Non Parametric Bayesian Acoustic Model Discovery for phoneme classification

Tanuj Jain

Department of Communications Engineering - University of Paderborn

November 17, 2015

Problem Description

Phoneme

- Basic unit of Language's Phonology
- Example: table = /t/, /a/, /bl/
- Multiple phonemes combine to form meaningful entity
- Number varies from language to language

Discovery of Phonemes

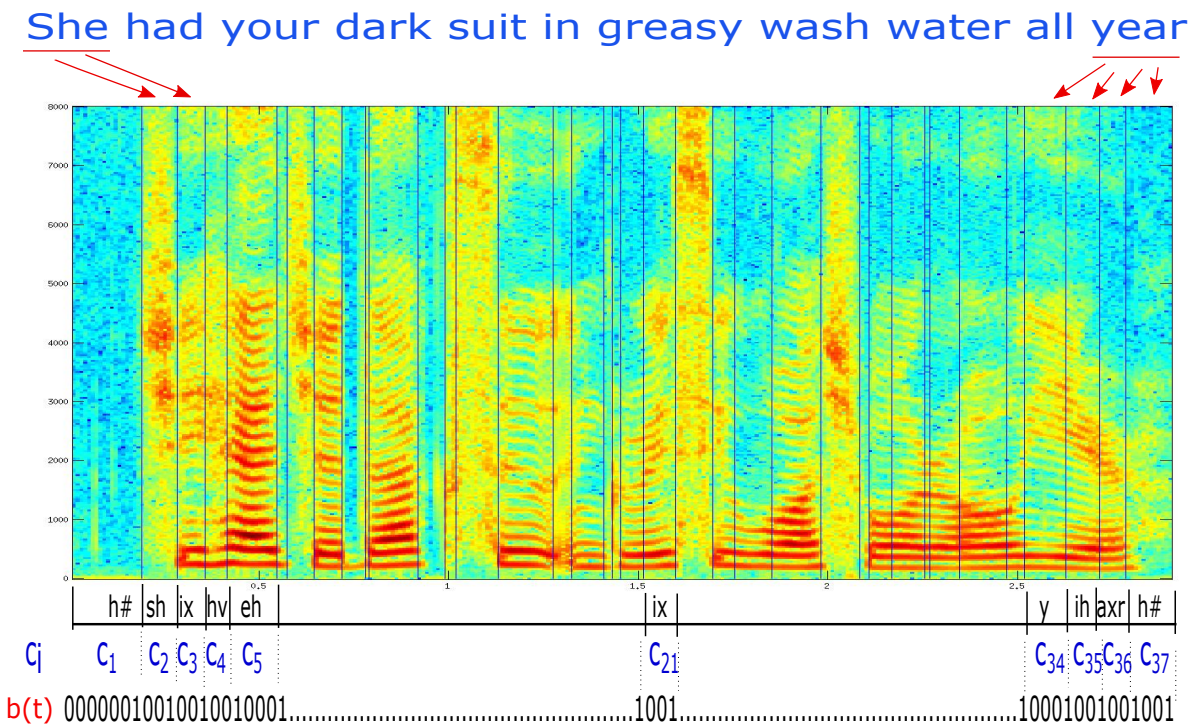
- Unsupervised: Labels absent
- Non-parametric: Varying number of parameters
- Language Independent

Overview

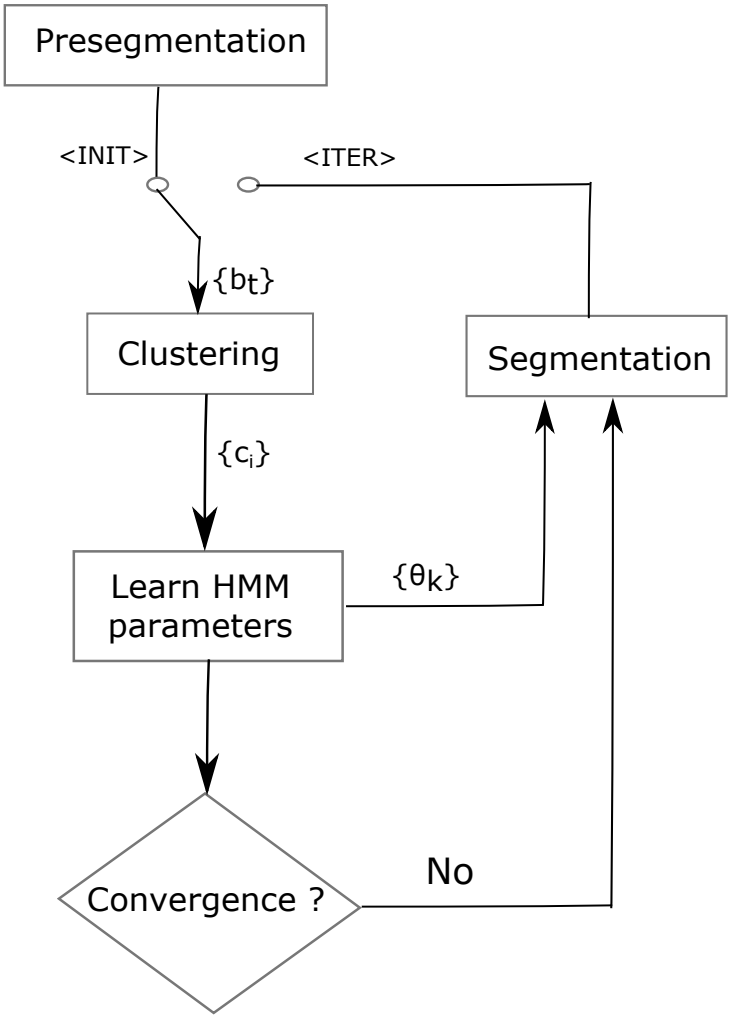
Basic sub-tasks

Three basic subtasks to solve the problem:

- Segmentation
- Clustering
- Learning

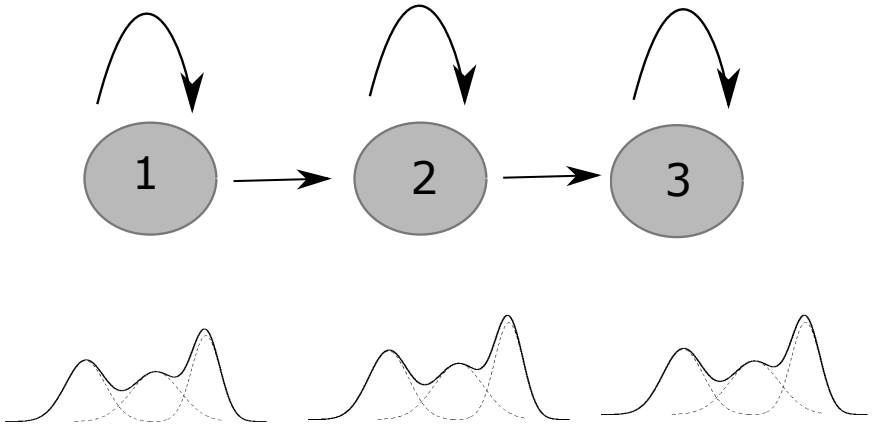


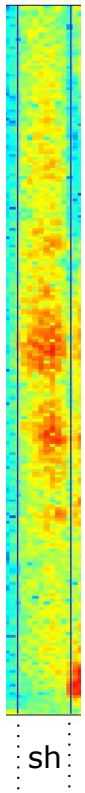
Complete Algorithm



Our model: HMM

- 1 phoneme = 1 Hidden Markov Model (HMM)
- 3 states
- Left-Right model
- State observation density = Gaussian Mixture Model (GMM)

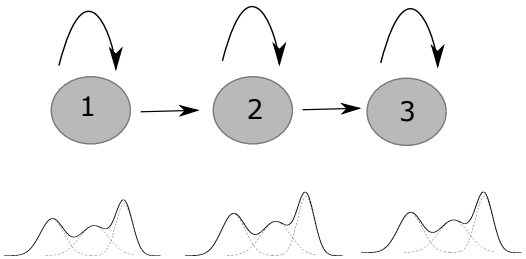




Model assumptions & implications

- First order assumption:
  $P(q_t|q_{t-1}, q_{t-2}, \cdots) = P(q_t|q_{t-1})$
- $P(X_t|q_1, \cdots, q_t, \cdots, q_T) = P(X_t|q_t = s_i) = b_i(X_t)$
- Left-Right assumption:

$$\Pi = \{\pi_1, \pi_2, \pi_3\} = \{1, 0, 0\}$$



- Left-right assumption + no state-skipping:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Problems of HMM

Given: HMM parameters $\theta = \{\mathbf{A}, \mathbf{B}, \Pi\}$
observation sequence $\mathbf{X} = \{X_1, X_2, \cdots, X_t, \cdots, X_T\}$

- Evaluation problem: Find $P(\mathbf{X}|\theta)$
- Learning Problem: Adjust θ

Segmentation

Input

Continuous stream of speech

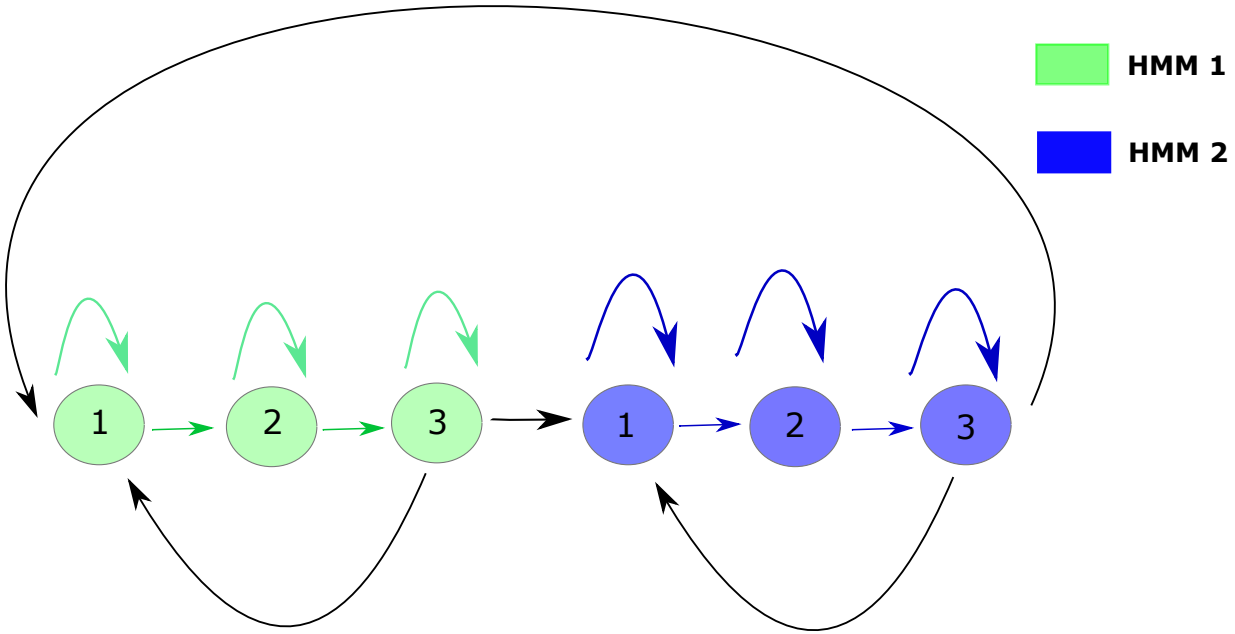
Desired

Sequences of speech samples to map to phoneme

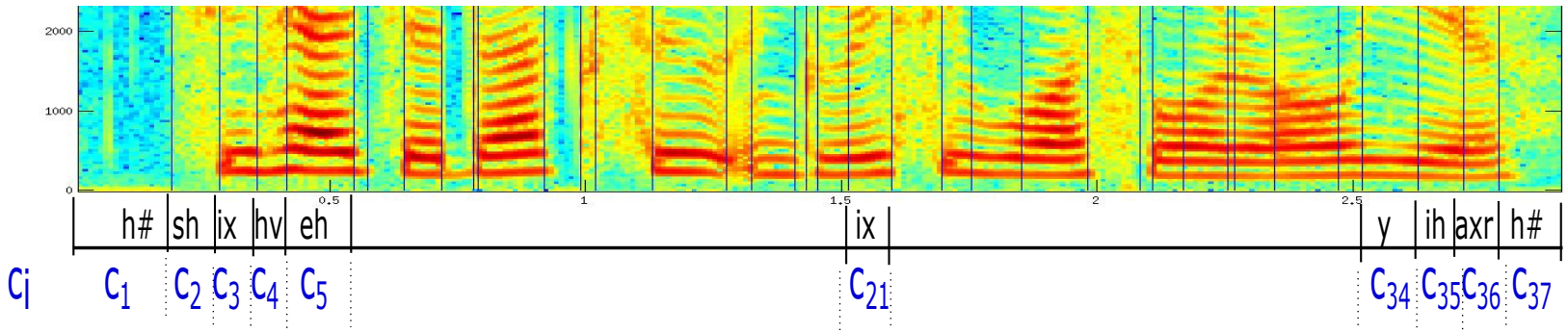
$$b_t = \begin{cases} 1, & \text{if sample is the last in a sequence} \\ 0, & \text{otherwise} \end{cases}$$

Segmentation

- First iteration: Pre segmentation algorithm
- Subsequent iterations: Viterbi algorithm on cascaded HMMs



Clustering



- Assignment of cluster labels to each sequence given $\{\theta_k\}$
- Aim: Same values to sequences belonging to same phoneme
- Cluster label values in range:

$$c_i \in \{1, 2, \cdots, K\}$$

- Done using Gibbs sampling

Gibbs sampling

- Markov Chain Monte Carlo method to generate samples from joint distribution $P(z_1, z_2, \cdots, z_D)$
- Basic idea: Sample from joint distribution given all conditional distributions

Algorithm 1 Gibbs sampling algorithm

Initialize $z^0_{1,2,\ldots,D}$
for $t = 0$ to $T - 1$ **do**
    Sample $z_1^{t+1} \sim P(z_1 | z_2^t, \ldots., z_D^t)$
    Sample $z_2^{t+1} \sim P(z_2 | z_1^{t+1}, \ldots., z_D^t)$
    .
    .
    Sample $z_D^{t+1} \sim P(z_D | z_1^{t+1}, z_2^{t+1}, \ldots., z_{D-1}^{t+1})$
end for

Clustering using Gibbs sampling

- Sample $\{c_i\}$ from $P(c_1, c_2, \cdots, c_i, \cdots, c_{N_{seq}} | \theta, \mathbf{X} \cdots)$
- Two cases:
  - Known number of HMMs (K)
  - Unknown number of HMMs = Known number of HMMs + new HMM

Known number of HMMs

Cluster label (c_i) sampled as per:

$$P(c_i = k | c_{-i}, \theta_k, \mathbf{X}_i \cdots) \propto \frac{N_{-k} + \gamma/K}{N_{seq} - 1 + \gamma} \cdot P(\mathbf{X}_i | \theta_k)$$

$$Posterior \propto Prior \times Likelihood$$

where,

- $\mathbf{X}_i$- i-th sequence
- N_{seq}- Total number of sequences
- γ- Dirichlet hyperparameter
- N_{-k} Number of sequences in k-th HMM excluding i-th sequence
- $P(\mathbf{X}_i | \theta_k)$ term obtained by solving the evaluation problem of the HMM

Unknown number of HMMs: Dirichlet Process

Also called 'Chinese Restaurant Process'

Chinese Restaurant Process

Chinese Restaurant Process

Chinese Restaurant Process

Unknown number of HMMs: Dirichlet Process

For existing HMMs:

$$P(c_i = k | c_{-i}, \theta_k, \mathbf{X}_i \cdots) \propto \frac{N_{-k}}{N_{seq} - 1 + \gamma} \cdot P(\mathbf{X}_i | \theta_k)$$

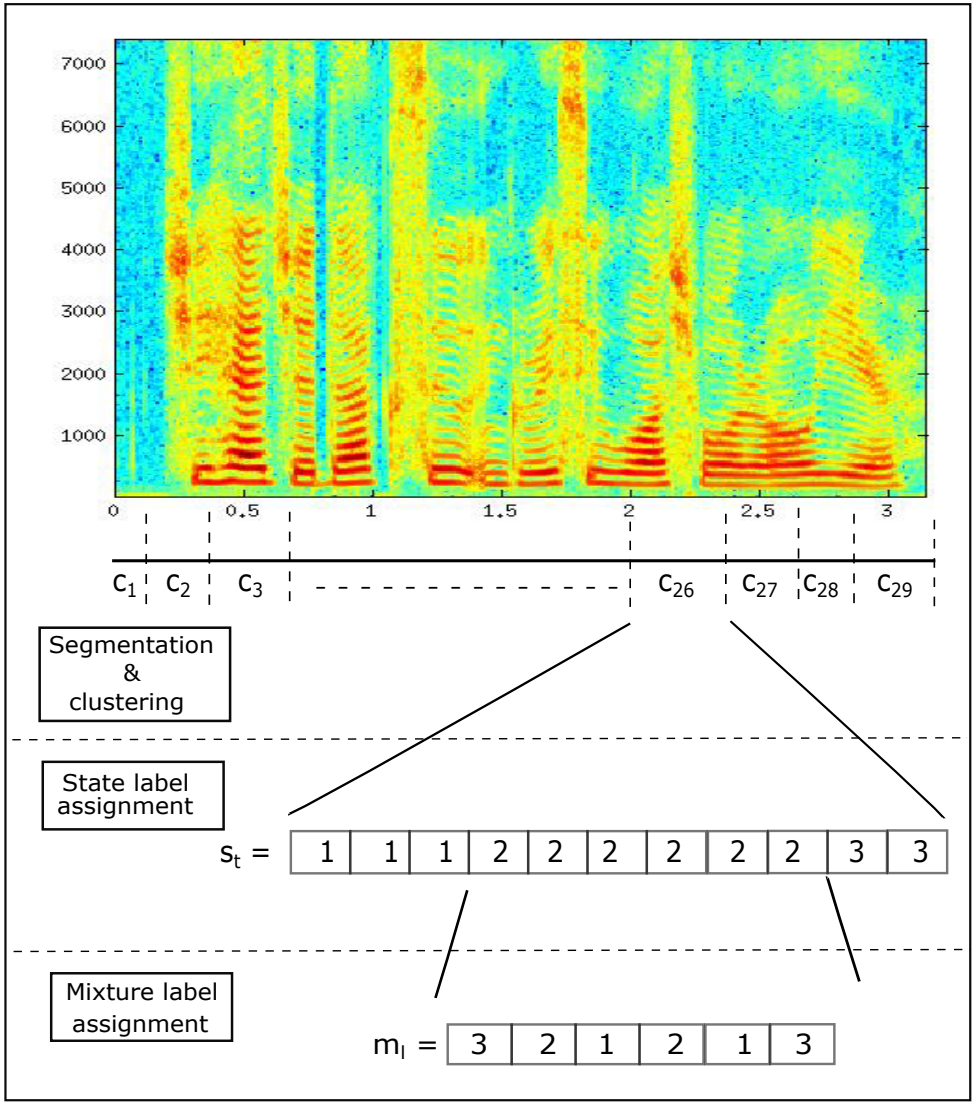
For the new HMM:

$$P(c_i = k_{new} | c_{-i}, \theta_{knew}, \mathbf{X}_i, \cdots) \propto \frac{\gamma}{N_{seq} - 1 + \gamma} \int P(\mathbf{X}_i | c_i = k_{new}, \phi) P(\phi) d\phi$$

Learning

Learning HMM attributes θ_k:

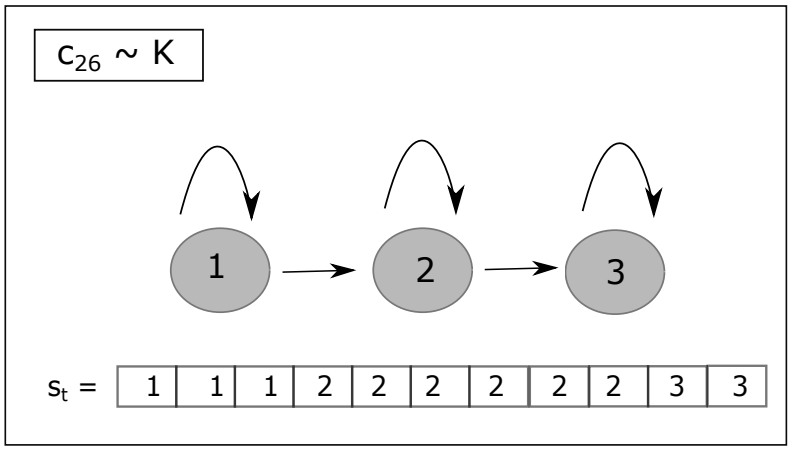
- Transition probabilities a_{ij}.
- GMM attributes for each state:
  - mixture weights (w)
  - mixture means (μ)
  - mixture precisions (λ)
- Can be learnt using:
  - Baum Welch algorithm: Traditional EM approach
  - Gibbs sampling



Learning state assignments (s_t)

- Assign states to each sample in each sequence for an HMM
- s_t takes value as:

$$s_t \in \{1, 2, 3\}$$

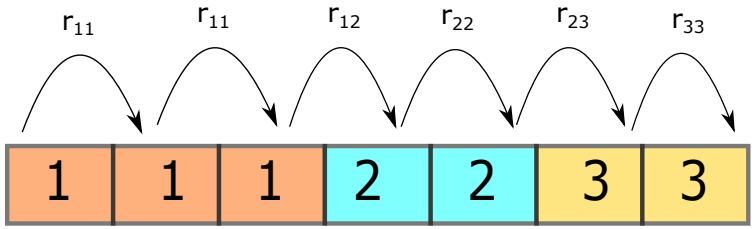


- Backward sampling used

$$P(s_t = j | s_{t+1} = l \cdots) \propto Cat(s_t | v_t(1), \cdots, v_t(N))$$

$$v_t(j) = P(X_1, X_2, \cdots, X_t, s_t = j | \theta) \cdot a_{jl}$$

Determining transition probabilities (a_{ij})



- Transitions from a state extracted:

$$\underline{r_1} = \{r_{11}, r_{11}, r_{12}\}$$
$$\underline{r_2} = \{r_{22}, r_{23}\}$$

- $\underline{a_i}$ sampled from:

$$P(\underline{a_i}|\underline{r_i}, \eta) \propto Dir(\underline{a_i}; \underline{\eta'})$$
$$\eta'_j = \eta_{0j} + n_{ij}$$
$$\underline{a_1} = \{a_{11}, a_{12}\}$$
$$\eta'_{11} = \eta_{01} + n_{11} = \eta_{01} + 2$$
$$\eta'_{12} = \eta_{02} + n_{12} = \eta_{02} + 1$$

Learning GMM attributes $w^m_{k,s_t}, \mu, \lambda$

- Use all samples belonging to one state
- Assign mixture labels:

$$P(m_l = m|\theta_k, s_t, o_l, \cdots) \propto P(m_l = m|\theta_k, s_t)P(o_l|\theta_k, s_t, m_l = m)$$
$$= w^m_{k,s_t} P(o_l|\mu^m_{k,s_t}, \lambda^m_{k,s_t})$$

- Update mixture weights using Dirichlet distribution:

$$P(\underline{w_{k,s}}|\underline{m}_{k,s}, \rho, \cdots) \propto P(\underline{w_{k,s}}; \rho) \cdot P(\underline{m}_{k,s}|\underline{w_{k,s}})$$
$$\rho'_m = \rho_m + \sum_{m_t \in \underline{m}_{k,s_t}} \delta(m_l, m)$$

- Update means and precisions:

$$P(\mu, \lambda|\mathbf{o}) = NG(\mu, \lambda; \mu', \kappa', \alpha', \beta')$$

Flat start initialization

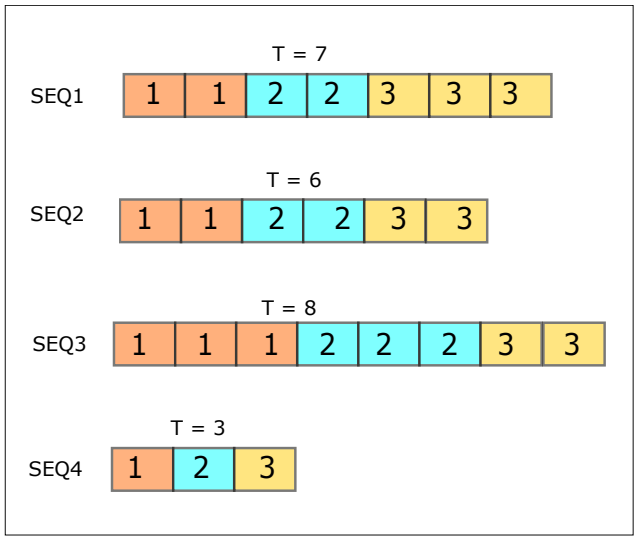
Initialization approach:

Step 1: Randomly assign cluster labels to each sequence

Step 2: Collect all sequences belonging to one cluster (HMM)

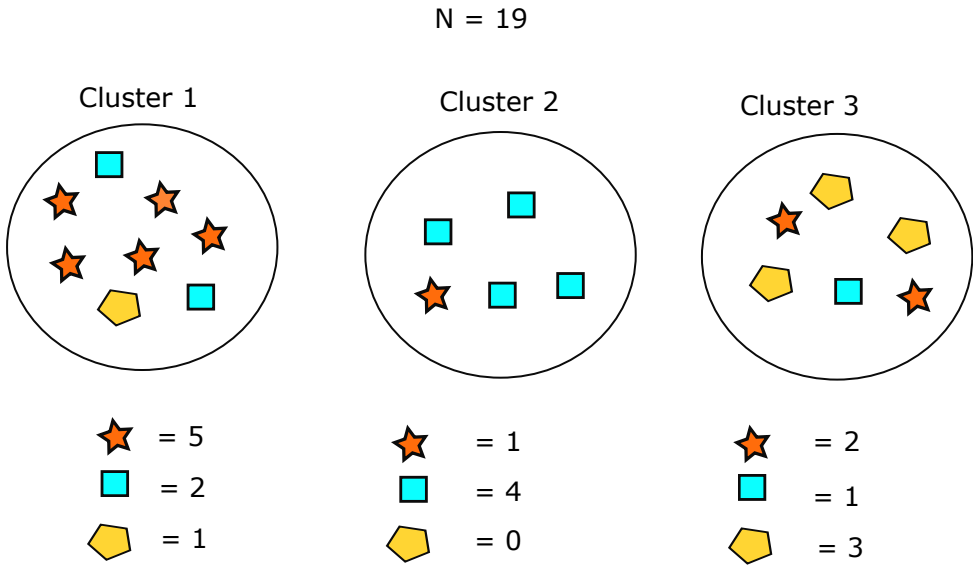
Step 3: Use left-right assumption to divide samples in each sequence

Step 4: Use samples for a state to initialise transition probabilities and mixture attributes



Evaluation measure: Cluster Purity

- Evaluate percentage of the majority classes in each cluster



$$CP = \frac{1}{19}(5 + 4 + 3)$$
$$= 0.6315$$
$$\Longrightarrow \quad CP = 63.15\%$$

Experimental setup

Simulations scenarios:

- Initialization method for known HMM case: Flat start vs. Random initialization
- Number of mixture components for GMM
- Learning algorithm: Gibbs sampling vs. Baum-Welch algorithm
- Supervised vs. unsupervised

Results

Training Database:

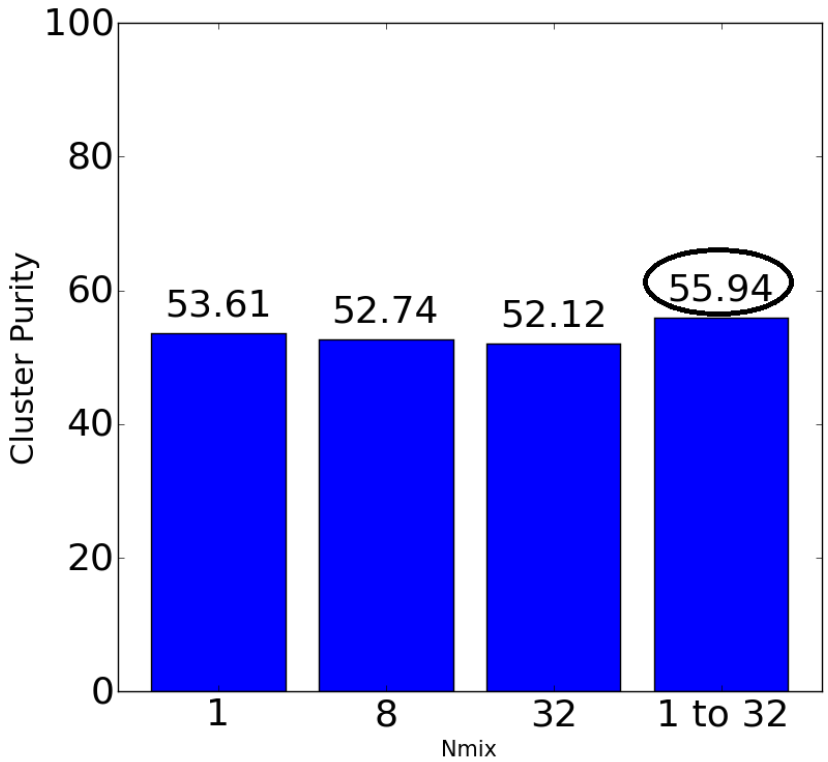
- Training sequences of TIMIT database used
- Number of phonemes used: 48 (folded from original 61)

Simulations divided into 3 main classes:

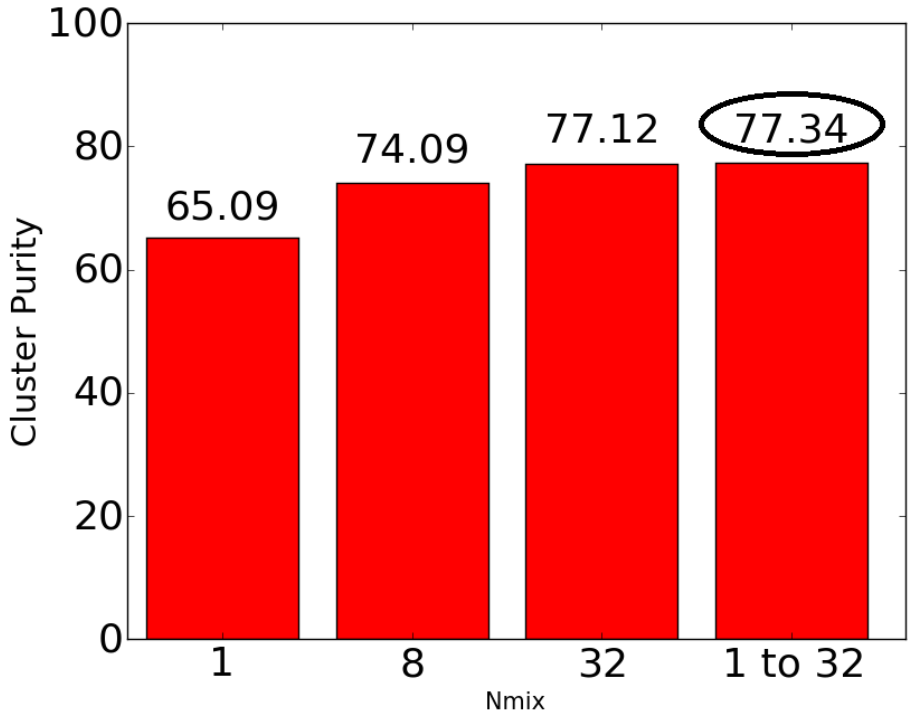
- Known number of HMMs with pre-segmented data
- Unknown number of HMMs with pre-segmented data
- Unknown number of HMMs with unsegmented data

Known HMM count: Effect of increasing mixture components

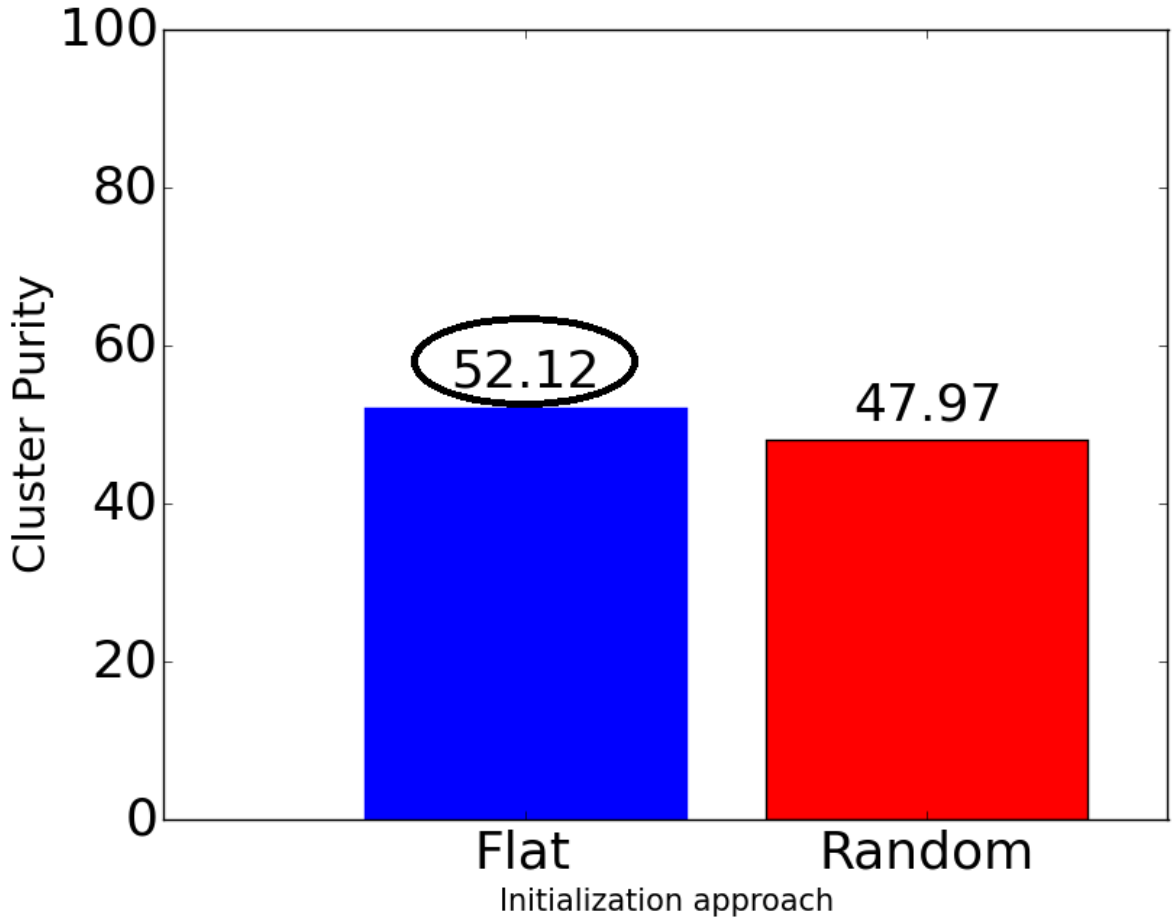
Unsupervised:



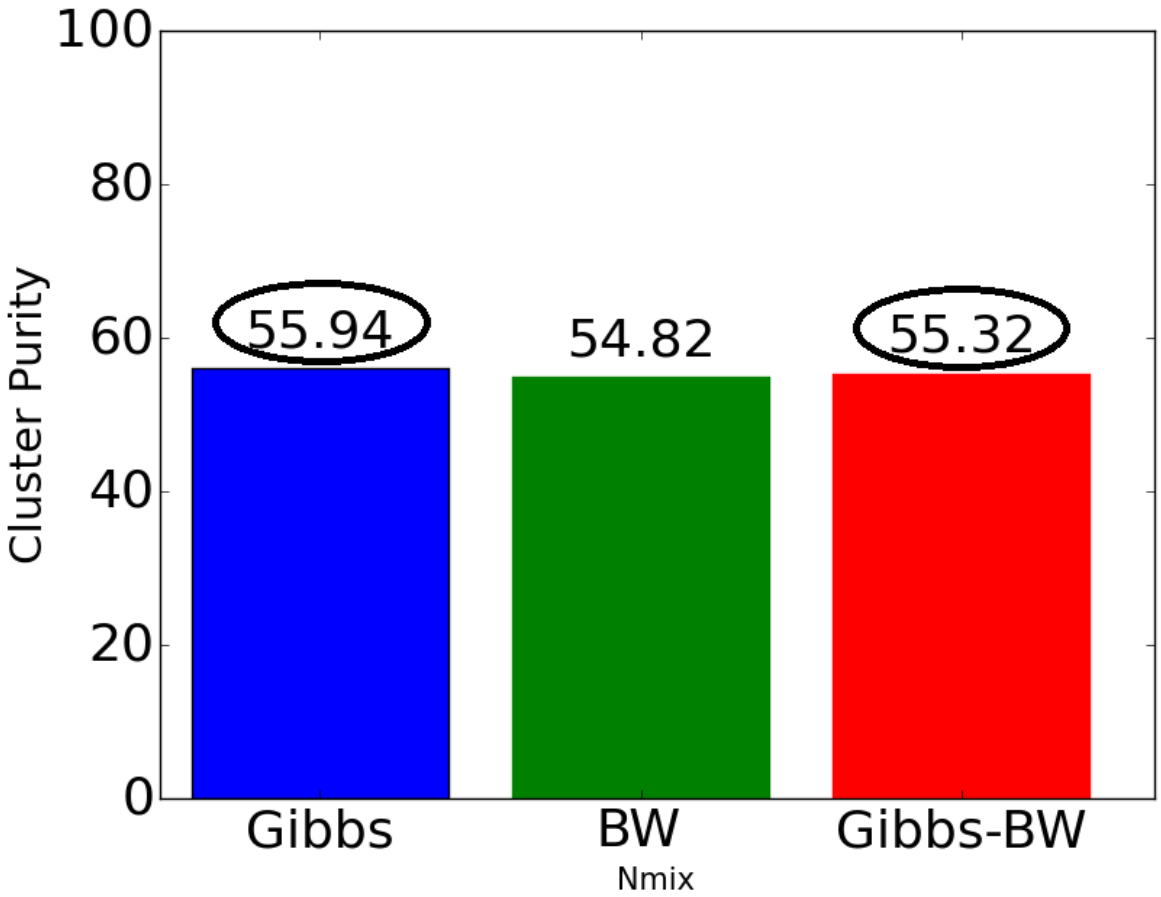
Supervised:



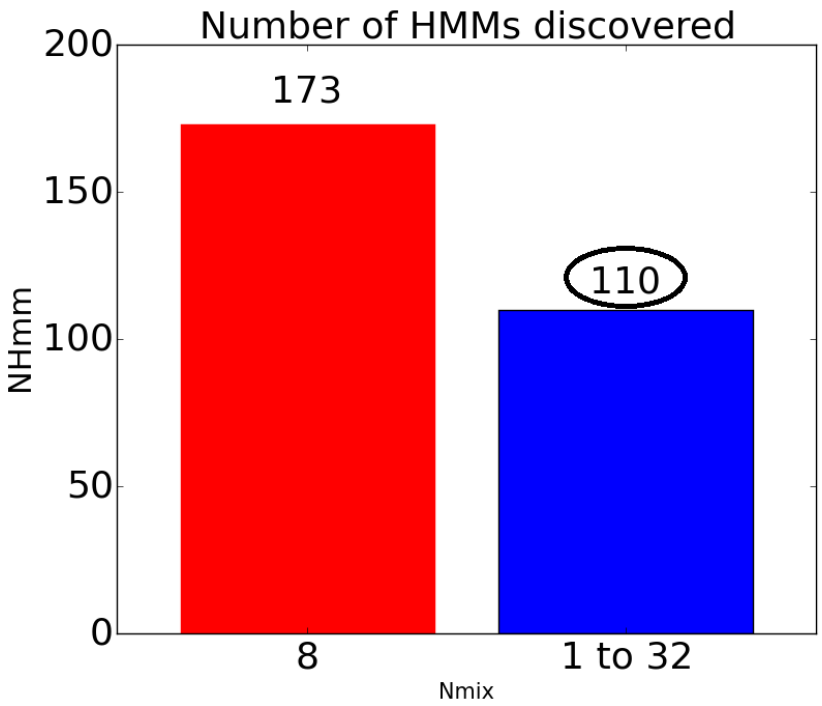
Known HMM count: Initialization method with $Nmix = 32$

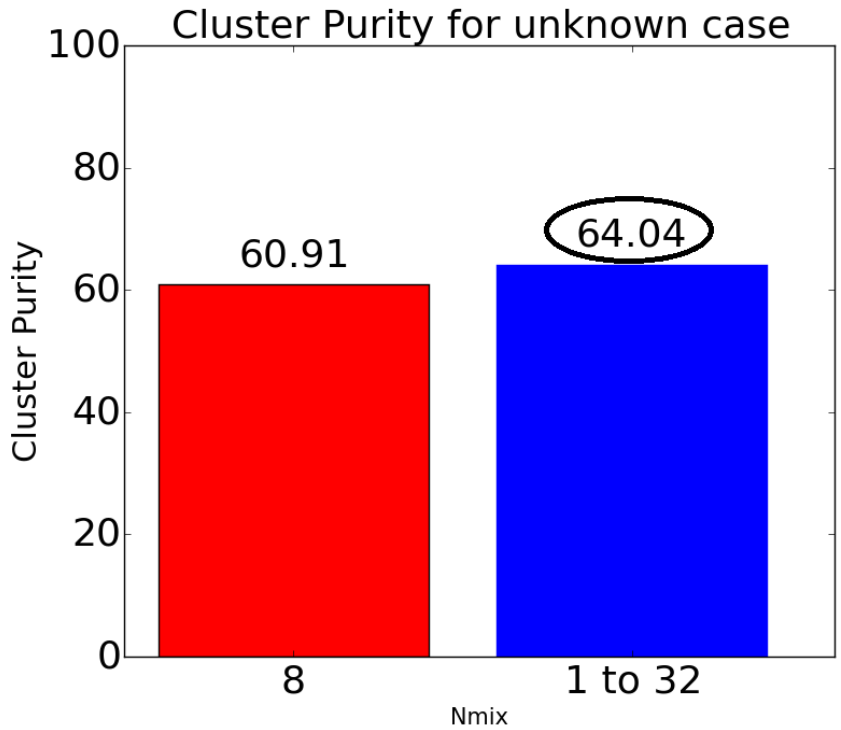


Known HMM count: Effect of learning algorithm with $Nmix = 1$ to 32



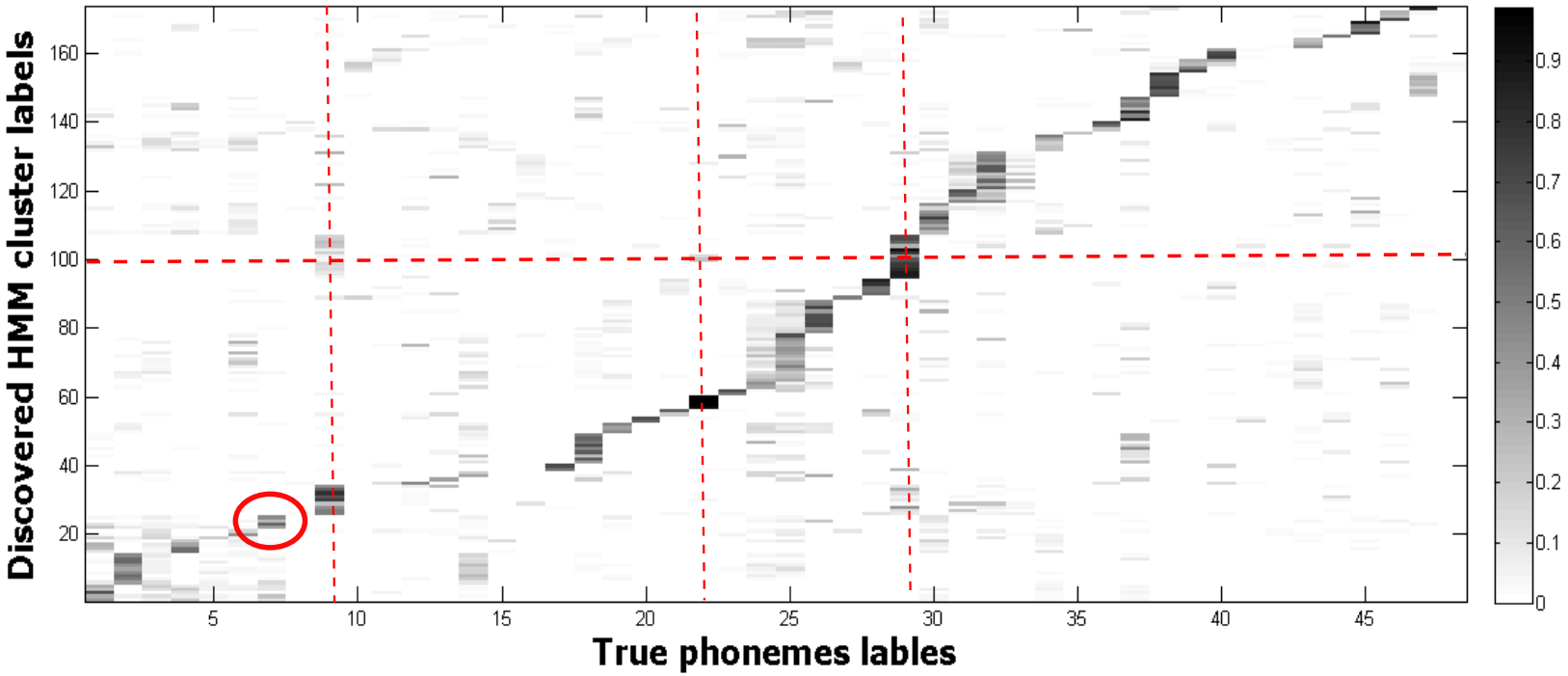
Results: Unknown number of HMMs with pre-segmented data





Unknown number of HMMs with pre-segmented data for $Nmix = 8$

- One cluster contains sequences from different phonemes
- Correlation between cluster labels



Results: Unknown number of HMMs with unsegmented data

- Segmentation performance:

$$recall = \frac{\text{number of correctly detected segmentation points}}{\text{number of true segmentation points}}$$

$$Oversegmentation(O_s) = \frac{\text{Number of segmentation points obtained}}{\text{Total number of true segmentation points}} \times 100$$

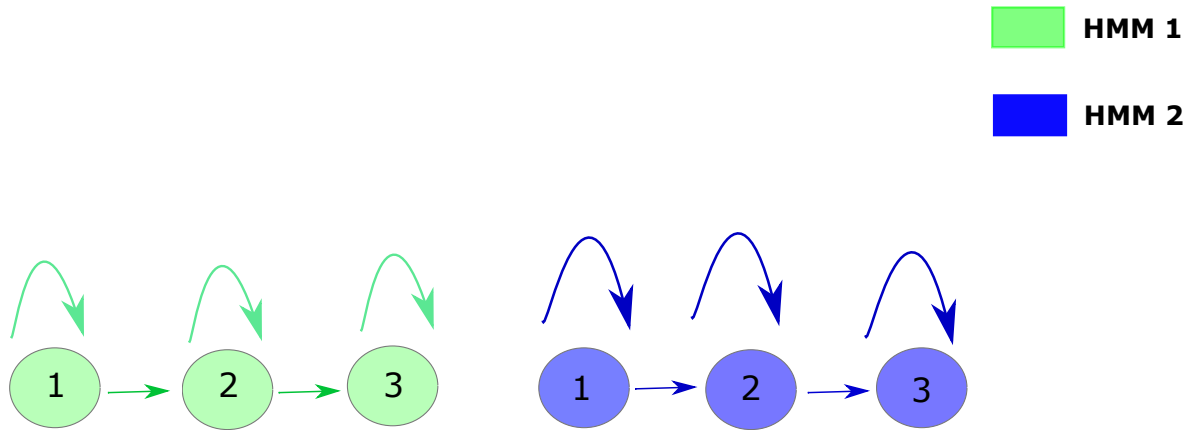
- Presegmentation algorithm performance:
  - Recall = 41.38%
  - Oversegmentation: 121.8 %
- Segmentation algorithm performance:
  - Recall = 6.51%
  - Oversegmentation = 14.12%
- Sequences kept getting longer with iterations

Conclusions

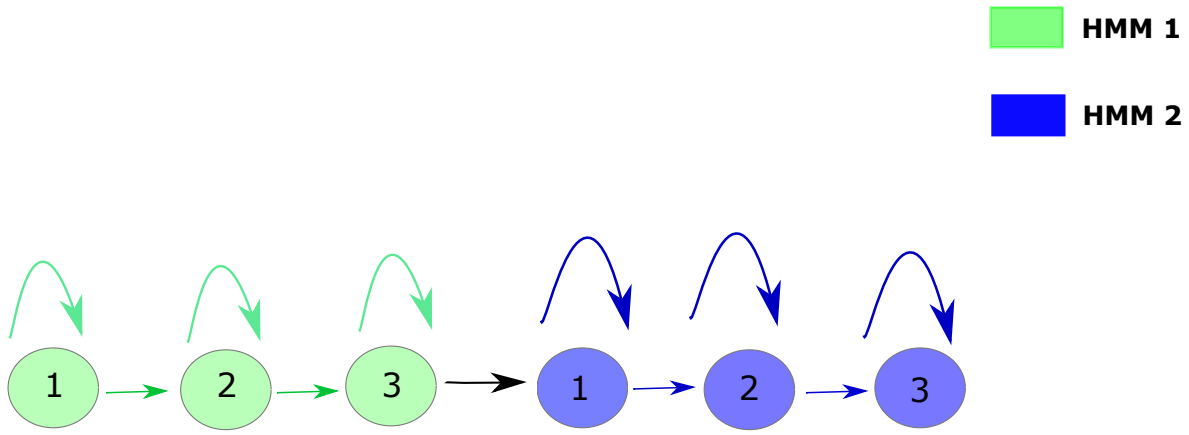
- Flat start better than random initialization
- Gibbs sampling, mixed learning approach works better than Baum Welch
- Gradual increase of GMM mixtures improve performance
- Current experiments with unsegmented data show increasing sequence lengths

Thanks!

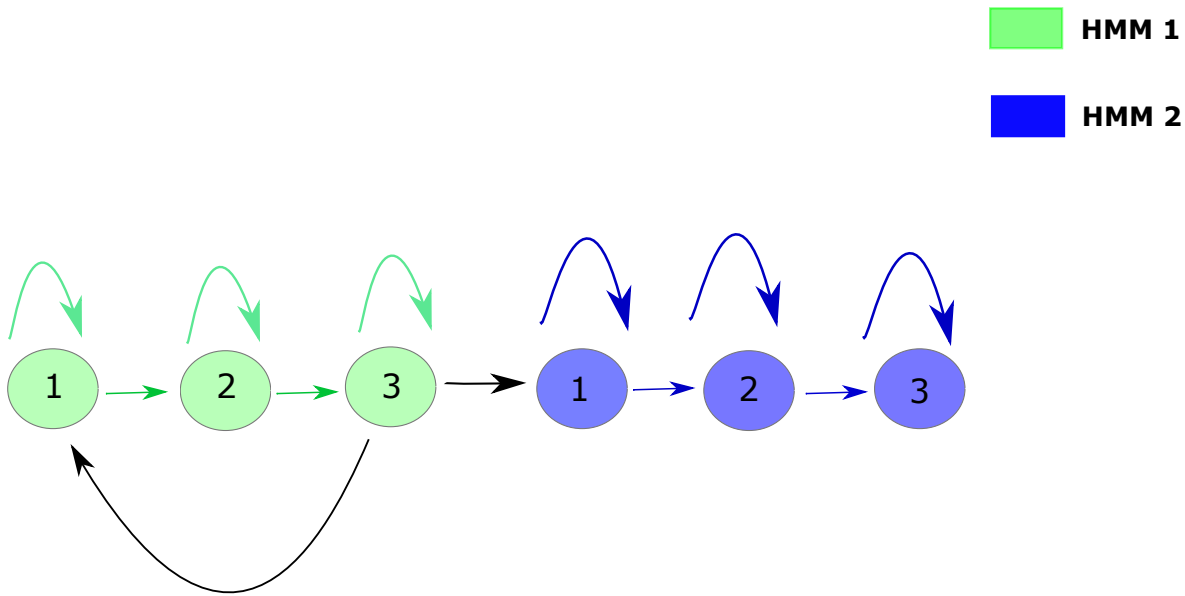
Questions? (Please ask easy ones!)


HMM 1
HMM 2
1
2
3
1
2
3

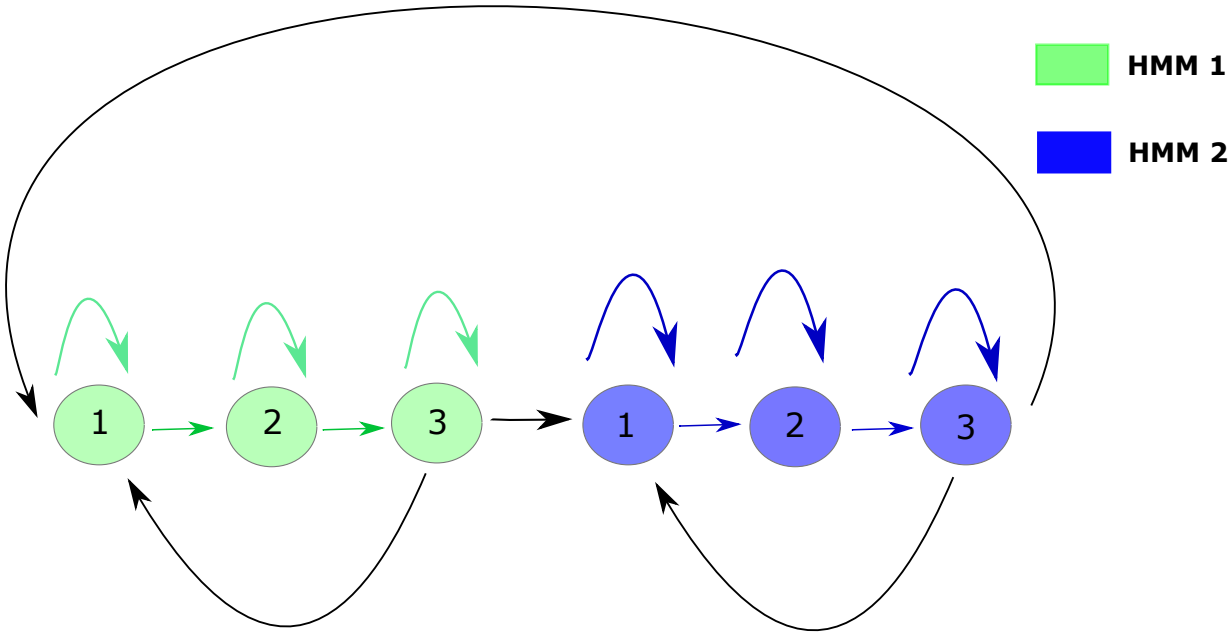
Segmentation



Segmentation



Segmentation



Segmentation

$$\Omega_{cas} = \{1, 2, 3, 4, 5, 6\}$$

$$\begin{aligned} \underline{\Pi}_{cas} &= \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\} \\ &= \{\epsilon_1, 0, 0, \epsilon_2, 0, 0\} \end{aligned}$$

$$\mathbf{A}_{cas} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 & 0 \\ \epsilon_1 \cdot (1 - z_1) & 0 & z_1 & \epsilon_2 \cdot (1 - z_1) & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 \\ 0 & 0 & 0 & 0 & a_{55} & a_{56} \\ \epsilon_1 \cdot (1 - z_2) & 0 & 0 & \epsilon_2 \cdot (1 - z_2) & 0 & z_2 \end{bmatrix}$$

$$b_t = \begin{cases} 1, & \text{if } mod(s_t, N) = 0 \text{ and } mod(s_{t+1}, N) = 1 \\ 0, & \text{otherwise} \end{cases}$$

$nmix = 8$,learning,unsupervised

Algorithm	Cluster purity (%)
Gibbs	52.74
Baum-Welch	53.39
Gibbs-BW	54.82

Table: Unsupervised case: Cluster purity values for different learning algorithms with $N_m = 8$.

supervised,nmix variation

N_m	CP validation(%)	CP training(%)
1	65.09	65.61
8	74.09	76.23
32	77.12	83.83
1 to 32	77.34	83.55

Table: Supervised case: Cluster purity values for varying number of GMM mixture components per state for all HMMs.

supervised, $nmix$ = 1to 32,learning algo

Algorithm	Cluster purity training (%)	Cluster purity validation (%)
Gibbs	83.55	77.34
Baum-Welch	84.88	77.64

Table: Supervised case: Cluster purity values for different learning algorithms with $N_m = 1$ to 32.