Thesis

Summaries on articles

Article analisys

1. Voice conversion based on DNN for time-variant linear transformation

- Linear transform is enough because we work with homo-domain mapping def 1) Cepstrum – result of inv. Fourier transform of the log of the estimated signal spectrum

2) Signal spectrum of a time series $x(t)$ – distribution of power into freq. components f, comp. that signal

- GMM based voice conversion approach:
(Gaussian mixture model)

joint proba of input and output features (on sepstrum space) is modeled by GMM; therefore, conversion input $\mapsto$ target is locally linear transformation

{can be interp. as the use of the homo-domain condition}

- Another way – voice conv. based on non-negative matrix factorization
{not much was said about it}

- also difference between speakers in cepstral space was discussed:

a wave transformation via warp function can be descr. as a linear transformation in cepstral space.

$$\hat{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \quad \text{where } z = e^{jw}, \ \hat{z} = e^{j\hat{w}}$$

α - warping param.
w - initial,
$\hat{w}$ - after transf.

(!) can be rewritten as

$$\hat{c} = Ac, \text{ where } \underbrace{A = \begin{pmatrix} 1-\alpha^2 & 2\alpha - 2\alpha^3 & \dots \\ -\alpha + \alpha^3 & \ddots & \\ \dots & & \ddots \end{pmatrix}}_{\text{rotation matr.}}$$

- Deep NN for time variant linear transform.

$$x_t \longmapsto \{ \text{lin. transform matrix}, \text{Biases}, \text{warping param.}, \text{dynamic feature} \}$$

Linear transform: $\hat{y}_t = (A_t + A_t^{(\alpha)})(x_t - b_t^{src}) + b_t^{tgt}$

where

$$\Delta\hat{y}_t = G_{\Delta y}(x_t)$$

$$\underbrace{G_{\Delta y}, G_A, G_\alpha, G_b}_{\text{Sub networks for parameters estimation}}$$

$$A_t = G_A(x_t) \quad A_t^{(\alpha)} = \begin{pmatrix} 1-\alpha_t^2 & \dots \\ \dots & \end{pmatrix}$$

$$\alpha_t = G_\alpha(x_t)$$

$$[b_t^{src}, b_t^{tgt}]^T = G_b(x_t)$$

2. Learning Latent Representations for style control and transfer in end to end speech synthesis.

- Main idea – learn latent reps. of the source audio to control style transfer

- Variational Autoencoder –
  reveals relationship betw. latent z and observed x.

True posterior $p_\theta(z|x)$ is impossible to find, therefore, likelihood $p_\theta(x)$ is indifferent. So $p_\theta(z|x)$ can be approximated by $q_\phi(z|x)$ and

variational lower bound $\mathcal{L}(\theta, \phi; x)$:

$$\log p_\theta(x) = KL[q_\phi(z|x) \| p_\theta(z|x)] + \mathcal{L}(\theta, \phi; x)$$
$$\geq \mathcal{L}(\theta, \phi; x)$$
$$= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x) \| p_\theta(z)]$$

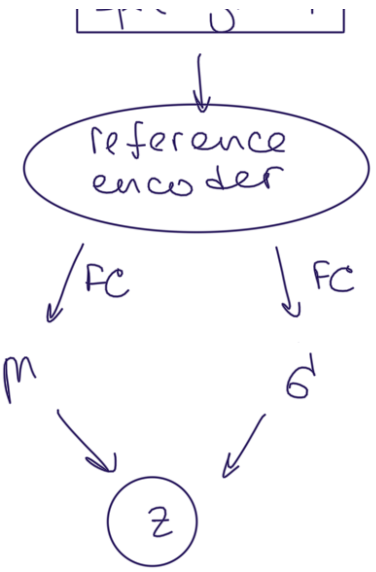
where $p_\theta(z) \sim \mathcal{N}(z; 0, I)$, $q_\phi(z|x) \sim \mathcal{N}(z; \mu(x), \sigma^2(x) \cdot I)$

sort of decoder to decode latent z to reconstruct x

Sampling z from $\mathcal{N}(\mu, \sigma^2 \cdot I)$ is decomposed to sampling $\varepsilon \sim \mathcal{N}(0, I)$ and $z = \mu + \sigma \odot \varepsilon$, $\odot$ – element-wise product

$\boxed{\text{spectrogram}}$

- KL collapse problem



- the convergence speed of KL loss surpasses that of the reconstruction loss

- Several dimensions of latent z could independently control different style attributes

Total loss: $KL[q_\phi(z|x) \| p_\theta(z)] - \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z,t)] + l_{stop}$

3. Deep learning for Audio style transfer

- Main idea – learning content and style separately using Gram Matrix for style retrieval and convolutional NN's for content

prepocessing – STFT (Short time Fourier transform)

therefore: spectogram

- Using different losses for style and content (still a lot is unclear)

mean squared error between Gram Matr. of image & white noise image

mean squared err. betw. source img. embed & white noise embe.

but why use Gram
matr.?

Gram matr. helps to numerically evaluate the correlation between features — the more they're similar the bigger the loss is.

So for convolved matr. the Gram matr. represents the "style", as it evaluates how these higher level features correlate

which repr. some "higher" level features, like, for images, edges, sharpness, blur, etc

the total loss — Σ(style_loss, content-loss)

Important — same approach as with images can be used

4. Linear transformation Approaches to Many-to-One Voice Conversion

Voice conversion:
1) training
2) adaptation (speaker selection, eigen voice techniques)
3) conversion

(1) Constrained Max. Likelyhood Linear Regression

$$\hat{x}_t = A x_t + b = W \xi_t \quad \left\{ \begin{array}{l} W = [b, A] \\ \xi_t = [1, x_t^T]^T \end{array} \right\}$$

$\sim$ constr. model-space transformation

$$P(x_t, y_t \mid \lambda, W) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix} ; \hat{\mu}_m^{xx}, \hat{\Sigma}_m^{xx} \right)$$

where $\hat{\mu}_m = \begin{bmatrix} A' \mu_m^x + b' \\ \mu_m^y \end{bmatrix}$

$$\hat{\Sigma}_m = \begin{bmatrix} A' \Sigma_m^{xx} A'^T & A' \Sigma_m^{xy} \\ \Sigma_m^{yx} A'^T & \Sigma_m^{yy} \end{bmatrix}$$

$\Rightarrow$ EM algorithm maximazises

$$\hat{W} = \underset{W}{\operatorname{argmax}} \prod_{t=1}^{T} P(x_t^{(new)} \mid \lambda, W)$$

(2) Mean Linear Transformation

$$\hat{\mu}_m^x = A' \mu_m^x + b' = W' \xi_m \qquad \begin{array}{l} W' = [b', A'] \\ \xi_m = [1, \mu_m^{x\,T}]^T \end{array}$$

$$\sim P(x_t, y_t \mid \lambda, W') = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix} ; \hat{\mu}_m^{xy}, \Sigma_m^{xy} \right)$$

where $\hat{\mu}_m^{xy} = [\mu_m^{x\,T}, \mu_m^{y\,T}]^T$

And EM algorithm max.

$$\hat{W}' = \operatorname*{argmax}_{W'} \prod_{t=1}^{T} P(x_t^{(new)} \mid \lambda, W')$$

SAT = Speaker adaptive training

In SAT realisation with EM-algorithm both the params. of canonical GMM and a set of speaker-dependent linear transforms are optimised:

$$\{\hat{\lambda}, \hat{W}\} = \operatorname*{argmax}_{\{\lambda, W\}} \prod_{s=1}^{S} \prod_{t=1}^{T} P(x_t^{(s)}, y_t \mid \lambda, W^{(s)})$$