An introduction of trajectory model into HMM-based speech synthesis

Heiga Zen　Keiichi Tokuda　Tadashi Kitamura　(Nagoya Institute of Technology, Japan)

Introduction

Corpus-based speech synthesis system

Unit selection and concatenation (e.g., CHATR)

- High quality (sometimes discontinuous)
- Require large memory
- Difficult to change voice characteristics

Speech synthesis from HMMs (e.g., HTS)

- Vocoded speech (smooth and stable)
- Small footprint
- Easy to change voice characteristics

Problems in HTS

- Excitation
  ⇒ Mixed excitation, Sinusoidal, etc.
- Formant structures get smooth
  ⇒ Post filtering based formant emphasis

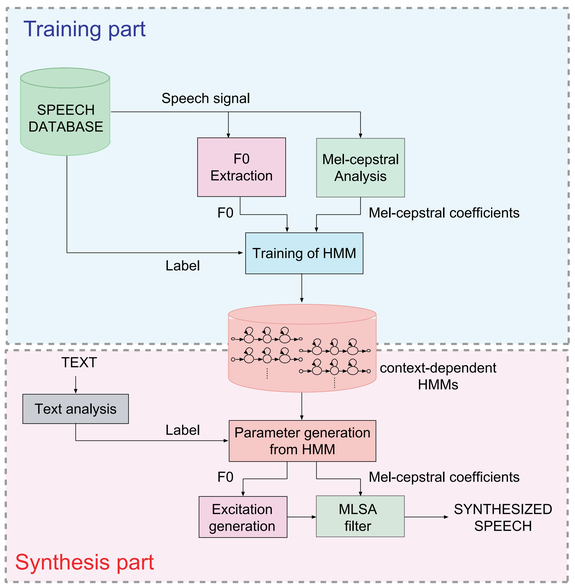
Trajectory-HMM and its training algorithm

(Tokuda et.al;Eurospeech2003, Zen et.al;ICASSP2004)

- Introduce dynamic feature constraints into HMM
  ⇒ Reformulate HMM as trajectory model
- Derive its training algorithm based on ML criterion
  ⇒ Closed-loop training for HTS

⇒ Apply trajectory-HMM training to HTS

System overview of the HTS



Dynamic feature constraints

Speech parameter generation from HMM

⇒ Generate most likely observation sequence

Without dynamic feature constraints

$$\boldsymbol{o}_{\max} = \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid \boldsymbol{q}, \lambda)$$

$\boldsymbol{o}$: observation sequence
$\boldsymbol{q}$: state sequence

⇒ $\boldsymbol{o}_{\max}$ becomes a sequence of mean vectors

Integrate dynamic feature constraints

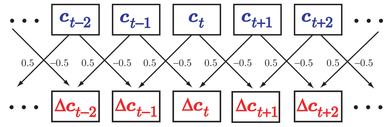
Observation consisted from static and dynamic features

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top$$

$\boldsymbol{c}_t$: static feature ($M \times 1$)
$\Delta\boldsymbol{c}_t$: 1st-order dynamic feature
$\Delta^2\boldsymbol{c}_t$: 2nd-order dynamic feature

Dynamic feature ⇒ computed from static features

Ex.) $\Delta\boldsymbol{c}_t = 0.5\boldsymbol{c}_{t+1} - 0.5\boldsymbol{c}_{t-1}$



Relationship between static & observation vector sequence

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$$

$\boldsymbol{W}$: Window matrix ($3MT \times MT$)

Under this relationship, we should regard only $\boldsymbol{c}$ as the random variable

With dynamic feature constraints

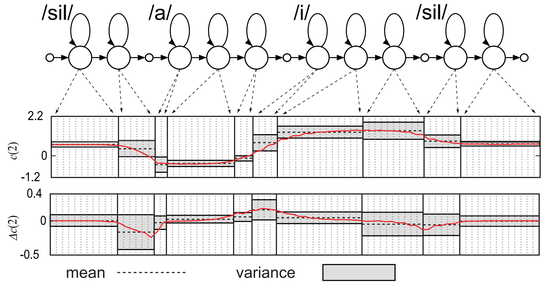
$$\boldsymbol{c}_{\max} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{q}, \lambda)$$

By setting $\frac{\partial}{\partial \boldsymbol{c}} \log P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = 0$

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

$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top\right]^\top$
$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \mathrm{diag}\left[\boldsymbol{\Sigma}_{q_1}, \ldots, \boldsymbol{\Sigma}_{q_T}\right]$

A sequence of speech parameter vector can be determined based on statistics of static and dynamic features.



Trajectory HMM

Dynamic feature constraints

⇒ ・ Only used in the synthesis stage of HTS
　　・ Inconsistency between training and synthesis

Trajectory-HMM and its training algorithm

(Tokuda et.al;Eurospeech2003, Zen et.al;ICASSP2004)

- Introduce dynamic feature constraints into HMM
  ⇒ Reformulate HMM as trajectory model
- Derive its training algorithm based on ML criterion
  ⇒ Closed-loop training for HTS

⇒ Apply trajectory-HMM training to HTS

Definition of trajectory-HMM

Output probability for given trajectory-HMM λ :

$$P(\boldsymbol{c} \mid \lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) P(\boldsymbol{q} \mid \lambda)$$

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

$\bar{\boldsymbol{c}}_{\boldsymbol{q}}$: Utterance mean vector for given state sequence

$\boldsymbol{P}_{\boldsymbol{q}}$: Utterance covariance matrix for given state sequence

$$\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$$

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}$$

⇒ $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is exactly the same to parameter trajectory obtained by speech parameter generation algorithm!

When $P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = \mathcal{N}\left(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right)$ is maximized ?

⇒ training data $\boldsymbol{c}$ and generated trajectory $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ (utterance mean vector) is equivalent.

Maximizing model likelihood of the trajectory-HMM
= Minimizing the error between training data and generated speech parameter trajectory

⇒ closed-loop training for HTS

Speech synthesis experiment

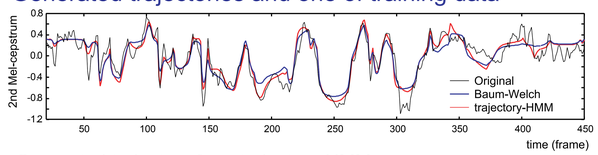
Training data	CMU ARCTIC database speaker AWB first 1096 utterances
Test data	Remaining 42 utterances
Sampling rate	16 kHz
Window	25-ms Blackman window
Frame rate	5-ms
Spectral analysis	24-order Mel-cepstral analysis
Dynamic feature	calculated from ±1 frames
Feature vector	$c(0)$〜$c(24)$, $\log F0$, and its Δ, $\Delta\Delta$
Topology	5-state left-to-right HMM with no skip Spectrum : single Gaussian distribution F0 : multi-space probability distribution

Trained models

Baum-Welch: Baum-Welch training
Viterbi: Viterbi training
trajectory-HMM: trajectory-HMM training

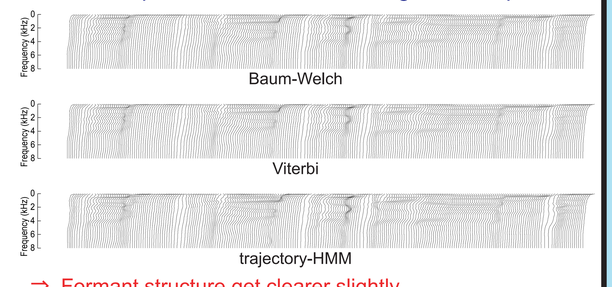
Prosodies generated from these models were exactly the same

Generated trajectories and one of training data



Generated trajectroy from trajectory-HMM
⇒ Closer to training data than that from Baum-Welch trained HMM

Generated spectra for a sentence fragment "tropic land"



⇒ Formant structure get clearer slightly

Subjective listening test

Test type	Paired comparison test
Subjects	8 persons
Test sentences	20 test sentences were chosen at random