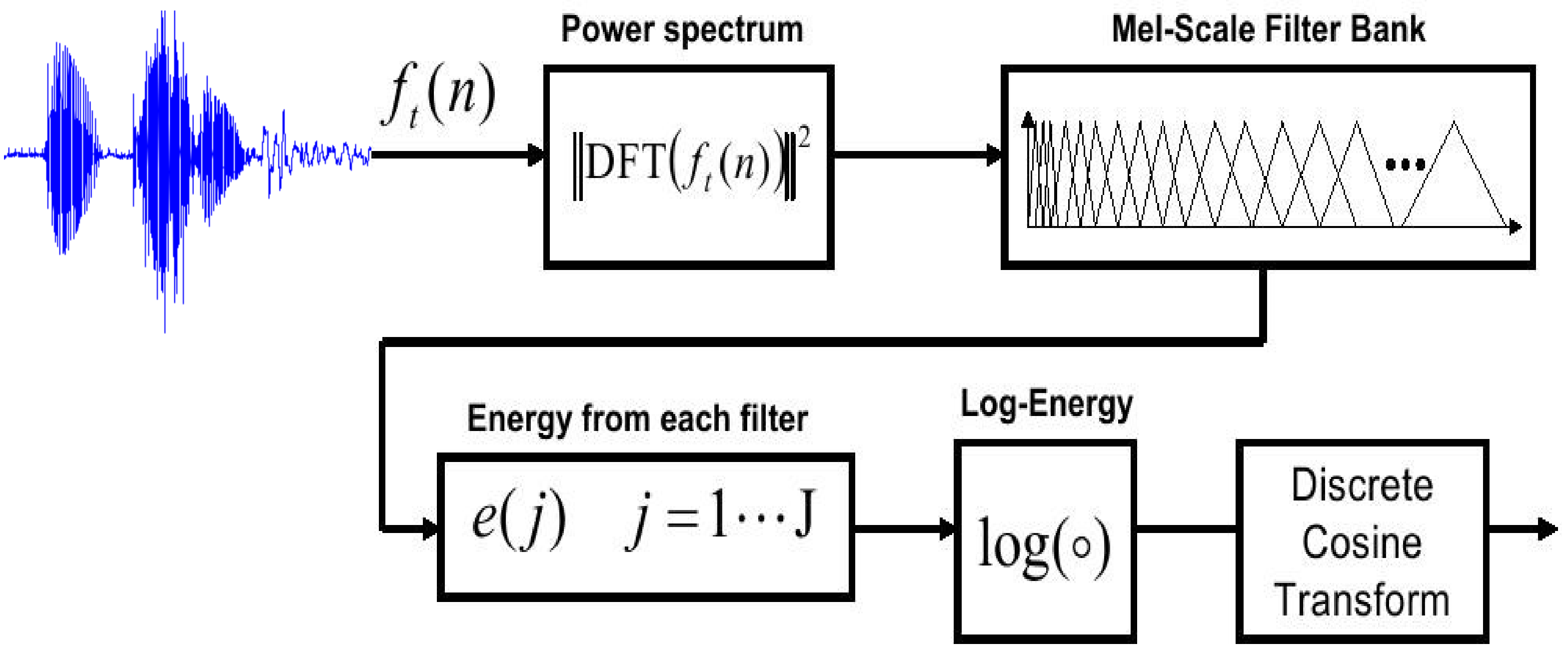
Timeline in the course

	Meetings		Home exercises	Project work
	Wednesdays	Thursdays		status
Week1	Speech features and classification		1.Feature classifier	Literature study
Week2	Phoneme modeling and recognition		2.Word recognizer	Work plan
Week3	Lexicon and language modeling		3.Text predictor	Analysis
Week4	Continuous speech and advanced search		4.Speech recognizer	Experimentation
Week5	End-to-end ASR		5.End-to-end recognizer	Preparing reports
Week6	Projects1	Projects2		Presentations
Week7	Projects3	Projects4		Report submissior
		Conclusion		

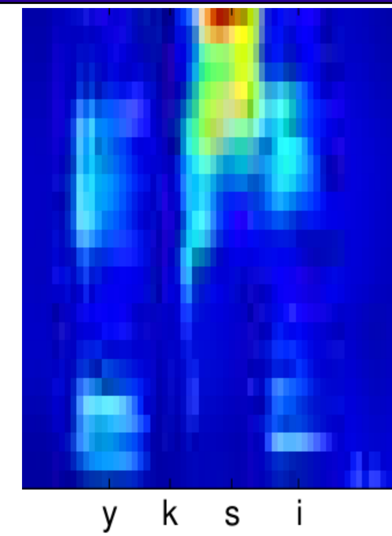
Learning goals for this week

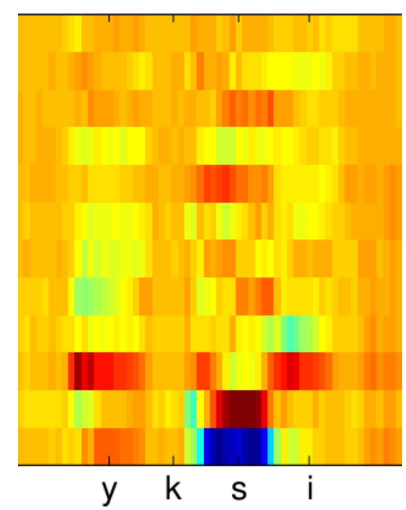
1. **Preprocessing, features, GMM**
   - remind of last week
2. Phonemes
   - know different units of speech
3. Hidden Markov model, HMM
   - learn to build a temporal model of speech units
   - learn how to train and use the model

Review: computation of MFCC

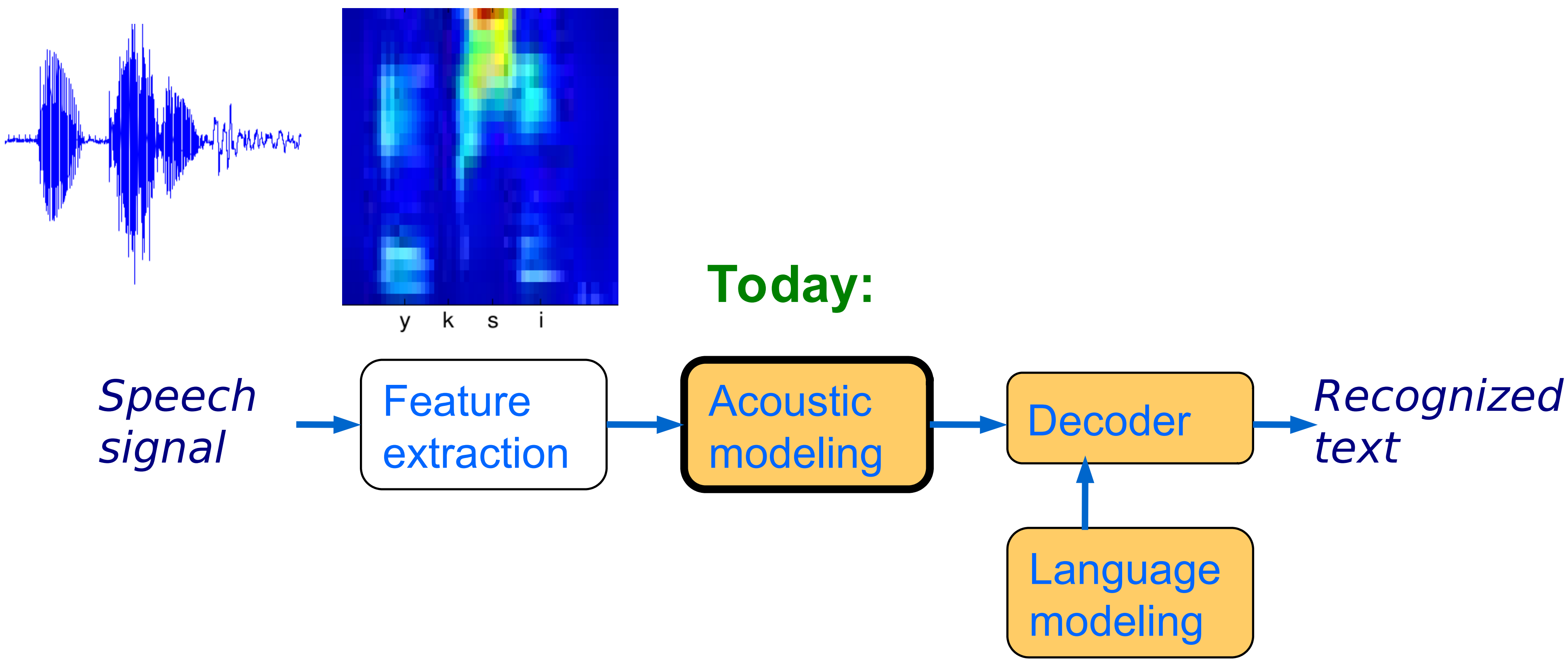



```
y = wavread('yksi.wav');

s = spectrogram(y,hamming(400),240);

spec = sqrt(abs(s));

mfcc = D*log(1+M*spec);
```





Review: speech recognition -from beginning to end



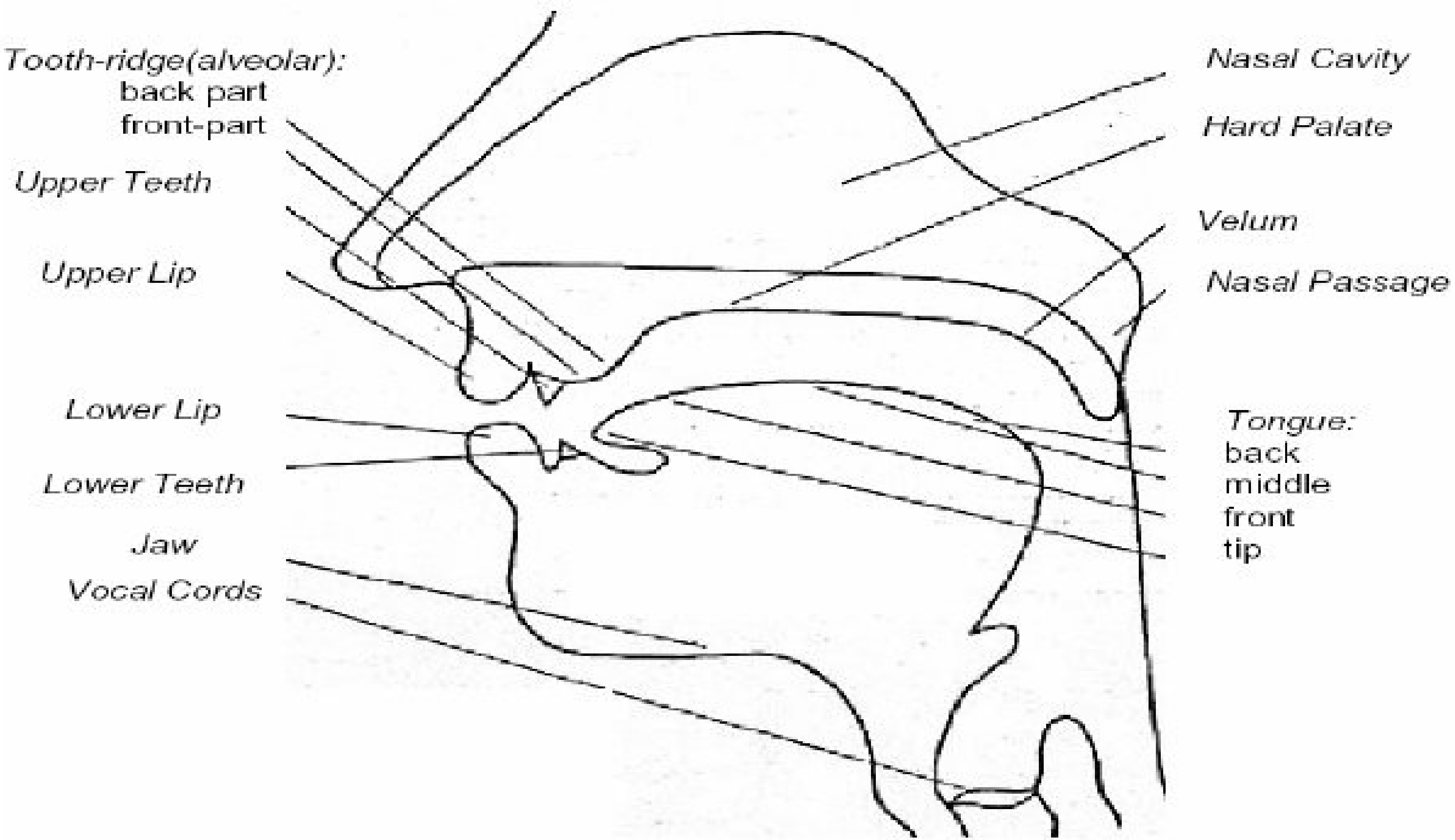
Content this week

1. Review: Preprocessing, features, GMM
2. **Phonemes**
3. Hidden Markov model, HMM
4. Home exercise 2: Build a GMM-HMM system to recognize spoken words
5. Feedback
6. Project status
    - Have a group meeting (as agreed with your tutor)

Description of speech sounds

- **Speech can be written down** using abstract units called phonemes
- **Phonemes** describe the sounds **by the way they are produced by human**
- Main classes:
  - vowels: air flow is not obstructed
  - consonants: air flow is partially or totally obstructed
- There are different writing systems, e.g. IPA (International Phonetic Alphabet)
- The phoneme sets differ depending on language

Production of speech sounds



Picture from Huang's text book (2001)

IPA symbols for US English

PHONEME	EXAMPLE	PHONEME	EXAMPLE	PHONEME	EXAMPLE
/i^{y}/	beat	/s/	see	/w/	wet
/ɪ/	bit	/š/	she	/r/	red
/e^{y}/	bait	/f/	fee	/l/	let
/ɛ/	bet	/θ/	thief	/y/	yet
/æ/	bat	/z/	z	/m/	meet
/ɑ/	Bob	/ž/	Gigi	/n/	neat
/ɔ/	bought	/v/	v	/ŋ/	sing
/ʌ/	but	/ð/	thee	/č/	church
/o^{w}/	boat	/p/	pea	/ǰ/	judge
/ʊ/	book	/t/	tea	/h/	heat
/u^{w}/	boot	/k/	key		
/ɝ/	Burt	/b/	bee		
/ɑ^{y}/	bite	/d/	Dee		
/ɔ^{y}/	Boyd	/g/	geese		
/ɑ^{w}/	bout				
/ə/	about				

Table from MIT Course Notes: 6.345 Automatic Speech Recognition, Spring 2003

CMU Sphinx ASR system symbols

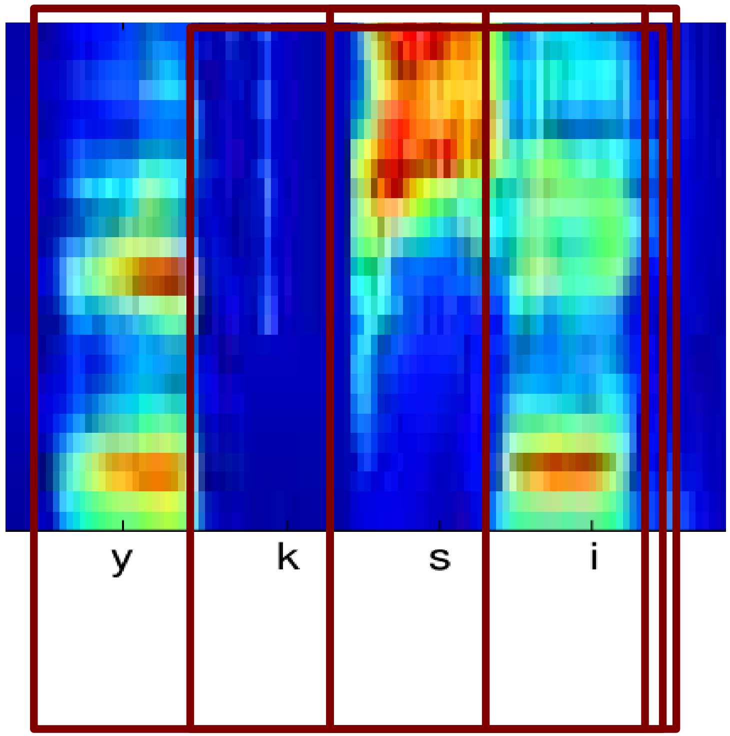
Phone	Example	Phone	Example	Phone	Example
AA	odd	EY	ate	P	pee
AE	at	F	fee	PD	lip
AH	hut	G	green	R	read
AO	ought	GD	bag	S	sea
AW	cow	HH	he	SH	she
AX	abide	IH	it	T	tea
AXR	user	IX	acid	TD	lit
AY	hide	IY	eat	TH	theta
B	be	JH	gee	TS	bits
BD	Dub	K	key	UH	hood
CH	cheese	KD	lick	UW	two
D	dee	L	lee	V	vee
DD	dud	M	me	W	we
DH	thee	N	note	Y	yield
DX	matter	NG	ping	Z	zee
EH	ed	OW	oat	ZH	seizure
ER	hurt	OY	toy	SIL	(silence)

Picture by B.Pellom

Acoustic model of speech

- Discussion: What speech units would suit for ASR?
- (how long, how many, language-dependence)
- (is the linguistic phoneme definition optimal?)

Why these discussions?
Learning happens, when:
+ brains are active and alert
+ new knowledge contradicts
your old believes



In ASR: Context-dependent phonemes

- **Context independent model,** Monophone /X/
  - Example: three => th + r + iy
  - does a phoneme sound the same in all contexts ?
- **Context dependent model,** Triphone /Left-X+Right/
  - Example: three => sil-**th**+r + th-**r**+iy + r-**iy**+sil
  - 25 phonemes => 25*25*25 = 15 625 triphones
  - do all the contexts exist ?
  - do all the contexts sound different ?
  - can we share parts of the model between some contexts, e.g. beginning, center, middle part?

Content this week

1. Preprocessing and features, GMM
2. Phonemes
3. **Hidden Markov Model**
4. Home exercise 2: Build a GMM-HMM system to recognize spoken words
5. Feedback
6. Project status
   - Have a group meeting (as agreed with your tutor)

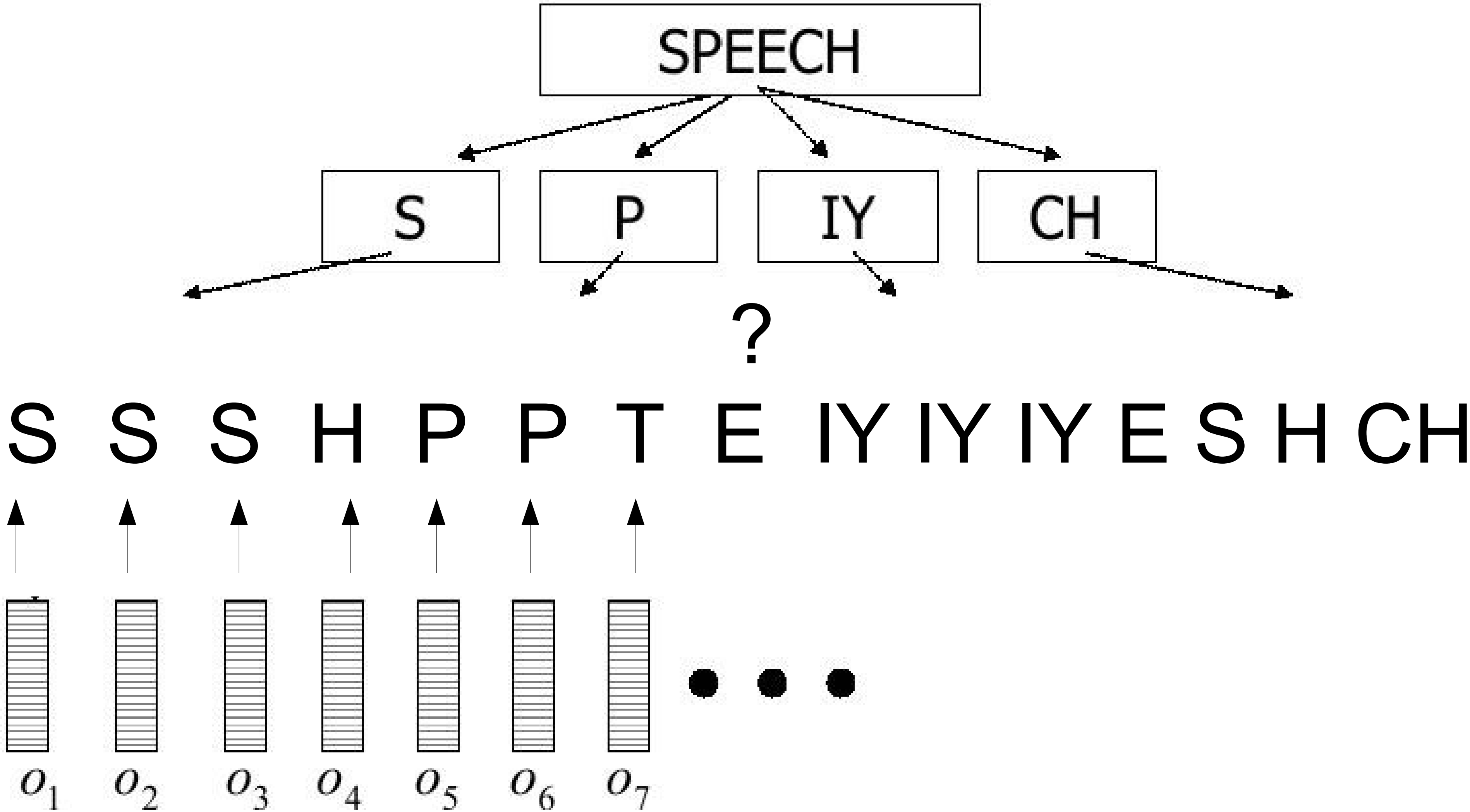
Results of GMM classification?

- This is not yet speech recognition, not even phoneme recognition!
- How to utilize this in phoneme recognition?

ssssssssssssssssssyeeeeeeeeeieeiiiiyiyiiiiiiiiiisssssssssssssssssssssstt

ttptpppptkkppkkkkkykoooooooouooooooooooolllllllllllaolmmmmmmmmmmmmmmmiiiieeeeyy

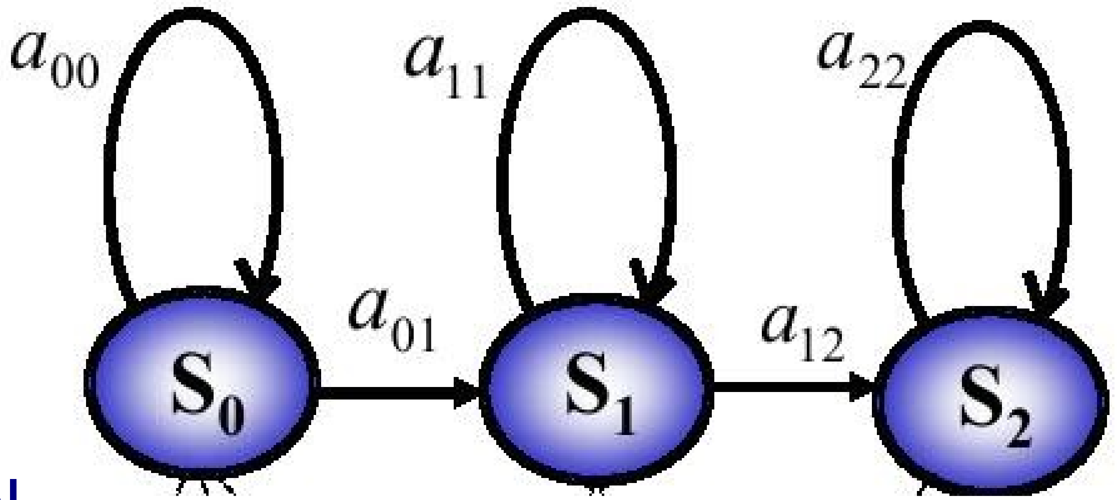
How to model a sequence of frames or phonemes?



Picture by B.Pellom

Hidden Markov model

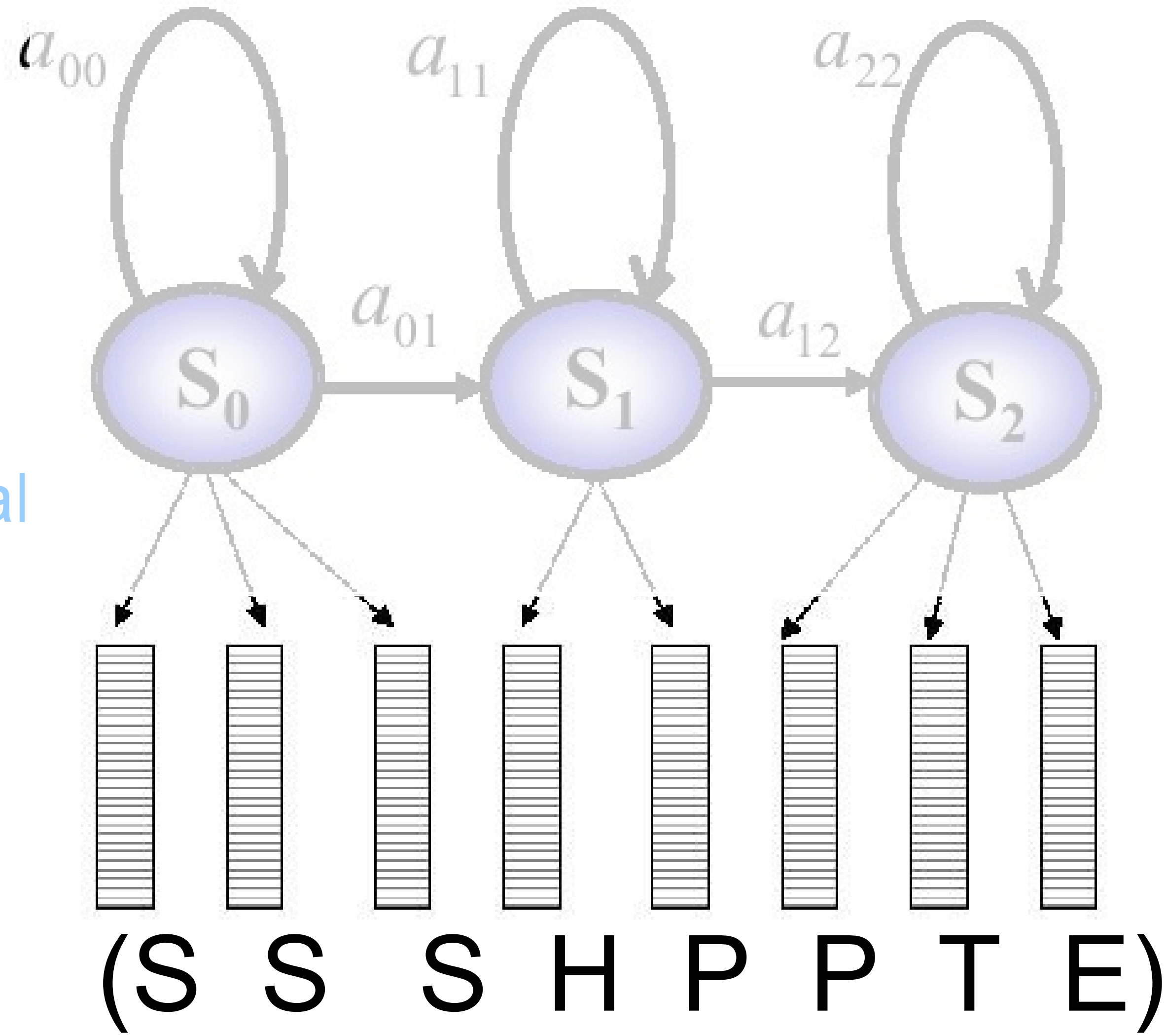
1. HMM is a system that has a set of operational states
2. From state *i* it moves to state *j* by probability *a(ij)*
3. Each state emits a characteristic sound signal
4. Signals are measured by feature vectors
5. The system's internal **state is hidden**, only the **feature vectors** are **measured**



Picture by B.Pellom

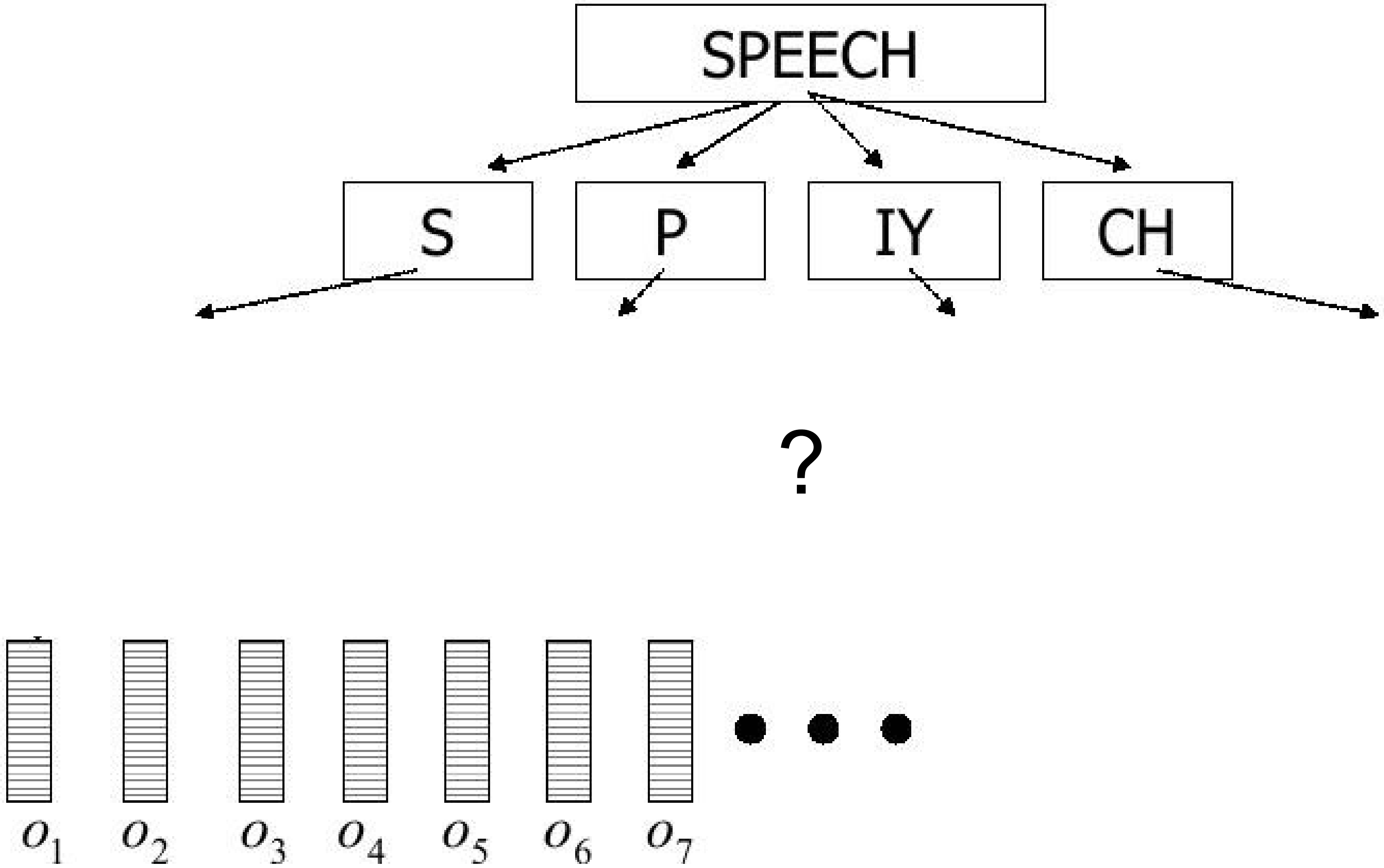
Hidden Markov model

1. HMM is a system that has a set of operational states
2. From state *i* it moves to state *j* by probability *a(ij)*
3. Each state emits a characteristic sound signal
4. Signals are measured by feature vectors
5. The system's internal **state is hidden**, only the **feature vectors** are **measured**

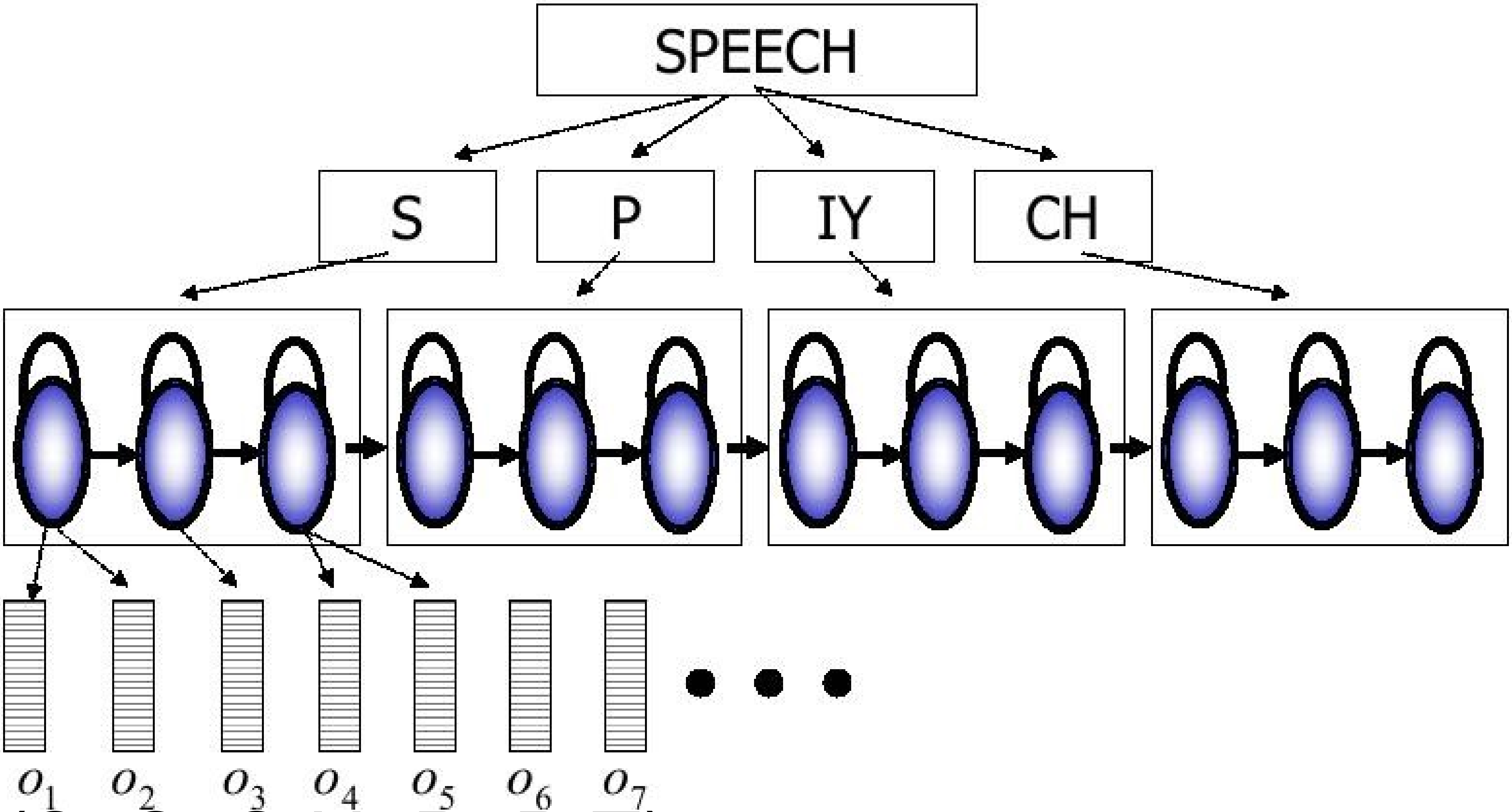


Picture by B.Pellom

How to model a sequence of frames or phonemes?



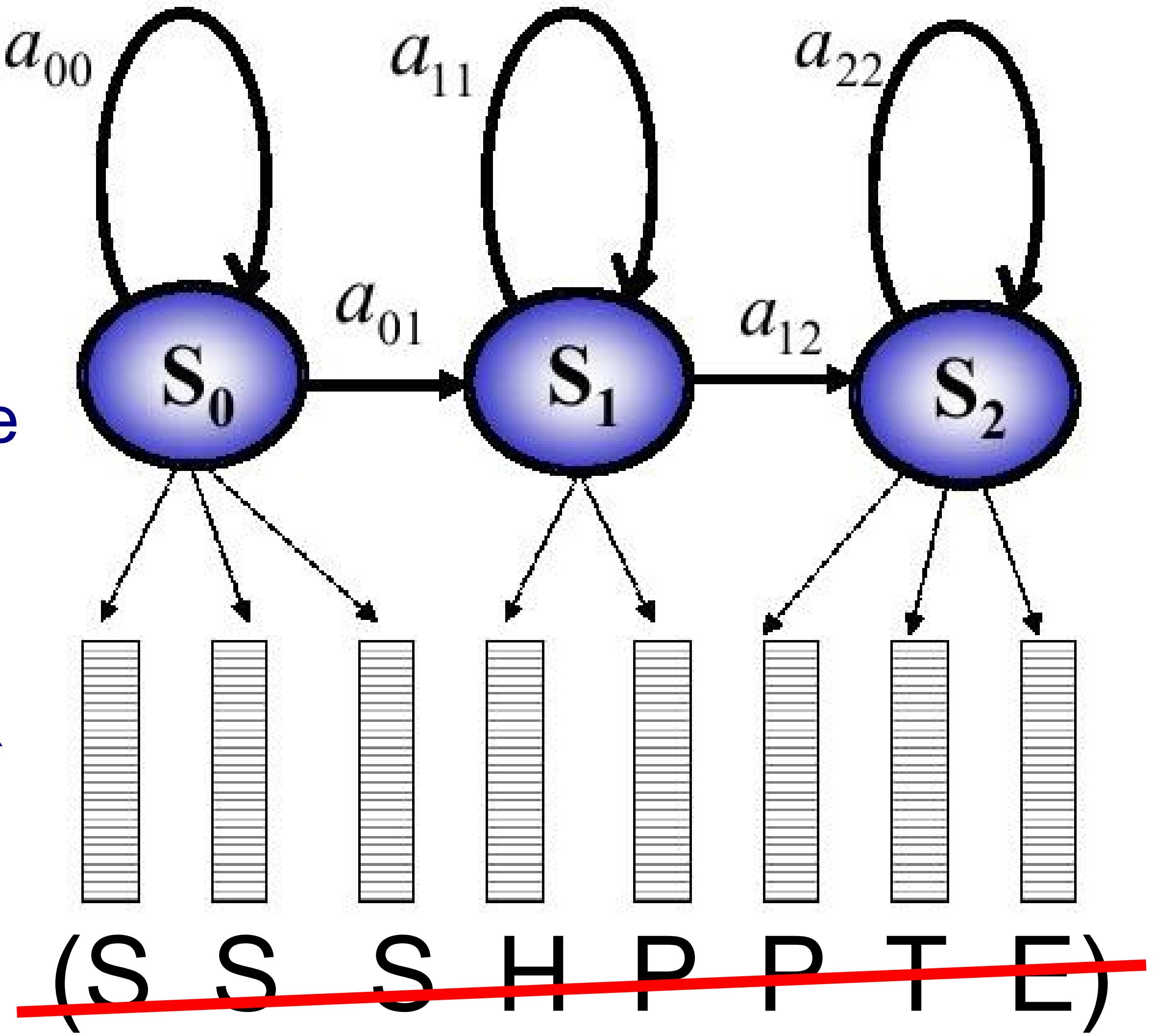
HMM as a phoneme model



(S S S H P P T)

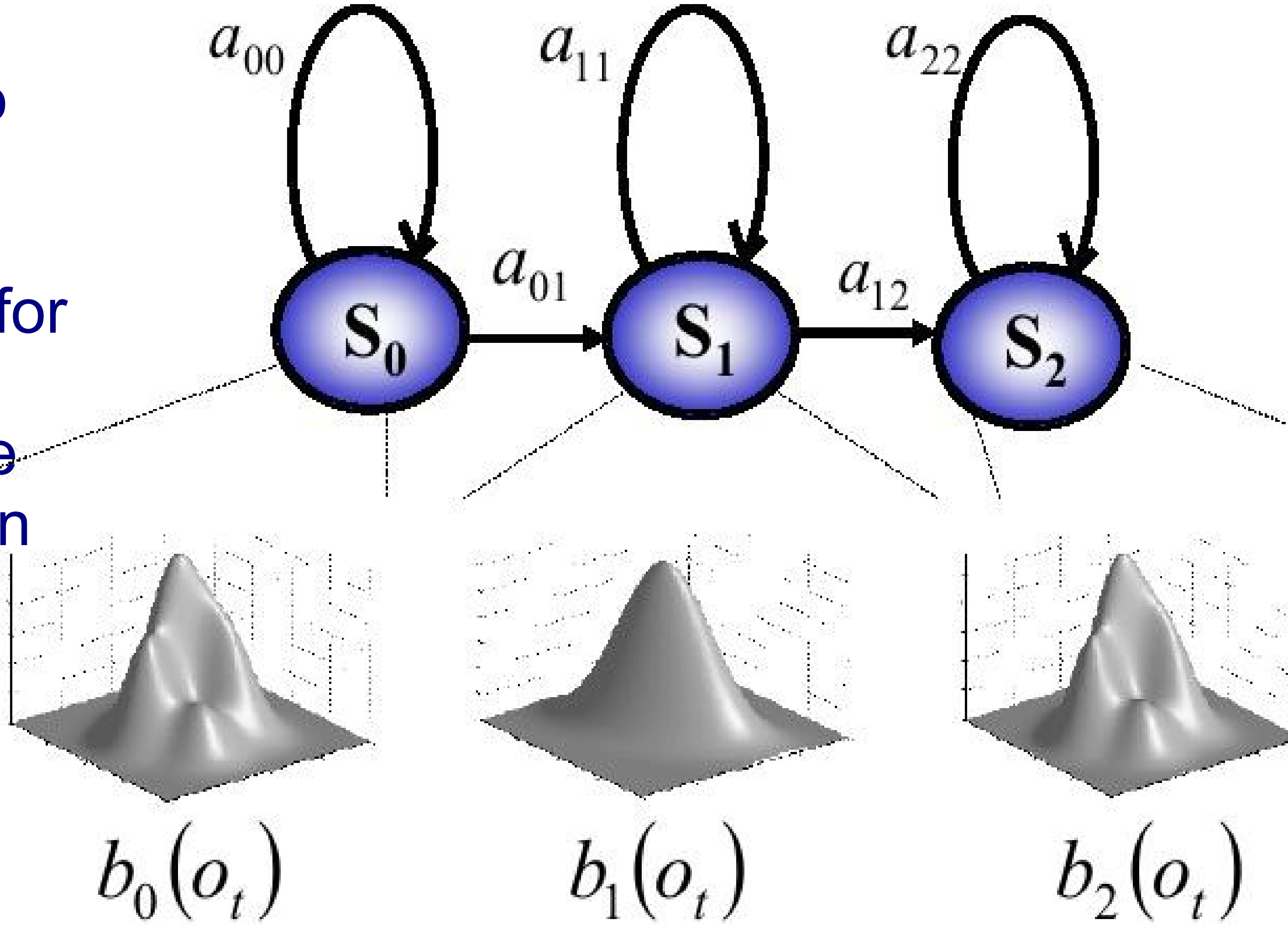
HMM as a phoneme model

- After **segmenting** each word sample into sounds, we find the set of feature vectors that represent a certain state
- These feature vectors are used to model the outputs in the state (by GMM e.g.)
- After modeling the states the HMM is ready for ASR



GMM-HMM system

- Each state emits sounds according to its GMM model
- This generative model can be used for **text-to-speech**, too
- The higher *a(ii)*, the longer is the duration



Picture by B.Pellom

Basic operations with HMMs

1. **Scoring**: - How to compute the probability of the observation sequence for a model?

2. **Decoding**: - How to compute the best state sequence for the observations?

3. **Training**: - How to set the model parameters to maximize the probability of the training samples?

GMM-HMM parameters

- Transition probability matrix **a**
  - Transition probability between state i and j is **a**(i,j)
- Observation probability function **b** of feature **x** is **b(x),** for example GMM:

$$f(x) = \sum_{m=1}^{M} w_m \mathrm{N}_m\left(x; \mu_m, \Sigma_m\right)$$

$$= \sum_{m=1}^{M} \frac{w_m}{(2\pi)^{n/2} \left|\Sigma_m\right|^{1/2}} \exp\left[-\frac{1}{2}\left(x-\mu_m\right)^{\mathrm{T}} \Sigma_m^{-1}\left(x-u_m\right)\right]$$

Basic operations with HMMs

1. **Scoring**: - How to compute the probability of the observation sequence for a model?

2. **Decoding**: - How to compute the best state sequence for the observations?

3. **Training**: - How to set the model parameters to maximize the probability of the training samples?

Article: Rabiner (1989), *Tutorial on hidden Markov models and selected applications*

1. Scoring

- Given an observation sequence,

$$\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$$

- Want to compute probability of generating it:

$$P(\mathbf{O} \mid \lambda)$$

- Let's assume a particular sequence of states,

$$q = \{q_1, q_2, \cdots, q_T\}$$

Scoring directly

- **Probability of the observation sequence given the state sequence,**

$$P(\mathbf{O} \mid q, \lambda) = \prod_{t=1}^{T} p(\mathbf{o}_t \mid q_t, \lambda)$$

$$= b_{q_1}(\mathbf{o}_1) \cdot b_{q_2}(\mathbf{o}_2) \cdots b_{q_T}(\mathbf{o}_T)$$

- **Probability of the state sequence,**

$$P(q \mid \lambda) = \pi_{q_1}(a_{q_1 q_2}) \cdot (a_{q_2 q_3}) \cdots (a_{q_{T-1} q_T})$$

Scoring directly?

- **Using the chain rule,**

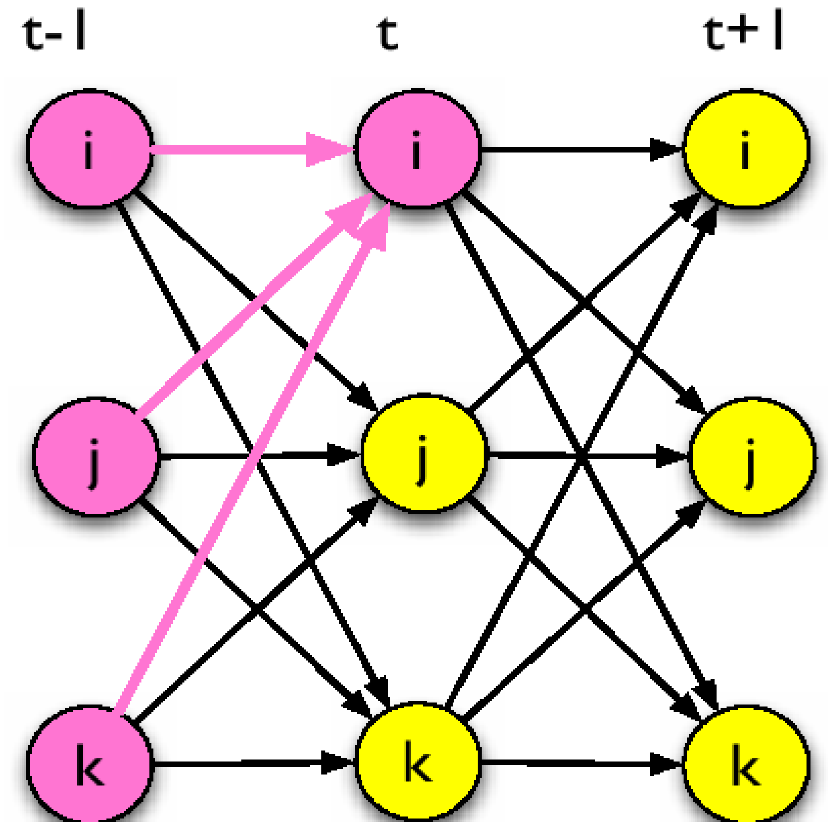
$$P(\mathbf{O} \mid \lambda) = \sum_{\text{all q}} P(\mathbf{O} \mid q, \lambda) P(q \mid \lambda)$$

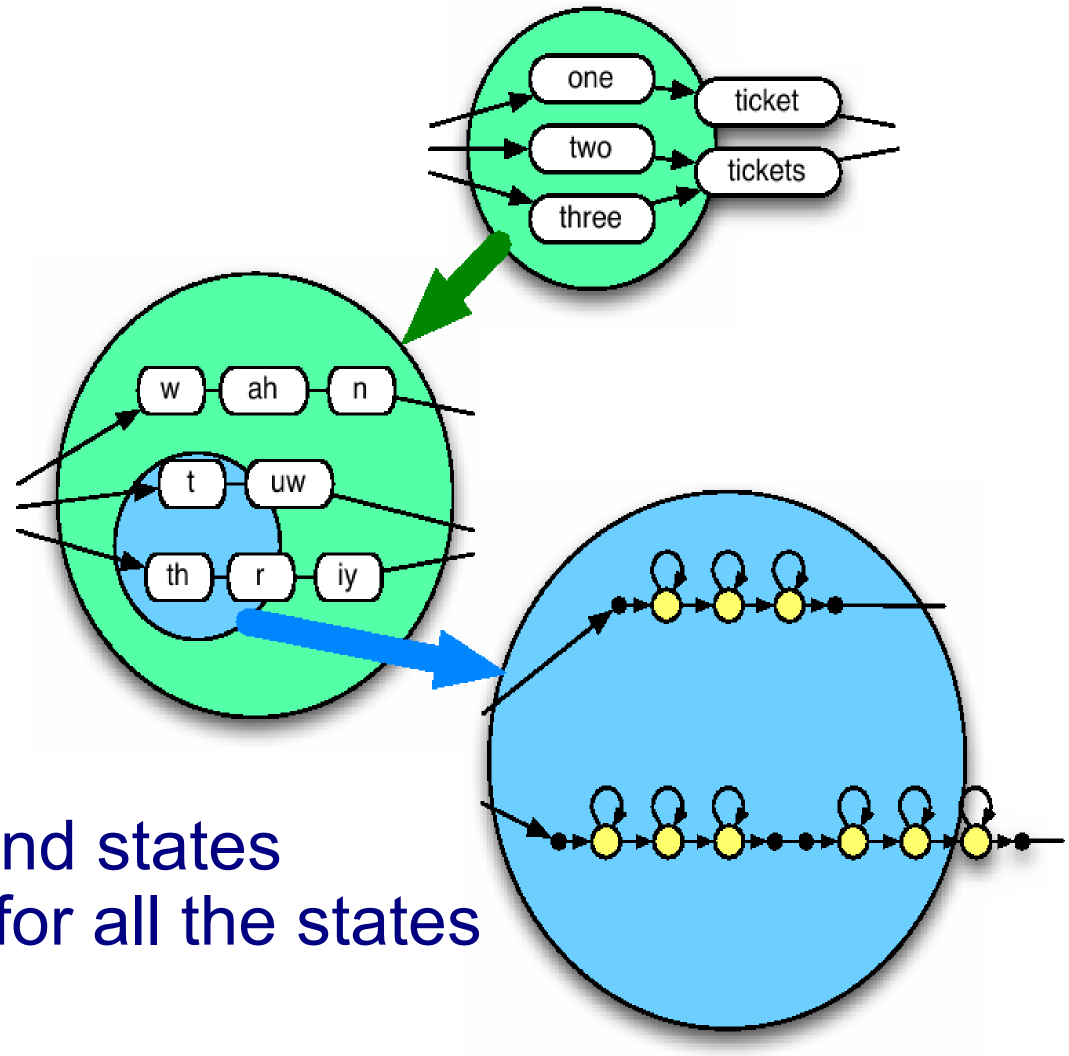
$$= \sum_{\text{all q}} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

- **This is not practical to compute. For N states, T observations, the number of state sequences is:**

$$O(2T * N^T)$$

Using induction in a search network





1. Map words into phonemes and states
2. Construct a search network for all the states

Picture by S.Renals

Forward algorithm

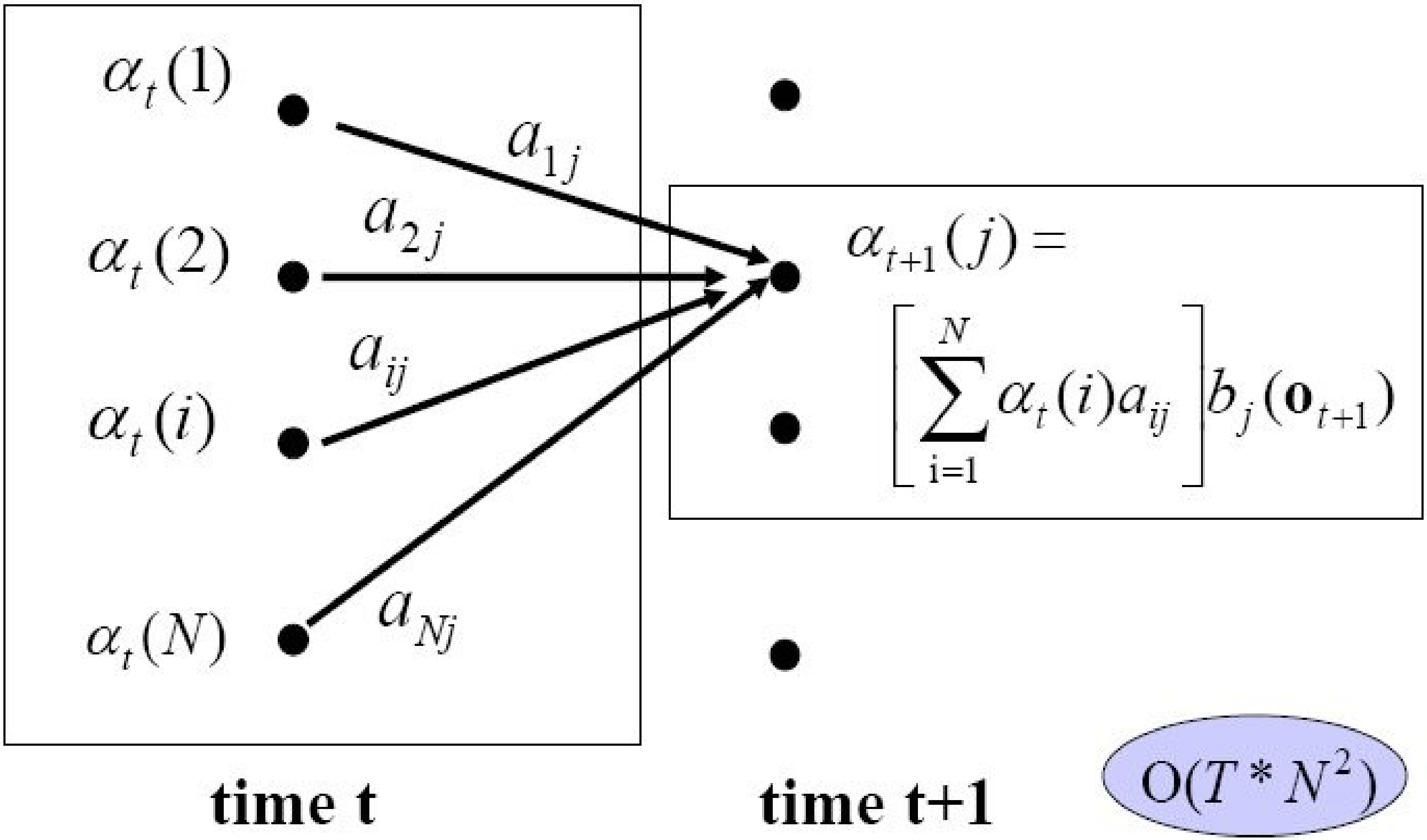
- **Definition:** $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2\ldots\mathbf{o}_t, q_t = i \mid \lambda)$

  (Probability of seeing observations o_1 to o_t and ending at state *i* given HMM λ)

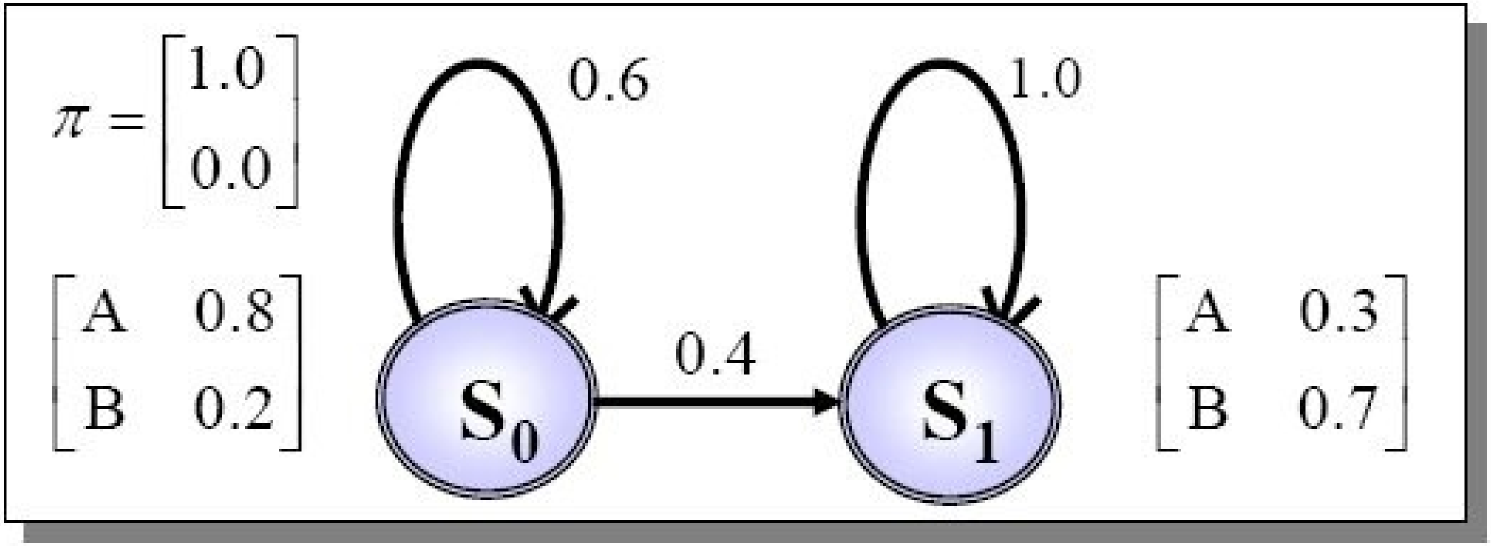
1. **Initialization** $\alpha_0(i) = \pi_i$
2. **Induction** $$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1})$$
3. **Termination** $$P(\mathbf{O} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Picture by B.Pellom

Forward step 2: Induction



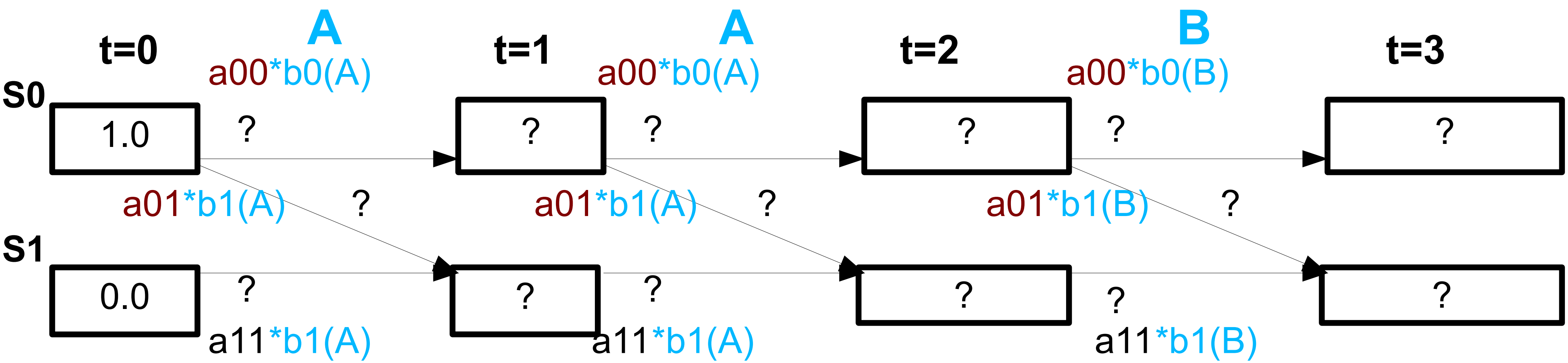
Forward example



- **Given the above HMM with discrete observations "A" and "B", what is the probability of generating the sequence "O = {A,A,B}"?**
- **In other words, find P(O={A,A,B} | λ)**

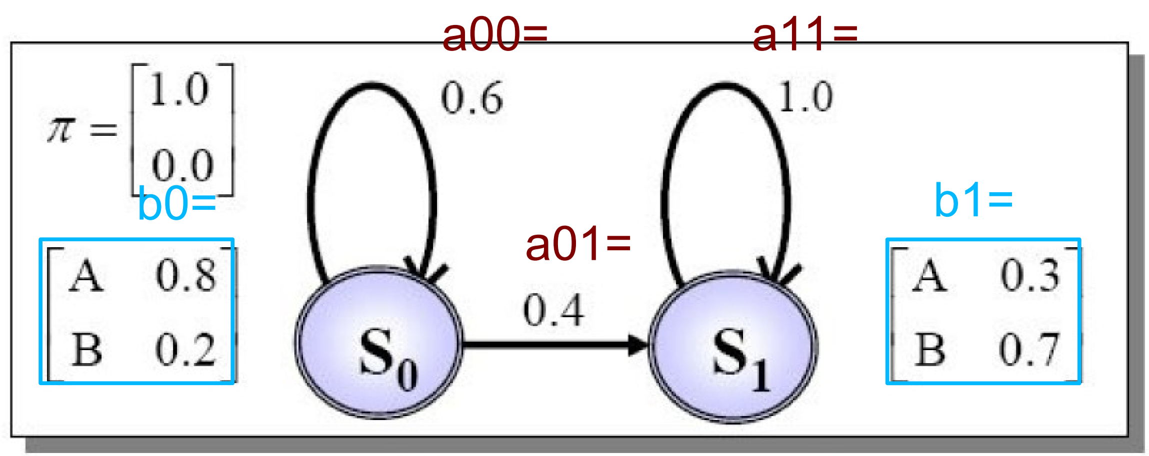
Picture by B.Pellom

Exercise1: Forward algorithm

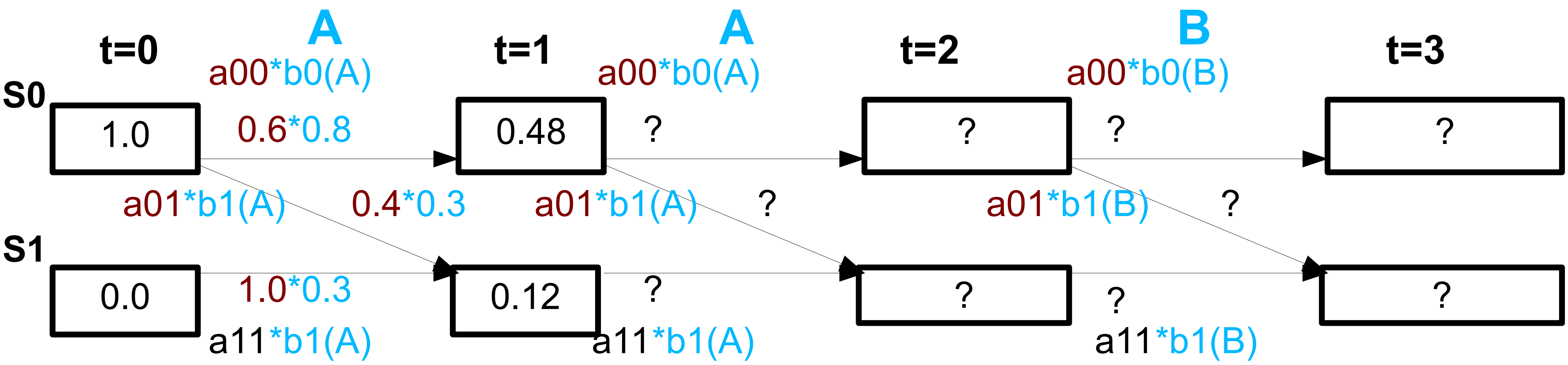


Answer:

?

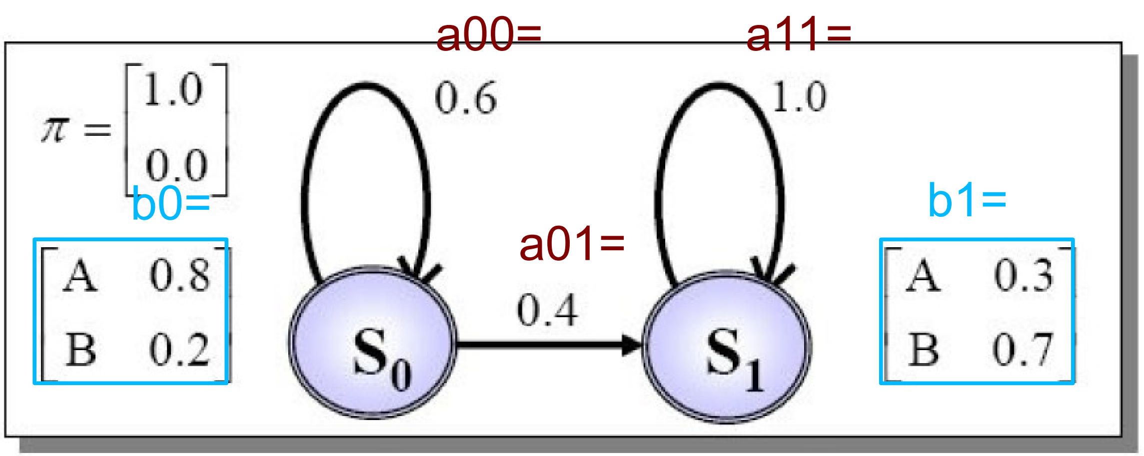


Exercise1: Forward algorithm



Answer:

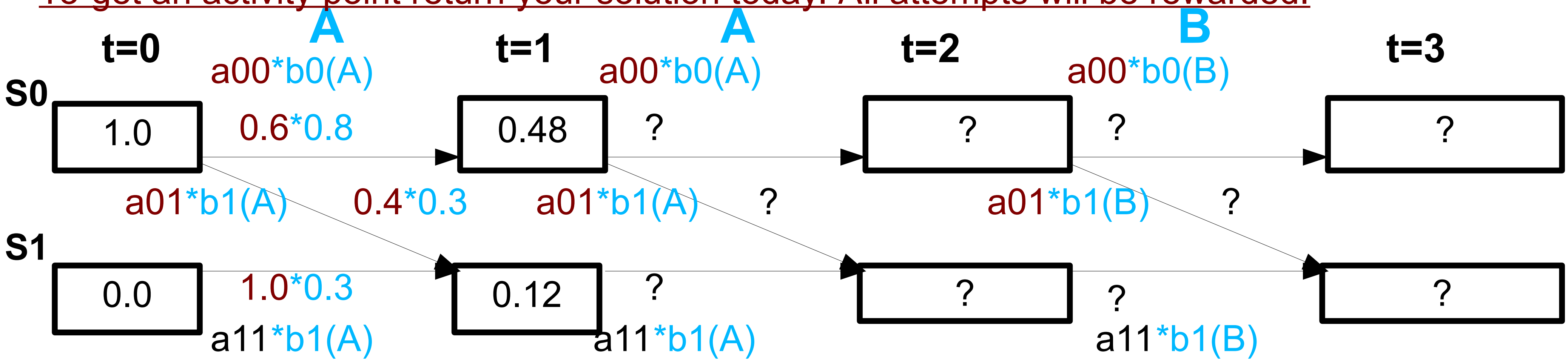
?



Exercise1: Forward algorithm

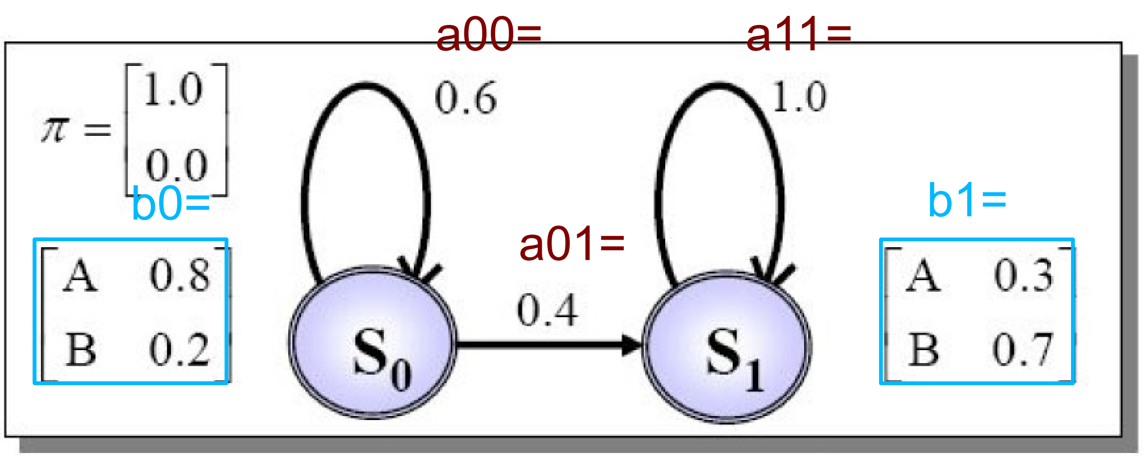
Now: Go to **MyCourses > Lectures > Lecture 3-4 exercise Forward** and open the return box

To get an activity point return your solution today. All attempts will be rewarded.



Answer:

?



2. Decoding

- **Given an observation sequence,**

$$\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$$

- **Find the single best sequence of states,**

$$q = \{q_1, q_2, \cdots, q_T\}$$

- **Which maximizes,**

$$P(\mathbf{O}, q \mid \lambda)$$

To be continued tomorrow...

Picture by B.Pellom

Viterbi algorithm

1. **Initialization** $\delta_1(i) = \pi_i b_i(\mathbf{o}_1) \quad \psi_1(i) = 0$

2. **Recursion**

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{o}_t)$$

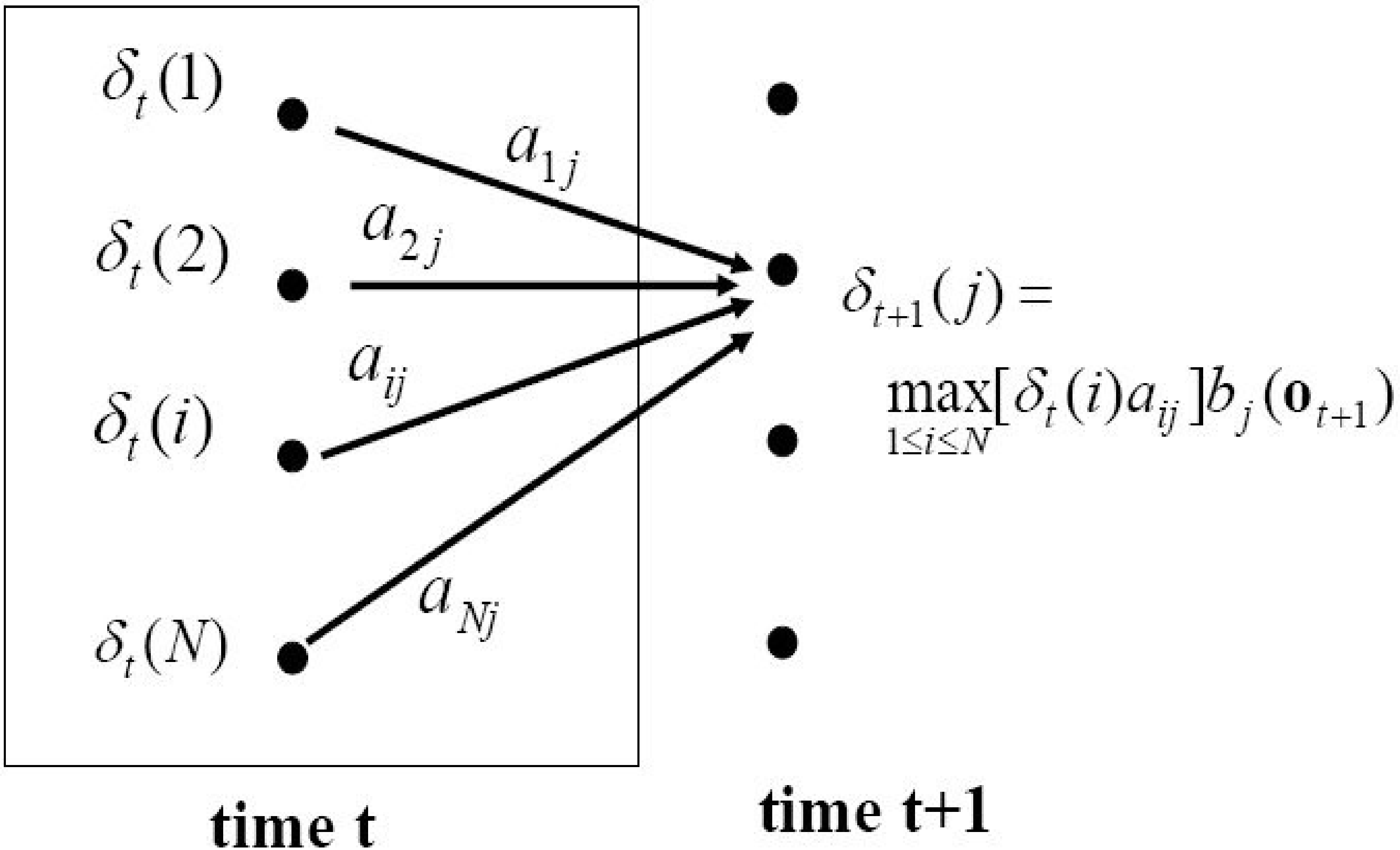
$$\psi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}]$$

3. **Termination** $P^* = \max_{1 \le i \le N} [\delta_T(i)] \quad q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$

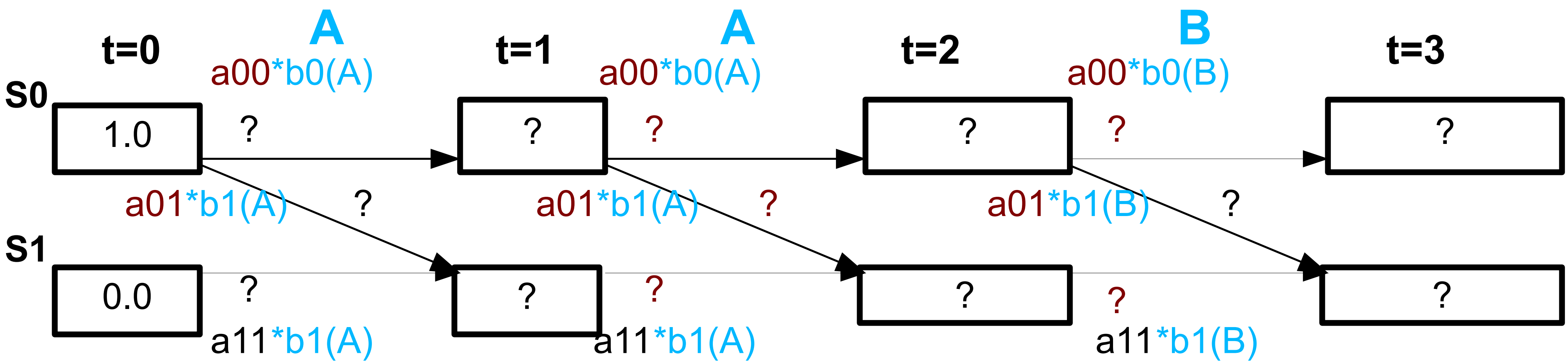
4. **Path Back trace** $q_t^* = \psi_{t+1}(q_{t+1}^*)$

Picture by B.Pellom

Viterbi step 2: Recursion

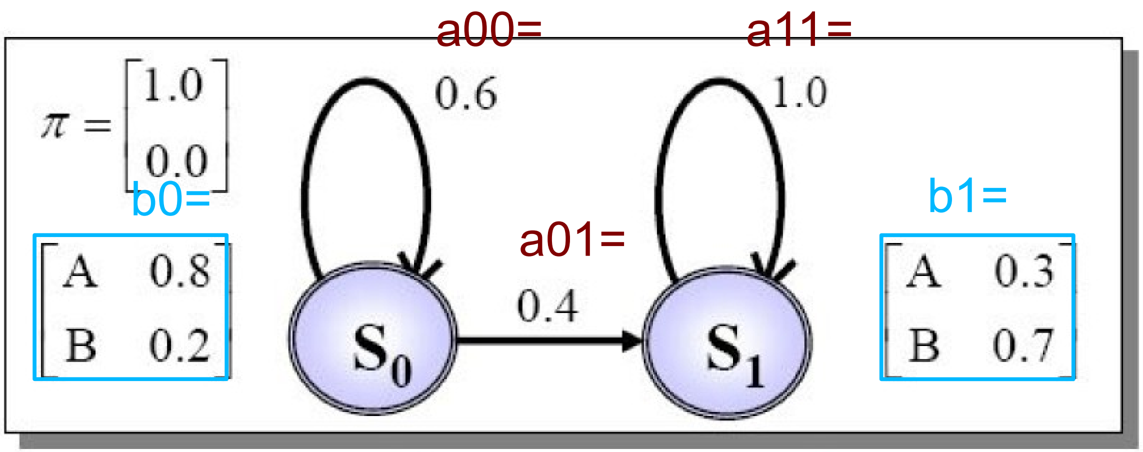


Exercise2: Viterbi search

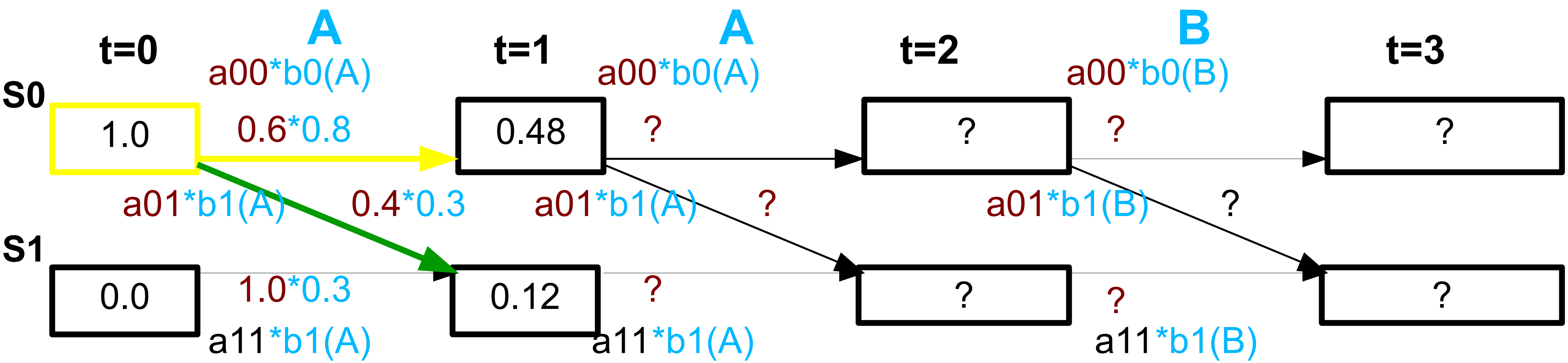


Answer:

?

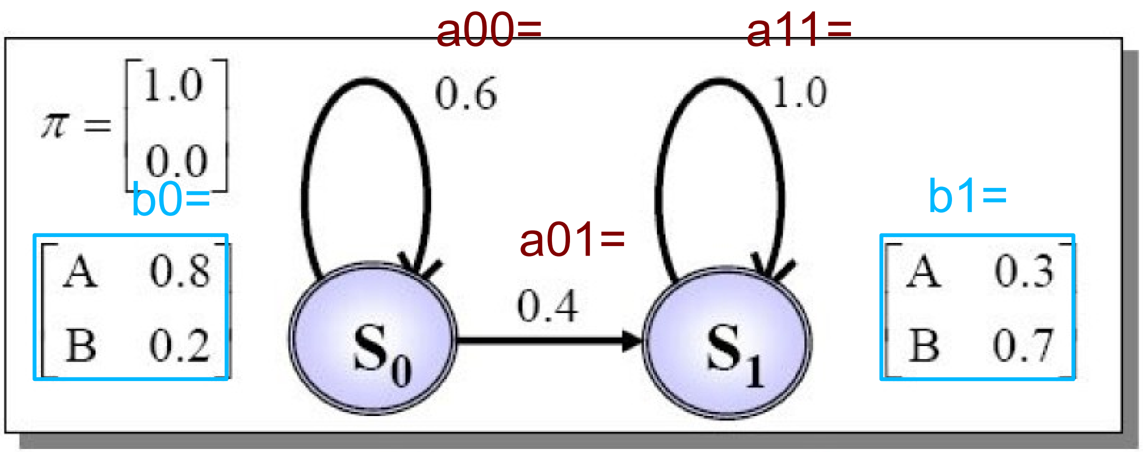


Exercise2: Viterbi search



Answer:

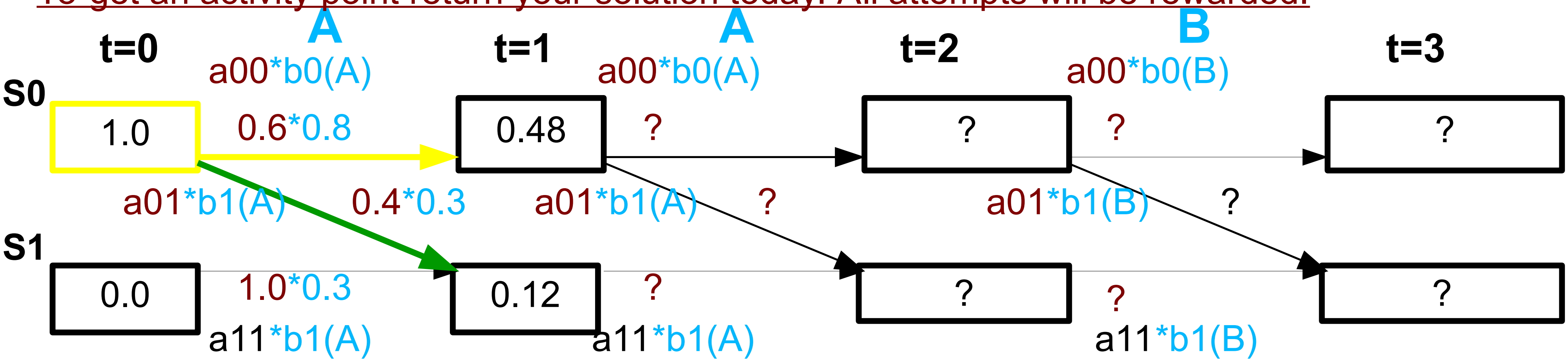
?



Exercise2: Viterbi search

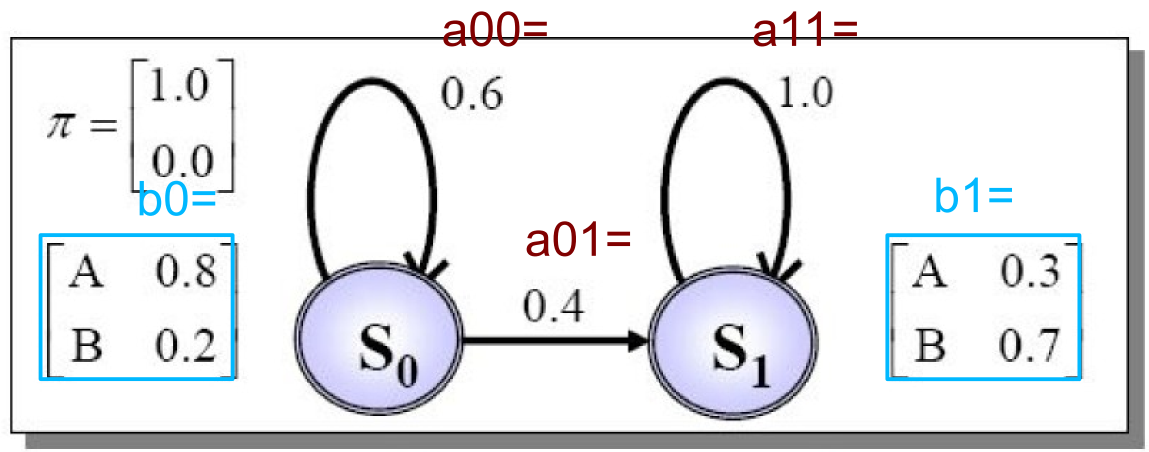
Now: Go to **MyCourses > Lectures > Lecture 3-4 exercise Viterbi** and open the return box

To get an activity point return your solution today. All attempts will be rewarded.



Answer:

?



3. Training

- **Forward-Backward** algorithm (a.k.a. Baum-Welch):
  - 1. Initialize the model parameters *(a,b)*
  - 2. Use the model and Forward (or Backward) algorithm to compute the probability matrix ***P(state=i,time=t |a,b)* for each sample**
  - 3. Use *P* to **enhance the model** parameters:
    - *a(ij)* : expected number of transitions from *i* to *j*
    - *b_i(o)* : expected pdf of features in state *i* (weighted by P)
  - 4. **Iterate** from 2.

Viterbi training

- Like Forward-Backward algorithm, but substitutes the **sum** operation by **max**
- Instead of summing probabilities over all HMM paths, **only use the best path** for each sample
- Like Viterbi decoding, but only the best **alignment** (segmentation) between the speech and text needed
- Simpler than F-B, but converges likewise to the (local) optimum
- *"Hard alignment"* in V, but *"soft alignment"* in F-B

Context-dependent HMM

- **Monophone** HMM = context-independent phoneme
  - three => th + r + iy
- **Triphone** HMM = context-dependent phonemes
  - three => sil-**th**+r + th-**r**+iy + r-**iy**+sil
- Difficult decisions needed in HMM design:
  - How many models, states and Gaussians?
  - Share models between some triphones?
  - Share states or Gaussians between models?

HMM assumptions

1. The HMM **topology** is usually fixed, e.g. left-to-right
2. The state **duration** is exponentially distributed
3. The **transition** between states is independent of time and state history: - It only depends on the current state
4. The **observations** are independent of time and each other: - They only depend on the current state

Content this week

1. Preprocessing and features, GMM
2. Phonemes
3. Hidden Markov model
4. **Home exercise 2**: Build a GMM-HMM system to recognize spoken words
5. Feedback
6. Project status
   - have a group meeting (as agreed with your tutor)

Home exercise 2

Build a GMM-HMM system to recognize words!

- Instructions and help given in Zulip by Ragheb Al-Ghezi and Yaroslav Getman
- Speech recognition by HTK toolkit
  - check *http://htk.eng.cam.ac.uk/docs/docs.shtml*
- This exercises is useful for most project works!
- To be returned by Wednesday **next week**

Feedback

Now: Go to **MyCourses > Lectures > Week 2 feedback** and fill in a feedback form to get an activity point.

Some of the feedback from the previous week:

+ interactive, which keeps me awake

+ many ways to get lecture points, could add even more

+ discussions in groups

- add references for self-study

- more focus on intuition behind formulas

- it was difficult to ask questions from physical to zoom

- possibility to attend remotely or have lectures recorded

Thanks for all the valuable feedback!

Project work receipt

1. Form a group (3 persons)
2. Get a topic (DL week 1)
3. Get reading material from M*ycourses* or your group tutor
4. 1st meeting: Specify the topic, start literature study (DL week 2)
5. **2nd meeting: Write a work plan (DL week 3)**
6. 3rd - 5th meetings: Perform analysis, experiments, and write a report
7. Book your presentation time for weeks 6 - 7 (DL week 4)
8. Prepare and keep your 15 min presentation
9. Return the report (DL week 7)

This week

Final project report

1. Abstract: (your working plan)
2. Introduction: (your literature review)
   - Remember to cite every article you read
3. Experiments: Describe what you did
4. Results: Describe the results you got
5. Conclusion: Your conclusion of the work
6. References: (list of articles that you read)