# Disclaimer

- The material provided in this document is not my original work and is a summary of some one else’s work(s).
- A simple Google search of the title of the document will direct you to the original source of the material.
- I do not guarantee the accuracy, completeness, timeliness, validity, non-omission, merchantability or fitness of the contents of this document for any particular purpose.
- 

Chapter # 8: HMM Parameter Estimation

# HMM TOOL KIT HTK

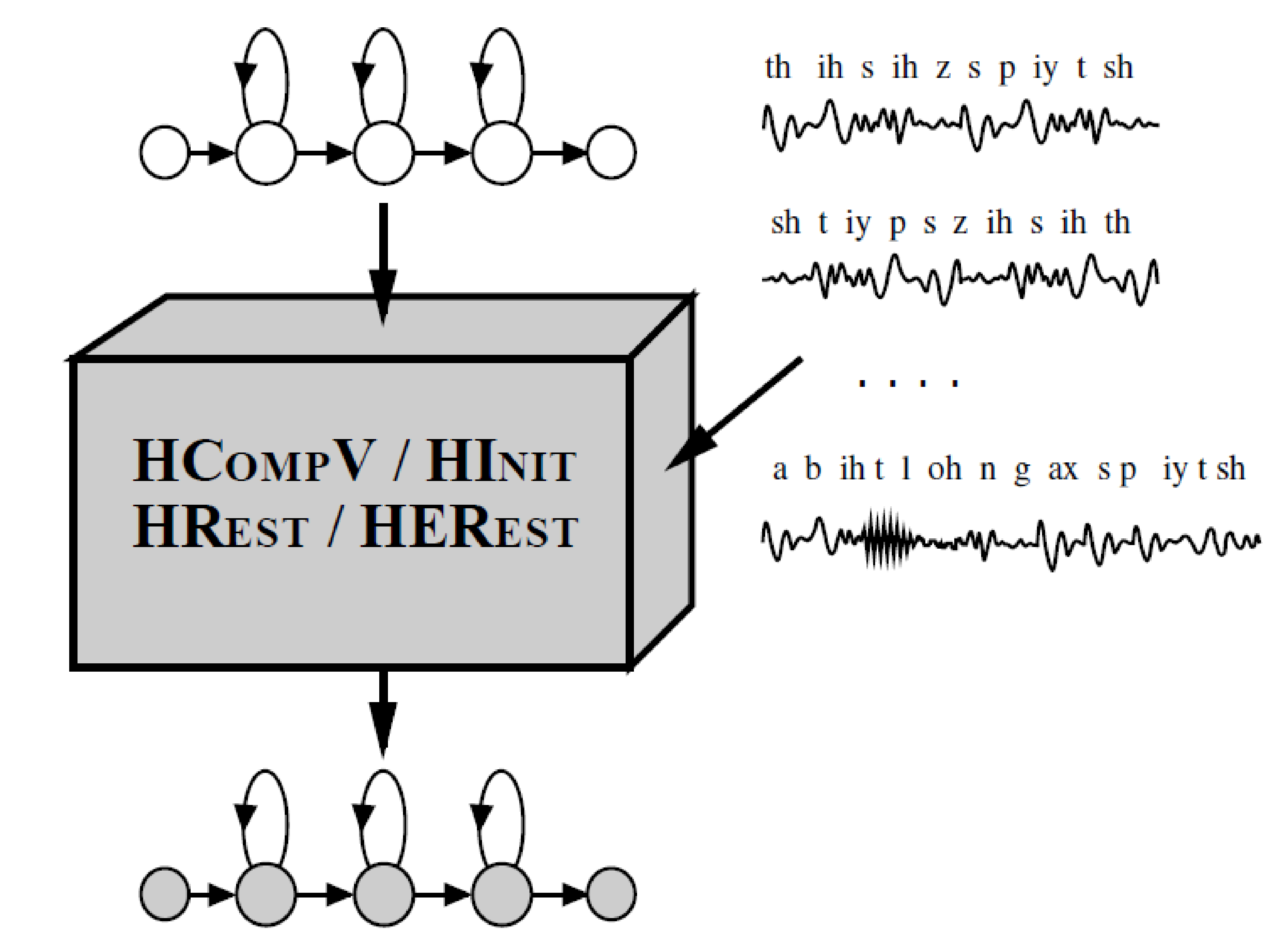
# Outline

- Introduction
- Training Strategies
- Initialization using Hinit
- Flat Starting with HCompV
- Isolated Unit Re-Estimation using Hrest
- Embedded Training using HERest
- Single-Pass Retraining
- Two-model Re-Estimation
- Parameter Re-Estimation Formulae

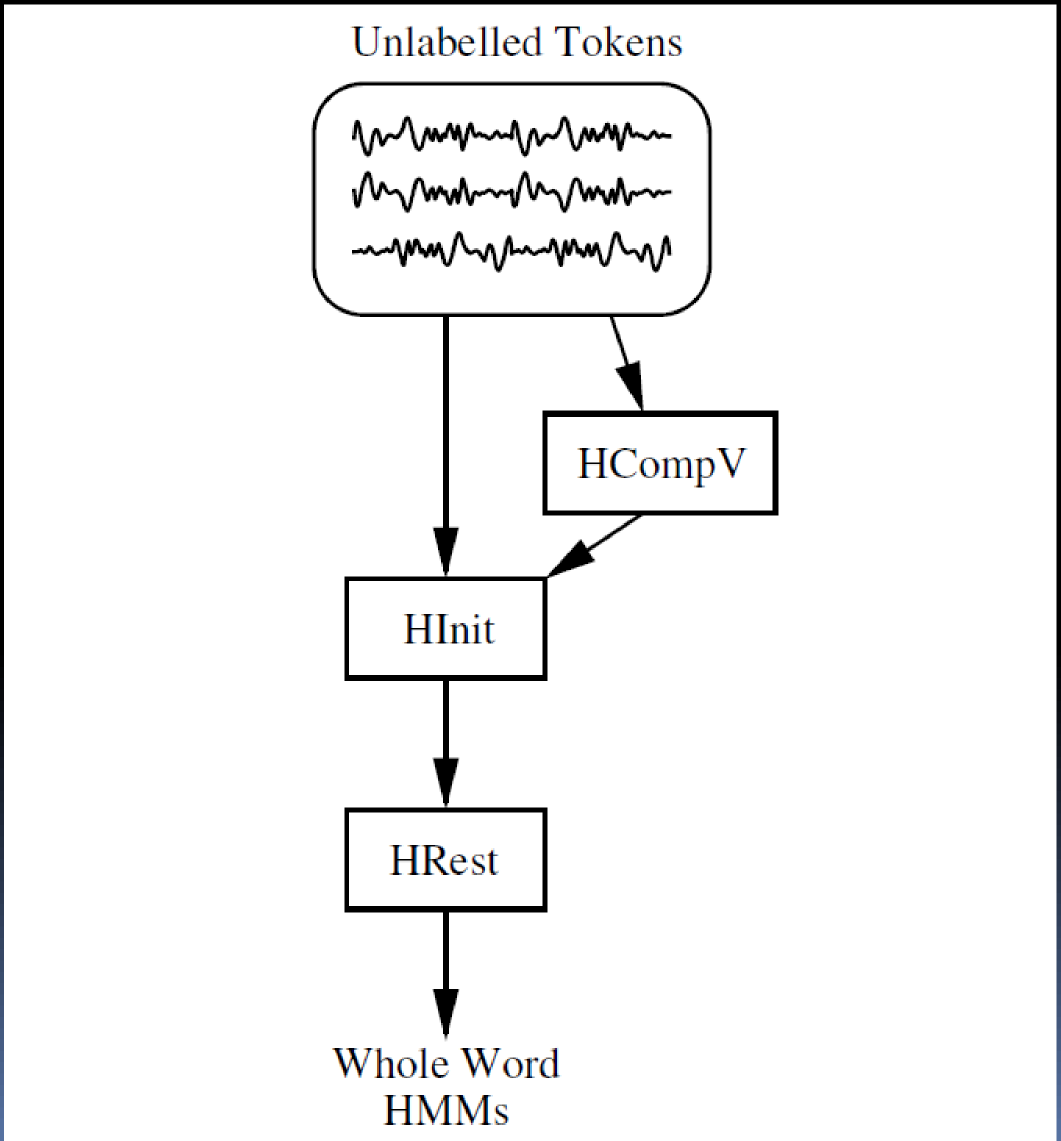
# Introduction

- Defining the structure and overall form of a set of HMMs is the first step towards building a recognizer
- The second step is to estimate the parameters of the HMMs from examples of the data sequences that they are intended to model
- HTK supplies four basic tools for parameter estimation
  - HCompV will set the mean and variance to be equal to the global mean and variance of the speech training data
  - HInit will compute the parameters of a new HMM using a Viterbi style of estimation
  - HRest and HERest are used to refine the parameters of existing HMMs using Baum-Welch Re-estimation

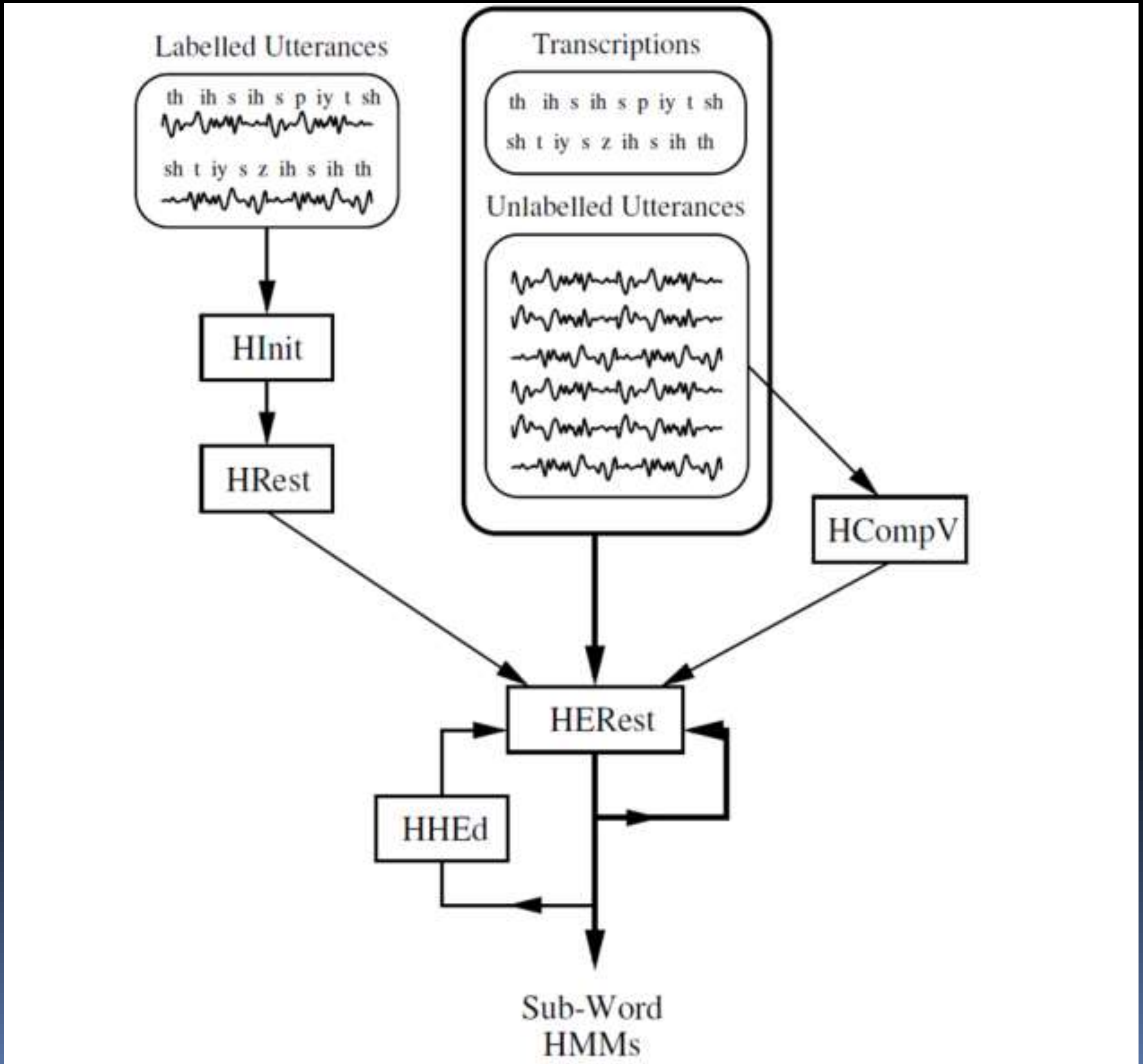
# Introduction



# Training Strategies



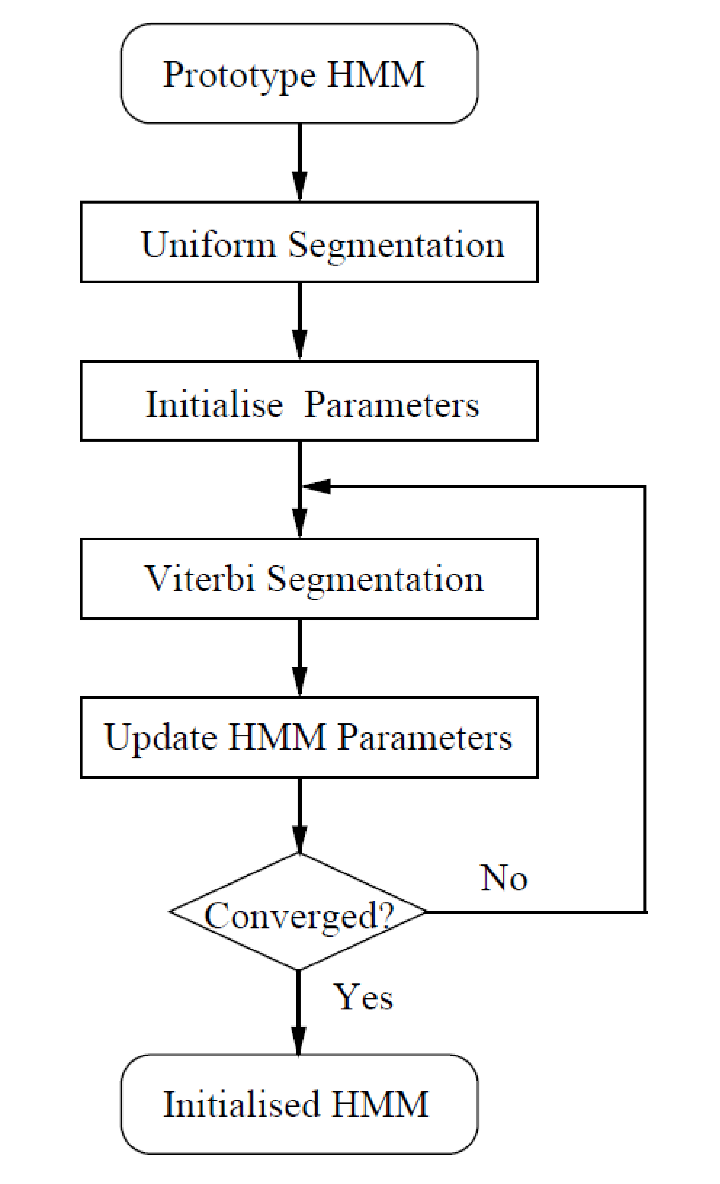
# Training Strategies



# Initialization using HInit

- The basic principle of HInit depends on the concept of a HMM as a generator of speech vectors
- Every training example can be viewed as the output of the HMM whose parameters are to be estimated
- If the state that generated each vector in the training data was known, then the unknown means and variances could be estimated by averaging all the vectors associated with each state
- The transition matrix could be estimated by simply counting the number of time slots that each state was occupied

# Initialization using HInit



# Initialization using HInit

- If any HMM state has multiple mixture components, then the training vectors are associated with the mixture component with the highest likelihood
- The number of vectors associated with each component within a state can then be used to estimate the mixture weights
- In the uniform segmentation stage, a K-means clustering algorithm is used to cluster the vectors within each state

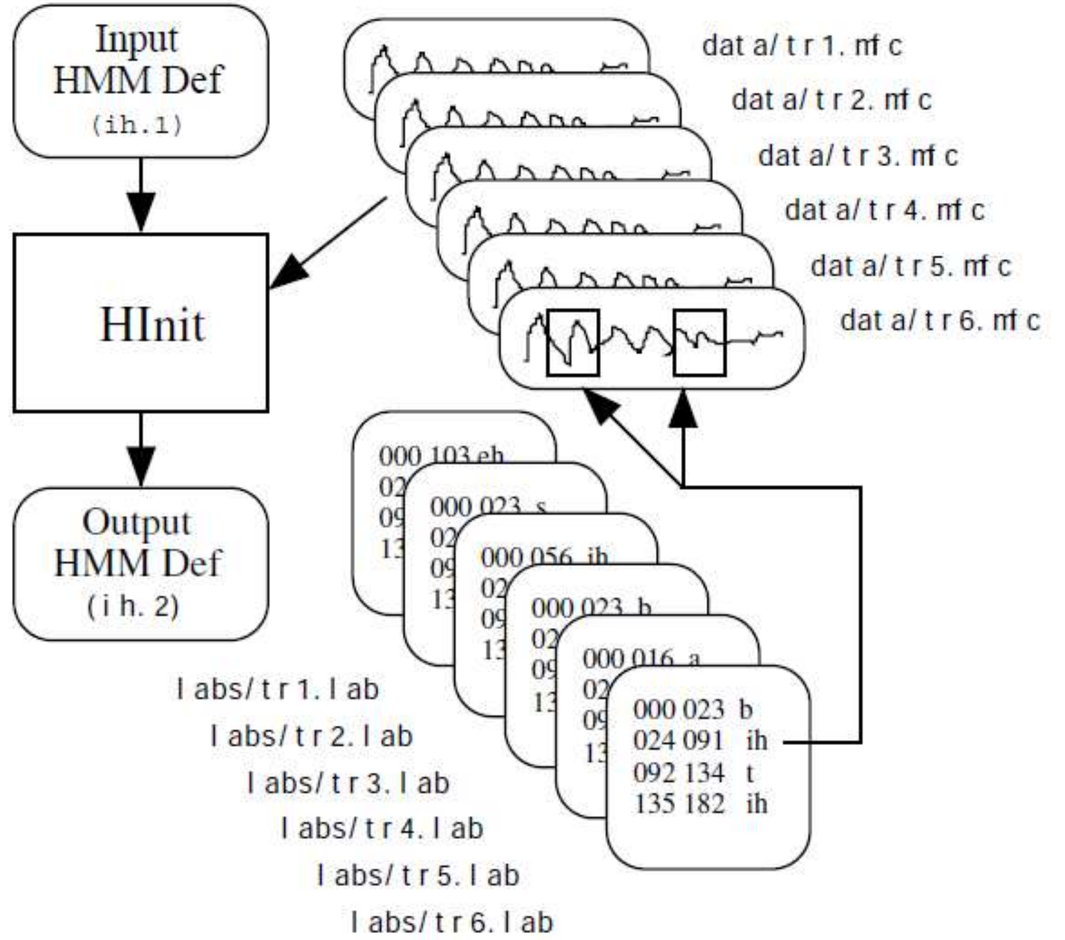
# Initialization using HInit

```
HInit hmm data1 data2 data3

HInit -H mac1 -H mac2 hmm data1 data2 data3 ...

HInit -H globals -M dir1 proto data1 data2 data3 ...
mv dir1/proto dir1/wordX
```

# Initialization using HInit

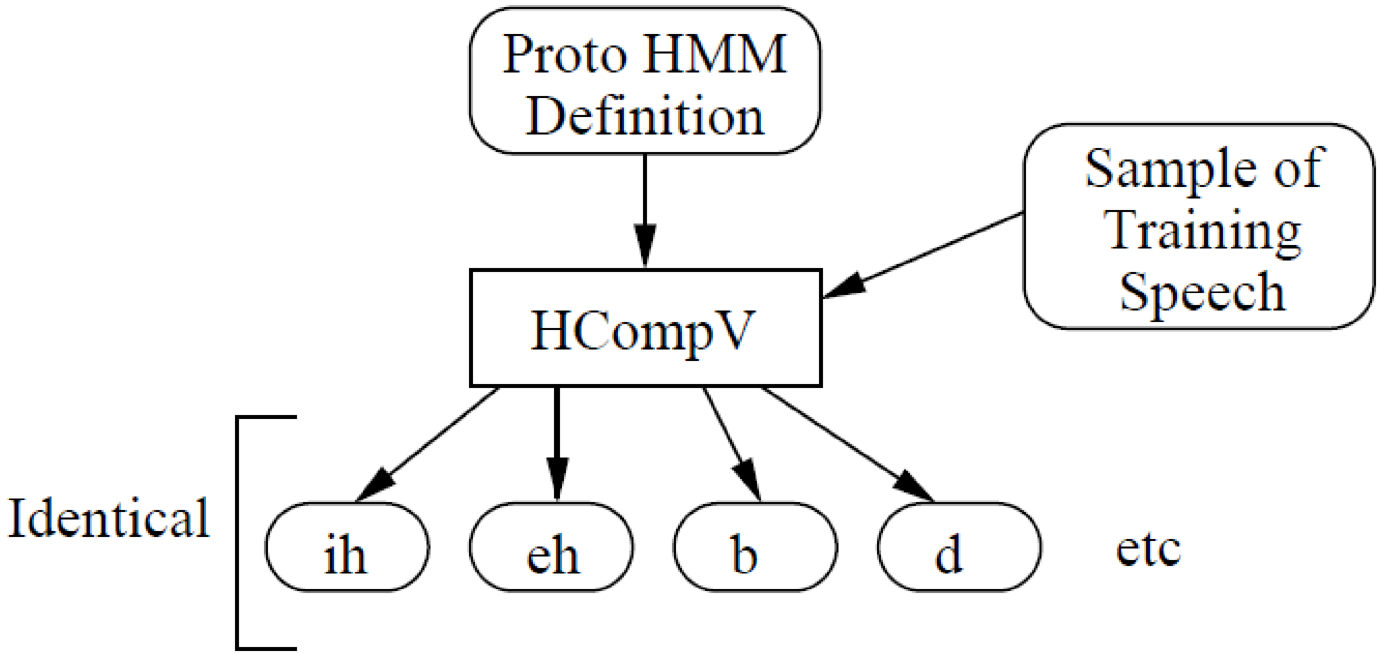



```
HInit -S trainlist -H globals -M dir1 -l ih -L labs proto
mv dir1/proto dir1/ih
```

# Initialization using HInit

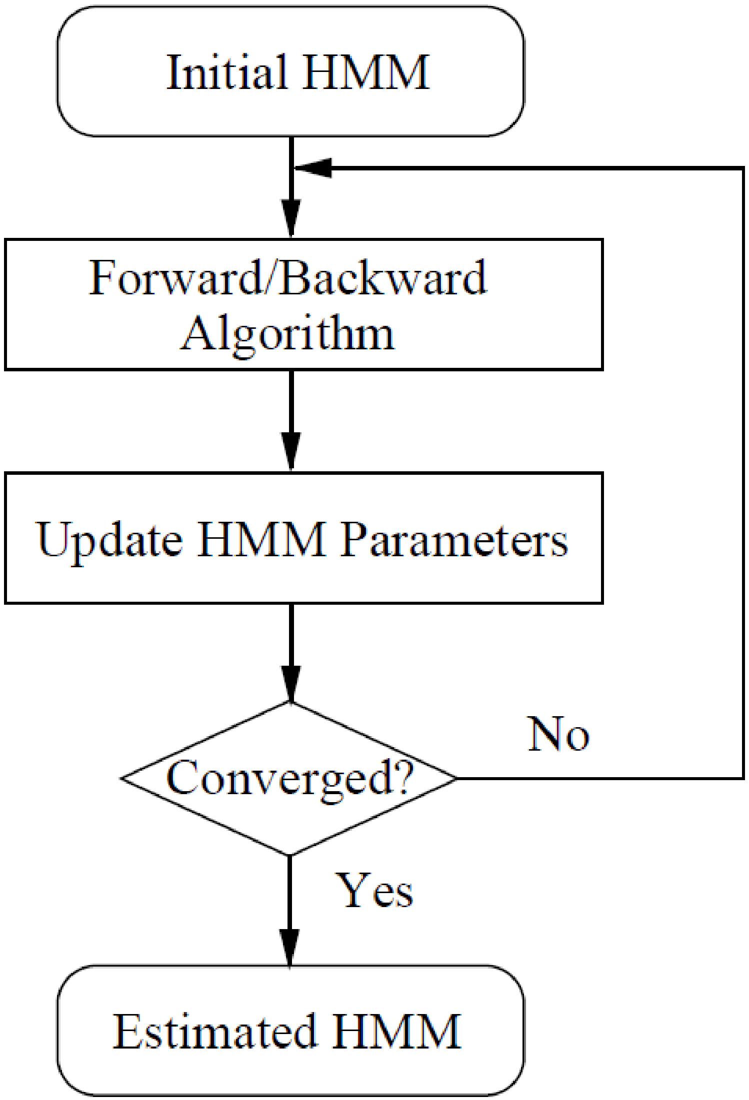
```
Initialising  HMM proto . . .
 States     :   2  3  4 (width)
 Mixes   s1:   1  1  1 ( 26  )
 Num Using:   0  0  0
 Parm Kind:  MFCC_E_D
 Number of owners = 1
 SegLab     :  ih
 maxIter    :  20
 epsilon    :  0.000100
 minSeg     :  3
 Updating :  Means Variances MixWeights/DProbs TransProbs
16 Observation Sequences Loaded
Starting Estimation Process
Iteration 1: Average LogP =  -898.24976
Iteration 2: Average LogP =  -884.05402  Change =    14.19574
Iteration 3: Average LogP =  -883.22119  Change =     0.83282
Iteration 4: Average LogP =  -882.84381  Change =     0.37738
Iteration 5: Average LogP =  -882.76526  Change =     0.07855
Iteration 6: Average LogP =  -882.76526  Change =     0.00000
Estimation converged at iteration 7
Output written to directory :dir1:
```

# Flat Starting with HCompV




```
HCompV -m -H globals -M dir1 proto data1 data2 data3 ...
```

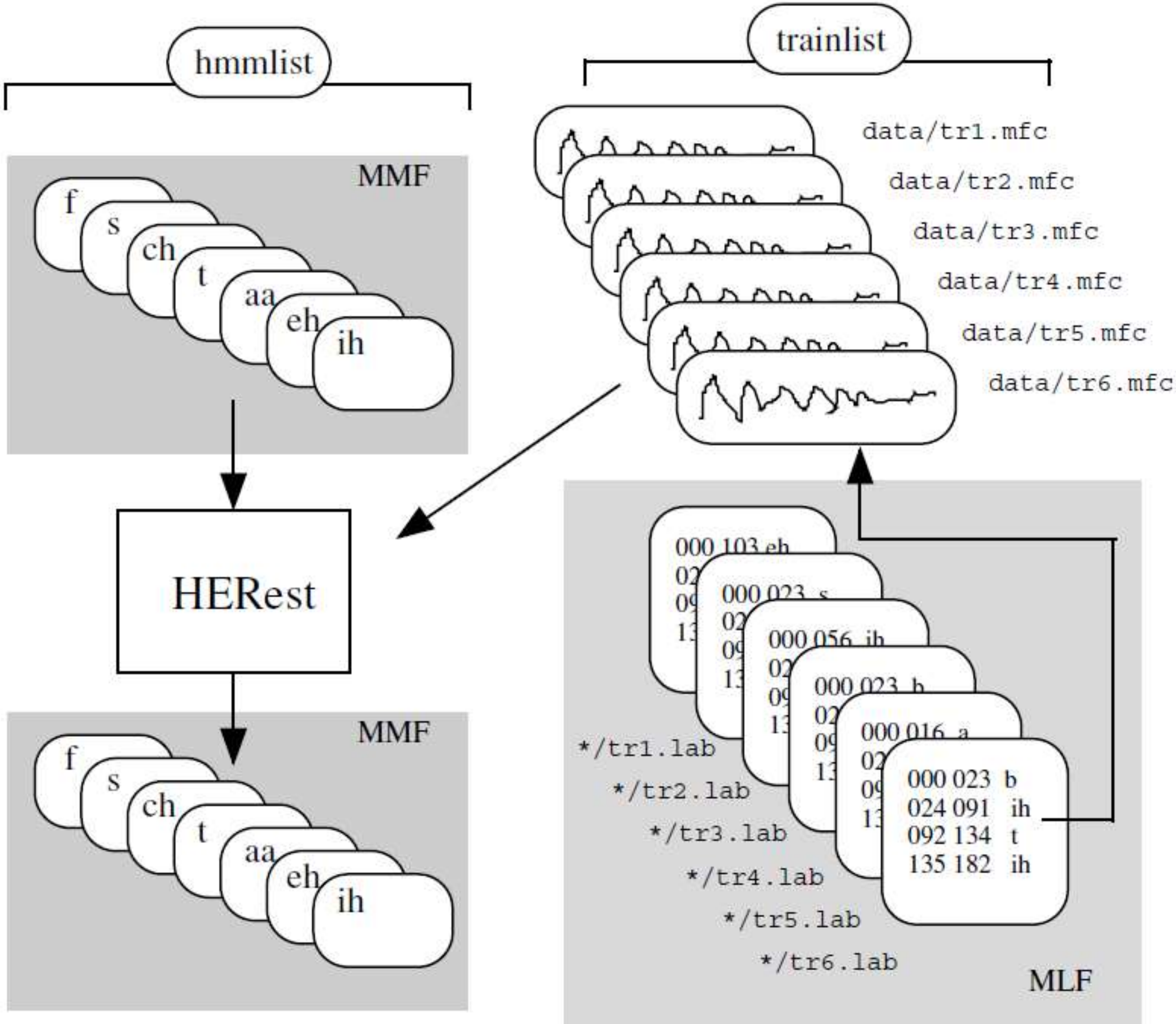
# Isolated Unit Re-Estimation using HRest




```
HRest -S trainlist -H dir1/globals -M dir2 -l ih -L labs dir1/ih
```

# Embedded Training using HERest

- Unlike the processes described so far, embedded training simultaneously updates all of the HMMs in a system using all of the training data
  - HERest loads in a complete set of HMM definitions
  - Every training file must have an associated label file which gives a transcription for that file
  - This composite HMM is made by concatenating instances of the phone HMMs corresponding to each label in the transcription
  - The Forward-Backward algorithm is then applied and the sums needed to form the weighted averages accumulated in the normal way




```
HERest -S trainlist -I labs -H dir1/hmacs -M dir2 hmmlist
```

# Embedded Training using HERest

- In order to get accurate acoustic models, a large amount of training data is needed
- There are two mechanisms for speeding up this computation
  - Beam Pruning
  - Parallel Computing
- Beam Pruning: HERest has a pruning mechanism incorporated into its forward-backward computation
- On the forward pass, HERest restricts the computation of the values $\alpha(t)$ to just those for which the total log likelihood as determined by the product $\alpha(t)\beta(t)$ is within a fixed distance from the total likelihood $P(O_j|M)$

# Embedded Training using HERest

- In order to get accurate acoustic models, a large amount of training data is needed
- There are two mechanisms for speeding up this computation
  - Beam Pruning

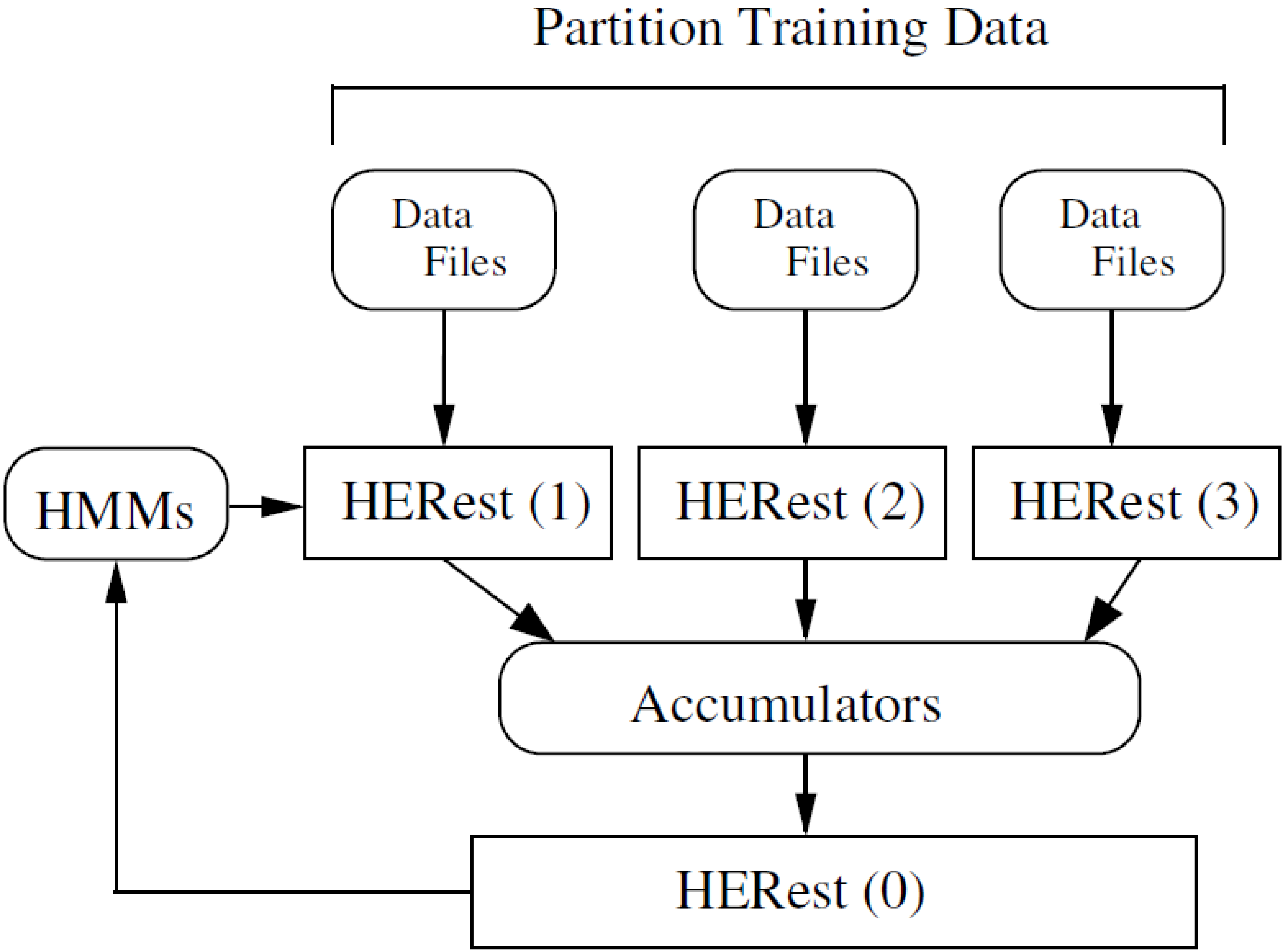
```
HERest -t 120.0 60.0 240.0 -S trainlist -I labs \
       -H dir1/hmacs -M dir2 hmmlist
```

incorporated into its forward-backward computation

- On the forward pass, HERest restricts the computation of the values α(t) to just those for which the total log likelihood as determined by the product α(t)β(t) is within a fixed distance from the total likelihood $P(O_j|M)$

# Embedded Training using HERest

- Parallel Computing




```
HERest -S trlist1 -I labs -H dir1/hmacs -M dir2 -p 1 hmmlist
HERest -S trlist2 -I labs -H dir1/hmacs -M dir2 -p 2 hmmlist

  HERest -H dir1/hmacs -M dir2 -p 0 hmmlist dir2/*.acc
```

# Single-Pass Retraining

- Suppose that a set of models has been trained on data with MFCC_E_D parameterization
- A new set of models using Cepstral Mean Normalization ( _Z) is required

```
# Single pass retraining
HPARM1: TARGETKIND = MFCC_E_D
HPARM2: TARGETKIND = MFCC_E_D_Z
```

```
HERest -r -C config -S trainList -I labs -H dir1/hmacs -M dir2 hmmList
```

# Two-model Re-Estimation

- Suppose that we would like to update a set of cloned single Gaussian monophone models in dir1/hmacs using the well trained state-clustered triphones in dir2/hmacs as alignment models
- Associated with each model set are the model lists hmmlist1 and hmmlist2 respectively

```
# alignment model set for two-model re-estimation
ALIGNMODELMMF = dir2/hmacs
ALIGNHMMLIST  = hmmlist2
HERest -C config -C config.2model -S trainlist -I labs -H dir1/hmacs -M dir3 hmmlist1
```

# Formulas

- Viterbi Training

$$\phi_N(T) = \max_i \phi_i(T) a_{iN}$$

$$\phi_j(t) = \left[\max_i \phi_i(t-1) a_{ij}\right] b_j(\boldsymbol{o}_t)$$

$$\phi_1(1) = 1$$

$$\phi_j(1) = a_{1j} b_j(\boldsymbol{o}_1)$$

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{k=2}^{N} A_{ik}}$$

$$\hat{\boldsymbol{\mu}}_{jsm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t) \boldsymbol{o}^r_{st}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t)}$$

$$\hat{\boldsymbol{\Sigma}}_{jsm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t) (\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})(\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})'}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t)}$$

$$\boldsymbol{c}_{jsm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{l=1}^{M_s} \psi^r_{jsl}(t)}$$

# Formulas

- Forward Backward Probabilities

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right] b_j(\boldsymbol{o}_t)$$

$$\alpha_1(1) = 1$$

$$\alpha_j(1) = a_{1j}b_j(\boldsymbol{o}_1) \quad \text{for } 1 < j < N$$

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN}$$

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_j(t+1)$$

$$\beta_i(T) = a_{iN}$$

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j}b_j(\boldsymbol{o}_1)\beta_j(1)$$

# Formulas

- Forward Backward Probabilities

Forward Probability

initial conditions at time $t = 1$

$$\alpha_1^{(q)}(1) = \begin{cases} 1 & \text{if } q = 1 \\ \alpha_1^{(q-1)}(1) a_{1N_{q-1}}^{(q-1)} & \text{otherwise} \end{cases}$$

$$\alpha_j^{(q)}(1) = a_{1j}^{(q)} b_j^{(q)}(\boldsymbol{o}_1)$$

$$\alpha_{N_q}^{(q)}(1) = \sum_{i=2}^{N_q-1} \alpha_i^{(q)}(1) a_{iN_q}^{(q)}$$

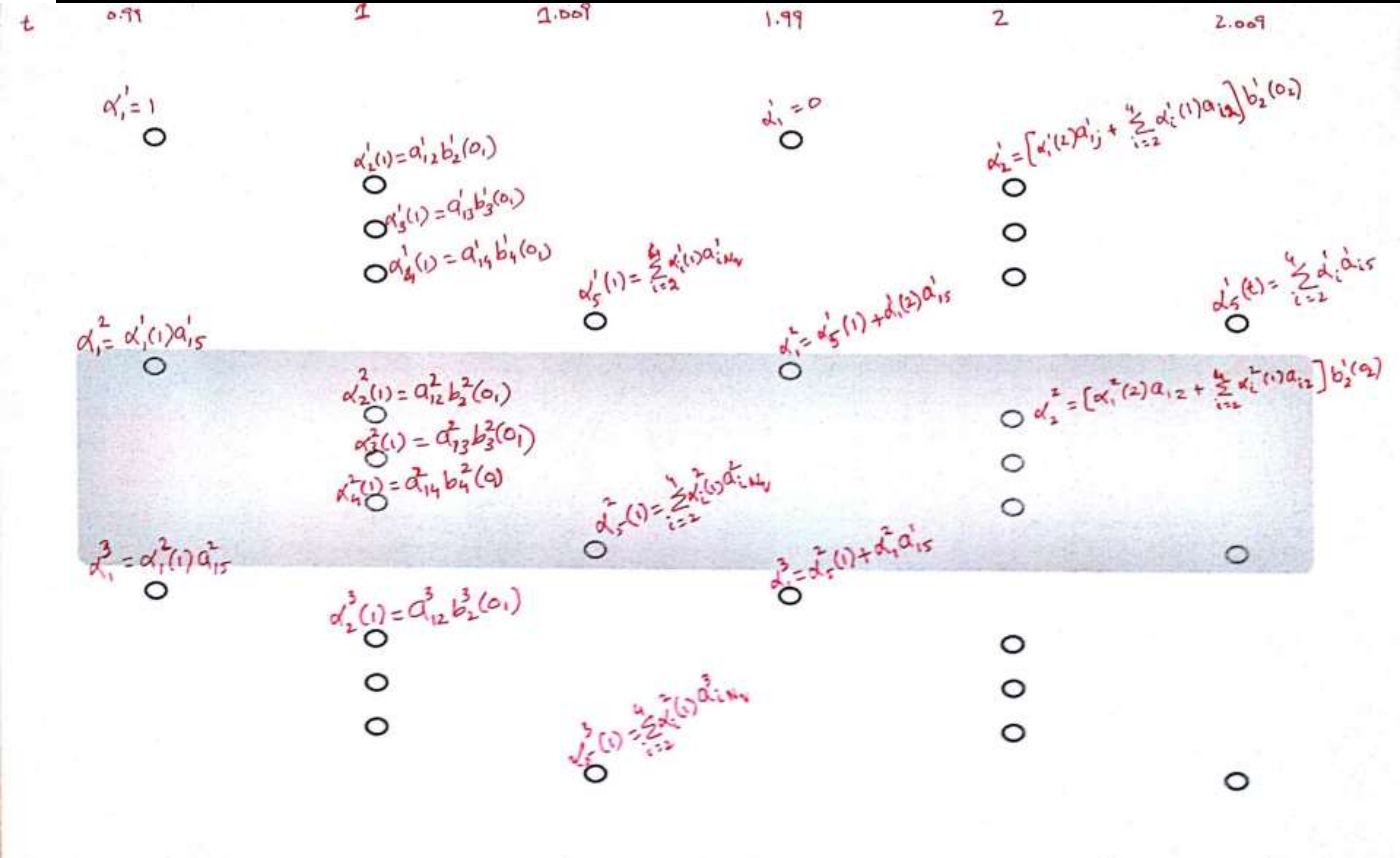
For time $t > 1$

$$\alpha_1^{(q)}(t) = \begin{cases} 0 & \text{if } q = 1 \\ \alpha_{N_{q-1}}^{(q-1)}(t-1) + \alpha_1^{(q-1)}(t) a_{1N_{q-1}}^{(q-1)} & \text{otherwise} \end{cases}$$

$$\alpha_j^{(q)}(t) = \left[ \alpha_1^{(q)}(t) a_{1j}^{(q)} + \sum_{i=2}^{N_q-1} \alpha_i^{(q)}(t-1) a_{ij}^{(q)} \right] b_j^{(q)}(\boldsymbol{o}_t)$$

$$\alpha_{N_q}^{(q)}(t) = \sum_{i=2}^{N_q-1} \alpha_i^{(q)}(t) a_{iN_q}^{(q)}$$

# Formulas



t
0.99
1
1.009
1.99
2
2.009
$\alpha_1^1 = 1$
$\alpha_2^1(1) = a_{12}^1 b_2^1(o_1)$
$\alpha_3^1(1) = a_{13}^1 b_3^1(o_1)$
$\alpha_4^1(1) = a_{14}^1 b_4^1(o_1)$
$\alpha_5^1(1) = \sum_{i=2}^{4} \alpha_i^1(1) a_{iN_4}^1$
$\alpha_1^1 = 0$
$\alpha_2^1 = [\alpha_1^1(2) a_{1j}^1 + \sum_{i=2}^{4} \alpha_i^1(1) a_{i2}] b_2^1(o_2)$
$\alpha_5^1(t) = \sum_{i=2}^{4} \alpha_i^1 a_{i5}^1$
$\alpha_1^2 = \alpha_1^1(1) a_{15}^1$
$\alpha_1^2 = \alpha_5^1(1) + \alpha_1(2) a_{15}^1$
$\alpha_2^2(1) = a_{12}^2 b_2^2(o_1)$
$\alpha_3^2(1) = a_{13}^2 b_3^2(o_1)$
$\alpha_4^2(1) = a_{14}^2 b_4^2(o_1)$
$\alpha_5^2(1) = \sum_{i=2}^{4} \alpha_i^2(1) a_{iN_4}^2$
$\alpha_2^2 = [\alpha_1^2(2) a_{12} + \sum_{i=2}^{4} \alpha_i^2(1) a_{i2}] b_2^1(o_2)$
$\alpha_1^3 = \alpha_1^2(1) a_{15}^2$
$\alpha_1^3 = \alpha_5^2(1) + \alpha_1^2 a_{15}^1$
$\alpha_2^3(1) = a_{12}^3 b_2^3(o_1)$
$\alpha_5^3(1) = \sum_{i=2}^{4} \alpha_i^2(1) a_{iN_4}^3$

# Formulas

- Forward Backward Probabilities

Backward Probability

initial conditions at time $t = T$

$$\beta_{N_q}^{(q)}(T) = \begin{cases} 1 & \text{if } q = Q \\ \beta_{N_{q+1}}^{(q+1)}(T)a_{1N_{q+1}}^{(q+1)} & \text{otherwise} \end{cases}$$

$$\beta_i^{(q)}(T) = a_{iN_q}^{(q)}\beta_{N_q}^{(q)}(T)$$

$$\beta_1^{(q)}(T) = \sum_{j=2}^{N_q-1} a_{1j}^{(q)} b_j^{(q)}(\boldsymbol{o}_T)\beta_j^{(q)}(T)$$

For time $t < T$,

$$\beta_{N_q}^{(q)}(t) = \begin{cases} 0 & \text{if } q = Q \\ \beta_1^{(q+1)}(t+1) + \beta_{N_{q+1}}^{(q+1)}(t)a_{1N_{q+1}}^{(q+1)} & \text{otherwise} \end{cases}$$

$$\beta_i^{(q)}(t) = a_{iN_q}^{(q)}\beta_{N_q}^{(q)}(t) + \sum_{j=2}^{N_q-1} a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1})\beta_j^{(q)}(t+1)$$

$$\beta_1^{(q)}(t) = \sum_{j=2}^{N_q-1} a_{1j}^{(q)} b_j^{(q)}(\boldsymbol{o}_t)\beta_j^{(q)}(t)$$

# Formulas

- Forward Backward Probabilities
- The total probability P = prob(O|λ) can be computed from either the forward or backward probabilities

$$P = \alpha_N(T) = \beta_1(1)$$

# Formulas

- Single Model Reestimation (HRest)

transition probabilities

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^r(t) a_{ij} b_j(\boldsymbol{o}_{t+1}^r) \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)}$$

transitions from the non-emitting entry state

$$\hat{a}_{1j} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P_r} \alpha_j^r(1) \beta_j^r(1)$$

transitions from the emitting states to exit state

$$\hat{a}_{iN} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \alpha_i^r(T) \beta_i^r(T)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)}$$

probability of occupying the $m$'th mixture component in stream $s$ at time $t$ for the $r$'th observation

$$L_{jsm}^r(t) = \frac{1}{P_r} U_j^r(t) c_{jsm} b_{jsm}(\boldsymbol{o}_{st}^r) \beta_j^r(t) b_{js}^*(\boldsymbol{o}_t^r)$$

$$U_j^r(t) = \begin{cases} a_{1j} & \text{if } t = 1 \\ \sum_{i=2}^{N-1} \alpha_i^r(t-1) a_{ij} & \text{otherwise} \end{cases}$$

$$b_{js}^*(\boldsymbol{o}_t^r) = \prod_{k \neq s} b_{jk}(\boldsymbol{o}_{kt}^r)$$

For single Gaussian streams

$$L_{jsm}^r(t) = L_j^r(t) = \frac{1}{P_r} \alpha_j(t) \beta_j(t)$$

# Formulas

- Single Model Reestimation (HRest)

re-estimation formulae in terms of $L^r_{jsm}(t)$

$$\hat{\boldsymbol{\mu}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jsm}(t)\boldsymbol{o}^r_{st}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jsm}(t)}$$

$$\hat{\boldsymbol{\Sigma}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jsm}(t)(\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})(\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})'}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jsm}(t)}$$

$$c_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jsm}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{j}(t)}$$

transition probabilities

$$\hat{a}_{ij}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^{(q)r}(t) a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}^r) \beta_j^{(q)r}(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}$$

transitions from the non-emitting entry states into the HMM

$$\hat{a}_{1j}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_1^{(q)r}(t) a_{1j}^{(q)} b_j^{(q)}(\boldsymbol{o}_t^r) \beta_j^{(q)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_1^{(q)r}(t) \beta_1^{(q)r}(t) + \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}$$

transitions out of the HMM

$$\hat{a}_{iN_q}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^{(q)r}(t) a_{iN_q}^{(q)} \beta_{N_q}^{(q)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}$$

transitions from entry to exit states

$$\hat{a}_{1N_q}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t) + \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}$$

transition probabilities

$$\hat{a}_{ij}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^{(q)r}(t) a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}^r) \beta_j^{(q)r}(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}$$

transitions from the non-emitting entry states into the HMM

$$\hat{a}_{1j}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_1^{(q)r}(t) a_{1j}^{(q)} b_j^{(q)}(\boldsymbol{o}_t^r) \beta_j^{(q)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_1^{(q)r}(t) \beta_1^{(q)r}(t) + \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}$$

$$U_j^{(q)r}(t) = \begin{cases} \alpha_1^{(q)r}(t) a_{1j}^{(q)} & \text{if } t = 1 \\ \alpha_1^{(q)r}(t) a_{1j}^{(q)} + \sum_{i=2}^{N_q-1} \alpha_i^{(q)r}(t-1) a_{ij}^{(q)} & \text{otherwise} \end{cases}$$

$$\hat{a}_{iN_q}^{(q)} = \frac{\cdots}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}$$

transitions from entry to exit states

$$\hat{a}_{1N_q}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t) + \alpha_1^{(q)r}(t) a_{1N_q}^{(q)} \beta_1^{(q+1)r}(t)}$$

# Discriminative Training

- HTK supports discriminative training using HMMIRest tool
- Maximum Mutual Information and Minimum Phone Error criteria are supported
- The form of MMI criteria to be maximized may be expressed as

$$
\begin{aligned}
\mathcal{F}_{\mathtt{mmi}}(\lambda) &= \frac{1}{R}\sum_{r=1}^{R}\log\left(P(\mathcal{H}^r_{\mathtt{ref}}|O^r,\lambda)\right) \\
&= \frac{1}{R}\sum_{r=1}^{R}\log\left(\frac{P(O^r|\mathcal{H}^r_{\mathtt{ref}},\lambda)P(\mathcal{H}^r_{\mathtt{ref}})}{\sum_{\mathcal{H}}P(O^r|\mathcal{H},\lambda)P(\mathcal{H})}\right)
\end{aligned}
$$

# Discriminative Training

- For MPE the expression to be minimized is

$$\mathcal{F}_{\mathtt{mpe}}(\lambda) = \sum_{r=1}^{R} \sum_{\mathcal{H}} P(\mathcal{H}|O^r, \lambda)\mathcal{L}(\mathcal{H}, \mathcal{H}^r_{\mathtt{ref}})$$

$$\hat{\lambda} = \arg\max_{\lambda} \left\{ 1 - \frac{1}{\sum_{r=1}^{R} Q^r} \mathcal{F}_{\mathtt{mpe}}(\lambda) \right\}$$

# Discriminative Training

- In HMMIREST implementation the language model scores, including the grammar scale factor are combined into the acoustic models to yield a numerator acoustic model and a denominator acoustic model

$$\mathcal{F}_{\mathtt{mmi}}(\lambda) = \sum_{r=1}^{R} \log \left( \frac{P(O^r|\mathcal{M}_r^{\mathtt{num}})}{P(O^r|\mathcal{M}_r^{\mathtt{den}})} \right)$$

$$\mathcal{F}_{\mathtt{mpe}}(\lambda) = \sum_{r=1}^{R} \sum_{\mathcal{H}} \left( \frac{P(O^r|\mathcal{M}_{\mathcal{H}})}{P(O^r|\mathcal{M}_r^{\mathtt{den}})} \right) \mathcal{L}(\mathcal{H}, \mathcal{H}_{\mathtt{ref}}^r)$$

# Discriminative Training

- Extended Baulm Welch algorithm

$$\hat{\mu}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}(L_{jm}^{\mathbf{num}r}(t) - L_{jm}^{\mathbf{den}r}(t))\mathbf{o}_t^r + D_{jm}\boldsymbol{\mu}_{jm} + \tau^{\mathrm{I}}\boldsymbol{\mu}_{jm}^{\mathrm{p}}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}(L_{jm}^{\mathbf{num}r}(t) - L_{jm}^{\mathbf{den}r}(t)) + D_{jm} + \tau^{\mathrm{I}}}$$

$$\hat{\Sigma}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}(L_{jm}^{\mathbf{num}r}(t) - L_{jm}^{\mathbf{den}r}(t))\mathbf{o}_t^r\mathbf{o}_t^{r\mathsf{T}} + D_{jm}\mathbf{G}_{jm}^{\mathrm{s}} + \tau^{\mathrm{I}}\mathbf{G}_{jm}^{\mathrm{p}}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}(L_{jm}^{\mathbf{num}r}(t) - L_{jm}^{\mathbf{den}r}(t)) + D_{jm} + \tau^{\mathrm{I}}} - \hat{\boldsymbol{\mu}}_{jm}\hat{\boldsymbol{\mu}}_{jm}^{\mathsf{T}}$$

$$\mathbf{G}_{jm}^{\mathrm{s}} = \boldsymbol{\Sigma}_{jm} + \boldsymbol{\mu}_{jm}\boldsymbol{\mu}_{jm}^{\mathsf{T}}$$

$$\mathbf{G}_{jm}^{\mathrm{p}} = \boldsymbol{\Sigma}_{jm}^{\mathrm{p}} + \boldsymbol{\mu}_{jm}^{\mathrm{p}}\boldsymbol{\mu}_{jm}^{\mathrm{p}\mathsf{T}}$$

# Thank You