Hidden Markov Models

Léon Bottou

COS 424 – 4/13/2010

Sequential data

Data often comes as sequences

– Speech and signal.
– Biological sequences.
– Textual data.

Tasks

Recognition

– From speech signal to sequence of words.
– From sequence of words to sequence of ideas.

Segmentation

– Locate the beginning and the end of a subsequence.

Time invariance

– Words sound the same over time.
– Both tasks are intimately connected.

Hidden Markov Models

The Annals of Mathematical Statistics
1970, Vol. 41, No. 1, 164–171

A MAXIMIZATION TECHNIQUE OCCURRING IN THE STATISTICAL ANALYSIS OF PROBABILISTIC FUNCTIONS OF MARKOV CHAINS

BY LEONARD E. BAUM, TED PETRIE, GEORGE SOULES, AND NORMAN WEISS

Institute for Defense Analyses, California Institute of Technology and Columbia University

Hidden Markov Models for Civilians

A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

LAWRENCE R. RABINER, FELLOW, IEEE

Although initially introduced and studied in the late 1960s and early 1970s, statistical methods of Markov source or hidden Markov modeling have become increasingly popular in the last several years. There are two strong reasons why this has occurred. First the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Second the models, when applied properly, work very well in practice for several important applications. In this paper we attempt to carefully and methodically review the theoretical aspects of this type

In this case, with a good signal model, we can simulate the source and learn as much as possible via simulations. Finally, the most important reason why signal models are important is that they often work extremely well in practice, and enable us to realize important practical systems—e.g., prediction systems, recognition systems, identification systems, etc., in a very efficient manner.

• • • •

[1]The idea of characterizing the theoretical aspects of hidden Markov modeling in terms of solving three fundamental problems is due to Jack Ferguson of IDA (Institute for Defense Analysis) who introduced it in lectures and writing.

The author gratefully acknowledges the major contributions of several colleagues to the theory of HMMs in general, and to the presentation of this paper, in particular. A great debt is owed to Dr. J. Ferguson, Dr. A. Poritz, Dr. L. Liporace, Dr. A. Richter, and to Dr. F. Jelinek and the various members of the IBM group for introducing the speech world to the ideas behind HMMs. In addition Dr. S. Levinson, Dr. M. Sondhi, Dr. F. Juang, Dr. A. Dembo, and Dr. Y. Ephraim have contributed significantly to both the theory of HMMs

Summary

– Speech recognition basics

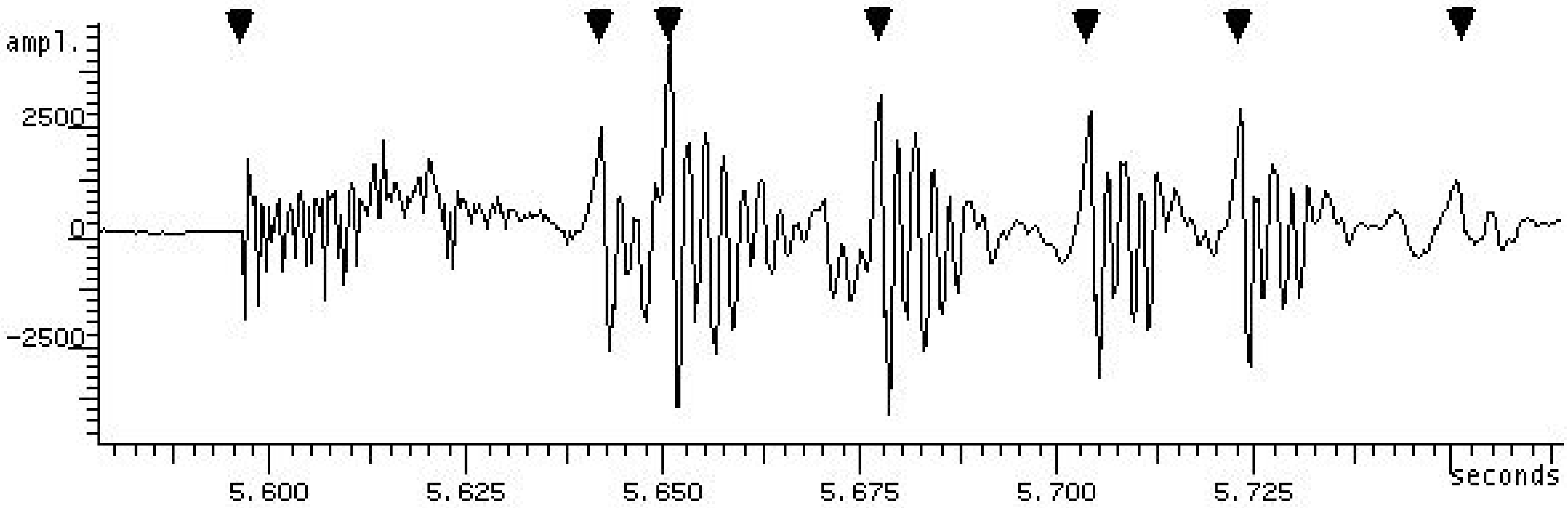
– Hidden Markov Models

– Segmentation and recognition

I. Speech recognition basics

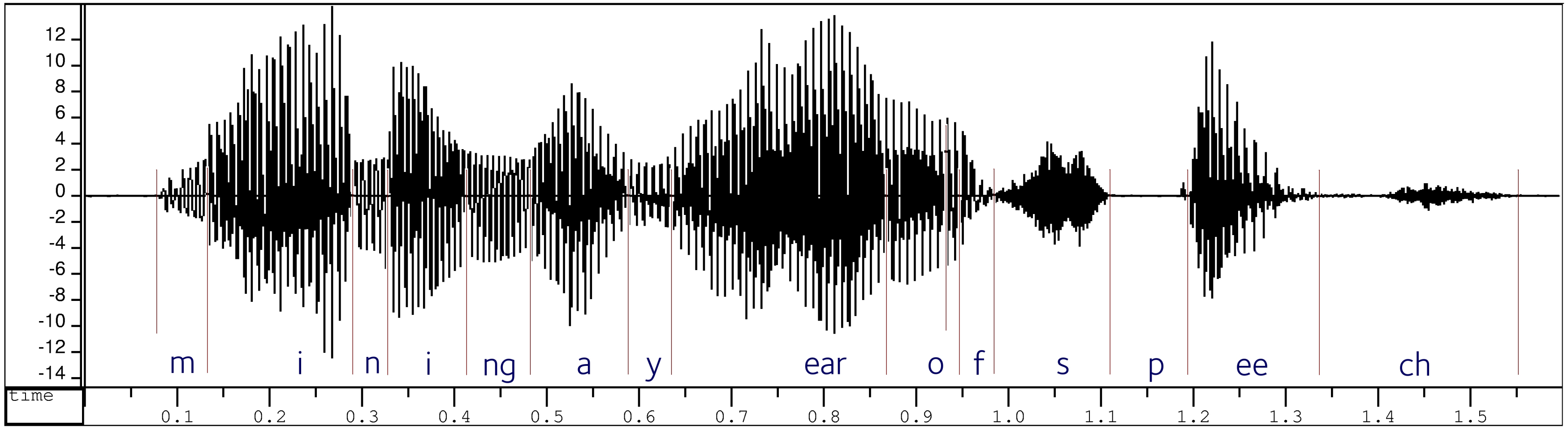
Sampling Waveforms

Sound is made of pressure variations.



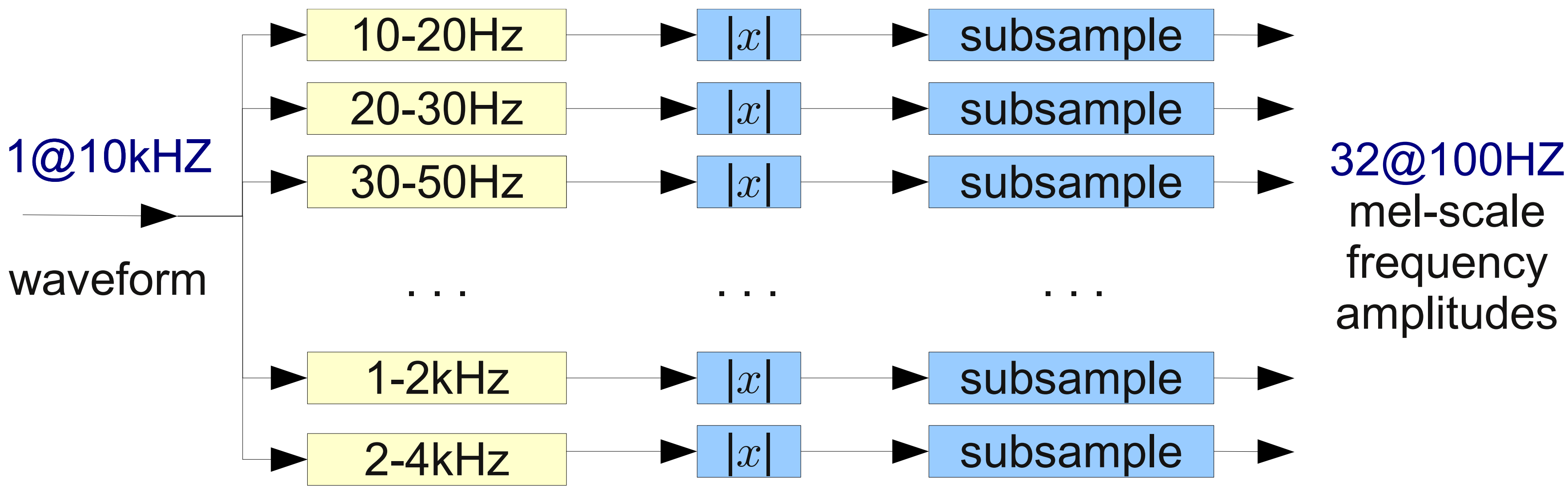
Digital waveforms

Digital speech waveform $\approx$ one number every 100 μsec.



Mel scaled filter bank

Preprocessing inspired by the human ear

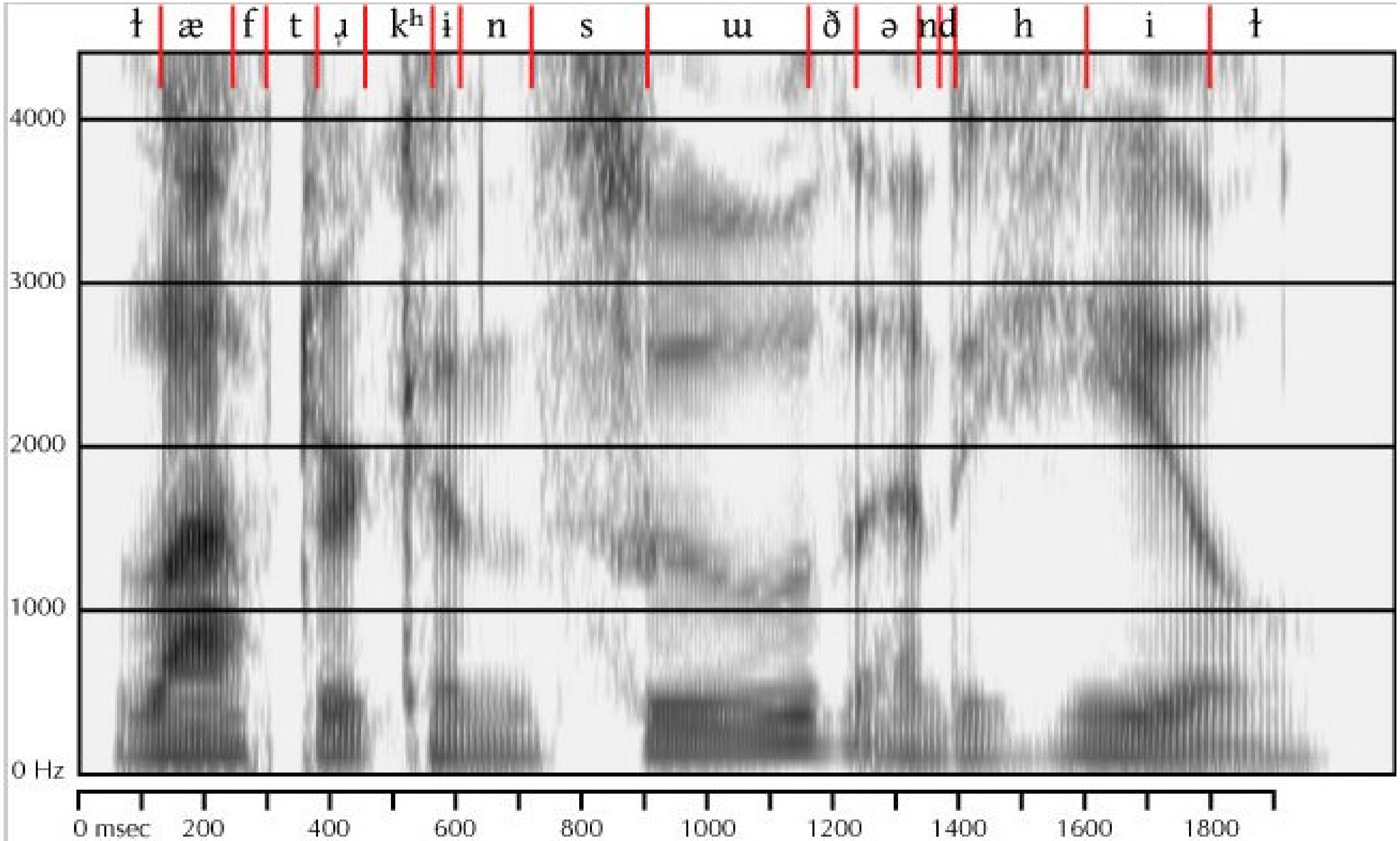


Resulting data

– Additional processing is common: MFCC, Delta encoding,. . .

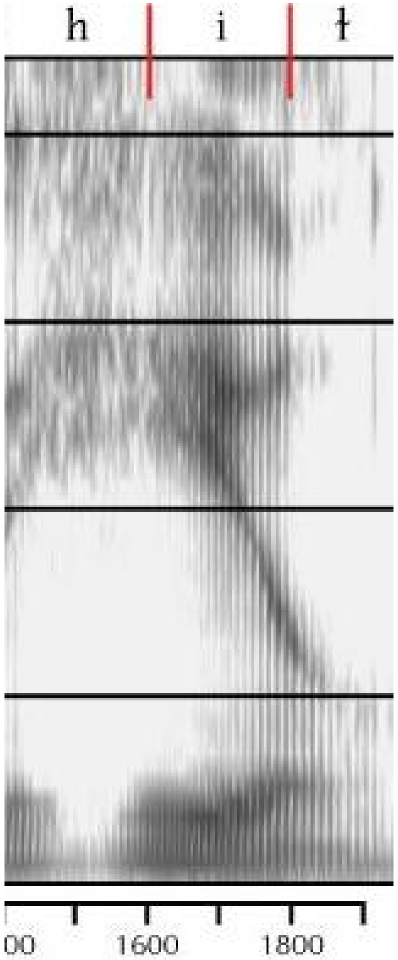
– One vector x_t with 16 to 48 coefficients every 5 to 20 ms.

Spectrogram



"Laughter can soothe and heal."

Coarticulation



“Heal”

Moving the mouth takes time.

– “h” shows the traces of the voiced “and”.
– “i” formants prepare the following “l”.

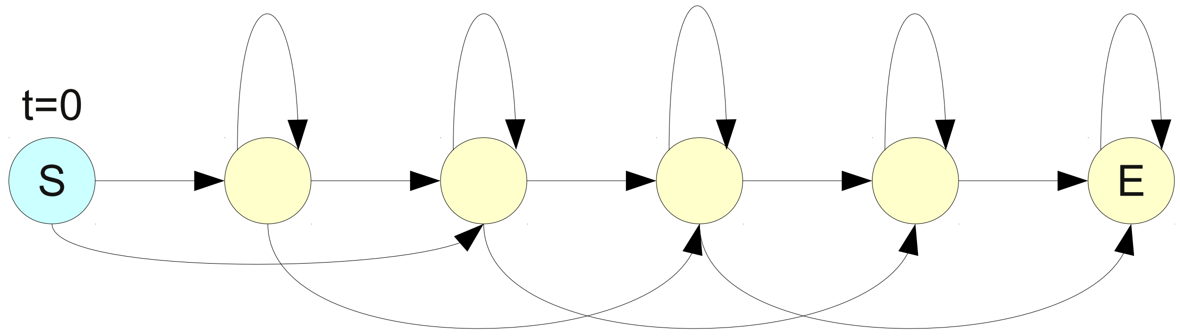
The sounds are all mixed.
Phoneme boundaries are an illusion!
Our brain reconstructs the phonemes.

But there is a clear sequential structure.

II. Hidden Markov Models

Generative word model

Markov state machine



Transition probabilities

– Markov assumption: s_t depends only on s_{t-1}.

– Invariance assumption: $P_\theta(s_t \,|\, s_{t-1}) \stackrel{\Delta}{=} a_{s_t,s_{t-1}}$ does not depend on t.

Emission probabilities

– Independence assumption: x_t depends only s_t (and sometimes s_{t-1})

– Continuous HMM: $P_\theta(x_t \,|\, s_t = s)$ is $\mathcal{N}(\mu_s, \Sigma_s)$.

– Discrete HMM: $P_\theta(x_t \in \mathcal{X}_c \,|\, s_t = s) \stackrel{\Delta}{=} b_{cs}$ with $\mathcal{X}_c$ defined by clustering.

The Ferguson problems

Likelihood

– Given a specific HMM,
compute the likelihood of an observation sequence.

$$P_\theta(x_1 \dots x_T) = \sum_{s_1 \dots s_T} P_\theta(x_1 \dots x_T,\, s_1 \dots s_T)$$

Decoding

– Given an observation sequence and an HMM,
discover the most probable hidden state sequence.

$$\arg\max_{s_1 \dots s_T} P_\theta(s_1 \dots s_T \,|\, x_1 \dots x_T) = \arg\max_{s_1 \dots s_T} P_\theta(s_1 \dots s_T\,,\, x_1 \dots x_T)$$

Learning

– Given an observation sequence, learn the HMM parameters.

– Like a mixture: learning would be easy if we knew $s_1 \dots s_T$.

$$\max_\theta \sum_{s_1 \dots s_T} P_\theta(s_1 \dots s_T)\, P_\theta(x_1 \dots x_T \,|\, s_1 \dots s_T)$$

Computing the likelihood

Exponential cost?

– The number of terms to sum grows exponentially with T.

$$L(\theta) \stackrel{\Delta}{=} P_\theta(x_1 \dots x_T) \;=\; \sum_{s_1 \dots s_T} P_\theta(x_1 \dots x_T,\, s_1 \dots s_T)$$

$$=\; \sum_{s_1 \dots s_T} \prod_{t=1}^{T} a_{s_{t-1} s_t}\, P_\theta(x_t | s_t)$$

– The sum runs over sequences $s_1 \dots s_T$ where $s_T \in \text{End}$.

Factoring the likelihood (1)

$$
\begin{aligned}
\forall t \quad L(\theta) \stackrel{\Delta}{=} P_\theta(x_1 \dots x_T) &= \sum_i P_\theta(x_1 \dots x_T,\, s_t{=}i) \\
&= \sum_i P_\theta(x_1 \dots x_t,\, s_t{=}i)\, P_\theta(x_{t+1} \dots x_T \,|\, x_1 \dots x_t\,,\, s_t{=}i) \\
&= \sum_i \underbrace{P_\theta(x_1 \dots x_t,\, s_t{=}i)}_{\stackrel{\Delta}{=}\, \alpha_t(i)}\ \underbrace{P_\theta(x_{t+1} \dots x_T \,|\, s_t{=}i)}_{\stackrel{\Delta}{=}\, \beta_t(i)}
\end{aligned}
$$

We have used the probabilistic relations
$P(A) = \sum_B P(A,B)$, $P(A,B) = P(A)\,P(B\,|\,A)$,
and the independence assumptions.

Factoring the likelihood (1bis)

Equivalent derivation:

$$
\begin{aligned}
L(\theta) \stackrel{\Delta}{=} P_\theta(x_1 \dots x_T) &= \sum_{s_1 \dots s_T} \; \prod_{t=1}^{T} a_{s_{t-1} s_t} \, P_\theta(x_t|s_t) \\
&= \sum_{s_t} \; \underbrace{\sum_{s_1 \dots s_{t-1}} \prod_{t'=1}^{t} a_{s_{t'-1} s_{t'}} \, P_\theta(x_{t'}|s_{t'})}_{\stackrel{\Delta}{=} \, \alpha_t(s_t)} \\
&\qquad \times \; \underbrace{\sum_{s_{t+1} \dots s_T} \prod_{t'=t+1}^{T} a_{s_{t'-1} s_{t'}} \, P_\theta(x_{t'}|s_{t'})}_{\stackrel{\Delta}{=} \, \beta_t(s_t)}
\end{aligned}
$$

We have only used the arithmetic relations: $AB + AC = A(B + C)$

Factoring the likelihood (2)

$$
\begin{aligned}
\alpha_t(s_t) &= P_\theta(x_1 \dots x_t,\, s_t) \\
&= \sum_{s_{t-1}} P_\theta(x_1 \dots x_t,\, s_t,\, s_{t-1}) \\
&= \sum_{s_{t-1}} P_\theta(x_1 \dots x_{t-1},\, s_{t-1}) \\
&\qquad\quad \times\; P_\theta(s_t \,|\, x_1 \dots x_{t-1},\, s_{t-1}) \\
&\qquad\quad \times\; P_\theta(x_t \,|\, x_1 \dots x_{t-1},\, s_{t-1},\, s_t) \\
&= \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})\, a_{s_{t-1}s_t}\, P_\theta(x_t|s_t)
\end{aligned}
$$

We have used the probabilistic relations
$P(A) = \sum_B P(A,B)$, $\quad P(A,B,C) = P(A)\,P(B\,|\,A)\,P(C\,|\,A,B)$
and the independence assumptions.

Factoring the likelihood (2bis)

Equivalent derivation:

$$
\begin{aligned}
\alpha_t(s_t) &= \sum_{s_1 \dots s_{t-1}} \prod_{t'=1}^{t} a_{s_{t'-1}s_{t'}} \, P_\theta(x_{t'} \,|\, s_{t'}) \\
&= \sum_{s_{t-1}} P_\theta(x_t|s_t) \, a_{s_{t-1}s_t} \sum_{s_1 \dots s_{t-2}} \prod_{t'=1}^{t-1} a_{s_{t'-1}s_{t'}} \, P_\theta(x_{t'} \,|\, s_{t'}) \\
&= \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1}) \, a_{s_{t-1}s_t} \, P_\theta(x_t|s_t)
\end{aligned}
$$

We have only used the arithmetic relations: $AB + AC = A(B+C)$

Factoring the likelihood (3)

$$\begin{aligned}
\beta_{t-1}(s_{t-1}) &= P_\theta(x_t \dots x_T \,|\, s_{t-1}) \\
&= \sum_{s_t} P_\theta(x_t \dots x_T \,|\, s_{t-1},\, s_t)\; P_\theta(s_t \,|\, s_{t-1}) \\
&= \sum_{s_t} P_\theta(x_{t+1} \dots x_T \,|\, s_{t-1},\, s_t) \\
&\qquad\quad \times\; P_\theta(x_t \,|\, x_{t+1} \dots x_T, s_{t-1},\, s_t)\; P_\theta(s_t \,|\, s_{t-1}) \\
&= \sum_{s_t} \beta_t(s_t)\; a_{s_{t-1}s_t}\; P_\theta(x_t|s_t)
\end{aligned}$$

Also derivable arithmetically.

Also with the chain rule: $\dfrac{\partial L}{\partial \alpha_{t-1}} = \beta_{t-1} = \left(\dfrac{\partial L}{\partial \alpha_t}\right)^\top \left(\dfrac{\partial \alpha_t}{\partial \alpha_{t-1}}\right) = \beta_t^\top \dfrac{\partial \alpha_t}{\partial \alpha_{t-1}}\,.$

Forward algorithm

Forward pass

$$\alpha_o(i) \;=\; \mathbb{I}\{i = \mathsf{Start}\}$$

$$\alpha_t(i) \;=\; \sum_j \alpha_{t-1}(j)\, a_{ji}\, P_\theta(x_t|s_t{=}i)$$

Likelihood

$$\beta_T(i) \;=\; \mathbb{I}\{i \in \mathsf{End}\}$$

$$P_\theta(x_1 \dots x_T) \;=\; \sum_i \alpha_T(i)\, \beta_T(i) \;=\; \sum_{i \in \mathsf{End}} \alpha_T(i)$$

Decoding

Forward works because $AB + AC = A(B + C)$.

But we also have $\max(AB, AC) = A \max(B, C)$ when $A, B, C \geq 0$.

$$\alpha_t(i) \triangleq \sum_{s_1 \dots s_{t-1}} \prod_{t'=1}^{t} a_{s_{t'-1}s_{t'}} P_\theta(x_{t'} \,|\, s_{t'})$$

$$\alpha^\star_t(i) \triangleq \max_{s_1 \dots s_{t-1}} \prod_{t'=1}^{t} a_{s_{t'-1}s_{t'}} P_\theta(x_{t'} \,|\, s_{t'})$$

Viterbi algorithm

$$\alpha^\star_o(i) = \mathbb{I}\{i = \text{Start}\}$$

$$\alpha^\star_t(i) = \max_j \alpha^\star_{t-1}(j)\, a_{ji}\, P_\theta(x_t | s_t = i)$$

$$\max_{s_1 \dots s_T} P_\theta(s_1 \dots s_T\,,\, x_1 \dots x_T) = \max_{i \in \text{End}} \alpha^\star_T(i)$$

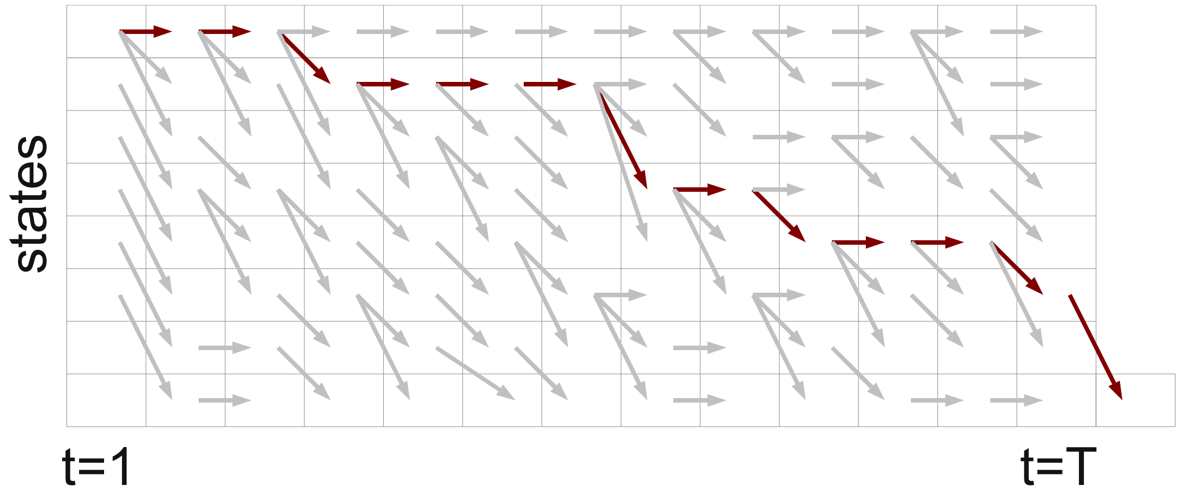
Viterbi Algorithm

Viterbi algorithm

$$\begin{aligned}\alpha_o^\star(i) &= \mathbb{I}\{i = \mathsf{Start}\} \\ \alpha_t^\star(i) &= \max_j \alpha_{t-1}^\star(j)\, a_{ji}\, P_\theta(x_t | s_t = i)\end{aligned}$$

$$\max_{s_1 \dots s_T} P_\theta(s_1 \dots s_T\,,\, x_1 \dots x_T) \;=\; \max_{i \in \mathsf{End}} \alpha_T^\star(i)$$

Viterbi backtracking



Learning

Expectation Maximization

– We only observe the $X = x_1 \dots x_T$.

– Learning would be easy if we knew $S = s_1 \dots s_T$.

Decomposition

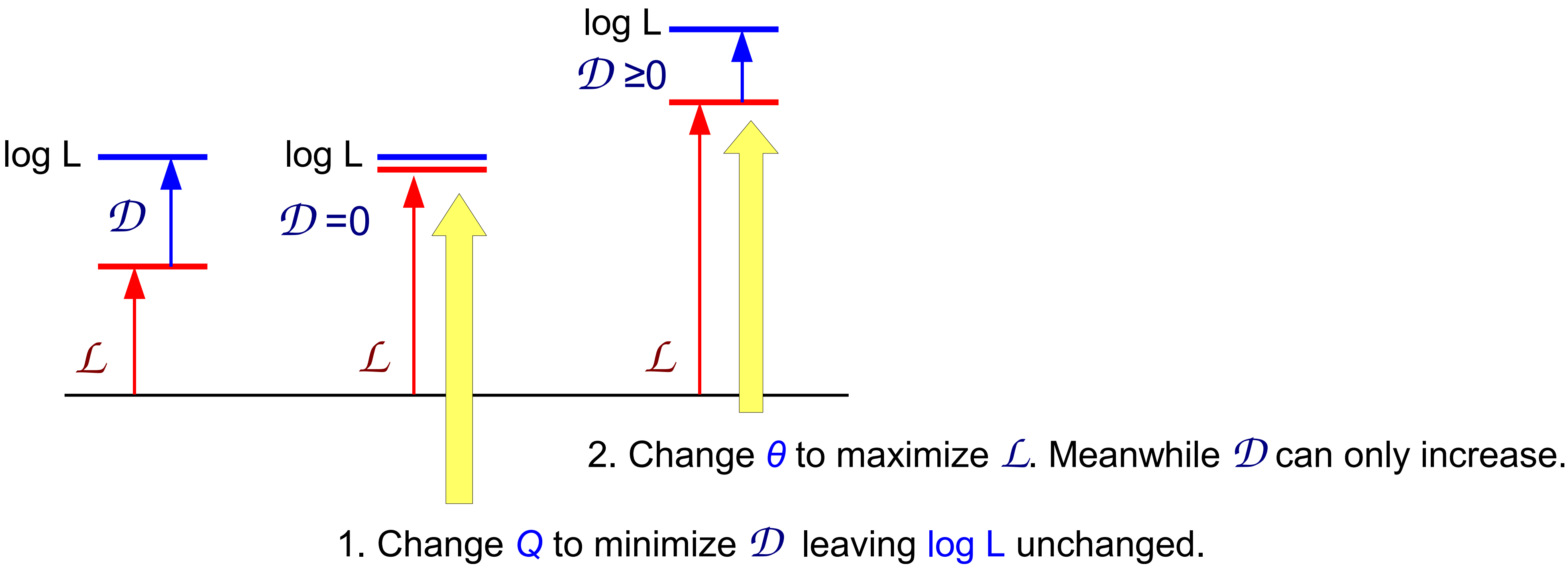
– For a given X, guess a distribution $Q(S|X)$.

– Regardless of our guess, $\log L(\theta) = \mathcal{L}(Q,\theta) + \mathcal{D}(Q,\theta)$

$$\mathcal{L}(Q,\theta) = \sum_{s_1 \dots s_T} Q(S\,|\,X) \log \frac{P_\theta(S)\, P_\theta(X\,|\,S)}{Q(S\,|\,X)} \qquad \text{Easy to maximize}$$

$$\mathcal{D}(Q,\theta) = \sum_{s_1 \dots s_T} Q(S\,|\,X) \log \frac{Q(S\,|\,X)}{P_\theta(S\,|\,X)} \qquad \text{KL divergence}$$

Expectation Maximization



E-Step: $Q(S \mid X) \propto P_\theta(S,\, X)$ — Memory?

M-Step: $a_{ij} \propto \sum_S \, Q(S|X)\ \text{Count}_S[i \to j]$ — Computation?

$\mu_i = \sum_S \, Q(S|X)\ \text{Avg}[x_t \text{ where } s_t = i]$

$\Sigma_i = \sum_S \, Q(S|X)\ \text{Avg}[(x_t - \mu_i)(x_t - \mu_i)^\top \text{ where } s_t = i]$

A closer look at the derivations (1)

$$\mathcal{L}(Q,\theta) \;=\; \sum_{s_1 \dots s_T} Q(S\,|\,X) \log \frac{P_\theta(S)\, P_\theta(X\,|\,S)}{Q(S\,|\,X)}$$

$$\;=\; \sum_{s_1 \dots s_T} Q(S\,|\,X) \left[\sum_t \log a_{s_{t-1}s_t} + \sum_t \log P_\theta(x_t|s_t) - \log Q(S|X) \right]$$

Since $\sum_j a_{ij} = 1$, the following relation holds at the optimum:

$$\frac{\partial \mathcal{L}}{\partial a_{ij}} \;=\; \sum_{s_1 \dots s_T} Q(S\,|\,X) \sum_t \frac{\mathbb{1}\{s_{t-1}\!=\!i\}\; \mathbb{1}\{s_t\!=\!j\}}{a_{ij}} \;=\; K_i$$

Therefore

$$a_{ij} \;\propto\; \sum_{s_1 \dots s_T} Q(S\,|\,X) \;\sum_{t=1}^{T} \mathbb{1}\{s_{t-1}\!=\!i\}\; \mathbb{1}\{s_t\!=\!j\}$$

A closer look at the derivations (2)

$$
\begin{aligned}
a_{ij} &\propto \sum_{t=1}^{T} \sum_{s_1 \dots s_T} Q(S \,|\, X)\ \mathbb{1}\{s_{t-1}=i\}\ \mathbb{1}\{s_t=j\} \\
&\propto \sum_{t=1}^{T} Q(s_{t-1}=i,\ s_t=j \,|\, x_1 \dots x_T) \ \propto\ \sum_{t=1}^{T} Q(s_{t-1}=i,\ s_t=j,\ x_1 \dots x_T) \\
&\propto \sum_{t=1}^{T} \underbrace{Q(x_1 \dots x_{t-1},\ s_{t-1}=i)}_{\alpha_{t-1}(j)}\ \underbrace{Q(s_t = j \,|\, s_{t-1}=i, \cdots)}_{a_{ij}} \\
&\qquad \times\ \underbrace{Q(x_t \,|\, s_t=j, \cdots)}_{P_\theta(x_t|s_t)}\ \underbrace{Q(x_{t+1} \dots x_T \,|\, s_t=j, \cdots)}_{\beta_t(i)}
\end{aligned}
$$

We do not need to store $Q(S|X)$

We only need to store $\alpha_t(s)$, $\beta_t(s)$, and $B_t(s) = P_\theta(x_t|s_t=s)$ for all t and s.

Forward backward algorithm

E-Step

Emission: $\forall t \ \forall i \quad B_t(i) = P_\theta(x_t | s_t{=}i)$

Forward pass: $\alpha_o(i) = \mathbb{1}\{i = \text{Start}\}$

for $t = 1 \dots T$, $\quad \forall i \quad \alpha_t(i) = \sum_j \alpha_{t-1}(j)\, a_{ji}\, B_t(i)$

Backward pass: $\beta_T(i) = \mathbb{1}\{i \in \text{End}\}$

for $t = T \dots 1$, $\quad \forall i \quad \beta_{t-1}(i) = \sum_j \beta_t(j)\, a_{ij}\, B_t(j)$

M-Step:

Baum-Welch formulas for continuous HMM.

$$a_{ij} \leftarrow \frac{\sum_t \alpha_{t-1}(i)\, a_{ij}\, B_t(j)\, \beta_t(j)}{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)}$$

$$\mu_i \leftarrow \frac{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)\, x_t}{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)} \qquad \Sigma_i \leftarrow \frac{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)\, x_t\, x_t^\top}{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)} - \mu_i\, \mu_i^\top$$

Forward backward algorithm

E-Step

Emission: $\forall t\ \forall i\quad B_t(i) = P_\theta(x_t | s_t{=}i)$

Forward pass: $\alpha_o(i) = \mathbb{1}\{i = \text{Start}\}$

for $t = 1 \dots T$, $\quad \forall i \quad \alpha_t(i) = \sum_j \alpha_{t-1}(j)\, a_{ji}\, B_t(i)$

Backward pass: $\beta_T(i) = \mathbb{1}\{i \in \text{End}\}$

for $t = T \dots 1$, $\quad \forall i \quad \beta_{t-1}(i) = \sum_j \beta_t(j)\, a_{ij}\, B_t(j)$

M-Step:

Baum-Welch formulas for discrete HMM.

$$a_{ij} \leftarrow \frac{\sum_t \alpha_{t-1}(i)\, a_{ij}\, B_t(j)\, \beta_t(j)}{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)}$$

$$b_{cs} \leftarrow \frac{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)\, \mathbb{1}\{x_t \in \mathcal{X}_c\}}{\sum_t \alpha_{t-1}(i)\, \beta_{t-1}(i)}$$

The Ferguson problems

Likelihood

– Given a specific HMM,
compute the likelihood of an observation sequence.

$\Longrightarrow$ Forward algorithm

Decoding

– Given an observation sequence and an HMM,
discover the most probable hidden state sequence.

$\Longrightarrow$ Viterbi algorithm

Learning

– Given an observation sequence, learn the HMM parameters.

$\Longrightarrow$ Forward-Backward algorithm

III. Segmentation and Recognition

Recognition only

Problem

– Classify sequence X as one of the categories $c \in \mathcal{C}$.
– Example: isolated word recognition.

Training

– Train HMM model W_c for each sequence category c.
– To train with multiple sequences for each category, accumulate numerator and denominator in the Baum-Welch formulas.

Prior probabilities

– Determine prior probabilities $P(C{=}c)$ for all categories.

Recognition

– Bayes rule says $P(C|x_1 \dots x_T) \;=\; \dfrac{P(X|C)\,P(C)}{P(X)}$.

– Therefore return category $\boldsymbol{\arg\max_c P_\theta(x_1 \dots x_T \,|\, W_c)}$ as computed using the forward algorithm in model W_c.

Simultaneous recognition and segmentation

Problem

– Split sequence X into segments belonging to categories $c \in \mathcal{C}$.

– Example: continuous speech recognition.

Training the HMM

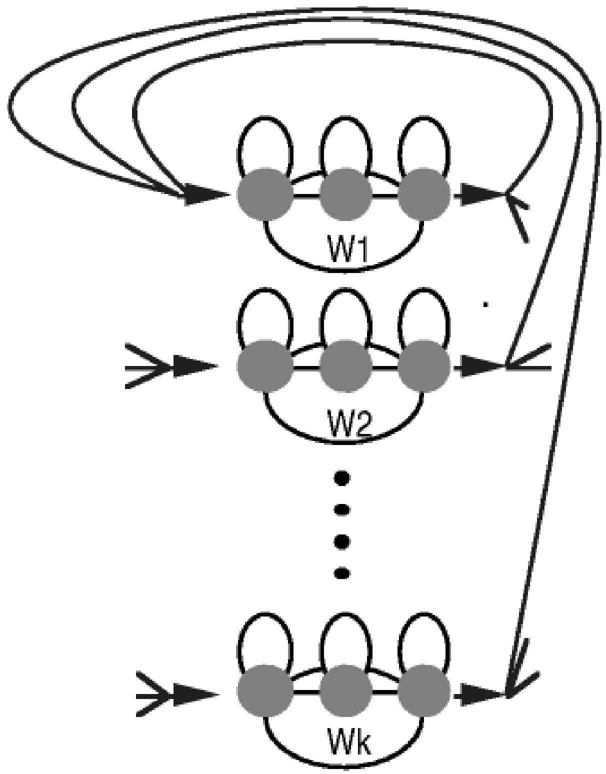
– Train HMM model W_c for each sequence category c.

Prior probabilities

– Prepare a bigram language model for sequences of categories.
Determine $P(c_{t+1} \,|\, c_t)$ using adequate data.

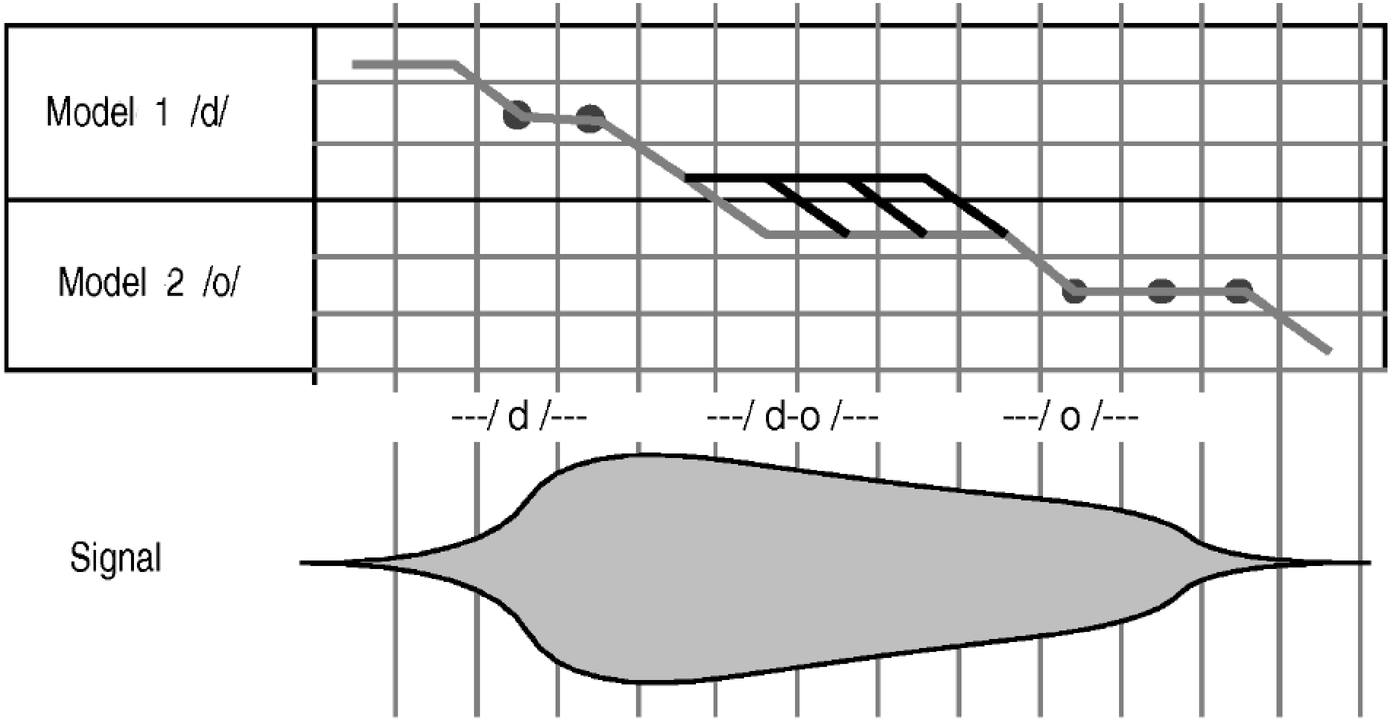
Simultaneous recognition and segmentation

Construct a super model



Simultaneous recognition and segmentation

Run Viterbi on the super model



No coarticulation modelling yet.

Training with unsegmented data

Problem

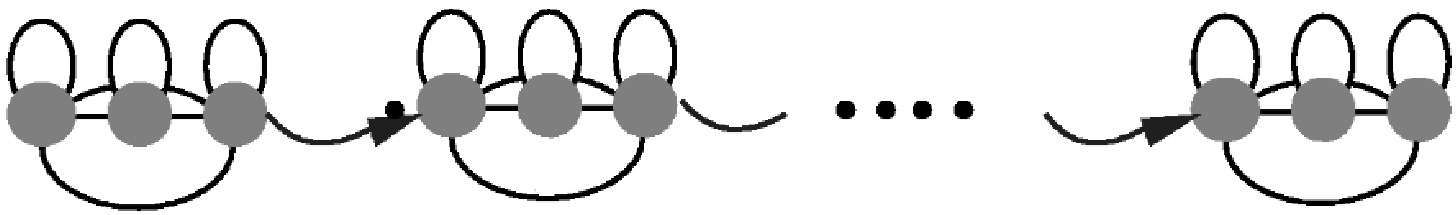
– Segmenting the training data is labor intensive

– Assume we have sequence of labels $c_1 \ldots c_S$ without segmentation.

Construct sequence model

– Concatenate models $W_{c_1} \ldots W_{c_S}$

– Run forward-backward algorithm.



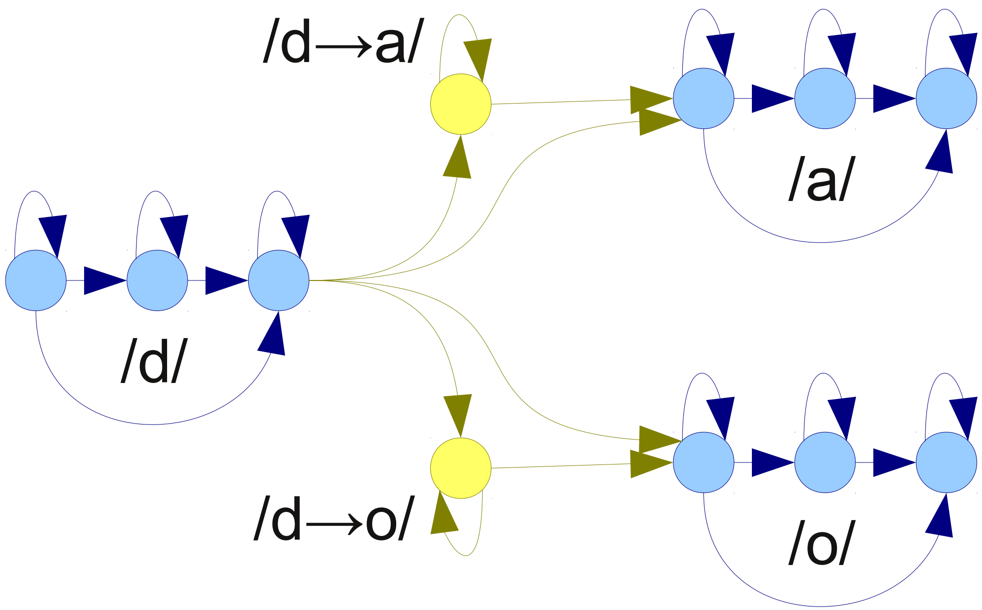
Dealing with coarticulation

Problem

– Transitions between categories need specific modelling.

Solution

– Develop more refined ways to combine models



Using more complex language models

Problem

– Bigram language models are rarely good enough.

Solution

– Develop more refined ways to combine models.

Finite state transducers

– A generic method to combine models
– We'll see them in a couple lectures.