專題研究
WEEK2

Prof. Lin-shan Lee
TA. Cheng-Chieh Yeh

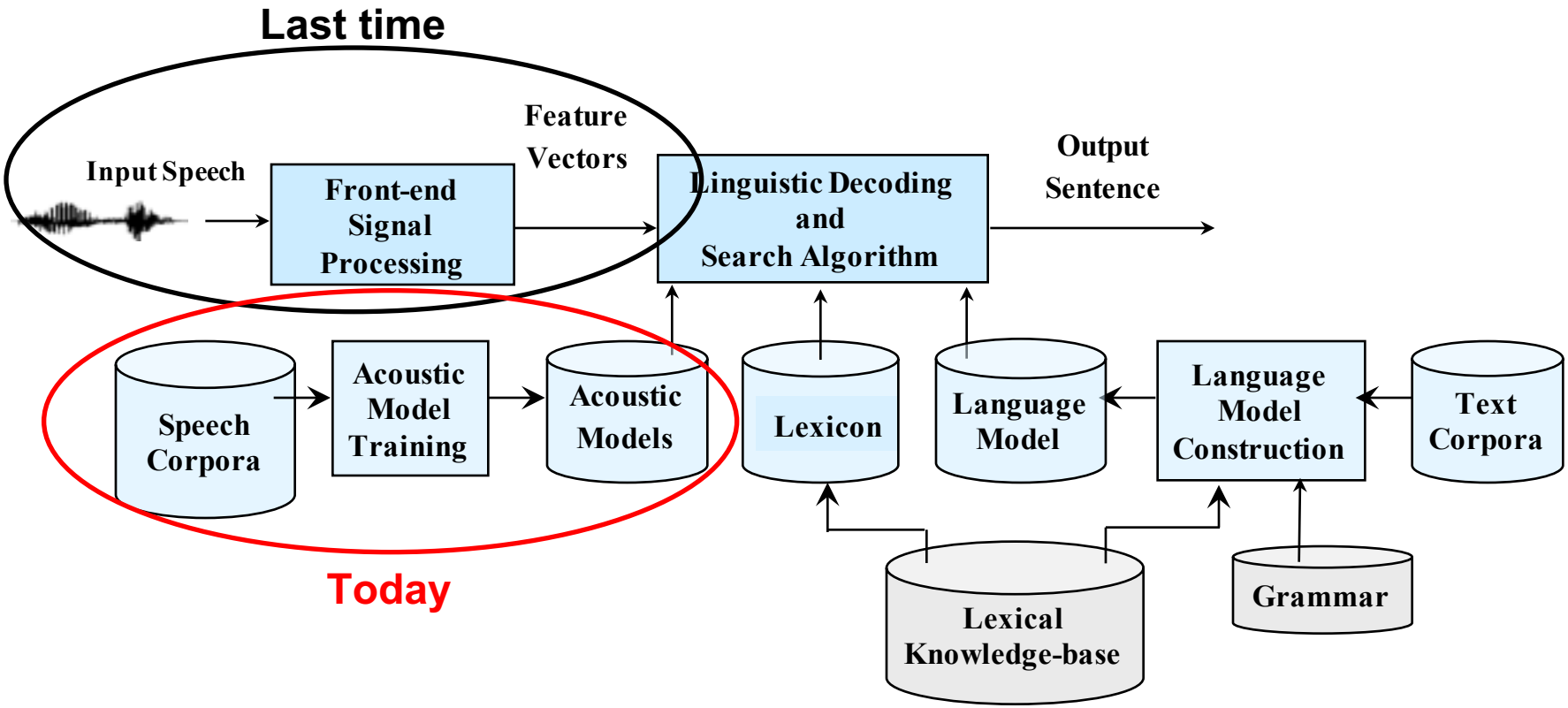
Outline

1. Recap
2. Apply HMM to Acoustic Modeling
3. Acoustic Model Training
4. Homework

3

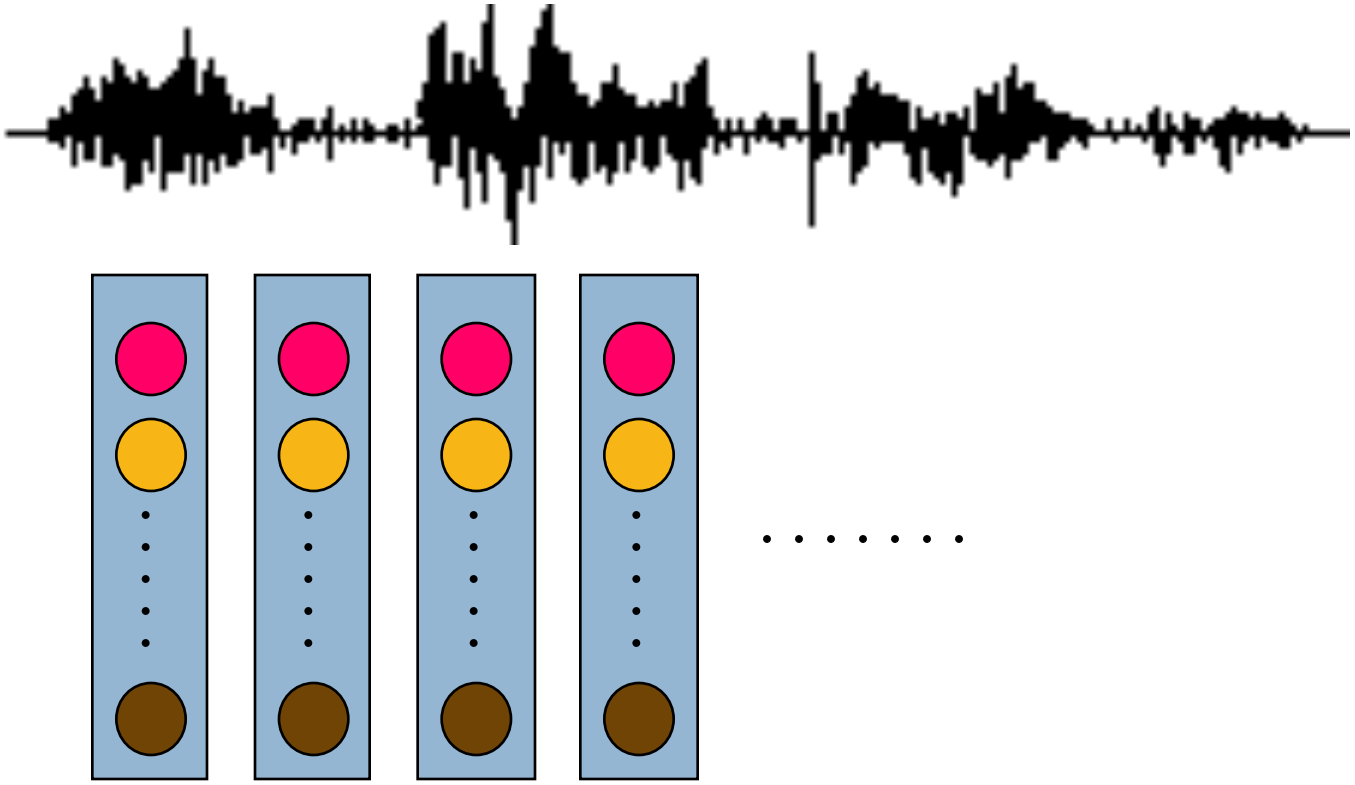
Recap

語音辨識系統


Last time
Input Speech
Front-end Signal Processing
Feature Vectors
Linguistic Decoding and Search Algorithm
Output Sentence
Speech Corpora
Acoustic Model Training
Acoustic Models
Lexicon
Language Model
Language Model Construction
Text Corpora
Today
Lexical Knowledge-base
Grammar

Feature Extraction

- Feature Extraction



How to do recognition?

- How to map speech O to a word sequence W ?

$$\begin{aligned}\hat{W} &= \arg\max_{W} P(W|O) \\ &= \arg\max_{W} \frac{P(O|W)P(W)}{P(O)} \\ &= \arg\max_{W} P(O|W)P(W)\end{aligned}$$

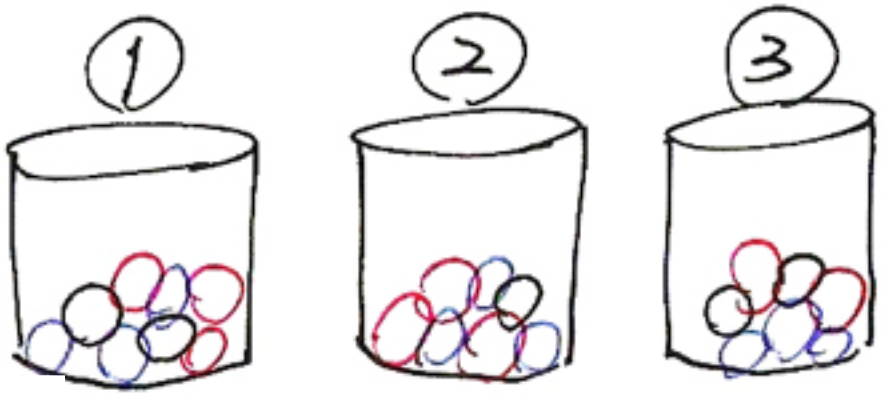
- P(O|W): acoustic model
- P(W): language model

Apply HMM to Acoustic Modeling

Hidden Markov Model

Simplified HMM Example

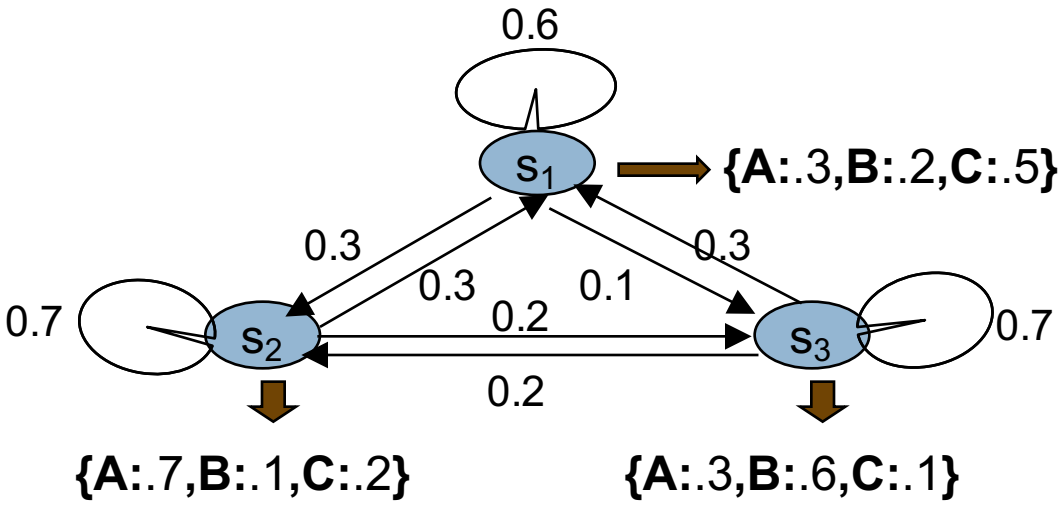
What problem can a HMM model?



RGBGGBBGRRR……

Hidden Markov Model

- **Elements of an HMM {S,A,B,π}**
  - **S** is a set of N states
  - **A** is the $N \times N$ matrix of state transition probabilities
  - **B** is a set of N probability functions, each describing the observation probability with respect to a state
  - **π** is the vector of initial state probabilities

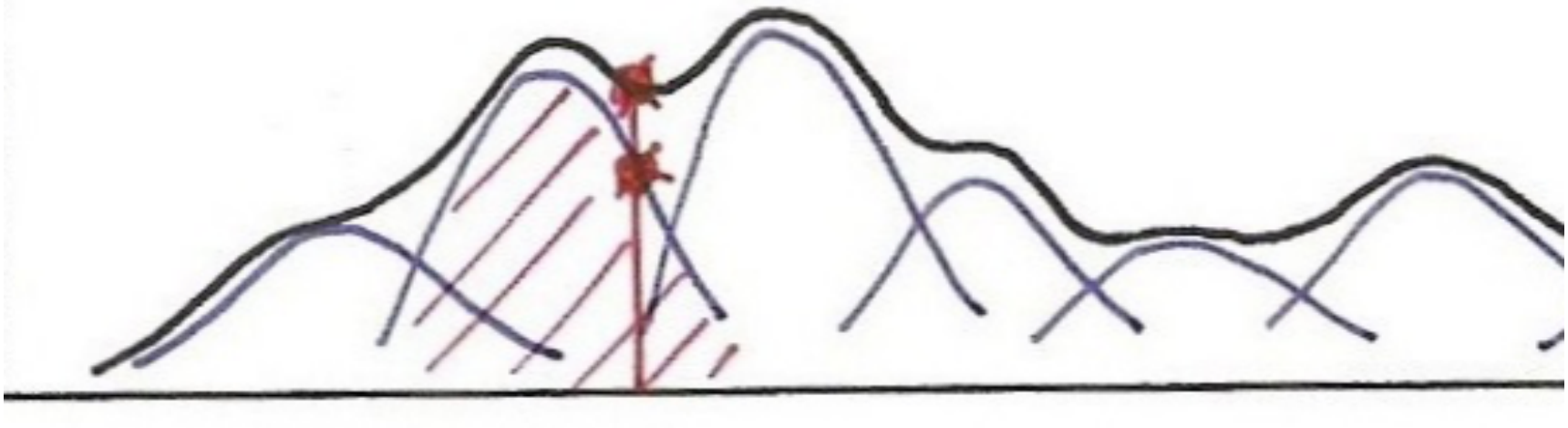


$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.4 & 0.5 & 0.1 \end{bmatrix}$$

Gaussian Mixture Model (GMM)

- What if observation is continuous? (Ex. MFCC feature vectors)
- Need a continuous probability density function to model observations, which is often assumed to be a **GMM**.



HMM: Three Basic Problems

- **Given an observation sequence** $O=(o_1,o_2,\ldots..,o_T)$ **and an HMM** $\lambda=(A,B,\pi)$
  - Problem *1* :

    How to *efficiently* compute $P(\mathbf{O}|\ \lambda)$?

    ⇨ *Evaluation problem*
  - Problem *2* :

    How to choose an optimal state sequence $\mathbf{q}=(q_1,q_2,\ldots\ldots, q_T)$?

    ⇨ *Decoding Problem*
  - Problem *3* :

    Given some observations O for the HMM λ , how to adjust the model parameter $\lambda=(A,B,\pi)$ to maximize $P(\mathbf{O}|\ \lambda)$?

    ⇨ *Learning /Training Problem*

HMM: Three Basic Problems

DP and variations

- **Given an observation sequence** $O=(o_1,o_2,\ldots..,o_T)$ **and an HMM** $\lambda=(A,B,\pi)$
  - Problem *1* :

    How to *efficiently* compute $P(\mathbf{O}|\ \lambda)$?

    ⇨ *Evaluation problem*
  - Problem *2* :

    How to choose an optimal state sequence $\mathbf{q}=(q_1,q_2,\ldots\ldots, q_T)$?

    ⇨ *Decoding Problem*
  - Problem *3* :

    Given some observations O for the HMM λ , how to adjust the model parameter $\lambda=(A,B,\pi)$ to maximize $P(\mathbf{O}|\ \lambda)$?

    ⇨ *Learning /Training Problem*

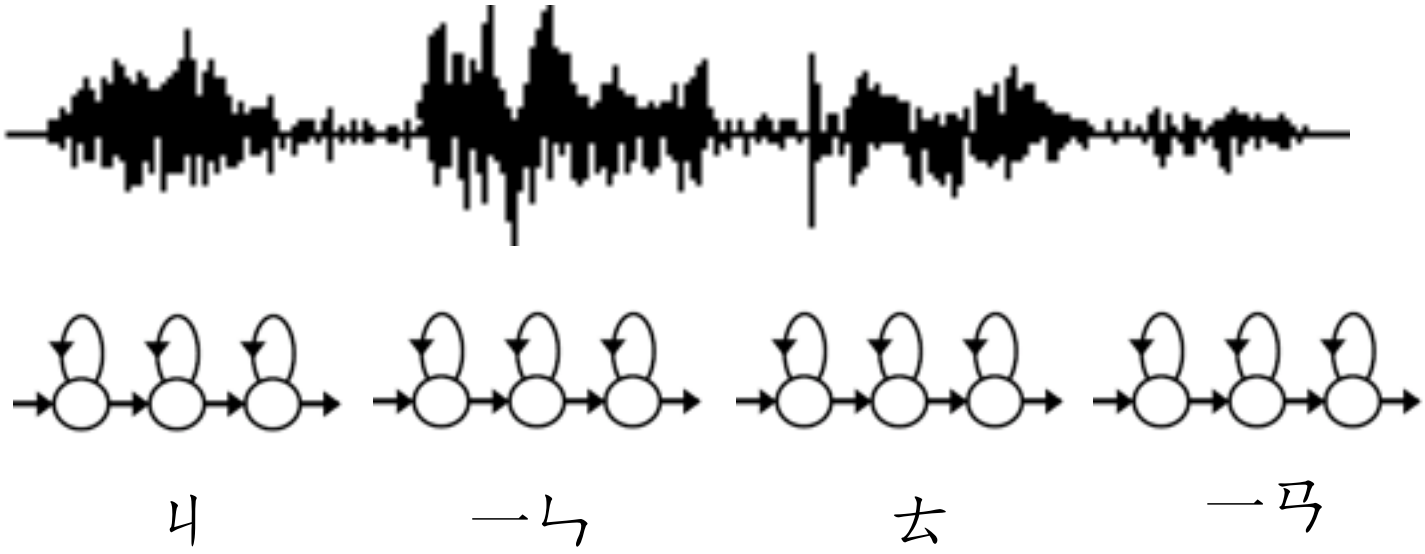
EM and variations

HMM Training

- Concept: EM-Training
- Ref: 老師上課投影片4.0
- 1. 粗調：Segmental K-means
- 2. 細調：Baum-Welch Algorithm

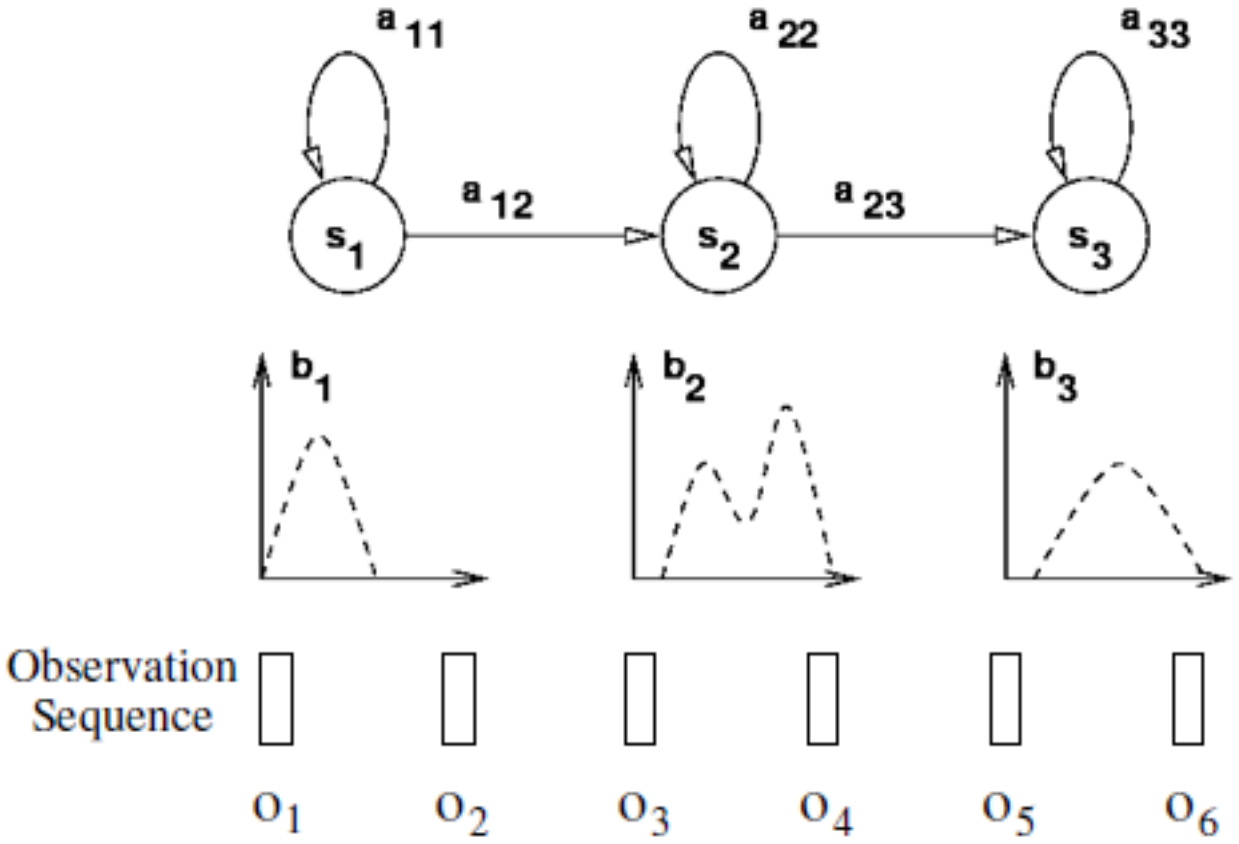
Acoustic Model P(O|W)

- How to compute P(O|W) ?



Acoustic Model P(O|W)

- Model of a phone

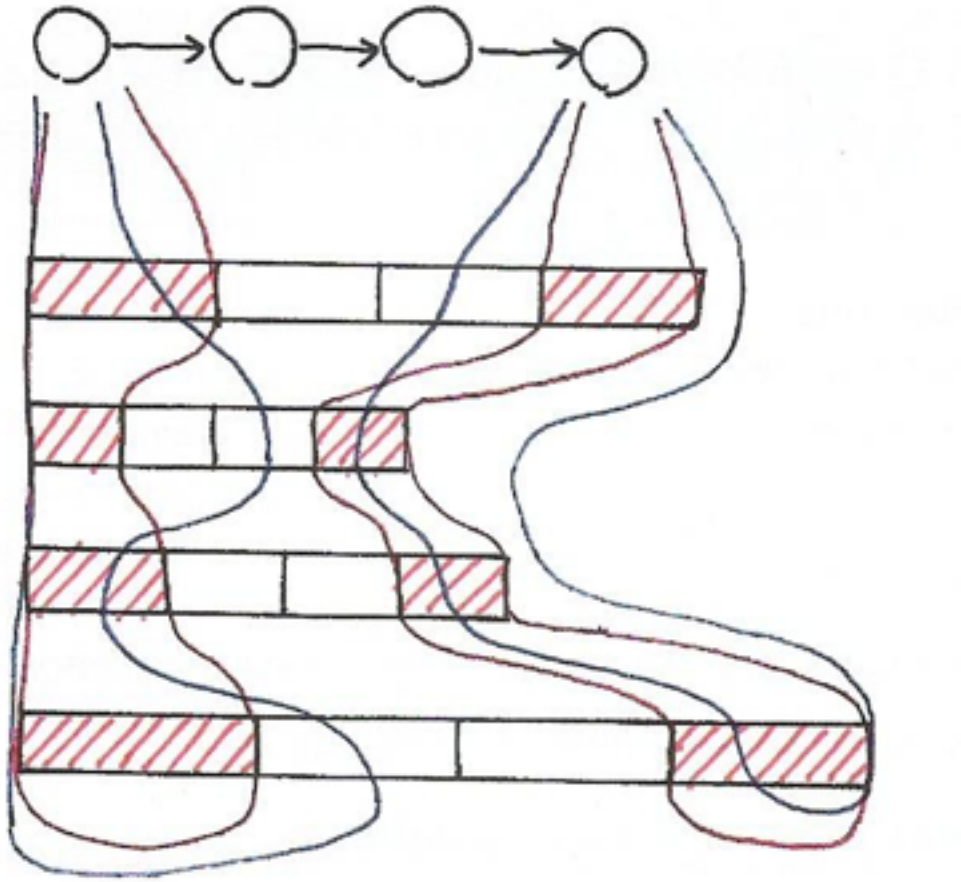


Markov Model

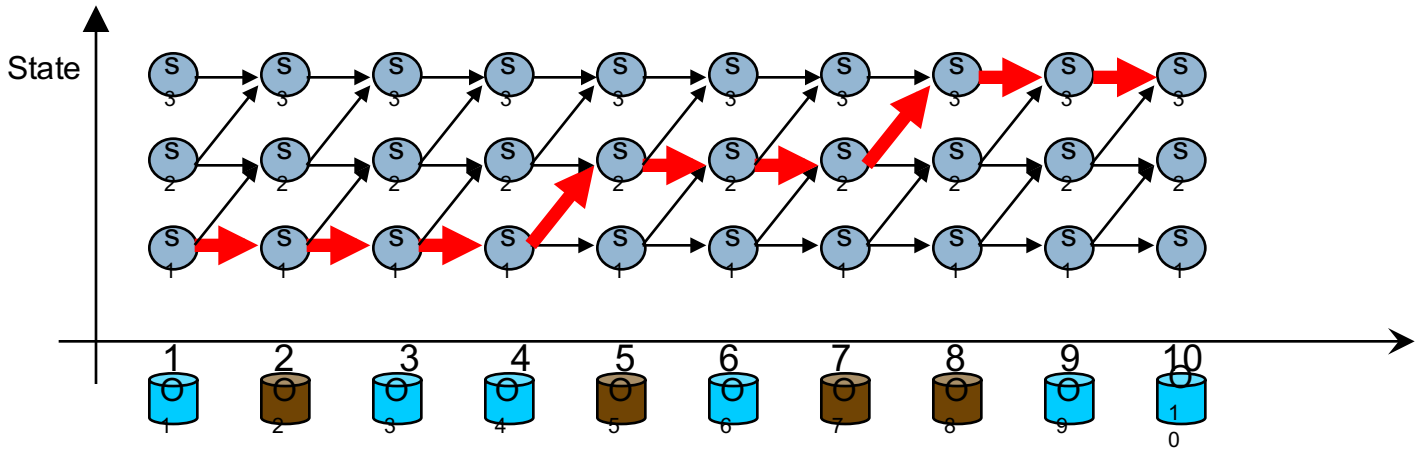
Gaussian Mixture Model

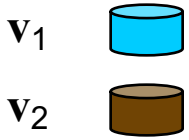
Segmental K-means

假設今天有四個人都發出‘ㄅ’這個音，但每個人念的長短不一



An example of Modifying HMM





$b_1(\mathbf{v}_1)=3/4, b_1(\mathbf{v}_2)=1/4$

$b_2(\mathbf{v}_1)=1/3, b_2(\mathbf{v}_2)=2/3$

$b_3(\mathbf{v}_1)=2/3, b_3(\mathbf{v}_2)=1/3$

Monophone vs. triphone

- Monophone

  consider only **one phone imformation** per model

- Triphone

  taking into consideration both left and right neighboring phones

Ex. Monophone

ㄧ、ㄨ、ㄩ

Triphone

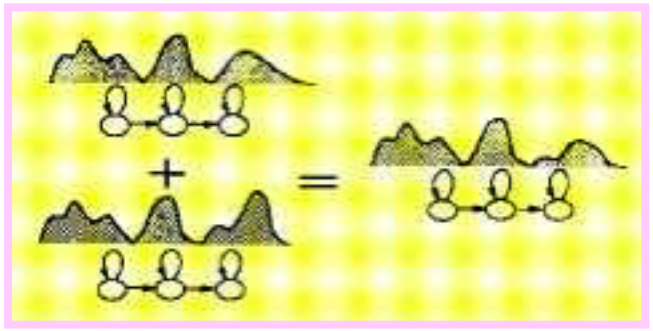
ㄐ-ㄧ+ㄢ、ㄐ-ㄧ+ㄤ

different models between these two 'ㄧ' !!!

Triphone

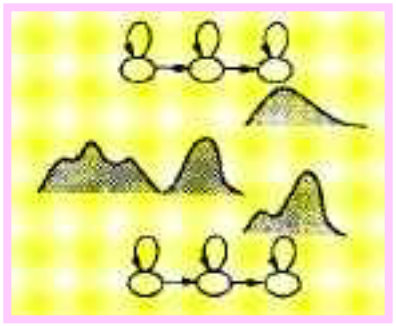
- A phone model taking into consideration both left and right neighboring phones $(60)^3 \rightarrow 216{,}000$
- We need to share rare observations' parameters with others.

- **Sharing at Model Level**



Generalized Triphone

- **Sharing at State Level**

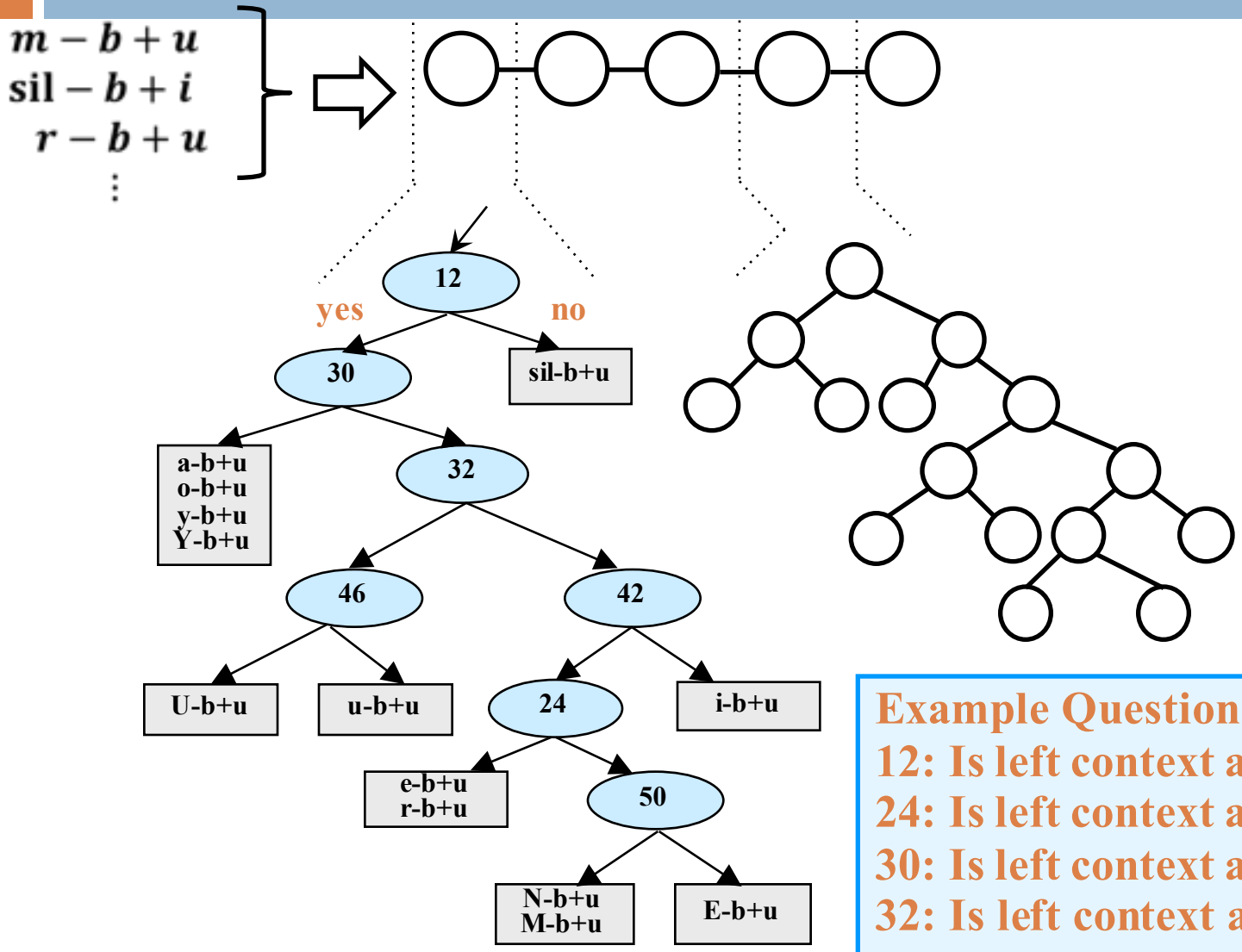


Shared Distribution Model (SDM)

Actually, we use DecisionTree-based method now.

Training Tri-phone Models with Decision Trees

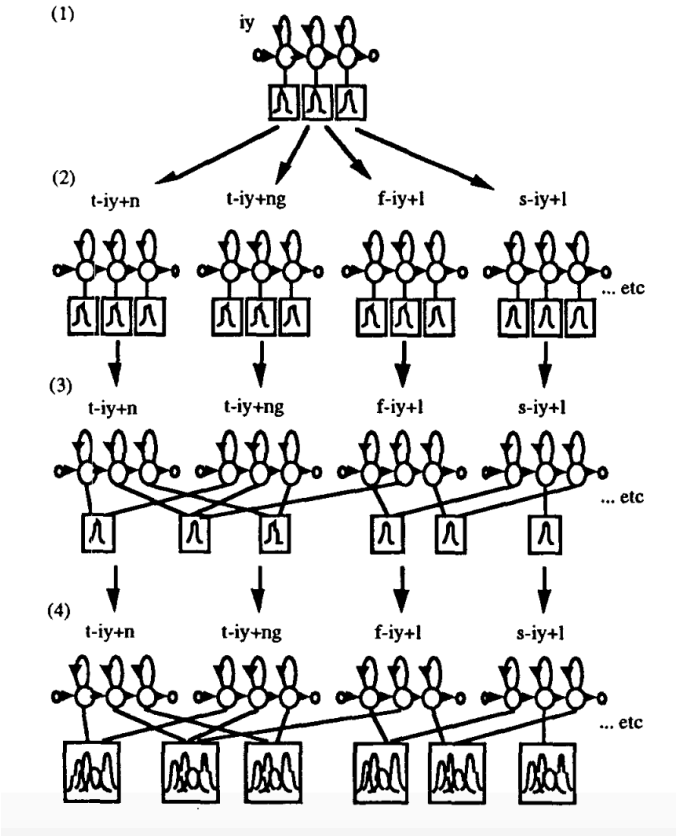
- **An Example: "(_ –) b (+_)"**



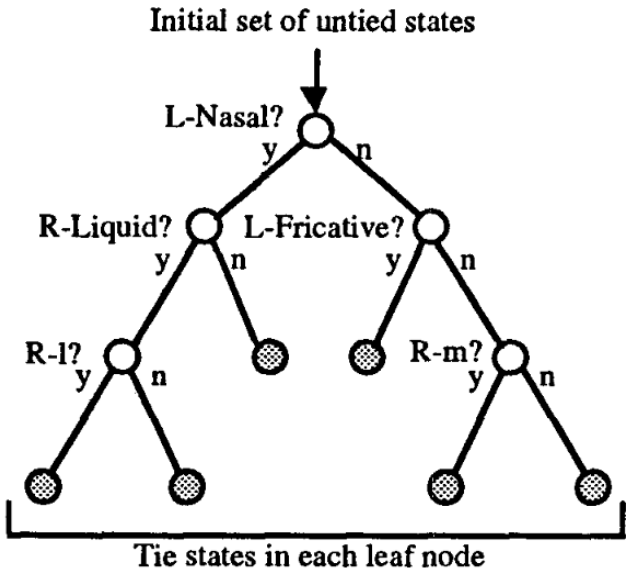
Example Questions:
12: Is left context a vowel?
24: Is left context a back-vowel?
30: Is left context a low-vowel?
32: Is left context a rounded-vowel?

How Kaldi Use Decision Trees to train Triphone

Cluster-based



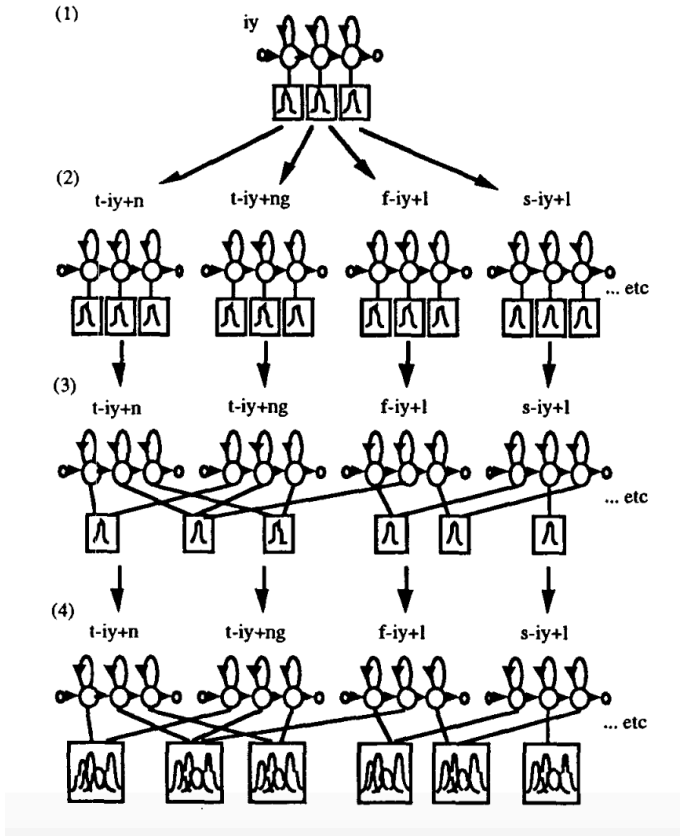
Tree-based Cluster



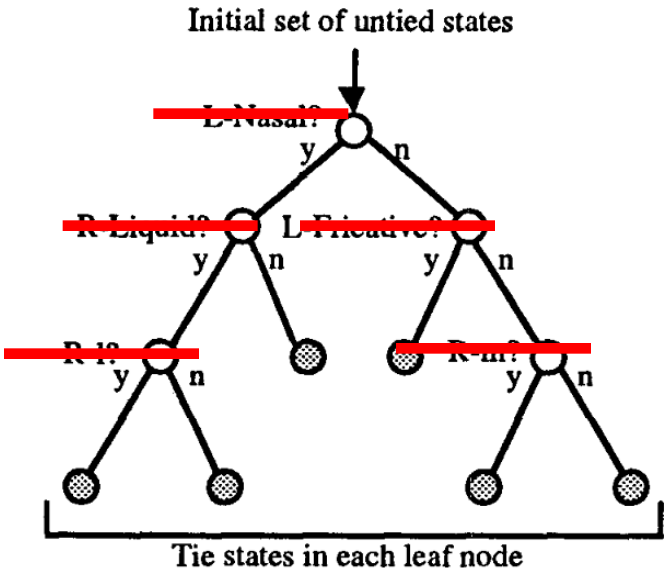
Ref: http://www.aclweb.org/anthology/H94-1062

How Kaldi Use Decision Trees to train Triphone

Cluster-based



Tree-based Cluster



No expert‘s rule questions

Fully data-driven split

Ref: http://www.aclweb.org/anthology/H94-1062

Acoustic Model Training

03.mono.train.sh

05.tree.build.sh

06.tri.train.sh

Acoustic Model Training Steps

- Step1: Train Monophone
- Step2: Build Decision Trees
- Step3: Train Triphone

Train Monophone

- Get features (last time)
- Train monophone model:
  - a. gmm-init-mono — initial monophone model
  - b. compile-train-graphs — get train graph
  - c. align-equal-compiled — model -> decode&align

    (use **gmm-align-compiled** instead when looping)
  - d. gmm-acc-stats-ali — EM training: E step
  - e. gmm-est — EM training: M step
  - f. go to step c

Train Triphone

- Train triphone model
  - a. gmm-init-model — Initialize GMM (from decision tree)
  - b. gmm-mixup — Gaussian merging (increase #gaussian)
  - c. convert-ali — Convert alignments(model <-> decisoin tree)
  - d. compile-train-graphs — get train graph
  - e. gmm-align-compiled — model -> decode&align
  - f. gmm-acc-stats-ali — EM training: E step
  - g. gmm-est — EM training: M step
  - h. Goto step e. — Train several times

How to get Kaldi usage?

source setup.sh

align-equal-compiled --help

```
align-equal-compiled

Write an equally spaced alignment (for getting training started)Usage:  align-equal-compiled <graphs
-rspecifier> <features-rspecifier> <alignments-wspecifier>
e.g.:
 align-equal-compiled 1.mdl 1.fsts scp:train.scp ark:equal.ali

Options:
  --binary                      : Write output in binary mode (bool, default = true)

Standard options:
  --config                      : Configuration file with options (string, default = "")
  --help                        : Print out usage message (bool, default = false)
  --print-args                  : Print the command line arguments (to stderr) (bool, default = true)
  --verbose                     : Verbose level (higher->more logging) (int, default = 0)
```

align-equal-compiled

Write an equally spaced alignment (for getting training started)

Usage: align-equal-compiled <graphs-rspecifier> <features-rspecifier> <alignments-wspecifier>

e.g.:

align-equal-compiled 1.mdl 1.fsts scp:train.scp ark:equal.ali

gmm-align-compiled $scale_opts --beam=$beam --retry-beam=$[$beam*4] <hmm-model*> ark:$dir/train.graph ark,s,cs:$feat ark:<alignment*>

For first iteration(in monophone) beamwidth = 6, others = 10;

Only realign at

$realign_iters="1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 23 26 29 32 35 38”

$realign_iters=“10 20 30”

gmm-acc-stats-ali

Accumulate stats for GMM training.(E step)

Usage: gmm-acc-stats-ali [options] <model-in> <feature-rspecifier> <alignments-rspecifier> <stats-out>

e.g.:

gmm-acc-stats-ali 1.mdl scp:train.scp ark:1.ali 1.acc

gmm-acc-stats-ali --binary=false <hmm-model*> ark,s,cs:$feat ark,s,cs:<alignment*> <stats>

gmm-est

Do Maximum Likelihood re-estimation of GMM-based acoustic model

Usage: gmm-est [options] <model-in> <stats-in> <model-out>

e.g.: gmm-est 1.mdl 1.acc 2.mdl

gmm-est --binary=false --write-occs=<*.occs> --mix-up=$numgauss <hmm-model-in> <stats> <hmm-model-out>

--write-occs : File to write pdf occupation counts to.

$numgauss increases every time.

Homework

03.mono.train.sh

05.tree.build.sh

06.tri.train.sh

閱讀：數位語音處理概論ch4, ch5 (opt.)

ToDo

- Step0. Make sure last time‘s results exist.
  - ex. feat/train.39.cmvn.ark ...
- Step1. Execute the following commands.
  - script/03.mono.train.sh | tee log/03.mono.train.log
  - script/05.tree.build.sh | tee log/05.tree.build.log
  - script/06.tri.train.sh | tee log/06.tri.train.log
- Step2. finish code in ToDo and redo Step1.
  - script/03.mono.train.sh
  - script/06.tri.train.sh
- Step3. Observe the output and results.
- Step4.(Opt.) tune #gaussian and #iteration.

Hint (extremely important!!)

- 03.mono.train.sh
  - Use the variables already defined.

```
numiters=40                                    # Number of iterations of training
maxiterinc=30                                  # Last iter to increase #Gauss on.
numgauss=300                                   # Initial num-Gauss (must be more than #states=3*phones)
totgauss=1000                                  # Target #Gaussians.
incgauss=$[($totgauss-$numgauss)/$maxiterinc] # per-iter increment for #Gauss
realign_iters="1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 23 26 29 32 35 38";
scale_opts="--transition-scale=1.0 --acoustic-scale=0.1 --self-loop-scale=0.1"
```

  - Use these formula:

```
x=`printf "%02g" $iter`
y=`printf "%02g" $[$iter+1]`

if echo $realign_iters | grep -w $iter >/dev/null ; then
```

  - Pipe for error
    - compute-mfcc-feats … 2> $log

Kaldi HMM Resource

- http://kaldi-asr.org/doc/hmm.html
- http://blog.csdn.net/u010731824/article/details/69668765#transitionmodel
- http://blog.csdn.net/u010731824/article/details/70161677

Questions?

- Try drawing the workflow of training.