Acoustic and Lexical Modelling

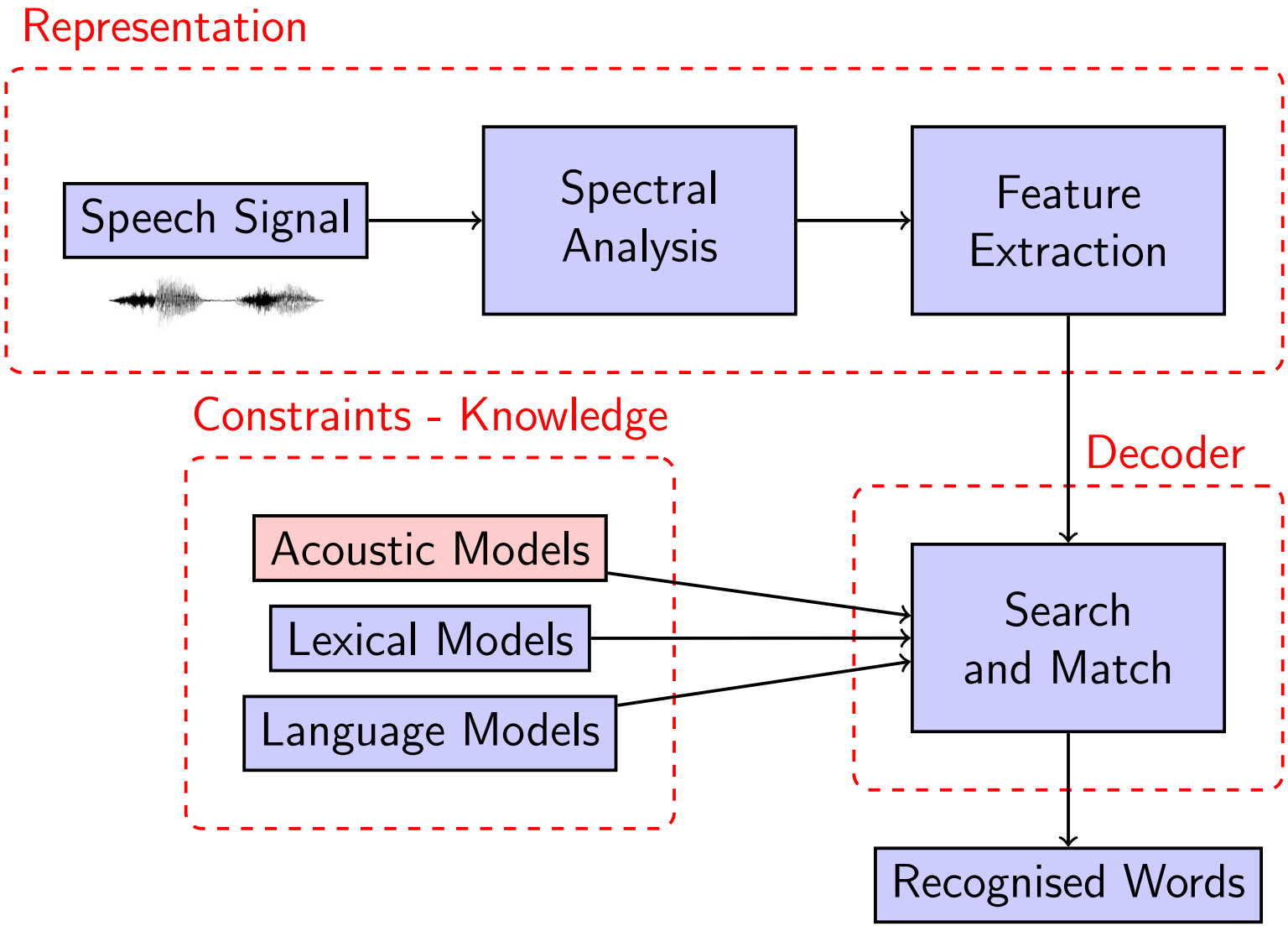
DT2119 Speech and Speaker Recognition

Giampiero Salvi

KTH/CSC/TMH giampi@kth.se

VT 2018

Components of ASR System



Outline

A probabilistic perspective: Bayes' rule

$$P(\text{words}|\text{sounds}) = \frac{P(\text{sounds}|\text{words})P(\text{words})}{P(\text{sounds})}$$

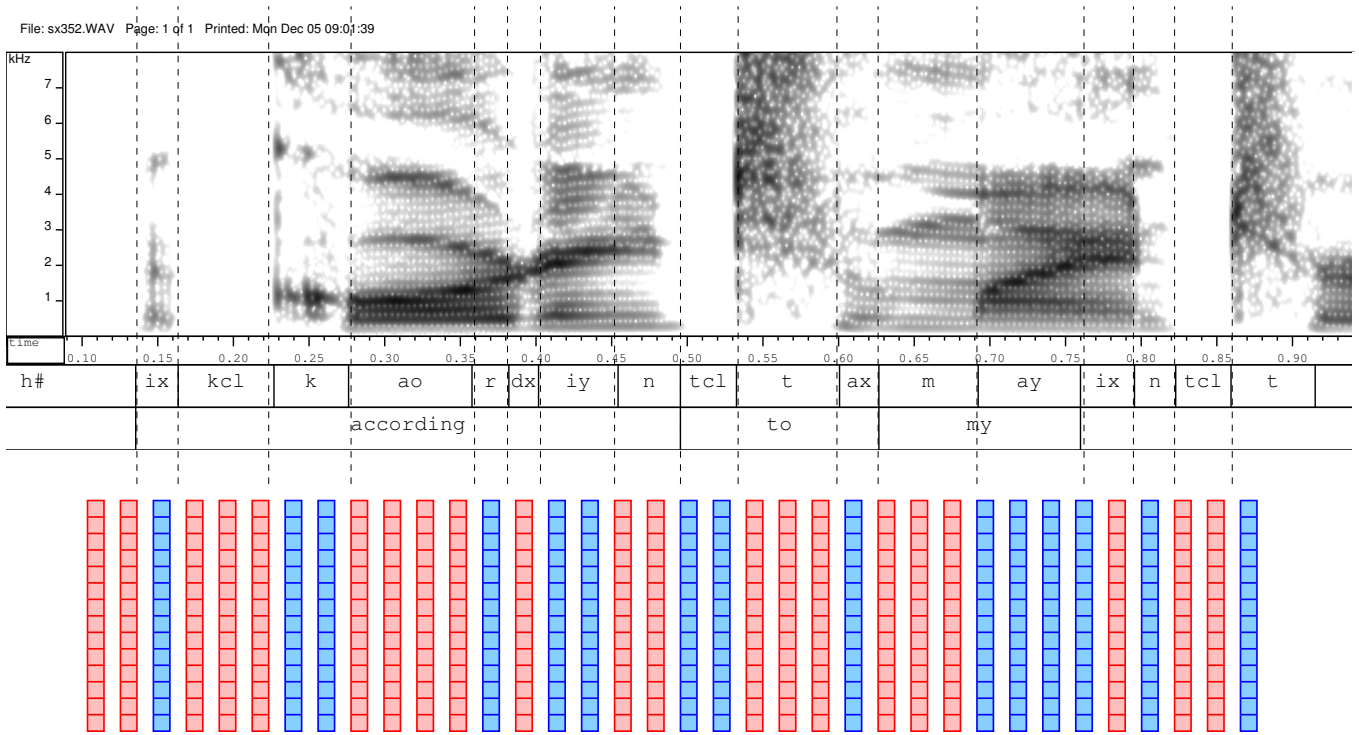
- $P(\text{sounds}|\text{words})$ can be estimated from training data and transcriptions
- $P(\text{words})$: *a priori* probability of the words (Language Model)
- $P(\text{sounds})$: *a priori* probability of the sounds (constant, can be ignored)

Probabilistic Modelling

Problem: How do we model $P(\text{sounds}|\text{words})$?

Probabilistic Modelling

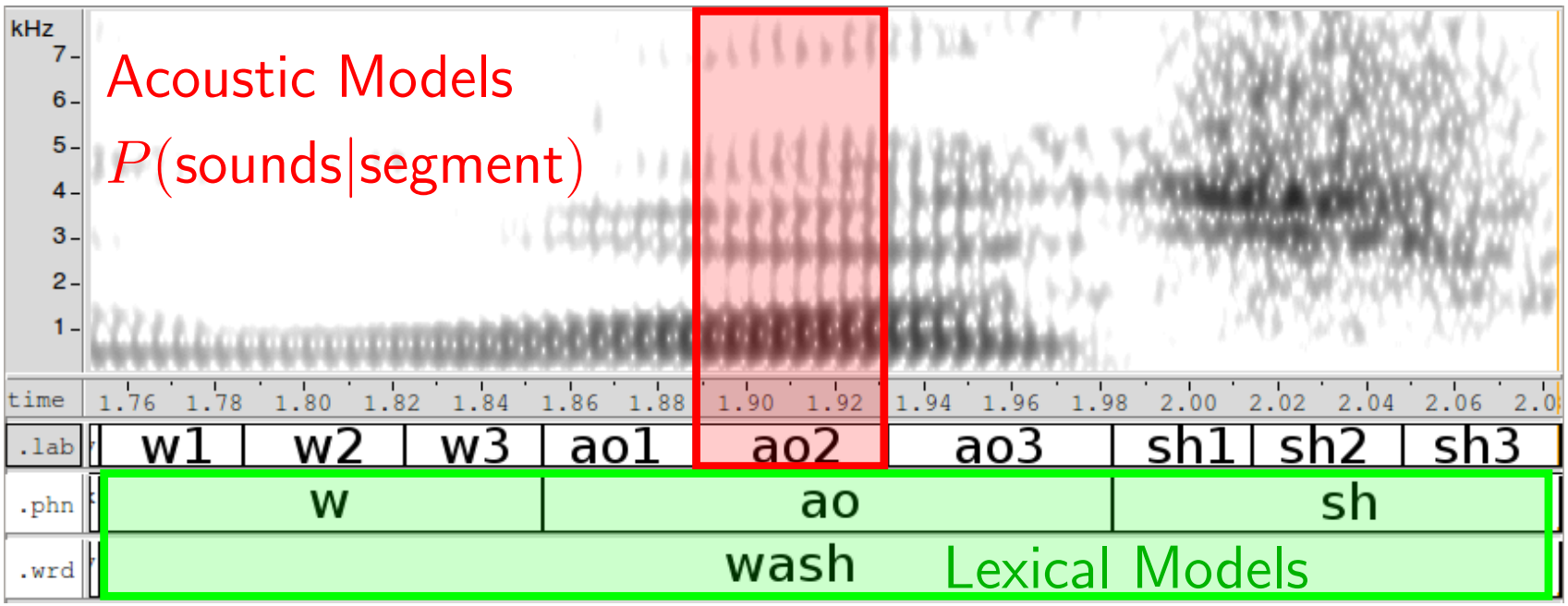
Problem: How do we model $P(\text{sounds}|\text{words})$?



Every feature vector (observation at time t) is a continuous stochastic variable (e.g. MFCC)

Stationarity

- we need to model short segments independently
- the fundamental unit can not be the word, but must be shorter
- usually we model three segments for each phoneme



Local probabilities (frame-wise)

If segment sufficiently short

$$P(\text{sounds}|\text{segment})$$

can be modelled with standard probability distributions

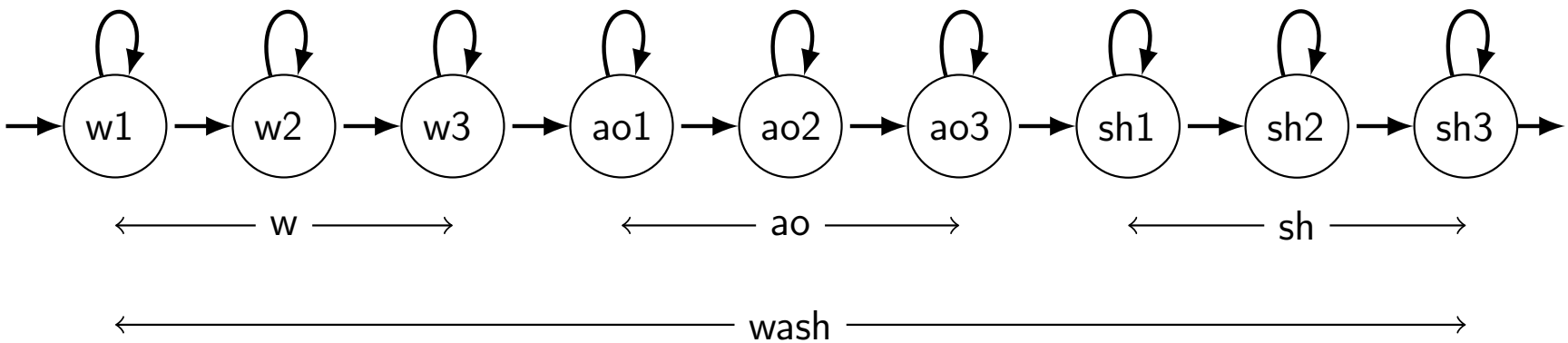
$$\phi_j(x_n) = P(x_n|z_n = s_j)$$

Usually Gaussian or Gaussian Mixture but also discrete distributions

Global Probabilities (utterance)

Problem: How do we combine the different $P(\text{sounds}|\text{segment})$ to form $P(\text{sounds}|\text{words})$?

Answer: Hidden Markov Model (HMM)

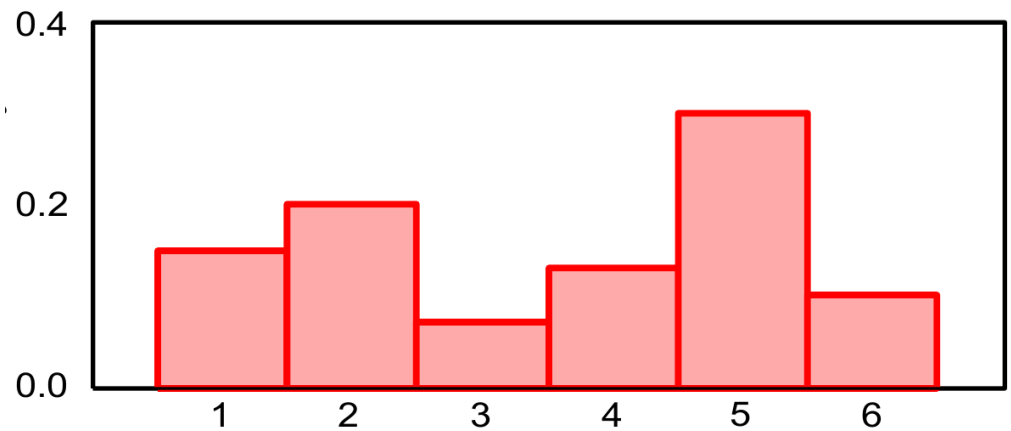


Emission probability model

- Discrete HMMs (DHMMs)
  - $\phi_j(x_n) = \text{Cat}(x_n|\lambda_{j1}, \ldots, \lambda_{jK})$
  - vector quantisation
- Continuous HMMs
  - Single Gaussian $\phi_j(x_n) = \mathcal{N}(x_n|\mu_j, \Sigma_j)$
  - Gaussian Mixture $\phi_j(x_n) = \sum_k \pi_{jk}\mathcal{N}(x_n|\mu_{jk}, \Sigma_{jk})$
- Semi-continuous HMMs (SCHMMs)
  - pool of shared Gaussians, categorical distribution for each state
- DNN-HMMs
  - interpret network output as probabilities

Discrete HMMs

- quantise feature vectors
- observation: sequence of discrete symbols
- $\phi_j(x_n)$ simple discrete probability distribution
- problem: quantisation error



Discrete HMMs: Update Rules

We know how to compute (forward-backward)

$$\gamma_n(j) = P(z_n = s_j | X, \theta)$$

are the posteriors of the latent variable
Update rule:

$$\phi_j(x_n = k) = \frac{E[x_n = k, z_n = s_j]}{E[z_n = s_j]} = \frac{\sum_{n:(x_n=k)} \gamma_n(j)}{\sum_{n=1}^{N} \gamma_n(j)}$$

HMMs with Gaussian Emission Probability

$$\phi_j(x_n) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

Update rules:

$$\begin{aligned}
\boldsymbol{\mu}_j &= \frac{\sum_{n=1}^{N} \gamma_n(j)\boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_n(j)} \\
\boldsymbol{\Sigma}_j &= \frac{\sum_{n=1}^{N} \gamma_n(j)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_j\right)^T}{\sum_{n=1}^{N} \gamma_n(j)} \\
&= \frac{\sum_{n=1}^{N} \gamma_n(j)\boldsymbol{x}_n\boldsymbol{x}_n^T}{\sum_{n=1}^{N} \gamma_n(j)} - \boldsymbol{\mu}_j\boldsymbol{\mu}_j^T
\end{aligned}$$

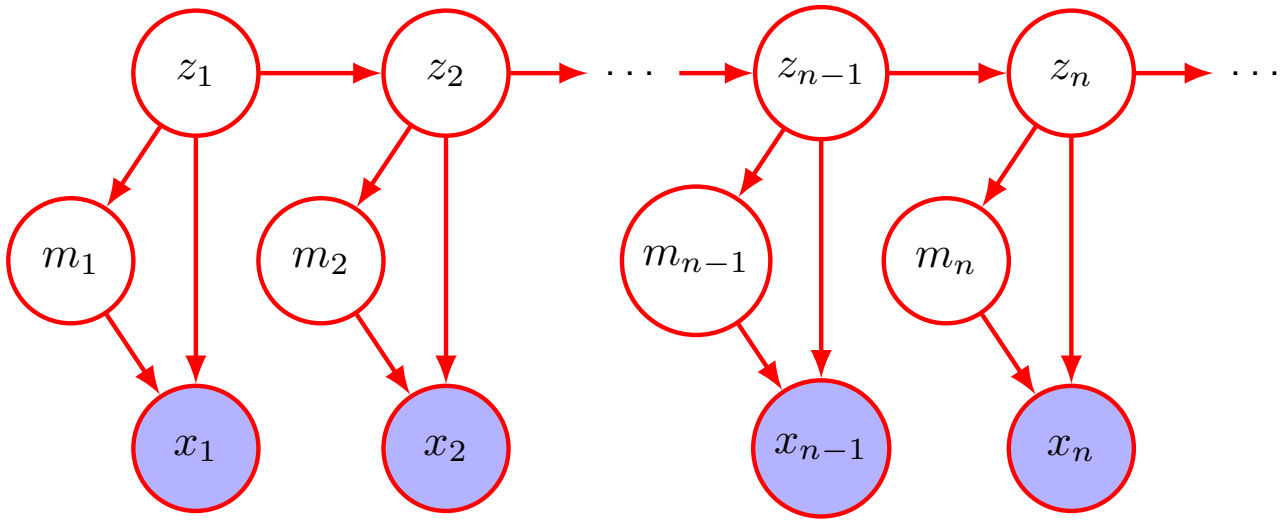
HMMs with Mixture Emission Probability

Often the Emission probability is modelled as a Mixture of Gaussians

$$\phi_j(x_n) = \sum_{k=1}^{K} w_{jk} \mathcal{N}(x_n | \mu_{jk}, \Sigma_{jk})$$

$$\sum_{k=1}^{M} w_{jk} = 1$$

HMMs with Mixture Emission Probability



Emission:

$$
\begin{aligned}
p(x_n|z_n, m_n) &= \mathcal{N}(x_n; \mu_{z_n,m_n}, \Sigma_{z_n,m_n}) \\
p(m_n|z_n) &= W(m_n, z_n)
\end{aligned}
$$

HMMs with Mixture Emission Probability

Training (hard to initialize):

1. start training single Gaussians
2. split each Gaussian into two
3. apply small perturbation and retrain
4. go back to 2. until desired number of terms is reached

HMMs with Mixture Emission Probability

Training (hard to initialize):

1. start training single Gaussians
2. split each Gaussian into two
3. apply small perturbation and retrain
4. go back to 2. until desired number of terms is reached

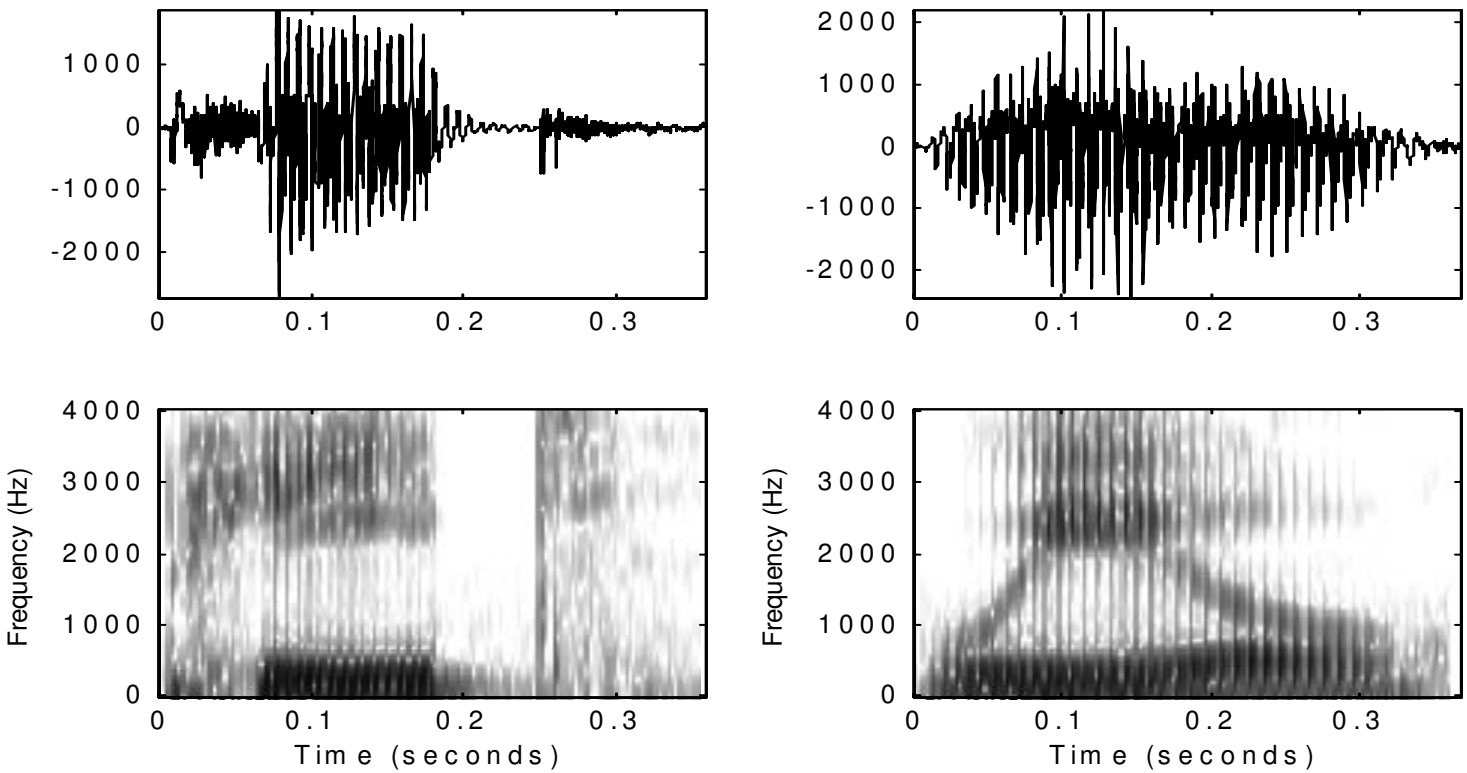
- The final number of Gaussians per state depends on the amount of data.
- Typical values are 32, 64 or 128

Semi-Continuous HMMs

- All Gaussian distributions in a pool of pdfs
- each $\phi_j(x_n)$ is a categorical probability distribution over the pool of Gaussians
- similar to quantisation, but probabilistic
- used for sharing parameters

Modelling Coarticulation

Example peat /piːt/ vs wheel /wiːl/



Modelling Coarticulation

Context dependent models (CD-HMMs)

- Duplicate each phoneme model depending on left and right context:
- from “a” monophone model
- to “d−a+f”, “d−a+g”, “l−a+s”... triphone models
- If there are $N = 50$ phonemes in the language, there are $N^3 = 125000$ potential triphones
- many of them are not exploited by the language

Amount of parameters

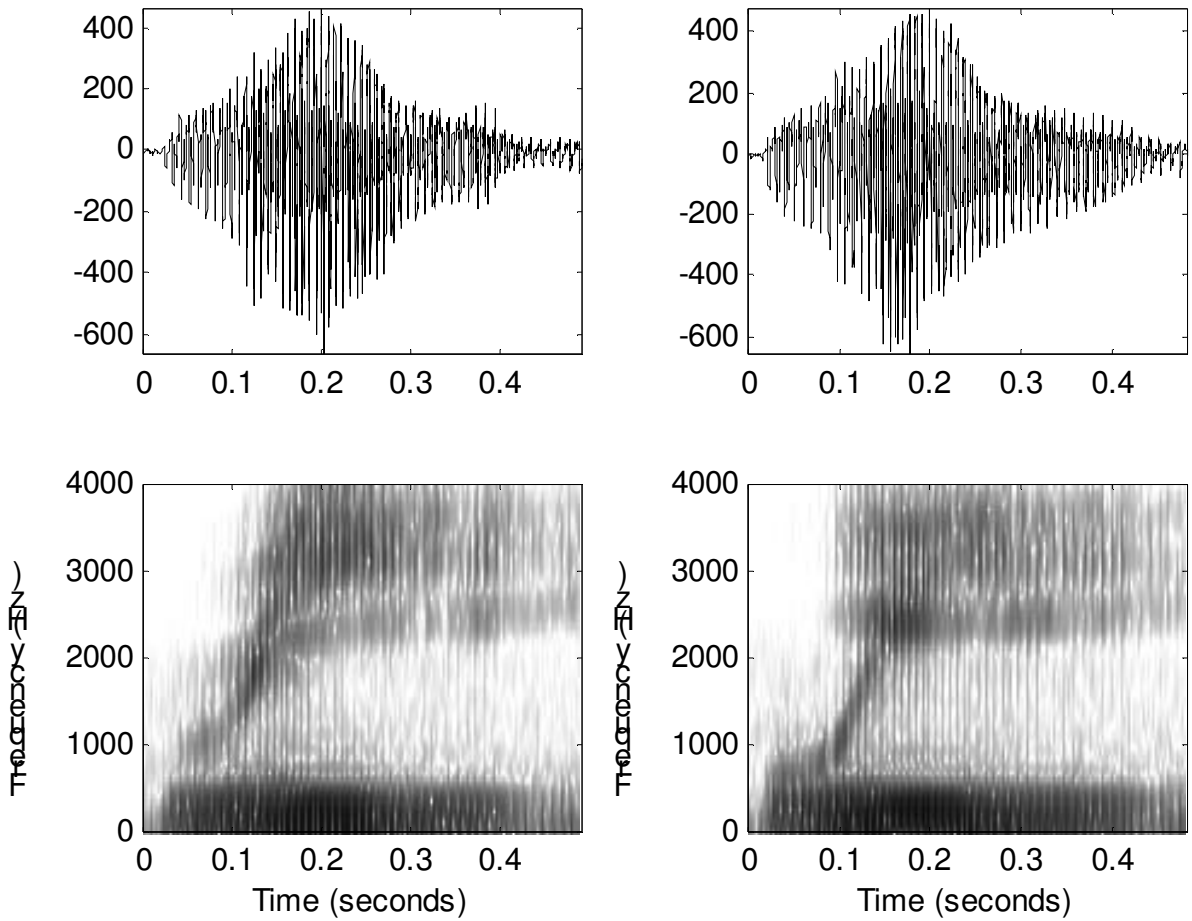
Example:

- a large vocabulary recogniser may have 60000 triphone models
- each model has 3 states
- each state may have 32 mixture components with $1 + 39 \times 2$ parameters each (weight, means, variances): $39 \times 32 \times 2 + 32 = 2528$

Totally it is $60000 \times 3 \times 2528 = 455$ million parameters!

Similar Coarticulation

/riː/ vs /wiː/



Tying to reduce complexity

Example: similar triphones d−a+m and t−a+m

- same right context, similar left context
- 3rd state is expected to be very similar
- 2nd state may also be similar

States (and their parameters) can be shared between models

+ reduce complexity
+ more data to estimate each parameter
– fine detail may be lost

Tying to reduce complexity

Example: similar triphones d−a+m and t−a+m

- same right context, similar left context
- 3rd state is expected to be very similar
- 2nd state may also be similar

States (and their parameters) can be shared between models

+ reduce complexity
+ more data to estimate each parameter
– fine detail may be lost

can be done data-driven, but usually
done with CART tree methodology

Data-Driven parameter tying

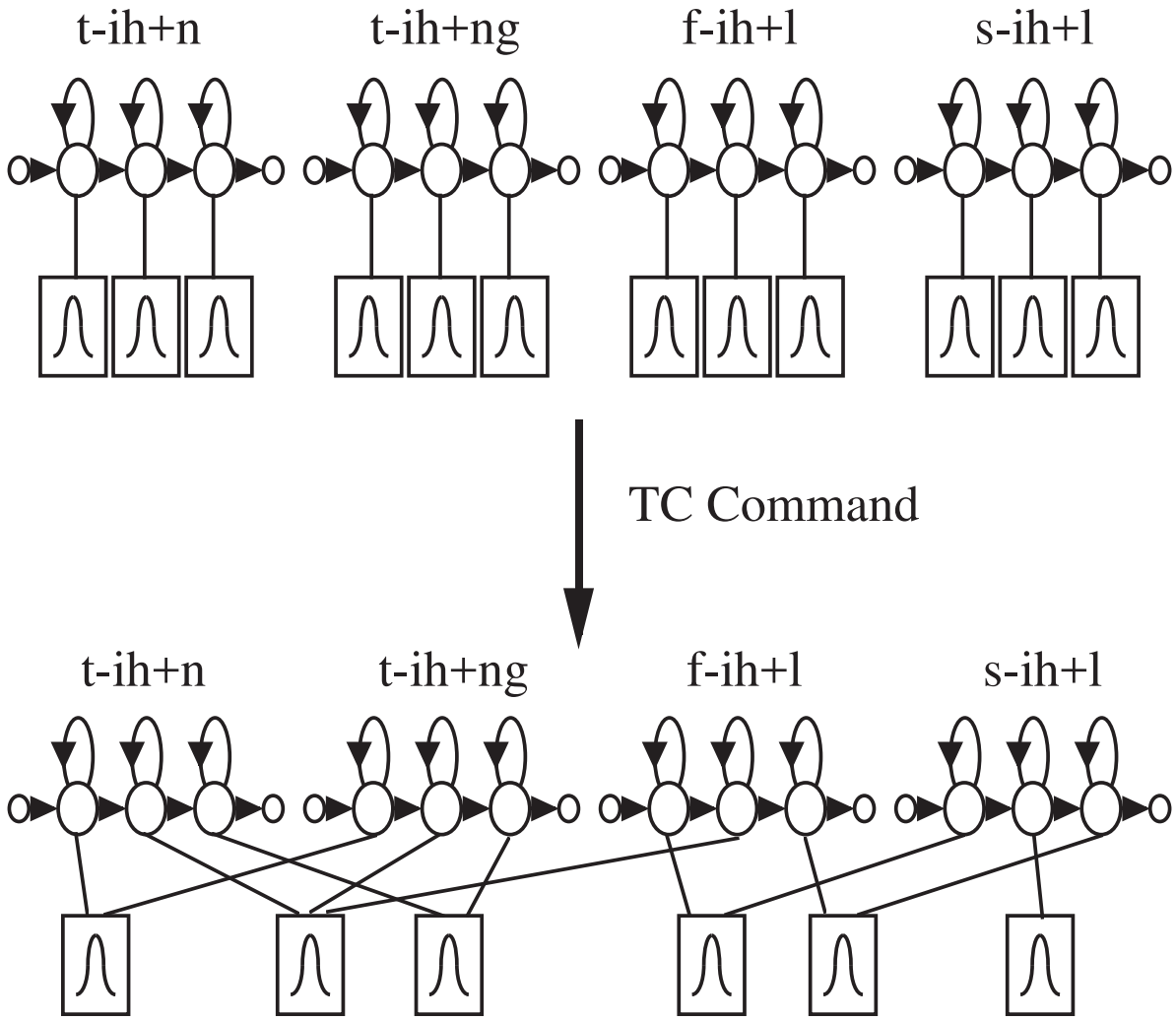


Figure from the HTK Book

Data-Driven parameter tying

- Hierarchical clustering with complete linkage
- States are compared using distance metric between emission distributions:
  - Single Gaussian: Mahalanobis distance between means
  - Gaussian Mixture: Euclidean distance between mixture weights
  - In general: Kullback-Leibler divergence
- Stopping criterion:
  - minimum number of clusters reached
  - maximum number of states per clusters reached

Data-Driven parameter tying: unseen triphones

Problem:
Not able to cope with triphones without examples?

- they might be in the test data, but not in the training data.
- we may want to add new words after training

Tree-Based Parameter Tying

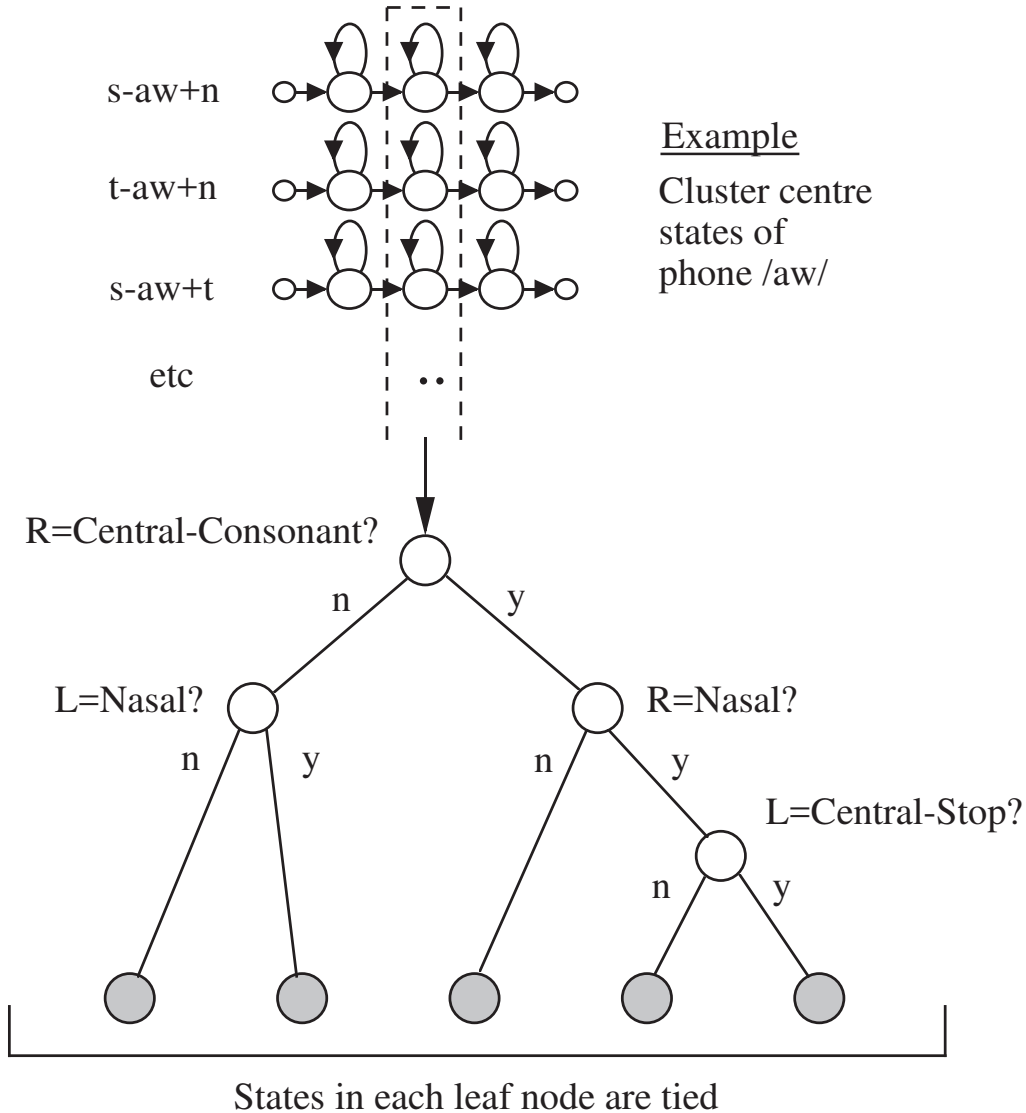


Figure from the HTK Book

Phonetic questions

- Consonant/Vowel
- Fricative/Plosive/Approximant/...
- Lateral/Labial/Velar/...
- Long/Short Vowel
- Front/Back Vowel
- ...

Example (triphone is `lc-ph+rc`)

```
QS "L_Nasal" { ng-*,n-*,m-* }
```

Tree-Based Parameter Tying: Sufficient Statistics

- Assume single Gaussian: $\phi_j(x_n) = \mathcal{N}(x_n|\mu_j, \Sigma_j)$
- we know the posterior for each state:
  $\gamma_n(j) = P(z_n = s_j|X, \theta)$

Sufficient statistics:

$$\Gamma_j = \sum_n \gamma_n(j) \quad \text{occupation count}$$

$$\boldsymbol{\nu}_j = \sum_n \gamma_n(j)\boldsymbol{x}_n \quad \text{first order stat.}$$

$$\boldsymbol{\Omega}_j = \sum_n \gamma_n(j)\boldsymbol{x}_n\boldsymbol{x}_n^T \quad \text{second order stat.}$$

Tree-Based Parameter Tying: Sufficient Statistics

- Assume single Gaussian: $\phi_j(x_n) = \mathcal{N}(x_n|\mu_j, \Sigma_j)$
- we know the posterior for each state:
  $\gamma_n(j) = P(z_n = s_j|X, \theta)$

Sufficient statistics:

$$\begin{aligned}
\Gamma_j &= \sum_n \gamma_n(j) \quad \text{occupation count} \\
\boldsymbol{\nu}_j &= \sum_n \gamma_n(j)\boldsymbol{x}_n \quad \text{first order stat.} \\
\boldsymbol{\Omega}_j &= \sum_n \gamma_n(j)\boldsymbol{x}_n\boldsymbol{x}_n^T \quad \text{second order stat.}
\end{aligned}$$

Gaussian parameters:

$$\boldsymbol{\mu}_j = \frac{\boldsymbol{\nu}_j}{\Gamma_j} \qquad \boldsymbol{\Sigma}_j = \frac{\boldsymbol{\Omega}_j}{\Gamma_j} - \boldsymbol{\mu}_j\boldsymbol{\mu}_j^T$$

Tree-Based Parameter Tying: Update Rules

For groups of states:

$$\Gamma_{\text{group}} = \sum_k \Gamma_k \quad \text{occupation count}$$

$$\boldsymbol{\nu}_{\text{group}} = \sum_k \boldsymbol{\nu}_k \quad \text{first order stat.}$$

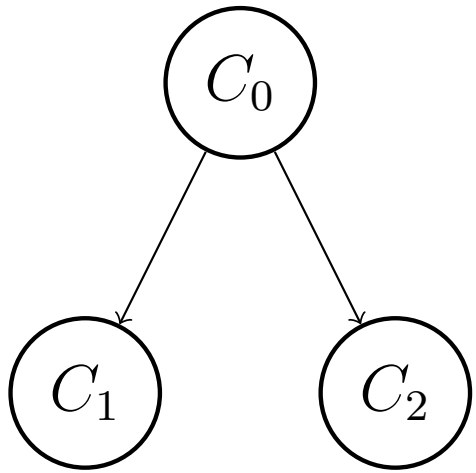
$$\boldsymbol{\Omega}_{\text{group}} = \sum_k \boldsymbol{\Omega}_k \quad \text{second order stat.}$$

Gaussian parameters:

$$\boldsymbol{\mu}_{\text{group}} = \frac{\boldsymbol{\nu}_{\text{group}}}{\Gamma_{\text{group}}} \qquad \boldsymbol{\Sigma}_{\text{group}} = \frac{\boldsymbol{\Omega}_{\text{group}}}{\Gamma_{\text{group}}} - \boldsymbol{\mu}_{\text{group}} \boldsymbol{\mu}_{\text{group}}^T$$

Tree-based parameter tying: Likelihood Gain

- parent node contains all states
- split to C_1 or C_2 based on question



Likelihood Gain:

$$\begin{aligned}\Delta\mathcal{L} &= \mathcal{L}_1 + \mathcal{L}_2 - \mathcal{L}_0 \\ &= \Gamma_1 \log|\mathbf{\Sigma}_1| + \Gamma_2 \log|\mathbf{\Sigma}_2| - \Gamma_0 \log|\mathbf{\Sigma}_0|\end{aligned}$$

Stopping Criteria

- threshold on the log likelihood increase
- avoid states with low occupation counts (expected value of number of training examples)

Unseen triphones

Use the tree to assign all states to most similar triphone.

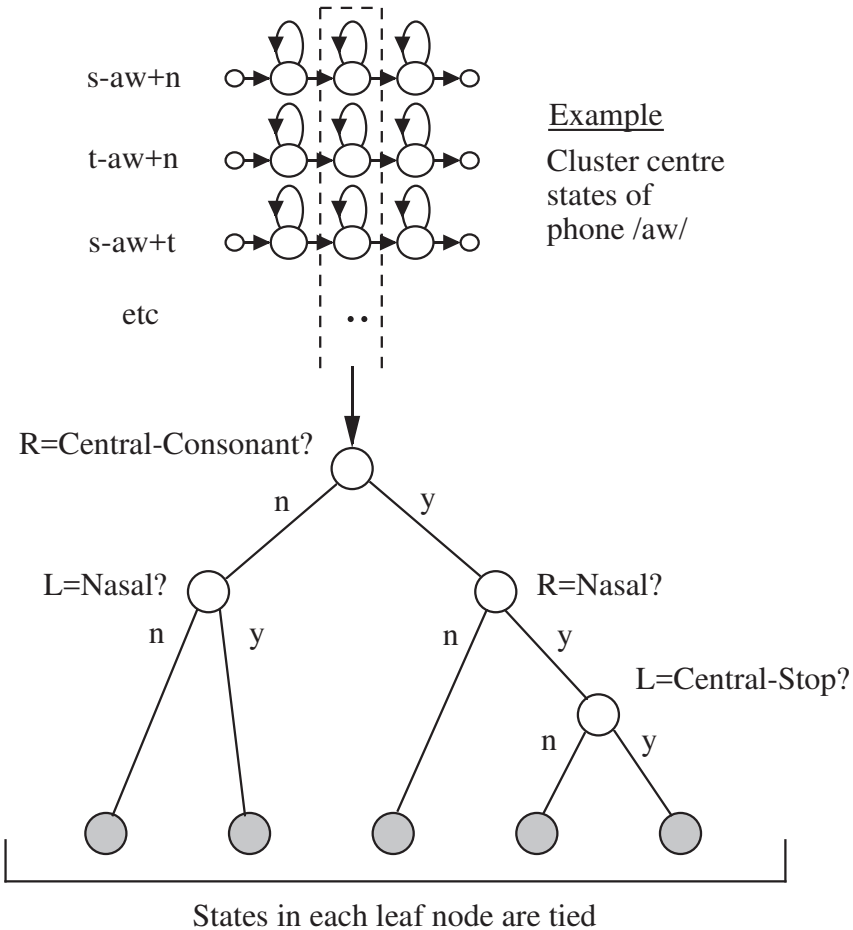


Figure from the HTK Book

Senones

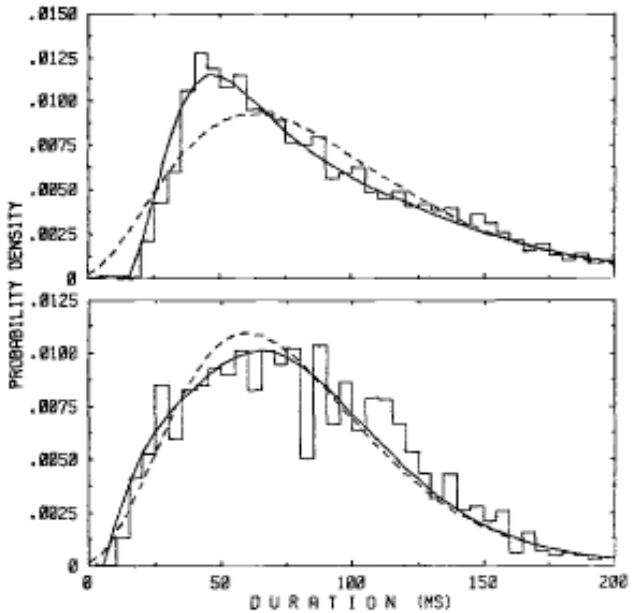
Pool of states after clustering[1]

- typically in the order of thousands
- may be shared between different phonetic models
- also used as targets in Deep Neural Networks

[1] M.-Y. Hwang, X. Huang, and F. A. Alleva. "Predicting Unseen Triphones with Senones". In: *IEEE Trans. Speech Audio Process.* 4.6 (1996).

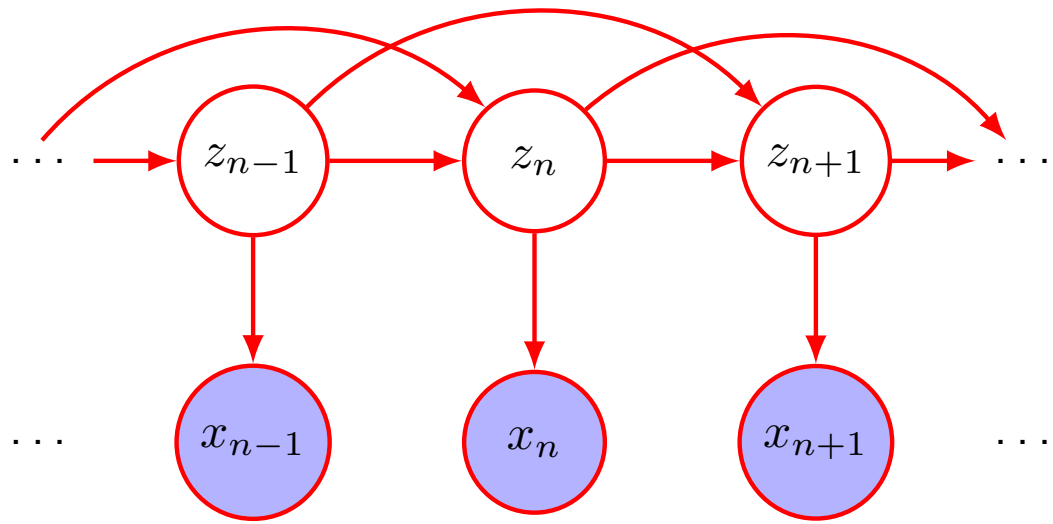
Outline

HMM Limitations: Duration modelling

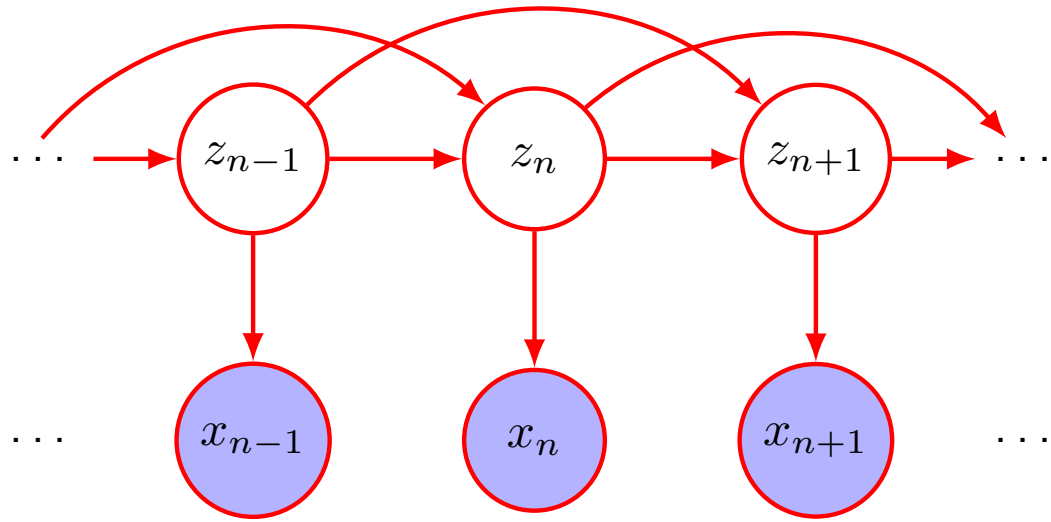


- $P(d_i = n) = a_{ii}^n(1 - a_{ii})$
- Several solutions proposed, but modest improvements

HMM Limitations: First Order Assumption

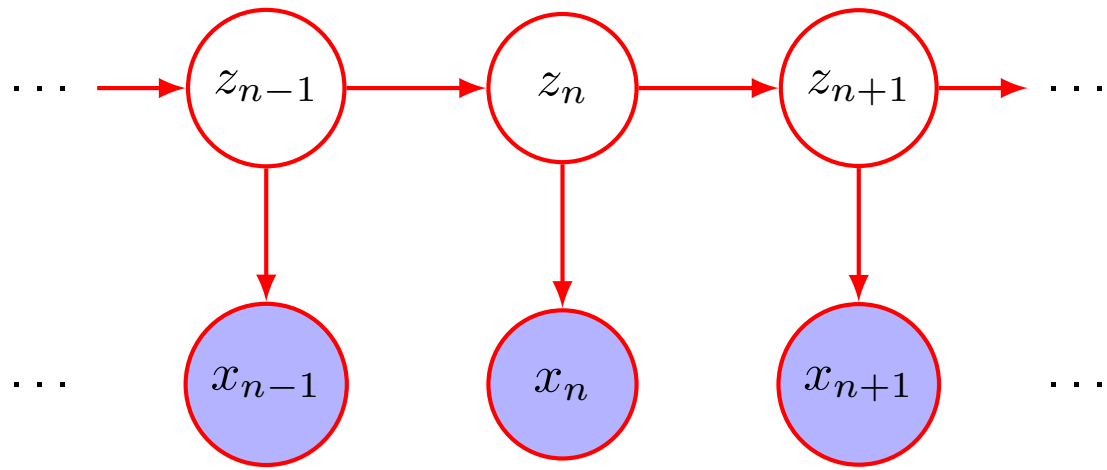


HMM Limitations: First Order Assumption

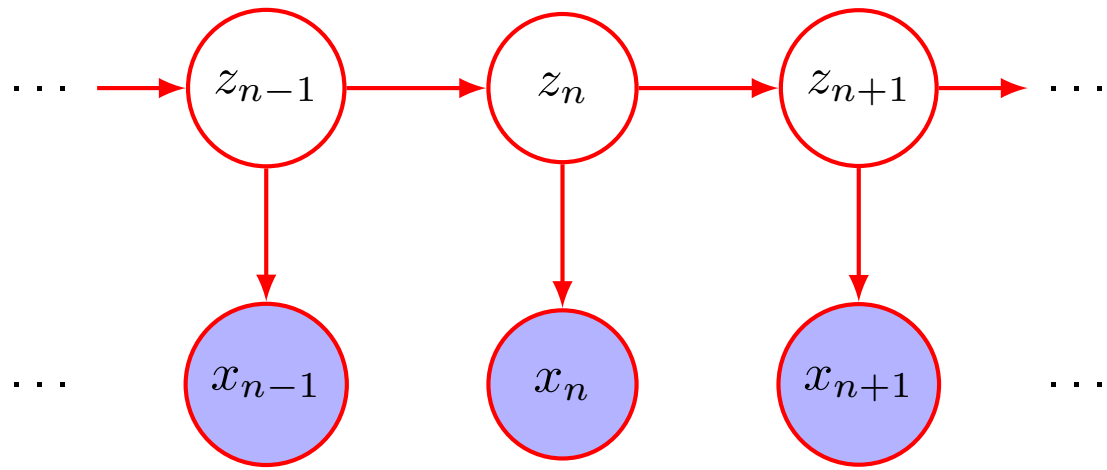


but: increasing order gives modest improvements

HMM Limitations: Conditional Independence Assumption



HMM Limitations: Conditional Independence Assumption



use dynamic features!

Dynamic Features

Concatenate static MFCCs (or LPCs) to Δ and $\Delta\Delta$ vectors. Δ_n computed as weighted sum of $d_k(n)$

$$\Delta_n = \frac{\sum_{k=1}^{K} w_k d_k(n)}{\sum_{k=1}^{K} w_k}$$

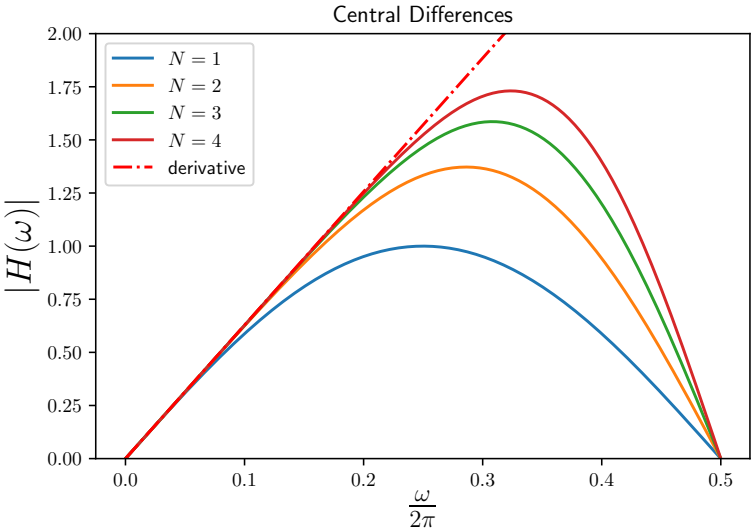
$d_k(n)$: finite differences centered around n with interval $2k$:

$$\begin{aligned} d_k(n) &= \frac{c_{n+k} - c_{n-k}}{2k} \\ w_k &= 2k^2 \end{aligned}$$

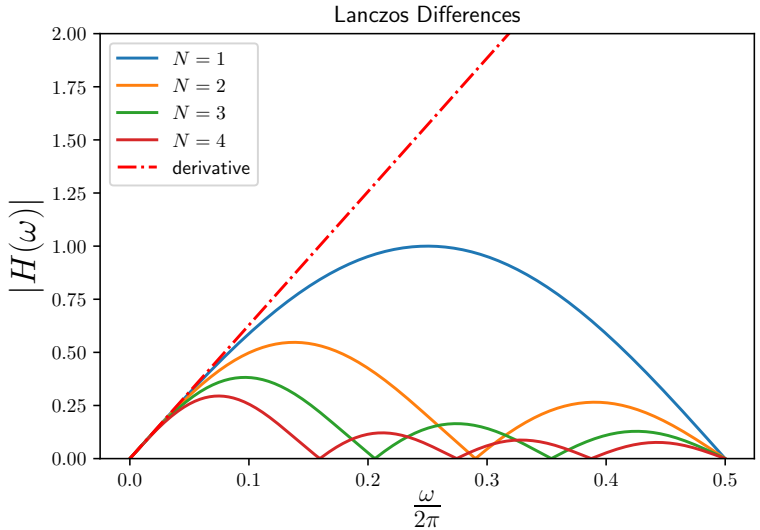
Similarly for $\Delta\Delta_n$

Dynamic Features: Motivation

Central Differences



Lanczos Differences

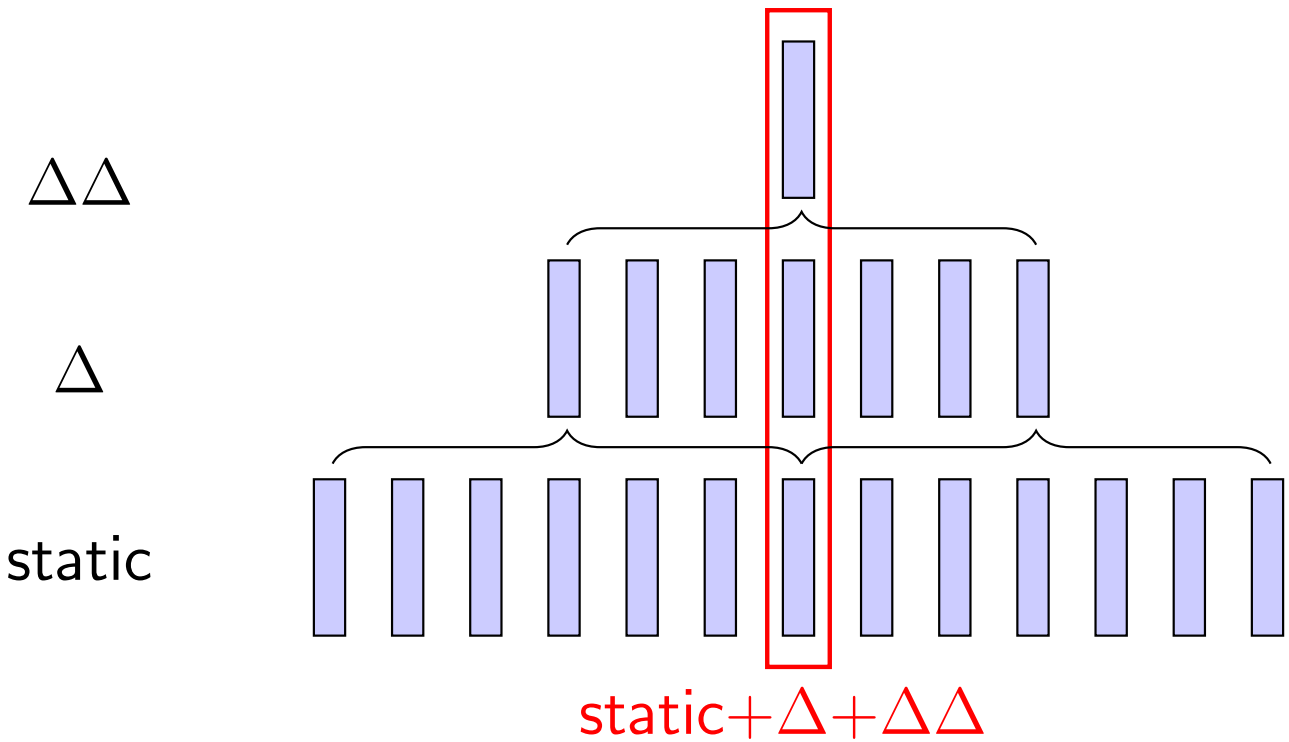


Polynomial fit with or without error

Detailed explanation in Canvas (usually not in the literature)

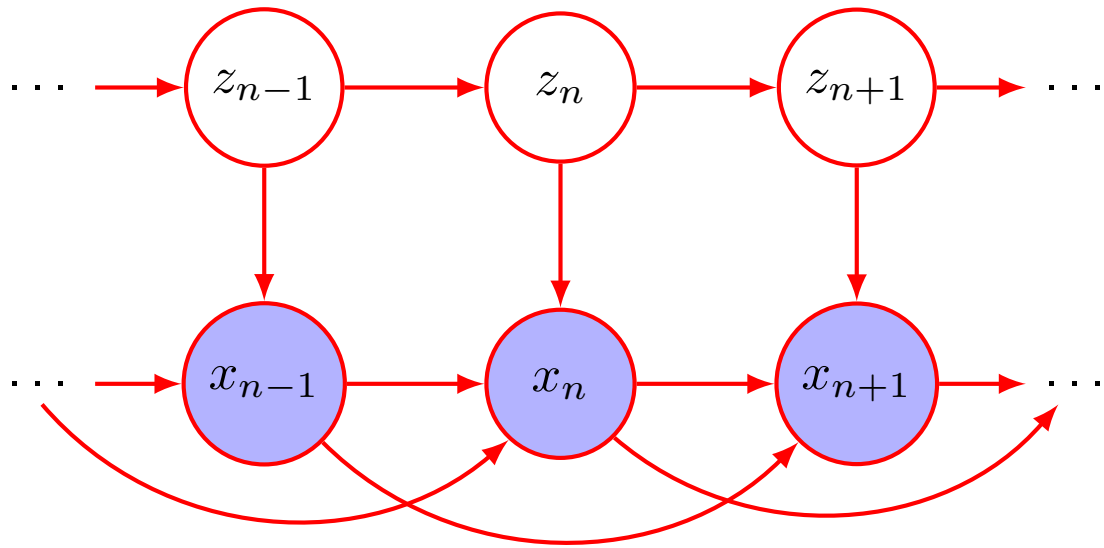
Dynamic Features: Common values

- Usually k goes from 1 to 3
- to compute static+Δ+$\Delta\Delta$ we need 13 consecutive static vectors (around 130 msec).



HMM Limitations: Conditional Independence Assumption

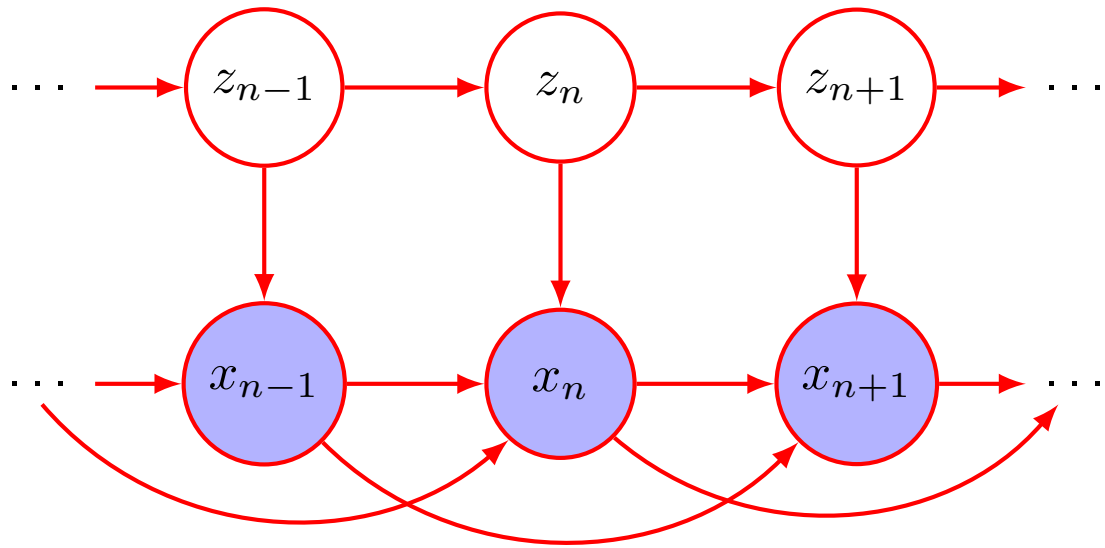
Autoregressive HMM[2]



[2] M. Shannon and W. Byrne. "Autoregressive HMMs for speech synthesis". In: *Proc. Interspeech*. Brighton, U.K., 2009.

HMM Limitations: Conditional Independence Assumption

Autoregressive HMM[2]



Also interesting results with Time Delay and Recurrent Neural Networks (TDNNs, RNNs, LSTMs)

[2] M. Shannon and W. Byrne. "Autoregressive HMMs for speech synthesis". In: *Proc. Interspeech*. Brighton, U.K., 2009.

HMMs: Practical Issues

- Initialisation
- Training Criteria

Initialisation

Important in order to reach a high local maximum

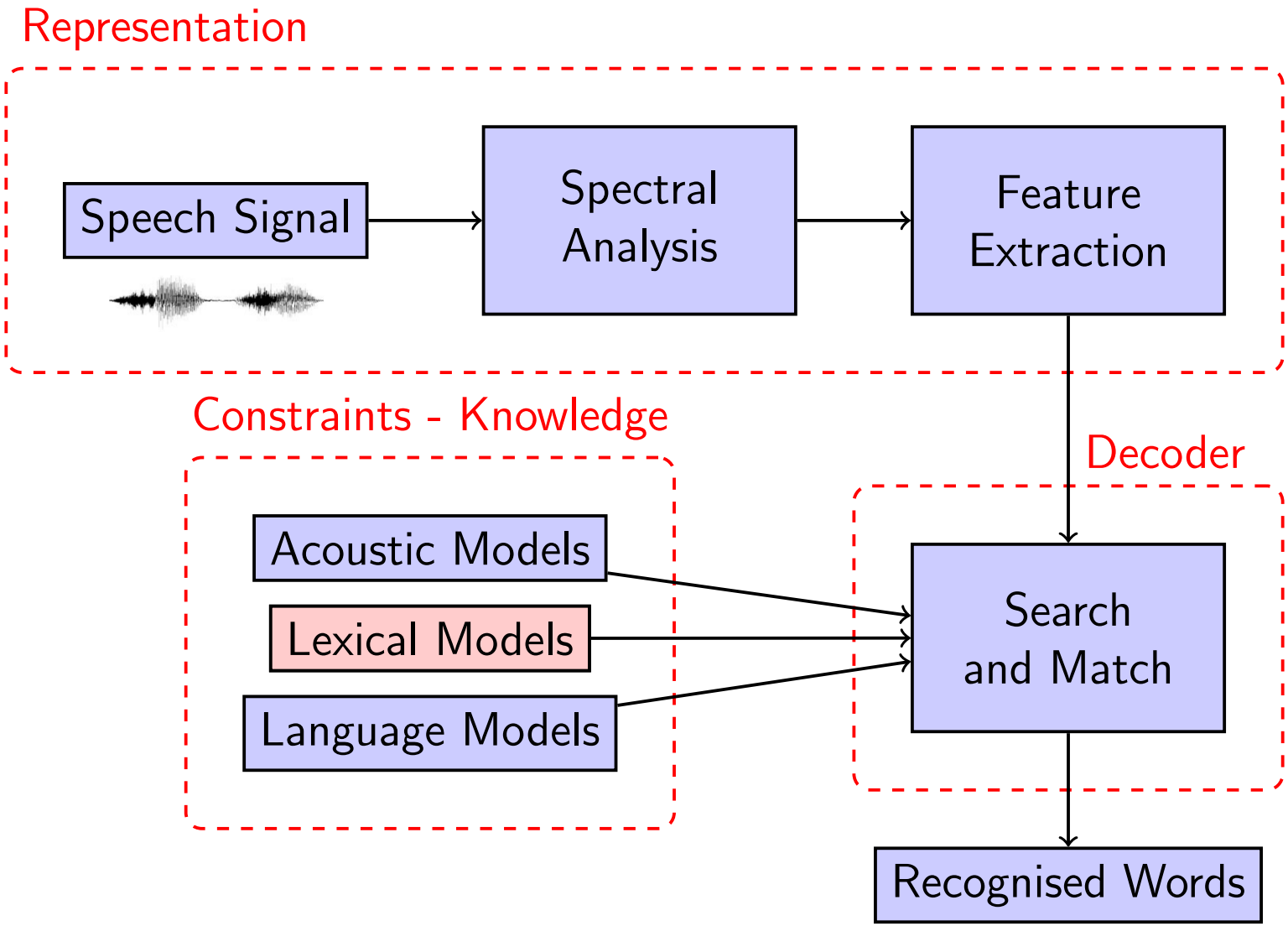
- Discrete HMM
  - Initial zero probability remains zero
  - Uniform distribution works reasonably well
- Continuous HMM methods
  - k-means clustering
  - Proceed from discrete HMM to semi-continuous to continuous
  - Start training single Gaussian models.
- Use previously segmented data or “flat start” (equal distribution for all states in the training data)

Training Criteria

- Maximum Likelihood Estimation (MLE)
  - Sensitive to inaccurate Markov assumptions
  - Maximises model likelihood rather than discrimination between models
- Minimum Classification Error (MCE) and Maximum Mutual Information Estimation (MMIE) might work better
- Maximum A Posteriori (MAP) if we have prior knowledge
  - for adaptation and small training data

Outline

Components of ASR System



Lexical Models

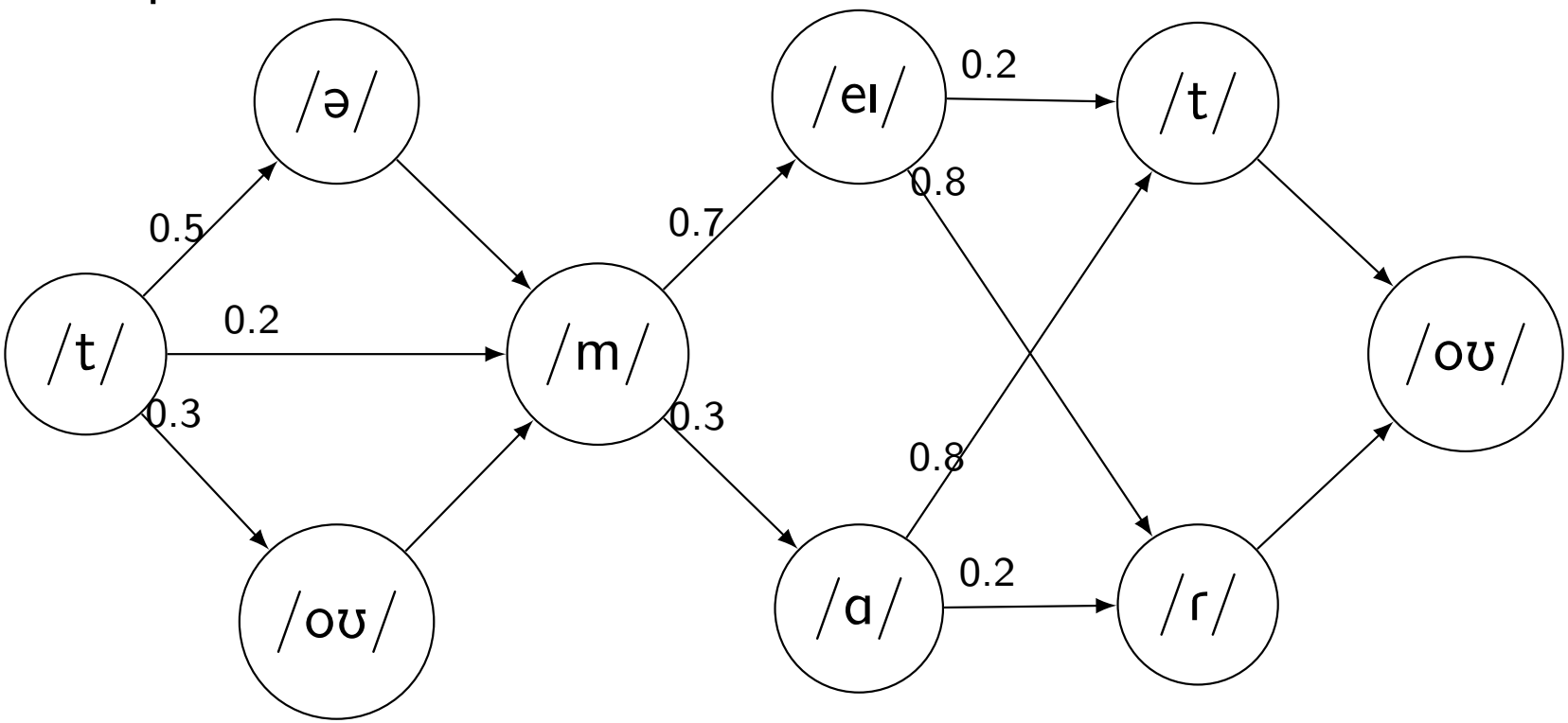
- in general specify sequence of phoneme for each word
- example:

"dictionary"	IPA	X-SAMPA
UK:	/d ɪ k ʃ ə n (ə) ɹ i/	/d I k S @ n (@) r i/
USA:	/d ɪ k ʃ ə n ɛ ɹ i/	/d I k S @ n E r i/

- expensive resources
- include multiple pronunciations
- phonological rules (assimilation, deletion)

Pronunciation Network

Example: tomato



Assimilation

did you	/d ɪ dʒ j ə/
set you	/s ɛ tʃ ɜ/
last year	/l æ s tʃ iː ɹ/
because you've	/b iː k ə ʒ uː v/

Deletion

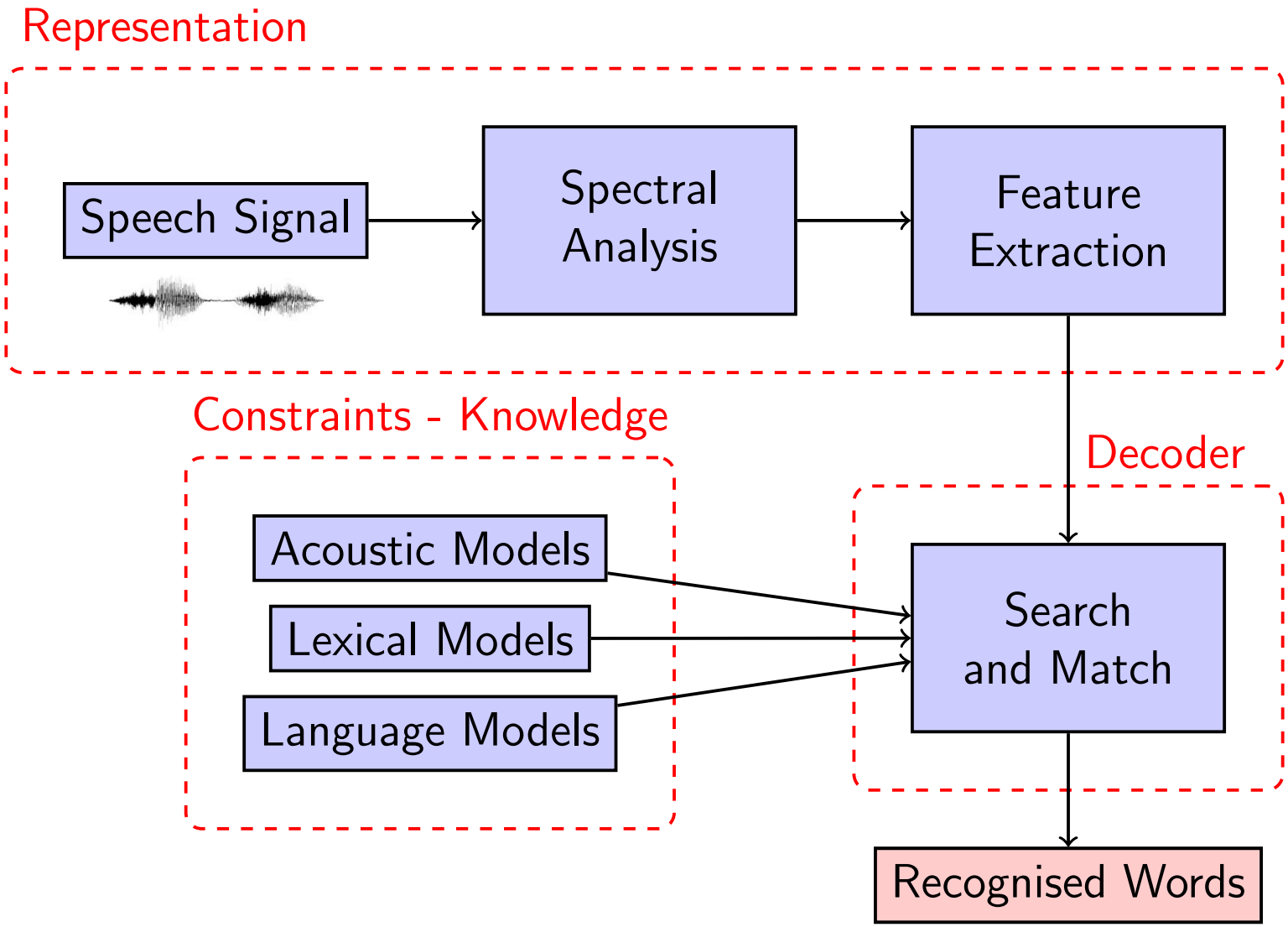
find him	/f a ɪ n ɪ m/
around this	/ə ɹ aʊ n ɪ s/
let me in	/l ɛ m iː n/

Out of Vocabulary Words

- Proper names often not in lexicon
- derive pronunciation automatically
- English has very complex grapheme-to-phoneme rules
- attempts to derive pronunciation from speech recordings

Outline

Components of ASR System



ASR Evaluation

- recognition results are sequences of words
- evaluation is non-trivial
- need to realign the recognised sequence to the transcription
- example:

  ref: I really wanted to see you
  rec: I wanted badly to meet you

- possible to use detailed time alignment
- usually only symbolic level is used
- dynamic programming

Word Accuracy and Word Error Rate (WER)

$$A = 100\frac{N - S - D - I}{N}$$

Where

- N: total number of reference words
- S: substitutions
- D: deletions
- I: insertions

$$\text{WER} = 100 - A$$

Word Accuracy: example

Ref/Rec	I	wanted	badly	to	meet	you
I	corr					
really	del					
wanted		corr				
to			ins	corr		
see					sub	
you						corr

6 words, 1 substitution, 1 insertion, 1 deletion

$$A = 100\frac{6-1-1-1}{6} = 50\%$$

requires dynamic programming

Effects of Sampling Rate on WER

Sampling Rate (kHz)	Relative Error Reduction (%)
8	baseline
11	+10
16	+10
22	+0

(from Huang, Acero and Hon)

Effects of Feaures on WER

Feature Set	Relative Error Reduction (%)
13th order LPC cepstrum	baseline
13th order MFCC	+10
16th order MFCC	+0
with Δ and $\Delta\Delta$	+20
with $\Delta\Delta\Delta$	+0

(from Huang, Acero and Hon)

Effect of Modelling Context

Units	Relative Error Reduction (%)
Context-independent phone	baseline
Context-dependent phone	+25
Clustered triphone	+15
Senone	+24

(from Huang, Acero and Hon)[3]

[3] M.-Y. Hwang, X. Huang, and F. A. Alleva. "Predicting Unseen Triphones with Senones". In: *IEEE Trans. Speech Audio Process.* 4.6 (1996).