{\theta}_0\right)$ also must have greater log-likelihood $\mathcal{L}$ than $\boldsymbol{\theta}_0$. This is the basis of the EM algorithm.

By iteratively optimising Q under the previous best $\boldsymbol{\theta}$-value – a process that includes updating $\widehat{\boldsymbol{z}}_{\mathrm{MAP}}$ – a local stationary point (typically maximum) of the log-likelihood $\mathcal{L}$ is identified. Since Eqs. (7) (E-step) and (10) (M-step) update two different arguments of the same maximisation objective, namely $\ln f_{\boldsymbol{X},\,\boldsymbol{Z}} = \ln(f_{\boldsymbol{X}\,|\,\boldsymbol{Z}}\,f_{\boldsymbol{Z}})$, the fixed points of the iterative updates are the stationary points of $f_{\boldsymbol{X},\,\boldsymbol{Z}}$. If we assume a flat (uniform) prior $f_{\boldsymbol{Z}}$ over the region of relevant $\widehat{\boldsymbol{z}}_{\mathrm{MAP}}$-values, we have that $f_{\boldsymbol{X},\,\boldsymbol{Z}} = f_{\boldsymbol{X}\,|\,\boldsymbol{Z}}$. Thus choosing $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{z}}$ to jointly maximise the original objective Eq. (2) is the same as approximate maximum-likelihood parameter estimation with $\widehat{\boldsymbol{z}}$ playing the role of $\widehat{\boldsymbol{z}}_{\mathrm{MAP}}$, which proves the desired result.

Intuitively, we can think of learned control vectors in synthesis literature as "poor man's latent variables", that permit latent variation but do not account for uncertainty in the posterior value of $\boldsymbol{z}$. To our knowledge, this interpretation is not widely known among speech technologists.

4. Application to controllable synthesis

4.1. Preliminary definitions

Text-to-speech technology maps a sequential text representation $\underline{\boldsymbol{l}}$ to a sequential audio representation $\underline{\boldsymbol{x}}$ of that text being spoken out loud. In the special case of *statistical parametric speech synthesis* (SPSS) the speech representation is a sequence $\underline{\boldsymbol{x}} = [\boldsymbol{x}_t,\,\ldots,\,\boldsymbol{x}_T]$ of speech parameters (and possibly their Δ and Δ^2-values) $\boldsymbol{x}_t$ for controlling a speech-waveform generator (vocoder), while in *signal-level speech synthesis* t may index waveform sample values x_t. In both cases, *acoustic modelling* is the task of specifying and fitting a probabilistic model $f_{\underline{\boldsymbol{X}}}\left(\underline{\boldsymbol{x}}\,|\,\underline{\boldsymbol{l}}\right)$, where the representation $\underline{\boldsymbol{l}}$ typically is a sequence of phone identities and other so-called linguistic features derived from the text input, as opposed to a raw text string. Commonly, forced alignment is used to upsample $\underline{\boldsymbol{l}}$ so that each output vector $\boldsymbol{x}_t$ is associated with an input vector $\boldsymbol{l}_t$.

4.2. Controllable speech synthesis

For a controllable speech synthesiser, the text message $\underline{\boldsymbol{l}}$ is augmented with control parameters $(\boldsymbol{c},\,\boldsymbol{z})$, which we have separated into annotated values, $\boldsymbol{c}$, and unannotated variation, $\boldsymbol{z}$. In DNN-based TTS, each $\boldsymbol{l}_t$ vector is usually concatenated with one or both of the corresponding $\boldsymbol{c}$ and $\boldsymbol{z}$-values, allowing these extra inputs to directly influence the generated speech. The maximum time-resolution of the control parameters is thus constrained by the granularity of $\underline{\boldsymbol{x}}$, though in our experiments we let the control parameters be constant for each utterance and train a standard RNN-based acoustic regression model

$$\underline{\boldsymbol{X}}(\underline{\boldsymbol{l}},\,\boldsymbol{c},\,\boldsymbol{z};\,\boldsymbol{\theta}) = \underline{\boldsymbol{\mu}}(\underline{\boldsymbol{l}},\,\boldsymbol{c},\,\boldsymbol{z};\,\boldsymbol{\theta}) + \underline{\boldsymbol{\varepsilon}}, \tag{11}$$

where $\underline{\boldsymbol{\varepsilon}}$ is a zero-mean white Gaussian noise process and $\underline{\boldsymbol{\mu}}$ is an LSTM-based deep, recurrent neural network. The training data comprises a set of speech utterances $\underline{\boldsymbol{x}}^{(n)}$ with matching text $\underline{\boldsymbol{l}}^{(n)}$ and annotated labels $\boldsymbol{c}^{(n)}$. Based on the general principle from Sec. 3, model parameters (network weights) and latent control vectors can be simultaneously estimated through

$$\left(\widehat{\boldsymbol{\theta}}_{\mathrm{ML}},\,\{\widehat{\boldsymbol{z}}_{\mathrm{MAP}}\}\right) \approx \underset{(\boldsymbol{\theta},\,\{\boldsymbol{z}^{(n)}\})}{\operatorname{argmax}} \sum_{n} \big(\ln f_{\boldsymbol{Z}}(\boldsymbol{z}^{(n)}\,|\,\underline{\boldsymbol{l}}^{(n)},\,\boldsymbol{c}^{(n)};\,\boldsymbol{\theta})$$

$$+ \ln f_{\underline{\boldsymbol{X}}\,|\,\boldsymbol{Z}}(\underline{\boldsymbol{x}}^{(n)}\,|\,\underline{\boldsymbol{l}}^{(n)},\,\boldsymbol{c}^{(n)},\,\boldsymbol{z}^{(n)};\,\boldsymbol{\theta})\big). \tag{12}$$

Note that both the prior $f_{\boldsymbol{Z}}$ and the acoustic model $f_{\underline{\boldsymbol{X}}\,|\,\boldsymbol{Z}}$ may depend on $\underline{\boldsymbol{l}}$ and $\boldsymbol{c}$. Since we have assumed a Gaussian model, log-likelihood maximisation is the same as minimising the conventional mean-squared error (MSE).

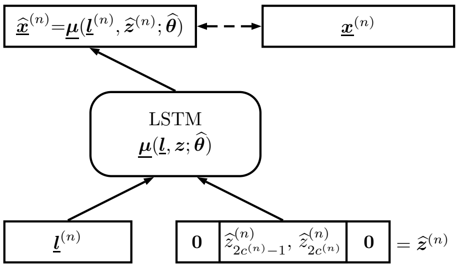
$\widehat{\underline{\boldsymbol{x}}}^{(n)}=\underline{\boldsymbol{\mu}}(\underline{\boldsymbol{l}}^{(n)}, \widehat{\boldsymbol{z}}^{(n)}; \widehat{\boldsymbol{\theta}})$
$\underline{\boldsymbol{x}}^{(n)}$
LSTM
$\underline{\boldsymbol{\mu}}(\underline{\boldsymbol{l}}, \boldsymbol{z}; \widehat{\boldsymbol{\theta}})$
$\underline{\boldsymbol{l}}^{(n)}$
$\mathbf{0}$
$\widehat{z}^{(n)}_{2c^{(n)}-1}, \widehat{z}^{(n)}_{2c^{(n)}}$
$\mathbf{0}$
$= \widehat{\boldsymbol{z}}^{(n)}$

Figure 1: *Proposed setup with $D = 2$ for an arbitrary training example n. Only D, contiguous elements of $\boldsymbol{z}$ are non-zero; the current emotion $c^{(n)}$ determines which ones. Hats denote estimated quantities ($\widehat{\boldsymbol{\theta}}$, $\widehat{\boldsymbol{z}}^{(n)}$) or predicted quantities ($\widehat{\underline{\boldsymbol{x}}}^{(n)}$). Learning is based on backpropagating the residual $\underline{\boldsymbol{x}}^{(n)} - \widehat{\underline{\boldsymbol{x}}}^{(n)}$.*

4.3. Learning emotional nuances

Essentially, learned control vectors $\widehat{\boldsymbol{z}}$ estimate the "extra bits" that, given the text and the annotated features, increase acoustic prediction accuracy the most. Control vectors different from $\widehat{\boldsymbol{z}}_{\text{MAP}}$ – even ones based on explicit labellings – should not be able to produce as high prediction accuracy on the training set. However, we have no direct influence on the nature of the variation learned; instead, the controllable synthesiser is indirectly defined by the data distribution, parametrisation, and model structure. Nonetheless, it is possible to tweak the implementation specifics of $\boldsymbol{c}$ and $\boldsymbol{z}$ in order to obtain a controllable synthesiser that is more likely to meet certain design goals.

In this paper, we wish to learn a synthesiser that can manipulate nuances in its emotional expression while ensuring that the specified (discrete) emotion expressed remains as constant and unambiguous as possible. To do this, we propose partitioning the elements of $\boldsymbol{z} \in \mathbb{R}^{E \times D}$ into E sets of D values each; one set for each emotion $c \in \{1, \ldots, E\}$. During training, only the D elements that correspond to the current emotion c are updated and learned, with the remainder held constant at zero, i.e., $z_i(c) = 0$ whenever $\lceil i/D \rceil \neq c$. This constrains the learned control vectors for emotional nuances to lie on axis-aligned orthogonal subspaces, one for each emotion. This way, we can identify and learn differences between each utterance, while ensuring that different input emotions remain separable throughout training. Fig. 1 illustrates our proposed setup.

An elegant and principled interpretation of our subspace-based proposal is to view it as the effect of a sparse prior in Eq. (12), e.g., the flat (improper) prior $f_{\boldsymbol{Z}}(\boldsymbol{z} \mid c; \boldsymbol{\theta}) = \prod_{i: \lceil i/D \rceil \neq c} I(z_i = 0)$, with $I(\cdot)$ being a binary indicator function. The prior thus provides a straightforward way to influence the learned control vectors. This illustrates how the connection between control vectors and latent variables is helpful for designing new ways of influencing speech synthesiser output.

Optimising (12) is a non-convex problem, and the resulting estimates are likely to depend on initialisation. Several previous studies in, e.g., multi-speaker synthesis [5, 6], have considered one-hot vectors $\boldsymbol{c}$ to encode available labels for input to a DNN, and as a starting point for optimisation [6]. Extending this to the case $D > 1$, we propose to initialise $\boldsymbol{z}$ using binary vectors that are constant and nonzero only on the elements matching the current emotion c. (We further normalised these to unit length in our experiments.) Thus all sentences are assumed to be similar in nuance unless there is evidence to the contrary.

In this setup, $\boldsymbol{c}$ is essentially redundant when given $\boldsymbol{z}$, since the zeroes of $\boldsymbol{z}$ uniquely determine the intended emotion for all training-data utterances. For this reason, we did not explicitly provide $\boldsymbol{c}$ as an input to our synthesiser with emotional nuance.

4.4. Synthesising emotional nuances

It should be noted that the latent-variable approach does not predict which $\boldsymbol{z}$-vectors that are appropriate for synthesising utterances outside the training set. A simple method to get a clear and consistent emotional expression is to synthesise from the centroid (mean) latent vector $\overline{\boldsymbol{z}}_c$ over each emotion c. For a more variable synthesis, while we have no explicit model of the posterior for $\boldsymbol{z}$, we can draw approximate samples from this posterior (assuming it does not depend on $\underline{\boldsymbol{l}}$) by taking random elements from the set of learned control vectors $\{\widehat{\boldsymbol{z}}\}$. This might offer a better approach than, e.g., sampling from a Gaussian approximation. This insight is another advantage of the connection to latent variables. Our sampling scheme also confines the control vectors to latent-space regions where training data is available, and modelling accuracy thus should be high.

5. Experiments

5.1. Setup

We evaluated our proposal on a Japanese-language database, detailed in [19], containing 8400 acted emotional-speech utterances from a professional voice actress. A list of the emotions and amounts of data can be found in Tab. 1. For the experiments the database was partitioned into training, validation, and test sets containing 80%, 10%, and 10% of the data, respectively.

For each phone in the database, 390 linguistic features (mean and variance normalised) were extracted using Open JTalk [20]. Acoustic features were extracted with a 5 ms time-resolution and consisted of a voiced/unvoiced flag, interpolated $\log F_0$, 25 band aperiodicities, 60 mel-cepstrum coefficients, and their Δ and Δ^2 for a total dimensionality of 259. The WORLD vocoder [21] was used, with MLPG [22] and postfiltering used for synthesis. Phone boundaries were estimated by forced alignment with an HSMM trained using HTS [23] on the training set, with HTS-predicted durations used for synthesis.

Two very similar neural-network acoustic models were trained on the data, differing only in their use (or not) of learned emotional nuance. Both models consisted of two feedforward layers with 512 nodes per layer followed by two bidirectional recurrent layers with 256 LSTM units [24] per layer and were implemented using the CURRENNT toolkit [25]. The difference is that the baseline network ("Ba") also took a one-hot vector encoding of the prompted emotion c as input, while the proposed network ("P") instead used learned 14-dimensional vectors of emotional nuances following Sec. 4.3, i.e., each emotion used a $D = 2$-dimensional latent space, similar to [8].

Networks weights were trained to minimise mean-squared prediction error on the training data using stochastic gradient descent for 40 epochs with a learning rate of 10^{-5}. A different learning rate of $2 \cdot 10^{-3}$ was used for the latent vectors $\widehat{\boldsymbol{z}}^{(n)}$ in P, since a given $\boldsymbol{z}$-vector is updated less often than a given weight. The ratio between the learning rates was tuned by looking at the training evolution of the fraction of intra-emotion variance to total variance of latent vectors, choosing a learning rate that gave smooth, monotonic convergence in 40 epochs. The limiting ratio of variances seemed to be about 0.25.

5.2. Evaluation of emotion recognition

Our first experiment compared the expressive accuracy of the two synthesisers against natural speech. Specifically, we performed an emotional classification test, in which listeners listen to speech stimuli one at a time and classify them into one of eight different categories (the seven emotions in Tab. 1 plus an

Table 1: *Database and results. Durations include silences. Italicised p-values are not statistically significant at level* $\alpha = 0.05$.

	Database		**Rec. test: % corr.**			**Rec. test: p-vals.**		**ABX test: % corr.**		**ABX test: p-vals.**	
Emotion	**Utts.**	**Dur. (min)**	**N**	**Ba**	**P**	**N vs. P**	**Ba vs. P**	**Rand**	**Far**	**Rand**	**Far**
Neutral	1200	147	88	69	86	*1.000*	*0.345*	74	70	$<10^{-3}$	0.003
Happy	1200	133	95	85	88	*1.000*	*1.000*	66	91	0.027	$<10^{-12}$
Calm	1200	158	71	63	46	*0.057*	*0.576*	61	90	*0.149*	$<10^{-11}$
Excited	1200	154	32	28	18	*0.576*	*1.000*	57	83	*0.364*	$<10^{-7}$
Sad	1200	141	93	72	70	0.045	*1.000*	80	100	$<10^{-5}$	≈ 0
Insecure	1200	136	71	61	59	*0.863*	*1.000*	57	88	*0.364*	$<10^{-10}$
Angry	1200	148	91	91	93	*1.000*	*1.000*	80	99	$<10^{-5}$	$<10^{-19}$
All	8400	1017	77	67	66	0.004	*1.000*	68	89	$<10^{-13}$	$<10^{-71}$

"other" option). The test was carried out over a web interface using native listeners crowdsourced through CrowdWorks$^{\text{LTD}}$.

The evaluation was separated into batches of 14 utterances each, two for each emotion in random order. Only batches where the listener scored all 14 utterances were included in the analysis. To limit the overall influence of any individual listener, no listener was allowed to classify more than five batches. In total, 75 listeners provided 1162 ratings.

The listening test contained three types of speech: synthetic speech from Ba and P, plus natural recordings (N) as a top line. Speech from P always used the mean control input $\overline{z}_c$ described in Sec. 4.4, to achieve the most standard emotional expression. All systems spoke the same sentences from the test set.

Listeners' recognition rates for the various speech types can be seen in Tab. 1, together with p-values for differences against P.[1] While synthetic emotions were about 10% more difficult to recognise than natural ones, there is no evidence that our proposal reduced recognition rates compared to the baseline.

5.3. Evaluation of emotional nuance control

Next, we performed an ABX test to confirm that the learned control space is useful for generating perceptible differences in emotional nuance. In this test, listeners heard three versions (A, B, X) of the same sentence and emotion generated by system P. Stimuli A and B used distinct emotional control vectors, while X used the same vector as either A or B. Listeners were asked to identify which of A and B that had the most similar emotional expression to X. 18 listeners provided 950 total responses, using a crowdsourced, batched, balanced setup like that in Sec. 5.2.

The pairs of emotional vectors used for the stimuli A and B in the test were selected from the set of learned emotional latent vectors $\{\widehat{z}\}$ for a given emotion c in two ways: either just as random (distinct) pairs ("Rand"), or by a random selection only from the 100 pairs of learned vectors that were the most distant in Mahalanobis distance ("Far").[2] This enables us to quantify the distinguishability both of random pairs of nuances and over entire latent subspaces, by seeing how often listeners correctly picked out the stimulus among A or B that was identical to X.

The results of the ABX test are shown in the two final sections of Tab. 1. The p-values are two-sided and have been corrected for multiple comparisons like before. It is clear that random emotional nuances (Rand) generally can be distinguished better than chance, and that distinguishability increases with greater separation between nuance vectors (Far).[3] This means our approach has successfully learned to express nuances within emotions. Different emotions also show different patterns. Since humans had difficulties classifying excited speech recordings, it is perhaps not surprising that they only were moderately successful in separating nuances in excited synthetic speech.

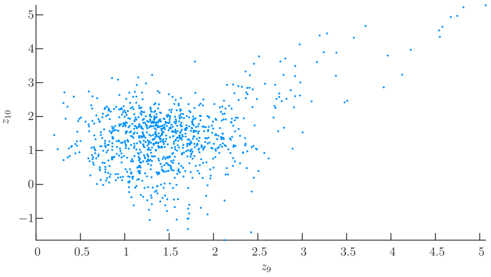

Figure 2: *Learned control vectors in Sad emotional subspace. All other point clouds are close to Gaussian in appearance.*

Unique to Sad speech, the set of learned control vectors exhibited a long, scattered tail (likely the consequence of a local optimum), as shown in Fig. 2. z-vector pairs for the Far condition consistently included a vector from the tip of this tail, where synthesis quality was noticeably poorer due to the paucity of data. This explains the ceiling rate of correct response for Sad speech in the Far condition in Tab. 1.

Informal listening suggests that the most perceptually salient effect of altering the control vector input might be that the emotional strength of the generated speech changes. This echoes the findings in [8]. Audio examples can be found at homepages.inf.ed.ac.uk/ghenter/.

6. Conclusions

We have described how the properties of latent, unannotated variation can be learned and recreated in speech synthesis. We presented a new argument interpreting established heuristics through latent-variable theory, and extended the approach to consider synthesis with mixed observed and unobserved control inputs. Furthermore, we detailed a novel method for learning nuances within speech expressions as subspaces in latent-variable space, interpretable as the actions of a sparse prior. Experiments confirm that the resulting synthesiser allows accurate generation of speech emotions while also permitting nuances within the emotional expression to be adjusted. For future work it is interesting to consider more advanced statistical methods for handling latent variables in neural networks, e.g., [28].

[1] Each system and emotion had between 54 and 56 responses. Significances were computed using Barnard's test [26], except for All, where the chi-squared approximation is accurate. A Holm-Bonferroni correction [27] for multiple comparisons clipped most p-values at 1.

[2] Full-covariance Mahalanobis distance was used since the latent space has scale, rotation, and translation ambiguities.

[3] Neutral speech is an exception, but its difference between Rand and Far is only a few percent and is in itself not statistically significant.

7. References

[1] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, “Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech,” in *Proc. Interspeech*, 2014, pp. 1504–1508.

[2] S. King and V. Karaiskos, “The Blizzard Challenge 2013,” in *Proc. Blizzard Chall. Workshop*, 2013.

[3] K. Tokuda and H. Zen, “Directly modeling voiced and unvoiced components in speech waveforms by neural networks,” in *Proc. ICASSP*, 2016, pp. 5640–5644.

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

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv preprint 1609.03499*, 2016.

[6] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, “Adapting and controlling DNN-based speech synthesis using input codes,” in *Proc. ICASSP*, 2017, pp. 4905–4909.

[7] S. Xue, O. Abdel-Hamid, H. Jiang, L.-R. Dai, and Q. Liu, “Fast adaptation of deep neural network based on discriminant codes for speech recognition,” *IEEE/ACM T. Audio Speech*, vol. 22, no. 12, pp. 1713–1725, 2014.

[8] O. Watts, Z. Wu, and S. King, “Sentence-level control vectors for deep neural network speech synthesis,” in *Proc. Interspeech*, 2015, pp. 2217–2221.

[9] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama, “Multiple-regression hidden Markov model,” in *Proc. ICASSP*, 2001, pp. 513–516.

[10] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[11] S. King, “An introduction to statistical parametric speech synthesis,” *Sadhana*, vol. 36, no. 5, pp. 837–852, 2011.

[12] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, “A style control technique for HMM-based expressive speech synthesis,” *IEICE T. Inf. Syst.*, vol. 90, no. 9, pp. 1406–1413, 2007.

[13] Z.-H. Ling, K. Richmond, and J. Yamagishi, “Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression,” *IEEE T. Audio Speech*, vol. 21, no. 1, pp. 207–219, 2013.

[14] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, “A study of speaker adaptation for DNN-based speech synthesis,” in *Proc. Interspeech*, 2015, pp. 879–883.

[15] B. Potard, P. Motlicek, and D. Imseng, “Preliminary work on speaker adaptation for DNN-based speech synthesis,” Idiap Research Institute, Martigny, Switzerland, Tech. Rep. Idiap-RR-02-2015, 2015.

[16] N. Hojo, Y. Ijima, and H. Mizuno, “An investigation of DNN-based speech synthesis using speaker codes,” in *Proc. Interspeech*, 2016, pp. 2278–2282.

[17] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Stat. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[19] J. Lorenzo-Trueba, C. Valentini-Botinhao, G. E. Henter, and J. Yamagishi, “Misperceptions of the emotional content of natural and vocoded speech in a car,” *submitted to Interspeech 2017*.

[20] K. Oura, S. Sako, and K. Tokuda, “Japanese text-to-speech synthesis system: Open JTalk,” in *Proc. ASJ Spring*, 2010, pp. 343–344.

[21] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE T. Inf. Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.

[22] 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.

[23] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, “The HMM-based speech synthesis system (HTS) version 2.0,” in *Proc. SSW*, 2007, pp. 294–299.

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

[25] F. Weninger, J. Bergmann, and B. W. Schuller, “Introducing CURRENNT: the Munich open-source CUDA recurrent neural network toolkit,” *J. Mach. Learn. Res.*, vol. 16, no. 3, pp. 547–551, 2015.

[26] G. A. Barnard, “Significance tests for 2×2 tables,” *Biometrika*, vol. 34, no. 1, pp. 123–138, 1947.

[27] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scand. J. Stat.*, vol. 6, no. 2, pp. 65–70, 1979.

[28] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proc. ICLR*, 2014.