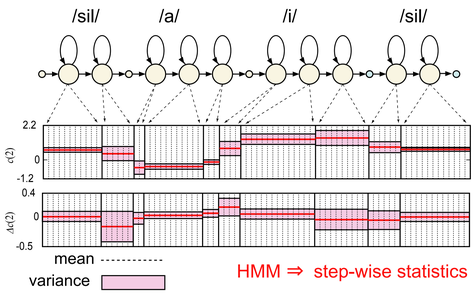
ESTIMATING TRAJECTORY HMM PARAMETERS USING MONTE CARLO EM WITH GIBBS SAMPLER

Heiga Zen Yoshihiko Nankaku Keiichi Tokuda Tadashi Kitamura (Nagoya Institute of Technology)

1. Background

Limitations of HMMs for modeling speech

- Constant statistics within a state
- Conditional independence assumption



Alternative approach

- Segment models (HDM, SSM, SLDS, etc.)

⇒ Increase #parameters and computational cost

- Use of dynamic features (Δ and ΔΔ coefficients)

⇒ Improper in the sense of statistical modeling

Trajectory HMM (Tokuda et. al, Eurospeech03)

- Impose relationships between static & dynamic

⇒ Reformulate an HMM as a trajectory model

- Overcome the above two limitations
- Viterbi-type training algorithm has been derived
- Lack of the EM-type training algorithm

⇓

2. Objective

EM-type algorithm for the trajectory HMM

- Temporal covariance matrix is generally full

⇒ Impossible to perform the exact EM algorithm

- Approximate training
  - Viterbi approximation
  - Variational EM
  - Monte Carlo EM (Sampling)

    ⇒ Obtain sub-optimal state sequences
- Evaluation in a speech recognition experiment

3. Definition of the trajectory HMM

Output probability of observation $\boldsymbol{c}$ from a trajectory HMM Λ

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

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

$$P(\boldsymbol{q} \mid \Lambda) = \prod_{t=1}^{T} P(q_t \mid q_{t-1}, \Lambda)$$

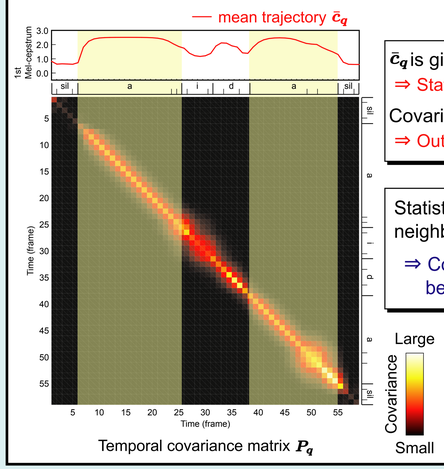
$\boldsymbol{c}$: static feature vector seq. ($MT \times 1$)

$\boldsymbol{q}$: state (mixture) seq.

$\bar{\boldsymbol{c}}_{\boldsymbol{q}}$: mean trajectory (vector) ($MT \times 1$)

$\boldsymbol{P}_{\boldsymbol{q}}$: temporal covariance mat. ($MT \times MT$)

M : #dim. of static feat. vec. (ex. MFCC)



$\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is given as a *smooth trajectory*

⇒ Statistics vary within a state

Covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$ is *full*

⇒ Output probs. depend on each other

Statistics vary according to their neighboring models

⇒ Coarticulation effect can naturally be captured

4. Estimating parameters of trajectory HMMs

Auxiliary function (hidden variable: state seq. $\boldsymbol{q}$)

$$\mathcal{Q}(\Lambda, \Lambda') = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda)\, \log P(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda')$$

However, applying the exact EM algorithm is intactable... (>_<)
(Temporal covariance matrix is full)

⇒ Approximate training

- Viterbi approximation (Viterbi training) [Zen, et. al; ICASSP04]
- Markov Chain Monte Carlo (Monte Carlo EM; MCEM)

$$\mathcal{Q}(\Lambda, \Lambda') \approx \sum_{n=1}^{N} \frac{1}{N} \log P(\boldsymbol{c}, \boldsymbol{q}_n \mid \Lambda')$$

□□ - Approximate using N state (Gaussian) seqs.

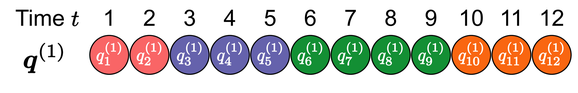
□□ - Generating $\boldsymbol{q}$ ⇒ Gibbs sampling

ESTIMATING TRAJECTORY HMM PARAMETERS USING MONTE CARLO EM WITH GIBBS SAMPLER

Heiga Zen Yoshihiko Nankaku Keiichi Tokuda Tadashi Kitamura (Nagoya Institute of Technology)

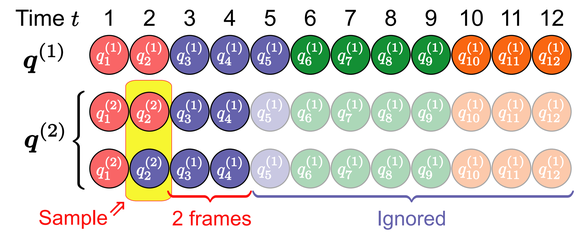
5. Generating state seqs. using Gibbs sampler

Step 1.



Initialized by the delayed decision Viterbi algorithm

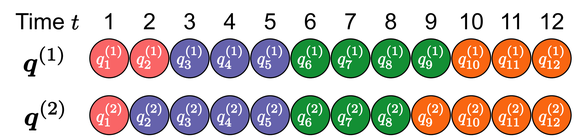
Step 2.



- Only consider states at the following J frames (ex. J=2)
- Generate random value using $P\left(q_2 \mid \boldsymbol{c}, q_1^{(2)}, q_3^{(1)}, q_4^{(1)}, \Lambda\right)$

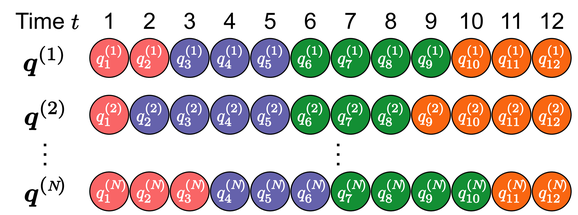
  ⇒ Choose $q_2^{(2)}$ according to generated random value
- Repeat until reaching the final frame

Step 3.



Final frame ⇒ 2nd state seq. sample is obtained

Step 4.



Repeat N times ⇒ N state seqs. can be sampled

Taking partial derivative of the approx. auxiliary function with respect to trajectory HMM parameters

□⇒ Parameter reestimation formulas

6. Speech recognition experiment

Conditions

Database	ATR Japanese continuous speech database b-set speaker MHT
Training data	450 utterances
Test data	53 utterances
Feature vector	0～18 order Mel-cepstral coefficients and its delta and delta delta
Model structure	3-state left-to-right monophone model with single Gaussian state output distribution

Training procedure

Estimate trajectory HMMs using baseline HMMs as their initial models

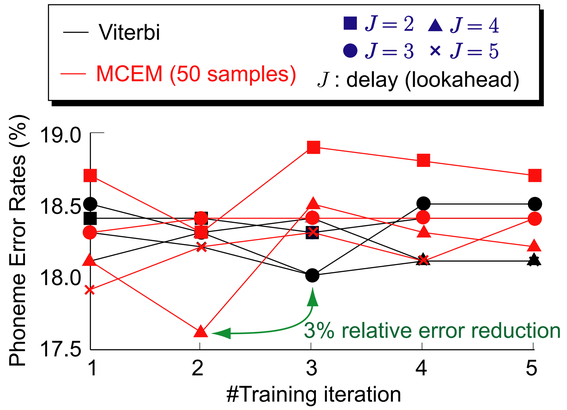
- Viterbi training × 5 iterations
- MCEM algorithm (50 samples) × 5 iterations

Recognition procedure

1. Generate 100-best hypotheses
2. Rescore using trajectory HMMs
   - Delayed decision Viterbi algorithm + beam search
   - Sampling was not performed during rescoring

Experimental results

HMM: 19.7% Best: 13.9% Worst: 27.4%



The performance of the MCEM-trained trajectory HMMs were relatively unstable (17.6%～18.9%)

- Mismatch between training and recognition?
- Sampling ⇒ Stochastic method?