DEEP GENERATIVE FACTORIZATION FOR SPEECH SIGNAL

Haoran Sun, Lantian Li, Yunqi Cai, Yang Zhang, Thomas Fang Zheng, Dong Wang

Center for Speech and Language Technologies, Tsinghua University

ABSTRACT

Various information factors are blended in speech signals, which forms the primary difficulty for most speech information processing tasks. An intuitive idea is to factorize speech signal into individual information factors (e.g., phonetic content and speaker trait), though it turns out to be highly challenging. This paper presents a speech factorization approach based on a novel factorial discriminative normalization flow model (factorial DNF). Experiments conducted on a two-factor case that involves phonetic content and speaker trait demonstrates that the proposed factorial DNF has powerful capability to factorize speech signals and outperforms several comparative models in terms of information representation and manipulation.

Index Terms— deep generative model, speech factorization

1. INTRODUCTION

Speech signal is highly complex and involves variations from multiple sources [1]. Decomposing speech signal into elementary components is perhaps the most important idea in the history of speech processing research.

Fourier transform is such a decomposition, which decomposes speech signals into a set of simple periodical functions, each with a single frequency. This Fourier decomposition, also known as spectrum analysis [2], reveals the frequency-dependent energy distribution, a fundamental property behind the complex vibration of the waveform. This decomposition, however, does not employ any knowledge on the speciality of speech signals, and so cannot offer deep understanding for speech. The source-filter model solves this problem by formulating speech production as an excitation and modulation process [3]. Inversely, speech signal can be decomposed into excitation and modulation according to the production model. This excitation-modulation decomposition 'was the basis of practically all the work on speech signal processing that followed' [4].

In spite of the predominant importance, the excitation-modulation decomposition plays a role on low-level signals and thus, is not directly related to high-level information such as phonetic contents and speaker traits. In fact, both the excitation and modulation components derived from the excitation-modulation decomposition are irrelated to speaker traits, which means that this decomposition is less useful for identifying speaker identity. For the sake of speech information processing, we hope to decompose speech signals into high-level information factors.

In this paper, we will present such a speech information decomposition approach. We will formally define this decomposition as a nonlinear extension of factorization analysis, and propose an implementation based on the normalization flow model.

2. DEEP GENERATIVE FACTORIZATION

Fujisaki [5] is perhaps the first scholar mentioning the information-based production/decomposition. In his proposal, speech signal can be regarded as a composition of three types of information factors: linguistic factors that determine what to speak, paralinguistic factors that determine how to speak, and non-linguistic factors that determine the residual properties that are unrelated to the speech content.

Although the concept is clear, it is not easy to turn Fujisaki's proposal to a computational model, due to the complex and unknown convolution amongst information factors. In fact, the concept of *information factor* per se is not easy to define. Fortunately, the recent development on deep generative models offers a renewed possibility. Basically, we shall represent each elementary variation c (e.g., phonetic content and speaker trait) in speech signal as *a set of random variables*, denoted by v_c. We will call v_c the information factor corresponding to variation c. By this definition, the distribution of speech signal can be modeled by a *deep composition* of the distributions among the information factors. Formally, it will take the following form:

$$x = G(v_{c_1}, v_{c_2}, ...), \tag{1}$$

where G is a deep generative model, and each information factor v_{c_i} is a standard multivariate Gaussian. Note that different information factors are independent. As a simplest case, we assume speech signal can be decomposed into a phonetic content factor v_q, a speaker factor v_s, and a Gaussian noise ϵ, following the linear form shown below:

$$x = M_q v_q + M_s v_s + D\epsilon. \tag{2}$$

where D is a diagonal matrix. This can be reformulated to the following simple form:

$$x = \begin{bmatrix} M_q \; M_s \end{bmatrix} \begin{bmatrix} v_q \\ v_s \end{bmatrix} + D\epsilon, \tag{3}$$

Note that this is just the standard factorization analysis [6], where v_q and v_s form the factors. Therefore, the deep generation model in Eq.(1) can be regarded as a nonlinear extension of factorization analysis. For this reason, the decomposition according to deep generative model in Eq.(1) can be called **deep generative factorization**.

3. DEEP FACTORIZATION BASED ON DNF

3.1. Revisit NF and DNF

Normalization flow (NF) [7, 8, 9] is a popular deep generative model. The basic idea of NF is to transform a normal distribution $p(\boldsymbol{z})$ to match the data by an *invertible* function f. Due to the invertibility, the latent variable $\boldsymbol{z}$ can be obtained exactly by $\boldsymbol{z} = f^{-1}(\boldsymbol{x})$. This means that an extra inference network is not necessary, and we can compute the likelihood function $p(\boldsymbol{x})$ exactly.

Suppose the latent variable $\boldsymbol{z}$ and the observation $\boldsymbol{x}$ are linked by an invertible transform f_θ, then $p(\boldsymbol{z})$ and $p(\boldsymbol{x})$ hold the following relation [10]:

$$\log p(\boldsymbol{x}) = \log p(\boldsymbol{z}) + \log \left| \det(\frac{\mathrm{d}{f_\theta}^{-1}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}}) \right|, \tag{4}$$

where $\det(\cdot)$ represents determinant of the Jacobian matrix of f^{-1}. In the above equation, the two terms on the right hand side are often called the *prior term* and the *entropy term*, respectively. The NF model is trained by maximizing the likelihood of the training data with respect to the parameter θ, and the log likelihood function can be computed as follows:

$$\mathcal{L}(\theta) = \sum_i \log p(\boldsymbol{z}_i) + \sum_i \log \left| \det(\frac{\mathrm{d}{f_\theta}^{-1}(\boldsymbol{x}_i)}{\mathrm{d}\boldsymbol{x}_i}) \right|. \tag{5}$$

This function can be maximized by any numerical optimization method, for instance stochastic gradient descend (SGD).

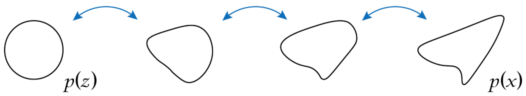


Fig. 1. Distribution transform with normalization flow.

In practice, the invertible function f_θ is often implemented as a sequence of simple invertible functions, which transforms a simple distribution on $\boldsymbol{z}$ *gradually* to a complex distribution on $\boldsymbol{x}$, as shown in Figure 1. It is shown that if f is powerful enough, any complex distribution can be obtained from a simple Gaussian [11].

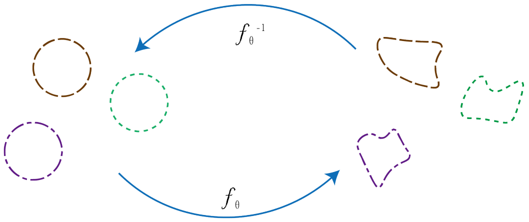


Fig. 2. The DNF architecture, where each class has its individual prior distribution.

Discriminative normalization flow (DNF) [12] is an extension of the NF model. As shown in Figure 2, different classes share the same NF network, but the prior distributions are different. Following the notations of NF, and assuming the prior of class y to be $\mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{I})$, the probability of an observation $\boldsymbol{x}$ is:

$$\log p(\boldsymbol{x}) = \log \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{y(\boldsymbol{x})}, \boldsymbol{I}) + \log J(\boldsymbol{x}), \tag{6}$$

where $\log J(\boldsymbol{x})$ denotes the entropy term, and $y(\boldsymbol{x})$ denotes the class label of $\boldsymbol{x}$. Similar to NF, DNF can be trained by maximum likelihood.

3.2. Factorial DNF

The DNF model can encode a single information factor (corresponding to the class label). However, our purpose is to represent multiple information factors in a full generative model. To accommodate this request, we split the latent code into several *partial codes*, with each partial code corresponding to a particular information factor. This model is denoted by **factorial DNF**.

Taking the case of two information factors as an example, we split the latent code into two partial codes, $\boldsymbol{z}^A$ and $\boldsymbol{z}^B$, corresponding to the information factor A and B respectively. Since the prior distribution is a diagonal Gaussian, $\boldsymbol{z}^A$ and $\boldsymbol{z}^B$ are naturally independent, which means:

$$p(\boldsymbol{z}) = p(\boldsymbol{z}^A)p(\boldsymbol{z}^B). \tag{7}$$

We assume that the prior distributions for $\boldsymbol{z}^A$ and $\boldsymbol{z}^B$ depend on the labels corresponding to the information factors A and B, respectively. More precisely,

$$p(\boldsymbol{z}^A) = \mathcal{N}(\boldsymbol{z}^A; \boldsymbol{\mu}_{y_A(\boldsymbol{z})}, \boldsymbol{I})$$
$$p(\boldsymbol{z}^B) = \mathcal{N}(\boldsymbol{z}^B; \boldsymbol{\mu}_{y_B(\boldsymbol{z})}, \boldsymbol{I})$$

where $y_A(\boldsymbol{z})$ and $y_B(\boldsymbol{z})$ are the class labels of $\boldsymbol{z}$ for factor A and B respectively. The likelihood $p(\boldsymbol{x})$ can therefore be written by:

$$\log p(\boldsymbol{x}) = \log p(\boldsymbol{z}^A) + \log p(\boldsymbol{z}^B) + \log J(\boldsymbol{x}). \tag{8}$$

Once the model has been well trained, an observation $\boldsymbol{x}$ can be encoded to $\boldsymbol{z} = [\boldsymbol{z}^A \ \boldsymbol{z}^B]$ by the invertible transform f^{-1}, and the partial codes $\boldsymbol{z}^A$ and $\boldsymbol{z}^B$ encode the information factors A and B, respectively.

3.3. Speech factorization by factorial DNF

Factorial DNF can be used to implement the deep generative factorization shown in Eq.(1). We will describe the approach with a two-factor case that involves a phone factor and a speaker factor.

Firstly, speech signals are split into short segments and all the segments are labeled by phone and speaker classes, denoted by Q and S respectively. During training, treat each short segment as an observation. Select the partial means $\boldsymbol{\mu}_q$ and $\boldsymbol{\mu}_s$ for the partial codes $\boldsymbol{z}^Q$ and $\boldsymbol{z}^S$ according to its phone class q and speaker class s. The prior for the latent variable $\boldsymbol{z}$ is then formed to be a Gaussian where the mean vector is $[\boldsymbol{\mu}_q \ \boldsymbol{\mu}_s]$. Note that the likelihood function $p(\boldsymbol{x})$ can be computed, and so the model can be trained without principle difficulties.

By this setting, we have formulated the variation in speech signal into three parts: (1) the randomness on phone class means $\boldsymbol{\mu}_q$; (2) the randomness on speaker class means $\boldsymbol{\mu}_s$; (3) the residual randomness, reflected by the Gaussian distribution on $\boldsymbol{z}$ conditioned on $[\boldsymbol{\mu}_q \ \boldsymbol{\mu}_s]$.

We highlight that since the NF transform is invertible, all the variation/information in the original speech is retained in the code $\boldsymbol{z}$. However, factorial DNF conducts an interesting 'variation reshaping' that makes variation corresponding to different information factors uncorrelated and confined in their own dimensions. This factorization holds two significant advantages: (1) The inference is as simple as an inverse transform $f^{-1}(\boldsymbol{x})$, which is different from conventional shallow factorization models such as JFA that requires complex Bayesian inference. (2) The code corresponding to an information factor can be obtained easily by selecting its individual dimensions, and changing codes corresponding to different information factors will not impact each other.

4. RELATED WORK

Speech factorization has been studied in speaker recognition for a long time. The famous Gaussian mixture model-Universal background model (GMM-UBM) is an early example [13], which firstly represents the short-term phonetic factor by a discrete random variable represented by the Gaussian components, and then represents the speaker factor as a mean shift on each of the components. The succeeding subspace models such as eigenvoice model [14], i-vector model [15] and joint factor analysis (JFA) model [16] follow the same principle but regulate the variation of speakers and sessions in a low-dimensional space. Some researches has proposed to use deep generative models to factorize speech variation. For example, Hsu et al. [17] employ a sequential VAE to discriminate phone and speaker factors. Another noticeable work is multi-style speech synthesis using style tokes [18].

Our approach in this paper is based on deep generative models as [17], however it uses supervised training to ensure the quality of the factorization model, and the invertibility of the model guarantees perfect speech reconstruction. Note that a fully supervised deep speech factorization approach was proposed by Li et al. [19], however it is not a generative model and so cannot guarantee a perfect reconstruction.

Our work is also related to deep speech representation, in the sense that both learn latent codes to represent speech. Contrastive prediction coding (CPC) [20] and autoregressive prediction coding (APC) [21] are trained to produce latent codes that predict the future samples based on the past samples. Our work follows the theme of latent representation learning, whereas focuses on information factorization.

5. EXPERIMENTAL

5.1. Data

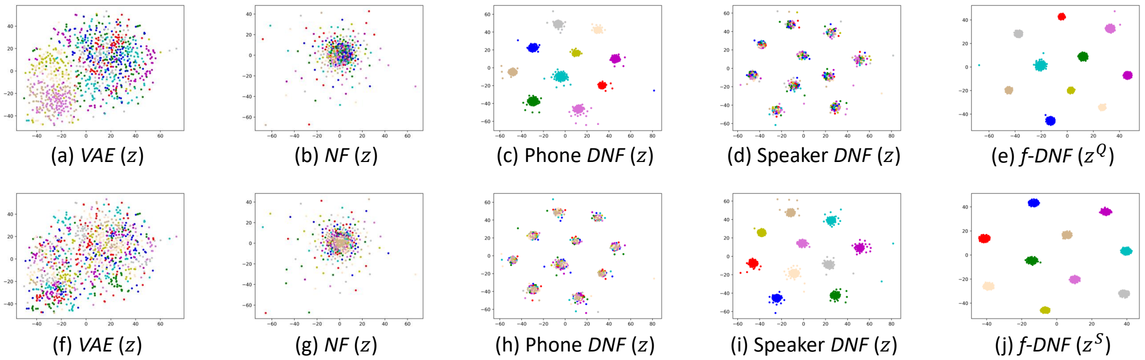
The TIMIT database is used in our experiments. The original 58 phones in the TIMIT transcription are mapped to 39 phones by Kaldi toolkit [22] following the TIMIT recipe, and 38 phones (*silence* excluded) are used as the phone labels. To balance the number of classes between phones and speakers, we select 20 female and 20 male speakers, resulting in 40 speakers in total.

All the speech utterances are firstly segmented into short segments according to the TIMIT phone transcriptions by force alignment. All the segments are trimmed to 200ms; if a segment is shorter than 200ms, we extend it to 200ms in both directions. All the segments are labeled by phone and speaker classes. Afterwards, every segment is converted to a 20×200 time-frequency spectrogram by FFT, where the window size is set to 25ms and the window shift is set to 10ms. The spectrograms are reshaped to vectors and are used as input features of deep generative models.

5.2. Model settings

The NF, DNF and factorial DNF models are based on the same *RealNVP* architecture [8]. It involves 6 blocks, and each block contains a coupling layer and a batch norm layer. The class means of DNF and factorial DNF are initialized from a normal distribution, and the variance of each class is set to I. For factorial DNF, the dimensions of the partial codes $\boldsymbol{z}^Q$ for phone and $\boldsymbol{z}^S$ for speaker are equal, and both are set to 2,000. The Adam optimizer [23] is used to train all these models.

For comparison, we also present the results with VAE. The source code released by Hsu et al. [17] is used to build the VAE model.



(a) *VAE* (z) (b) *NF* (z) (c) Phone *DNF* (z) (d) Speaker *DNF* (z) (e) *f-DNF* (z^Q)

(f) *VAE* (z) (g) *NF* (z) (h) Phone *DNF* (z) (i) Speaker *DNF* (z) (j) *f-DNF* (z^S)

Fig. 3. The latent codes generated by various models, plotted by t-SNE. In the first row (a) to (e), each color represents a phone; in the second row (f) to (j), each color represents a speaker. 'Phone DNF' denotes DNF trained with phone labels; 'Speaker DNF' denotes DNF trained with speaker labels.

5.3. Encoding

In the first experiment, we conduct a qualitative study on the latent codes generated by VAE, NF, DNF and factorial DNF. We use t-SNE [24] to draw the distributions of the latent codes generated by different models. Results are shown in Figure 3. It can be observed that the latent codes generated by VAE and NF almost lose the class structure; DNF can retain the class structure of the information factor corresponding to the class labels in the model training; Factorial DNF can retain the class structure corresponding to all the information factors.

5.4. Factor manipulation

The second experiment tests the quality of the factorization by manipulating the factors. Presumably, if the factorization is perfect, changing one factor will not modify the properties corresponding to other factors.

We use the mean-shift approach to conduct the manipulation. Given a factor A to manipulate, we compute the mean vectors of each class on factor A, denoted by $\{\mu_{A,i}\}$. Now for a sample x from one class c_1, we hope to change it to another class c_2. This can be obtained by moving its latent code z by a shift $\mu_{A,c_2} - \mu_{A,c_1}$ and then transforming it back to the observation space. In summary:

$$x' = f(f^{-1}(x) + \mu_{A,c_2} - \mu_{A,c_1}).$$

We will test the results when the phone factor and the speaker factor are manipulated respectively. Take the phone manipulation as an example, suppose the conversion for phone is from q_1 to q_2, and the speaker label s remains the same. We will report the posteriors $p(q_2|x)$, $p(q_2|x')$, $p(s|x)$ and $p(s|x')$. Experiments are conducted on all pairs of phones and the averaged posteriors are reported in Table 1. The same settings are applied to speaker manipulation. We trained a speaker and a phone MLP classifiers respectively to compute corresponding posteriors. Both classifiers contain one hidden layer with 800 hidden units.

Table 1. MLP posteriors on the target class before and after phone/speaker manipulation. 'f-DNF' denotes factorial DNF. $\delta(\cdot)$ denotes the difference on posteriors $p(\cdot|x')$ and $p(\cdot|x)$.

	Phone Manipulation					
Model	$p(q_2\|x)$	$p(q_2\|x')$	$\delta(q_2)$	$p(s\|x)$	$p(s\|x')$	$\delta(s)$
VAE	0.013	0.312	0.299	0.612	0.454	-0.158
NF	0.013	0.410	0.397	0.612	0.489	-0.123
DNF	0.013	0.619	0.606	0.612	0.335	-0.277
f-DNF	0.013	**0.636**	**0.623**	0.612	**0.536**	**-0.076**

	Speaker Manipulation					
Model	$p(s_2\|x)$	$p(s_2\|x')$	$\delta(s_2)$	$p(q\|x)$	$p(q\|x')$	$\delta(q)$
VAE	0.010	0.303	0.293	0.520	**0.509**	**-0.011**
NF	0.010	0.435	0.425	0.520	0.484	-0.036
DNF	0.010	0.700	0.690	0.520	0.349	-0.171
f-DNF	0.010	**0.710**	**0.700**	0.520	0.503	-0.017

It can be seen that DNF has a stronger capacity than VAE and NF to implement factor manipulation. However, the DNF-based manipulation tends to cause larger distortion on other factors. Factorial DNF has similar even better performance than DNF in terms of factor manipulation, but causes very little distortion on other factors. Speech examples can be found at http://project.cslt.org.

6. CONCLUSIONS

This paper presented a speech information factorization method based on a novel deep generative model that we called factorial discriminative normalization flow. Qualitative and quantitative experimental results show that compared to all other models, the proposed factorial DNF can retain the class structure corresponding to multiple information factors, and changing one factor will cause little distortion on other factors. This demonstrates that factorial DNF can well factorize speech signal into different information factors. Future work will test factorial DNF on larger datasets, and establish general theories for deep generative factorization.

7. REFERENCES

[1] James L Flanagan, *Speech analysis synthesis and perception*, vol. 3, Springer Science & Business Media, 2013.

[2] Sean A Fulop, *Speech spectrum analysis*, Springer Science & Business Media, 2011.

[3] Gunnar Fant, *Acoustic theory of speech production*, Number 2. Walter de Gruyter, 1970.

[4] Jacob Benesty, M Mohan Sondhi, and Yiteng Arden Huang, "Introduction to speech processing," in *Springer Handbook of Speech Processing*, pp. 1–4. Springer, 2008.

[5] Hiroya Fujisaki, "Prosody, models, and spontaneous speech," in *Computing prosody*, pp. 27–42. Springer, 1997.

[6] Christopher Bishop, *Continuous Lantent Variables*, Springer, 2006.

[7] Laurent Dinh, David Krueger, and Yoshua Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real NVP," *arXiv preprint arXiv:1605.08803*, 2016.

[9] Durk P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in neural information processing systems*, 2018, pp. 10215–10224.

[10] Walter Rudin, *Real and complex analysis*, Tata McGraw-hill education, 2006.

[11] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *arXiv preprint arXiv:1912.02762*, 2019.

[12] Yunqi Cai, Lantian Li, Dong Wang, and Andrew Abel, "Deep normalization for speaker vectors," *arXiv preprint arXiv:2004.04095*, 2020.

[13] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[14] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel, "Maximum likelihood estimation of eigenvoices and residual variances for large vocabulary speech recognition tasks," in *Seventh International Conference on Spoken Language Processing*, 2002.

[15] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[16] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[17] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," in *INTERSPEECH*, 2017, pp. 1273–1277.

[18] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[19] Lantian Li, Dong Wang, Yixiang Chen, Ying Shi, Zhiyuan Tang, and Thomas Fang Zheng, "Deep factorization for speech signal," in *ICASSP*. IEEE, 2018, pp. 5094–5098.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[21] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.