fit@hcmus

# Mô hình ngữ âm

Nguyễn Đức Hoàng Hạ
2022

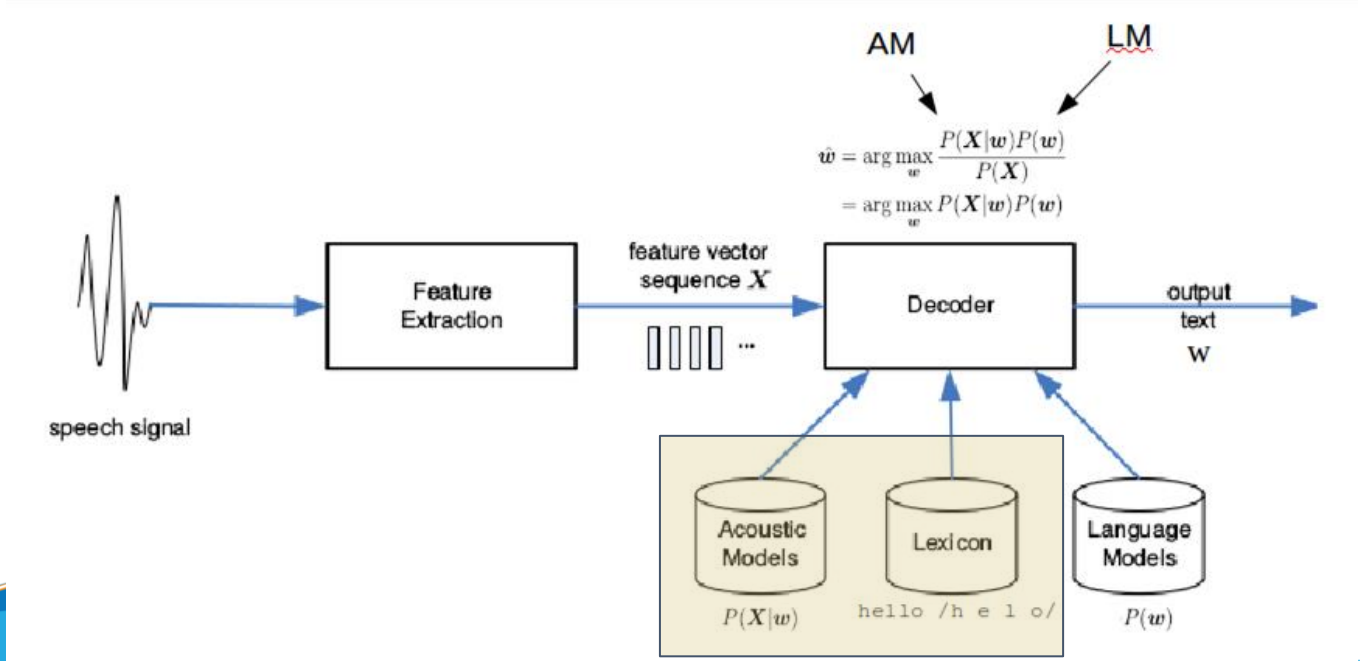
# Nội dung

- Bài toán nhận dạng tiếng nói
- Giới thiệu mô hình ngữ âm GMM-HMM
- Xây dựng mô hình GMM-HMM trên thư viện Kaldi
- Công cụ hỗ trợ tạo từ điển phiên âm

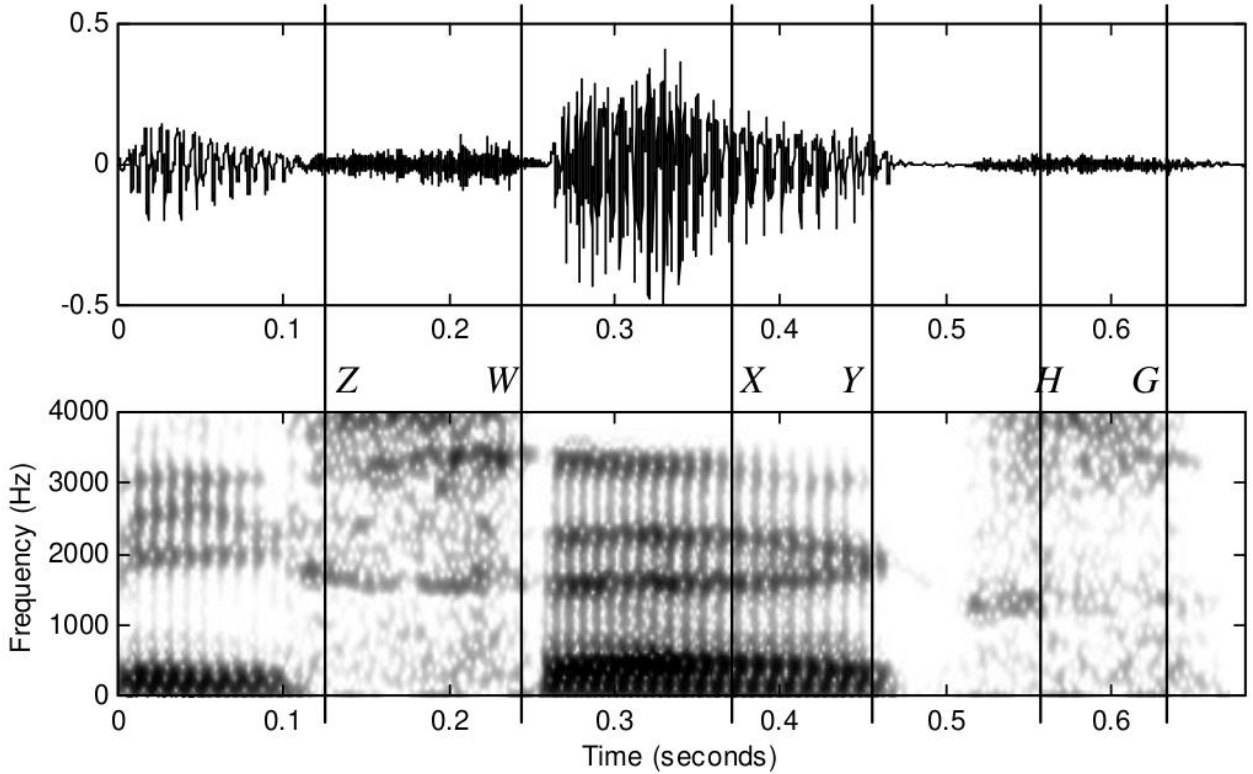
# Ôn tập toán xác suất

www.menti.com

# Bài toán nhận dạng tiếng nói (Automatic Speech Recognition)



# Đặc trưng của tiếng nói



# Đặc trưng của tiếng nói - Mel-Scale filter bank

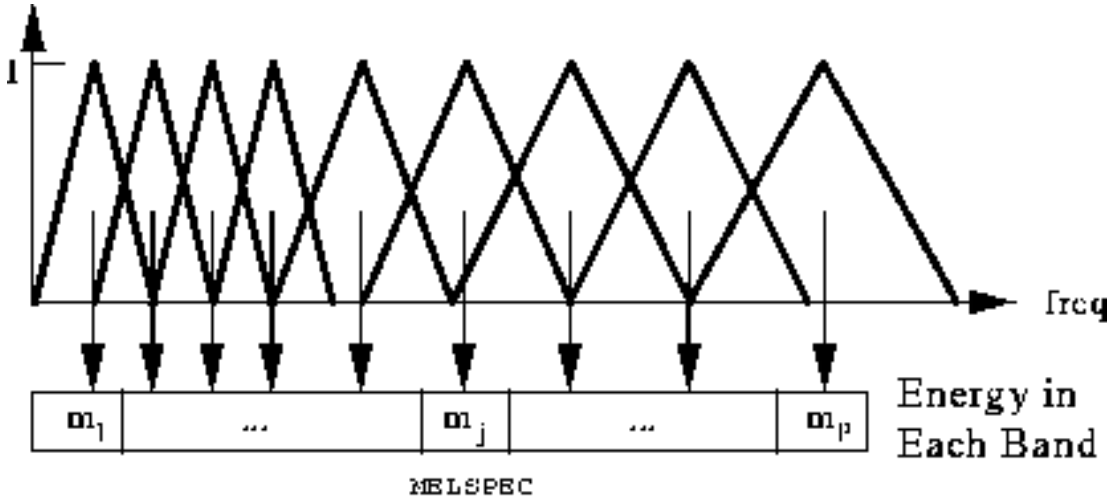
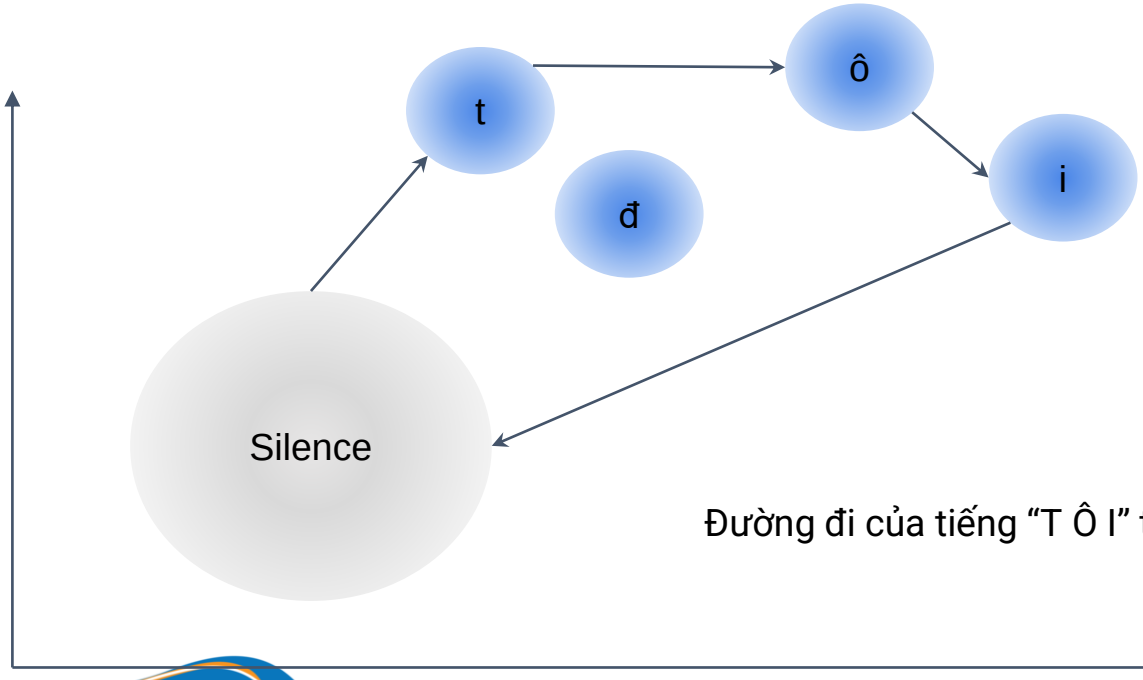



Fig. 5.3 Mel-Scale Filter Bank

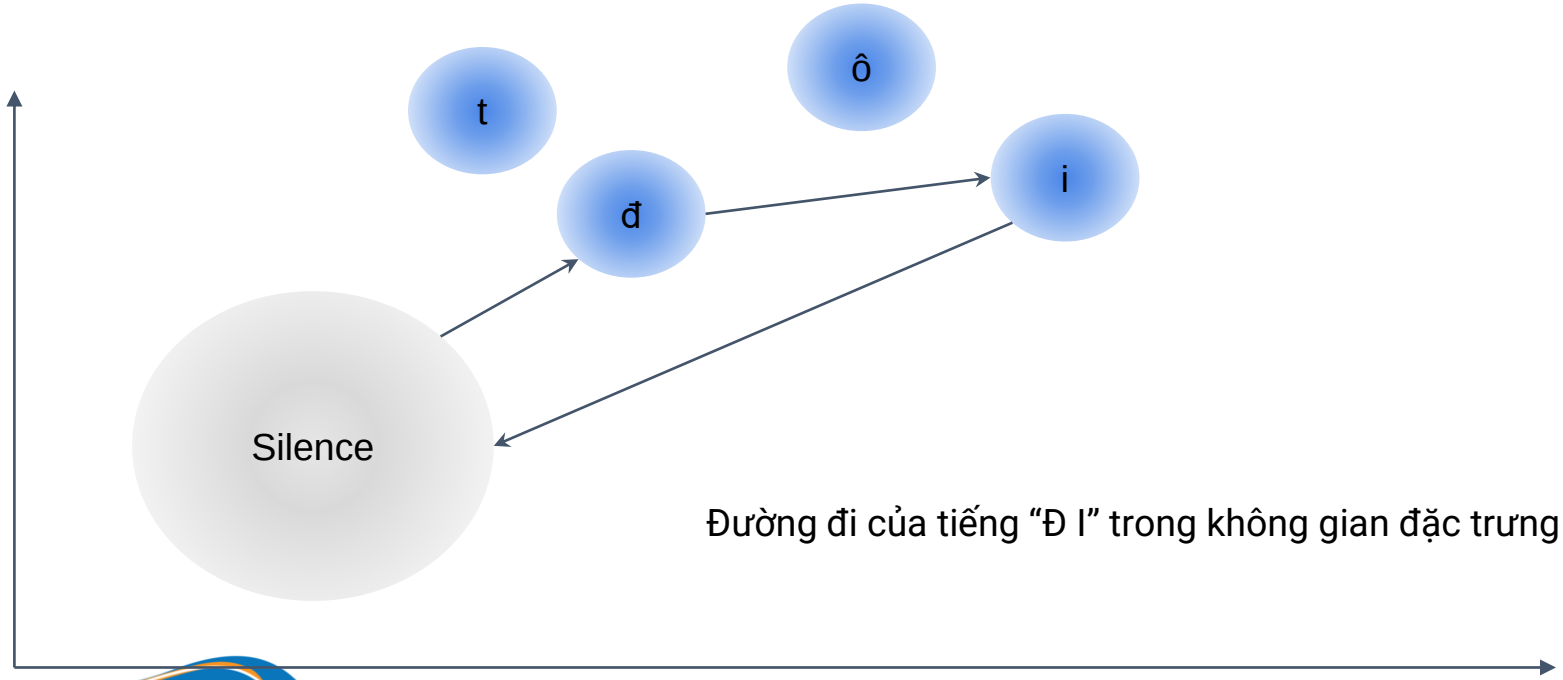
# Không gian đặc trưng của tiếng nói




Đường đi của tiếng “T Ô I” trong không gian đặc trưng

(Không gian đặc trưng N chiều)

# Không gian đặc trưng của tiếng nói



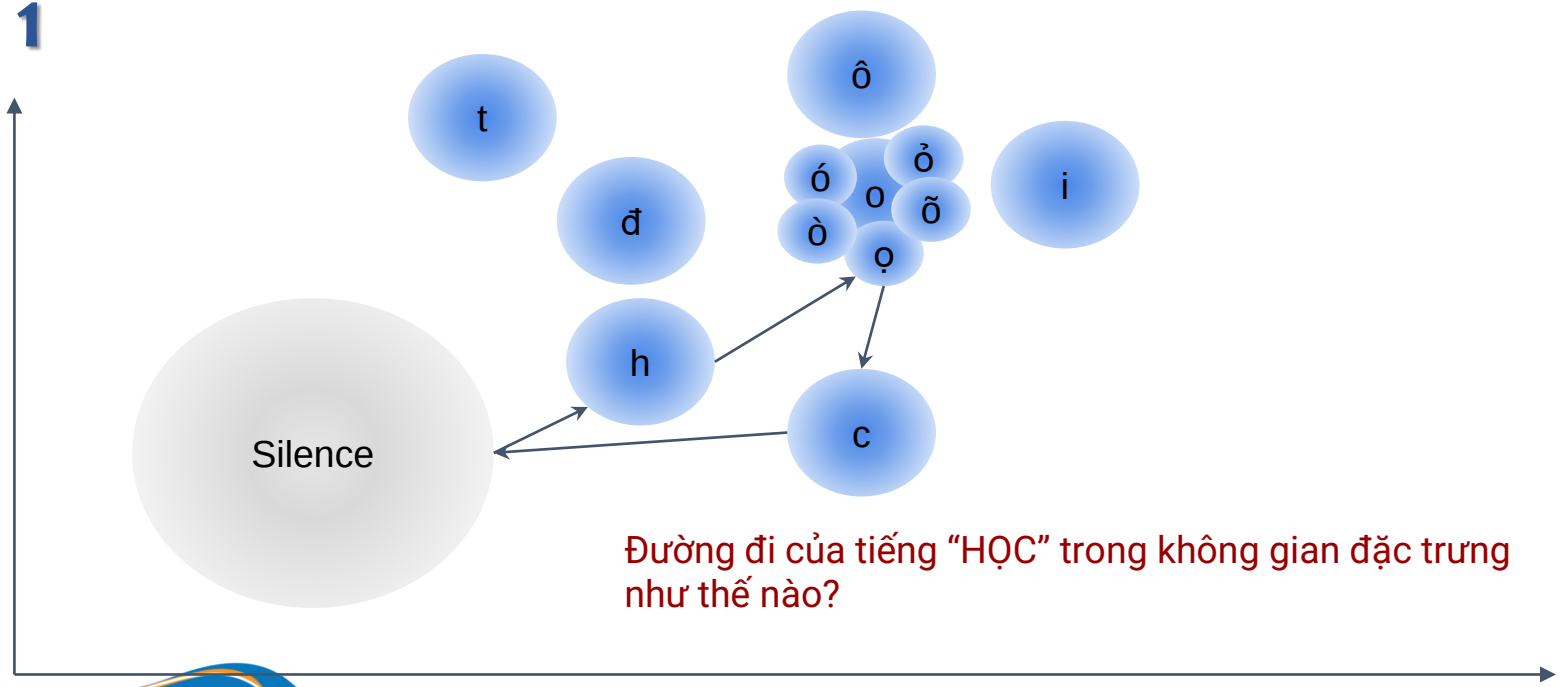

Đường đi của tiếng “Đ I” trong không gian đặc trưng

(Không gian đặc trưng N chiều)

# Không gian đặc trưng của tiếng nói


Đường đi của tiếng “HỌC” trong không gian đặc trưng như thế nào?

# Mô hình ngữ âm đơn mono-phone, tiếp cận 1




Đường đi của tiếng “HỌC” trong không gian đặc trưng như thế nào?

# Mô hình ngữ âm đơn mono-phone cho tiếng Việt

Có nhiều cách tiếp cận, ví dụ với từ trường

1. trường = tr, ư, ơ2, ng
2. trường = tr, ư, ơ, 2, ng
3. trường = tr, ư, ơ, ng2
4. trường = tr, ư, ơ, ng, 2
5. trường = tr, ư2, ơ2, ng2
6. trường = tr2, ư2, ơ2, ng2

# Mô hình ngữ âm tri-phone

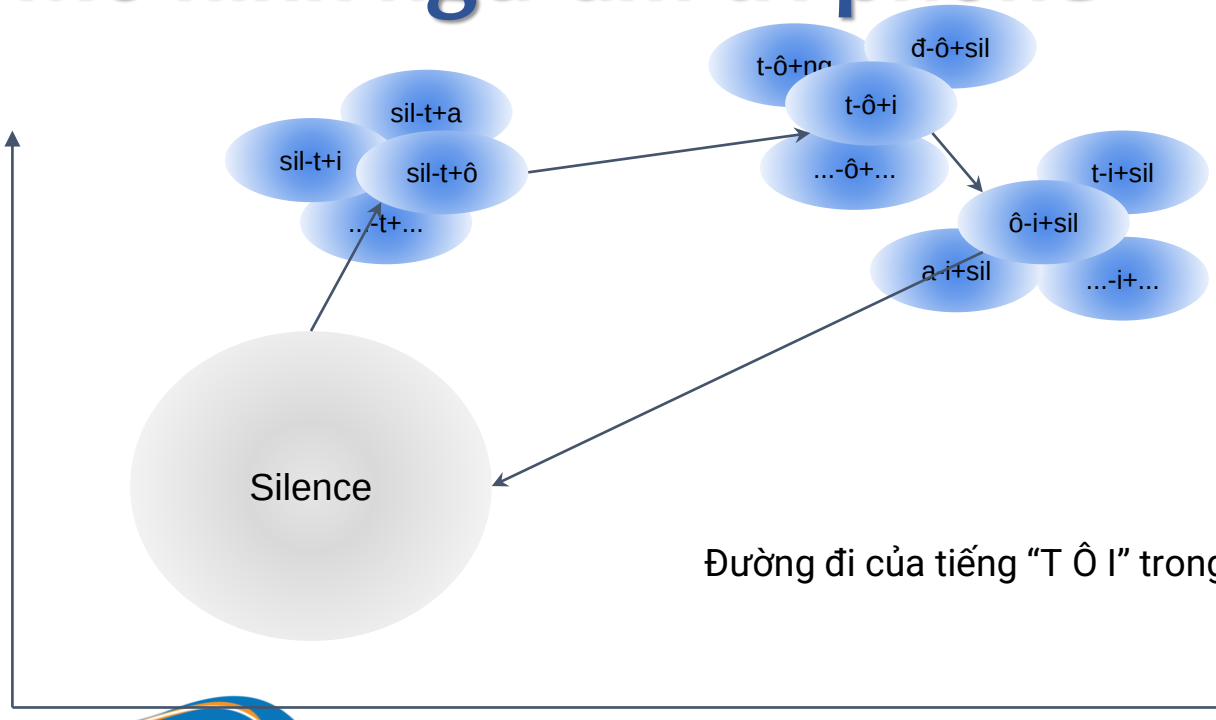
Mono-phone:

tôi = t, ô, i

Tri-phone:

tôi = sil-t+ô, t-ô+i, ô-i+sil

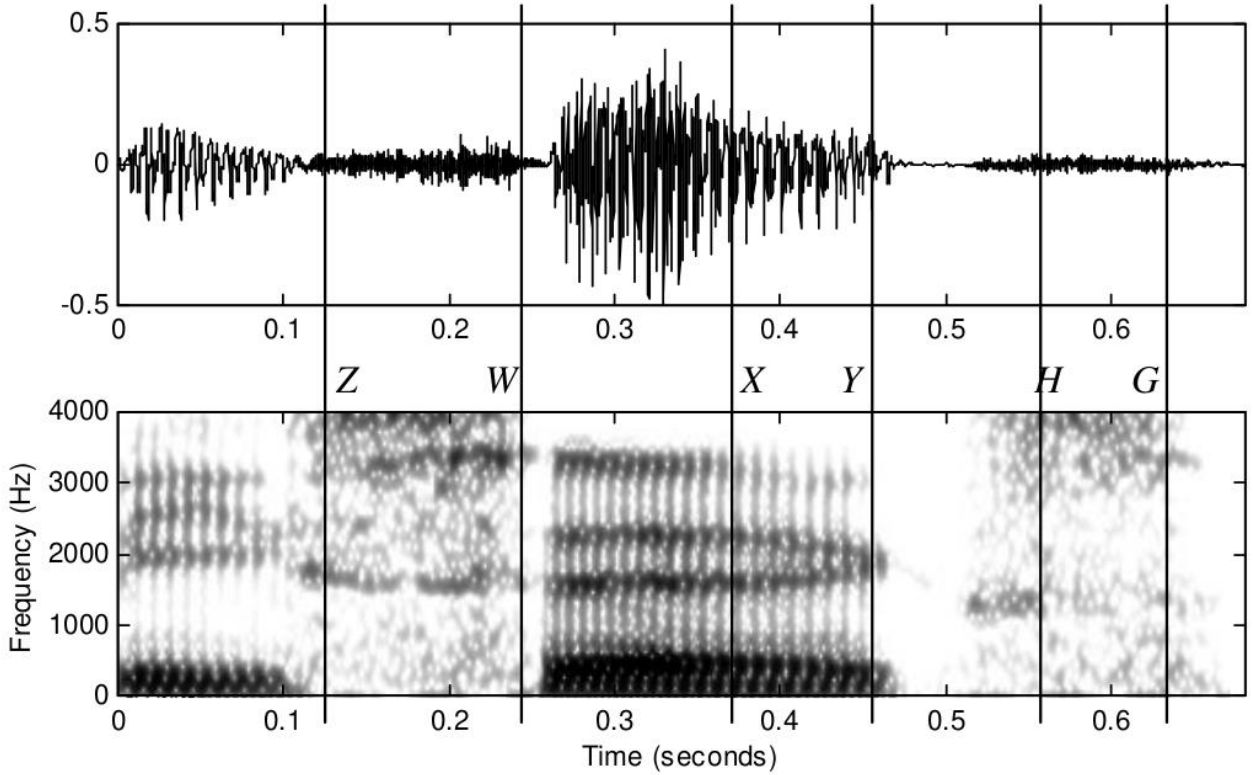
# Mô hình ngữ âm tri-phone



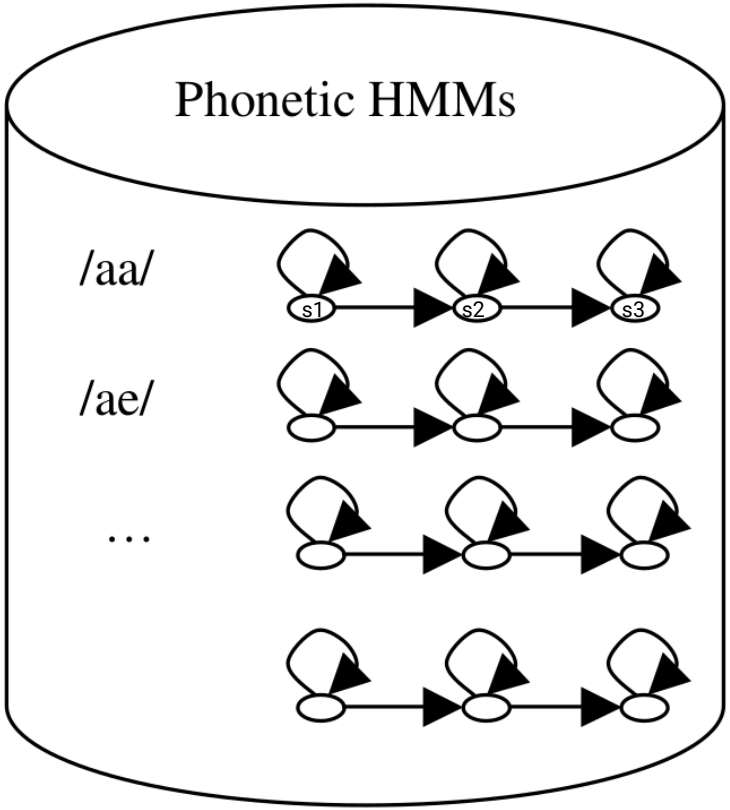

Đường đi của tiếng “T Ô I” trong không gian đặc trưng

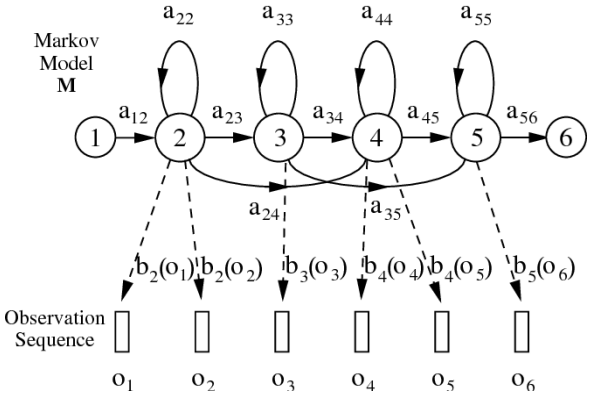
(Không gian đặc trưng N chiều)

# Bàn về âm dài, âm ngắn



# Mô hình ngữ âm dùng Hidden Markov

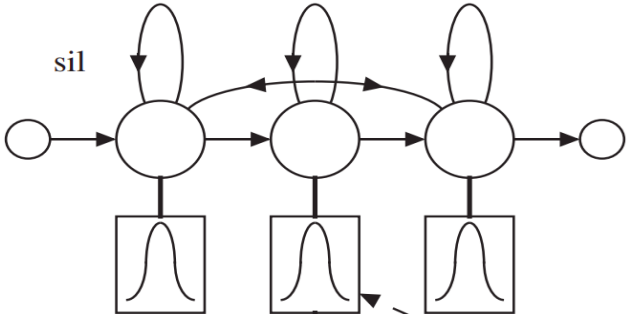


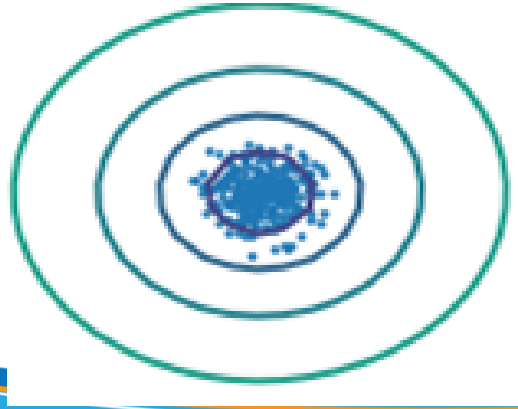




# Mô hình cho các state S

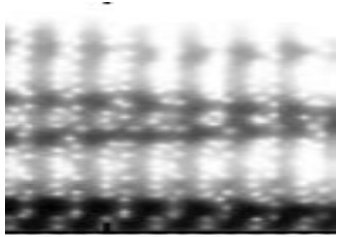
Sử dụng Gaussian model

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$



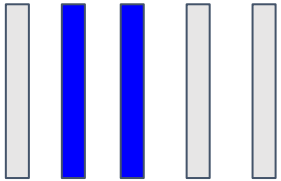



# Sử dụng Gaussian mixture model

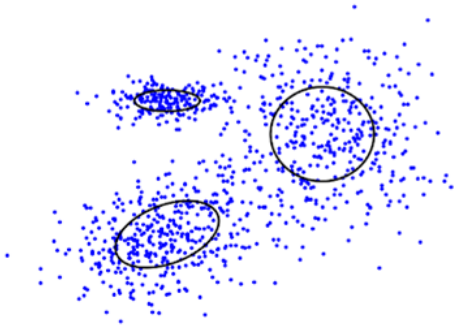


Trong tiếng nói

- Mỗi người sẽ có giọng nói khác nhau
- Cùng 1 người nhưng mỗi lần nói cho cùng 1 âm cũng không hoàn toàn giống nhau
- Môi trường thu âm có nhiều âm thanh nhiễu khác nhau



s1 s2 s2 s3 s3



Phân phối của s2

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

# Huấn luyện mô hình ngữ âm

Bài toán ASR, Bayes' rule

P(W | audio) = p(audio | W) P(W) / p(audio)

W : câu văn

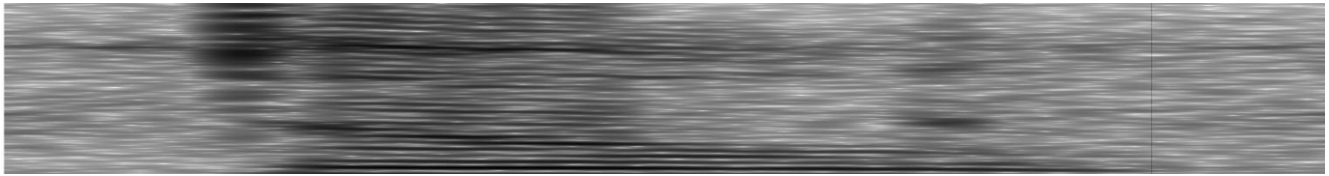
**p(audio | W): mô hình ngữ âm**

P(W): mô hình ngôn ngữ

# Huấn luyện mô hình ngữ âm

Đầu vào:

Đặc trưng đoạn âm thanh

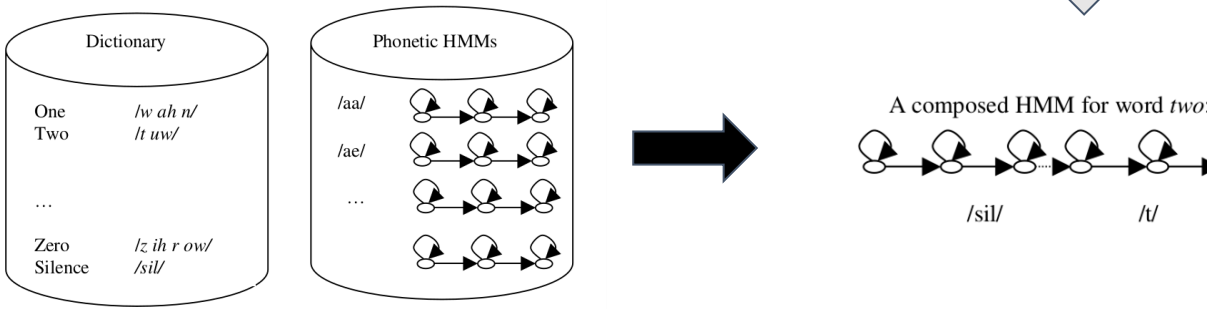


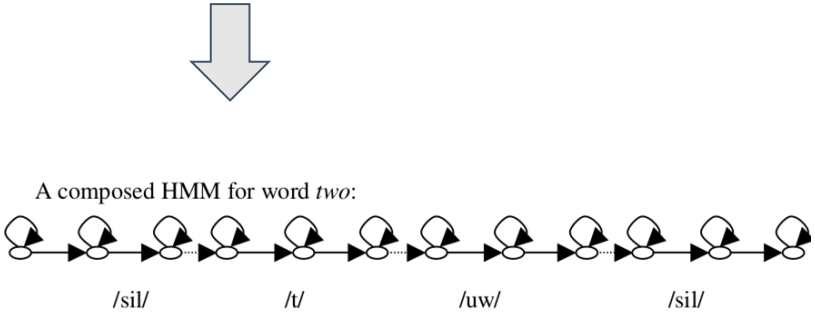
Văn bản đọc của đoạn âm thanh

"T W O"

Từ điển phiên âm

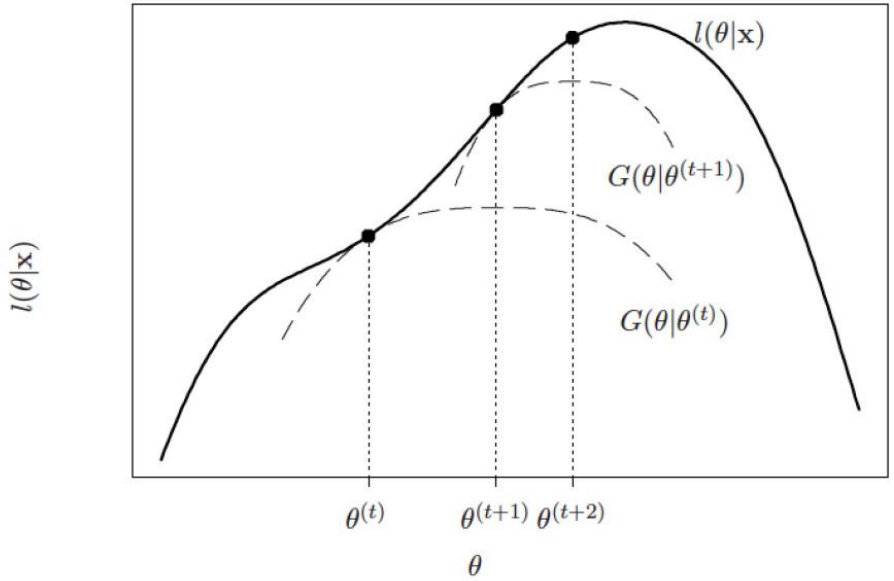
Mô hình ngữ âm



# Baum-Welch Algorithm

Là một trường hợp của expectation–maximization (EM) alg



# Huấn luyện mô hình ngữ âm

Tham khảo chính: Huang, Xuedong & Acero, Alex & Hon, Hsiao-Wuen. (2001). Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.

Baum-Welch Algorithm
https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm

Expectation–maximization algorithm

https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

Forward–backward algorithm

https://en.wikipedia.org/wiki/Forward%E2%80%93backward_algorithm

Jensen's inequality

https://en.wikipedia.org/wiki/Jensen%27s_inequality

# Kaldi Speech Recognition Toolkit

Installation

- Cài đặt từ source code
- Sử dụng Docker
- Sử dụng conda

# Cấu trúc một số thư mục, tập tin trong Kaldi

src: source code

tools: third party library

egs: Các ví dụ

→ yesno: ví dụ thường dùng để kiểm tra cài đặt thành công

→ an4: ví dụ xây dựng hệ thống ASR trên tập Alphanumeric database (http://www.speech.cs.cmu.edu/databases/an4/)

egs/an4/s5/cmd.sh: khai báo lệnh gọi chạy ở máy local (run.pl) hoặc trên GridEngine (queue.pl)

egs/an4/s5/path.sh: khai báo các đường dẫn đến các thư viện liên quan

egs/an4/s5/run.sh: bash script gọi các lệnh cần thiết để xây dựng hệ thống ASR

# Yes No example

Demo

# AN4 example

head -5 data/lang/phones.txt

head -2 data/lang/words.txt

head -2 data/train/wav.scp

head -2 data/train/text

head -2 data/train/utt2spk

# wspecifier

| | |
|---|---|
| “ark:foo.ark” | Write to archive “foo.ark” |
| “scp:foo.scp” | Write to files using mapping in foo.scp |
| “ark:-” | Write archive to stdout |
| “ark,t:\|gzip -c >foo.gz” | Write text-form archive to foo.gz |
| “ark,t:-” | Write text-form archive to stdout |
| “ark,scp:foo.ark,foo.scp” | Write archive and scp file (see below) |

# rspecifier

| | |
|---|---|
| “ark:foo.ark” | Read from archive foo.ark |
| “scp:foo.scp” | Read as specified in foo.scp |
| “ark:-” | Read archive from stdin |
| “ark:gunzip -c foo.gz\|” | Read archive from foo.gz |
| “ark,s,cs:-” | Read archive (sorted) from stdin... |

# Feature processing

FBANK features: compute-fbank-feats, steps/make_fbank.sh
MFCC features: compute-mfcc-feats, steps/make_mfcc.sh

```
# MFCC feature extraction
    mfccdir='mfcc'
    for x in test train; do
        steps/make_mfcc.sh --nj $nj --cmd
"$train_cmd" \
            data/$x exp/make_mfcc/$x
$mfccdir
    done
```

# Feature processing

**head -n 3 mfcc/raw_mfcc_test.1.scp**

fcaw-an406-b /opt/kaldi/egs/an4/s5/mfcc/raw_mfcc_test.1.ark:13

fcaw-an407-b /opt/kaldi/egs/an4/s5/mfcc/raw_mfcc_test.1.ark:5325

fcaw-an408-b /opt/kaldi/egs/an4/s5/mfcc/raw_mfcc_test.1.ark:10767

Xem giá trị mfcc của câu fcaw-an406-b

**copy-feats "scp:head -n 1 mfcc/raw_mfcc_test.1.scp|" ark,t:-**

# Train mono-phone

steps/train_mono.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono

Decode:

utils/mkgraph.sh data/lang exp/mono exp/mono/graph

steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" \

exp/mono/graph data/test exp/mono/decode

# Train tri-phone

```
steps/train_deltas.sh --cmd "$train_cmd" \
      1800 9000 data/train data/lang exp/mono_ali exp/tri1
utils/mkgraph.sh data/lang exp/tri1 exp/tri1/graph
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" \
      exp/tri1/graph data/test exp/tri1/decode
```

# Bài tập

Xây dựng mô hình nhận dạng đọc số điện thoại bằng tiếng Việt

- Thu dữ liệu đọc số điện thoại
- Tạo từ điển phiên âm cho từ điển số

  Tham khảo cách tạo tập tin lexicon.txt cho dữ liệu an4 (https://github.com/kaldi-asr/kaldi/blob/master/egs/an4/s5/local/lexicon_prep.py)

- Xây dựng mô hình ngôn ngữ

  Tham khảo thư viện pocolm (https://github.com/danpovey/pocolm)

- Huấn luyện mô hình ASR và kiểm chứng kết quả

# Epitran: Công cụ hỗ trợ tạo từ điển phiên âm

A library and tool for transliterating orthographic text as IPA (International Phonetic Alphabet).

https://pypi.org/project/epitran/

```
In [1]: import epitran

In [2]: epi = epitran.Epitran('vie-Latn')

In [3]: epi.trans_list("Chào các bạn ")

Out[3]: ['c', 'a', 'w', ' ', 'k', 'a', 'k', ' ', 'b', 'a', 'n', ' ']
```