UNIVERSITÀ
DEGLI STUDI DI TRIESTE

dipartimento
di ingegneria
e architettura

Speech recognition

A.Carini – Elettronica per l'audio e l'acustica

Speech recognition

- E’ l’area che ha maggiormente beneficiato delle tecniche di machine learning.
- E’ stato proprio grazie alle tecniche di machine learning che è stato possibile superare quel pleateau nelle prestazioni, durato un decennio (attorno alla fine del secolo), che ne limitava prestazioni e quindi la comparsa di nuove applicazioni.
- Sono stati scritti diversi libri sul riconoscimento vocale e questo argomento da solo è responsabile per la metà degli articoli su «*speech and audio processing*».
- Ci sono diversi motivi per questo interesse, primo tra tutti il desiderio di comunicare più naturalmente con un computer.

Gerarchia dei modi di interazione uomo-macchina

Hardwired: The computer designer (i.e. engineer) 'reprograms' a computer, and provides input by reconnecting wires and circuits.

Card: Punched cards are used as input, printed tape as output.

Paper: Teletype input is used directly, and printed paper as output.

Alphanumeric: Electronic keyboards and monitors (visual display units), alphanumeric data.

Graphical: Mice and graphical displays enable the rise of graphical user interfaces (GUIs).

WIMP: Standardised methods of windows, icons, mouse and pointer (WIMP) interaction become predominant.

Touch: Touch-sensitive displays, particularly on smaller devices.

Ian Vince McLoughlin

Gerarchia dei modi di interazione uomo-macchina

Speech commands: Nascent speech commands (such as voice dialling, voice commands, speech alerts), plus workable dictation capabilities and the ability to read back selected text.

Natural language: We speak to the computer in a similar way to a person, it responds similarly.

Anticipatory: The computer understands when we speak to it just like a close friend, husband or wife would, often anticipating what we will say, understanding the implied context as well as shared references or memories of past events.

Mind control: Our thoughts are the interface between ourselves and our computers.

Gerarchia dei modi di interazione uomo-macchina

- In questo momento la ricerca si posizione in prevalenza tra i *speech command* e il *natural language.*
- I ricercatori che lavorano sul futuro dei *digital assistant* (maggiordomi digitali) hanno fatto grandi progressi sui sistemi capaci di comprendere il contesto e riferimenti condivisi.
- Mentre il *mind control* rimane fantascienza, ricercatori che lavorano con fMRI (functional Magnetic Resonance Imaging) e strumenti EEG (electroencephalogram) hanno isolato dei *pattern* di attivazione del cervello associati a concetti fissi, attività, singole parole o frasi.
- Siamo agli inizi, ma 30 anni fa nel riconoscimento vocale si potevano riconoscere solo singole parole mentre oggi viene trattato con facilità il parlato continuo.

Speech recognition

- Nel seguito considereremo principalmente il riconoscimento vocale e il task di decidere quale parola è stata detta.
- Considereremo il problema collegato della rivelazione della presenza di voce (*voice activity detection*), dell'identificazione del parlatore e della lingua, di quante persone parlano e di cosa viene detto.
- Il riconoscimento vocale copre un campo estremamente vasto, normalmente categorizzato come segue …

Tipi di riconoscimento vocale

Automatic speech recognition (ASR) describes a system that can recognise what has been spoken, without requiring additional user input.[1]

Keyword spotting means looking out for particular words or phrases (e.g. 'attack' or 'plant a bomb'), as well as listening out for a particular phrase that signals the user's intention to begin speech communications (e.g. 'Okay Google' or 'Computer:'). This technology is used primarily for initiating vocal commands, which means recognition of single words delimited by pauses, as well as monitoring continuous telephone conversations for public security purposes.

[1] *Speech recognition* is the task of recognising speech, by both human and computer listeners, whereas *automatic* speech recognition implies that it is a computer doing the listening.

Ian Vince McLoughlin

Tipi di riconoscimento vocale

Continuous speech recognition is recognition of full sentences or paragraphs of speech rather than simple phrases or words. These applications do not require a user to pause when speaking (so the system must be quick enough to cope with their fastest rate of speaking – a real problem with current technology), and would encompass dictation and transcription systems.

Natural language processing (NLP), whilst not strictly limited to speech, describes the computational methods needed for a computer to understand the meaning of what is being said, rather than simply recognising what words have been spoken. Unlike automated transcription systems, where the meaning may be irrelevant as long as the words have been captured correctly, NLP enables the creation of a virtual butler who is able to cater to human needs, and so the semantics or meaning of what is said would be of primary importance.

Ian Vince McLoughlin

Speech recognition

- Ci concentreremo nell'obiettivo del riconoscimento delle parole (che dipendono dal riconoscimento dei fonemi).
- Questa tecnologia fornisce i fondamenti di tutte le applicazioni che abbiamo menzionato, ma non richiede l'interpretazione ad alto livello necessaria per il *natural language processing*, né richiede di preoccuparsi per i requisiti di elaborazione in tempo reale del riconoscimento continuo o della trascrizione automatica della voce.

Prestazioni di un sistema di speech recognition

- Sono stati identificati diversi parametri che possono essere usati per caratterizzare i sistemi di riconoscimento vocale e le loro prestazioni.

Table 9.1 Speech recognition system parameters.

Parameter	Typical range (easier	–	more difficult)
Vocabulary	small	–	large
Users	single	–	open access
Speech type	single words	–	continuous sentences
Training	in advance	–	continuous
SNR	high	–	low
Transducer	restricted	–	unrestricted

Vocabulary and SNR

- E’ ragionevole assumere che più grande sia il vocabolario e tanto più difficile sia il riconoscimento accurato di una parola:
- Ci sono più opportunità di fraintendere le parole e più grandi probabilità che ci siano suoni simili.
- Del resto la stessa situazione si presenta nel parlato umano.

Vocabulary and SNR

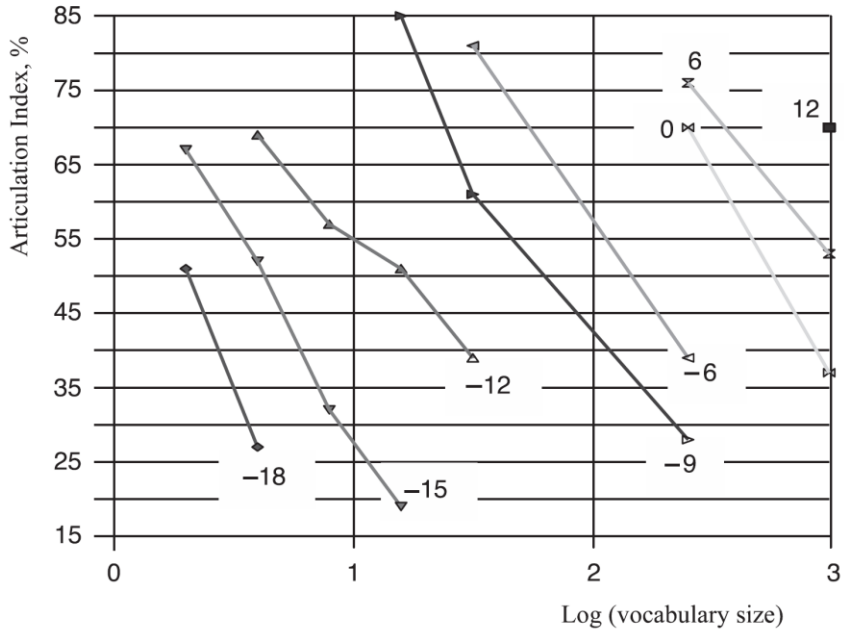


Figure 9.1 Plot of articulation index (a measure of recognition accuracy) versus the logarithm of vocabulary size for speech recognised at various levels of additive white noise, with SNR ranging from −18 dB to +12 dB.

Vocabulary and SNR

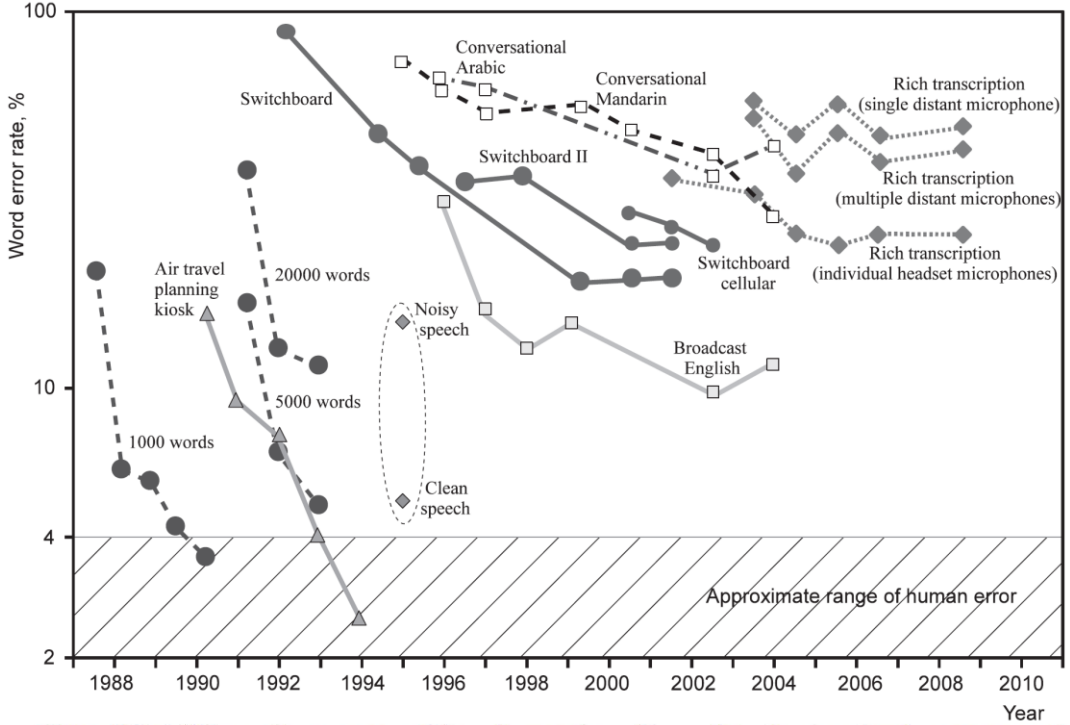


Figure 9.2 ASR word error rate achieved over time. Reproduced using the data presented in Figure 4 of [116], showing the historic results of various NIST ASR competitions and evaluations.

Vocabulary and SNR

- I valori peggiori nelle competizioni più recenti derivano dalla loro maggiore complessità rispetto alle prime.
- Alcune sono più complesse in termini di vocabolario.
- «Switchboard» contengono più di 3,000,000 di parole di conversazioni telefoniche spontanee da più di 500 parlatori americani di entrambi i generi, con un vocabolario di 6000 parole. Sono caratterizzati da una grande variabilità di parlatori, rumore, e problematiche legate alla vita reale.
- Uno studio del 1995 ha mostrato l'effetto negativo del rumore sul WER (circolo).
- Un simile effetto influenza le tre linee alla destra: mostrano i risultati di trascrizioni di riunioni usando rispettivamente un microfono singolo a distanza, un array di microfoni, un *headset microphone*. Più distante il microfono, maggiore il rumore.
- E' bene ricordare il salto di prestazioni che è stato ottenuto con i big data e le tecniche di machine learning, il quale non viene considerato nella figura.

Training

- La maggior parte dei sistemi di ASR richiede una qualche forma di addestramento per familiarizzare il sistema alla voce di un particolare individuo o di un gruppo di individui simili.
- Il training viene generalmente fatto in anticipo usando un ampio dataset di parlato simile (o di parlatori simili). Più grande è la variabilità delle condizioni che il sistema deve trattare, più difficile è il training e il raggiungimento di un dato livello d'errore. Inoltre, tanto più ampio deve essere il training set.
- Ci dovrebbero essere uguali quantitativi di materiale di training per ciascuno dei diversi tipi di voce che il sistema deve trattare. Non possiamo aspettarci il sistema riconosca un ingresso per il quale non è stato addestrato.
- Nel riconoscimento vocale, tutte le parole che vogliamo riconoscere dovrebbero essere nel training set (in alternativa, tutti i fonemi contenuti in quelle parole).

Training

- Oltre alla quantità, anche la qualità del training set è importante – deve essere rappresentativa della reale voce che il sistema verrà ad ascoltare. Questo vuol dire che le voci dovrebbero essere idealmente pronunciate in modo simile (con la stessa pronuncia) della voce che sarà riconosciuta.

Restricted topic

- Il riconoscimento continuo della voce include riconoscitori per argomenti specifici. L'idea è che restringendo il tipo di parole, la scelta del vocabolario, la grammatica, diviene possibile migliorare le prestazioni.
- Nella figura, il chiosco per la pianificazione dei viaggi aerei ha ottenuto una eccellente accuratezza in parte perché confinato a una specifica area di applicazione.
- Siccome questi riconoscitori «ristretti» lavorano così bene, sono stati anche sviluppati riconoscitori di argomento da usare come front-end di analisi per determinare quale sistema di riconoscimento di back-end dovrebbe essere usato per trattare una specifica frase in base all'argomento cui si riferisce.
- In pratica, il front-end decide quale vocabolario di ASR deve essere usato per la frase corrente.

Transducer

- Il processo di cattura della voce di ingresso da usare nel riconoscimento è molto importante: influenza non solo il rapporto segnale rumore, ma anche la distorsione della voce, parametri che determinano la qualità della voce catturata.
- Nell'ASR, i due attributi principali del processo di cattura sono:
    1. Il posizionamento del microfono rispetto alla sorgente della voce (la bocca) e ai rumori di fondo interferenti;
    2. La funzione di trasferimento tra il suono in ingresso e il segnale elettrico in uscita al microfono (e tutto quanto sta fino alla digitalizzazione).
- Il primo influenza il rapporto segnale rumore, il secondo la distorsione del segnale.

Transducer

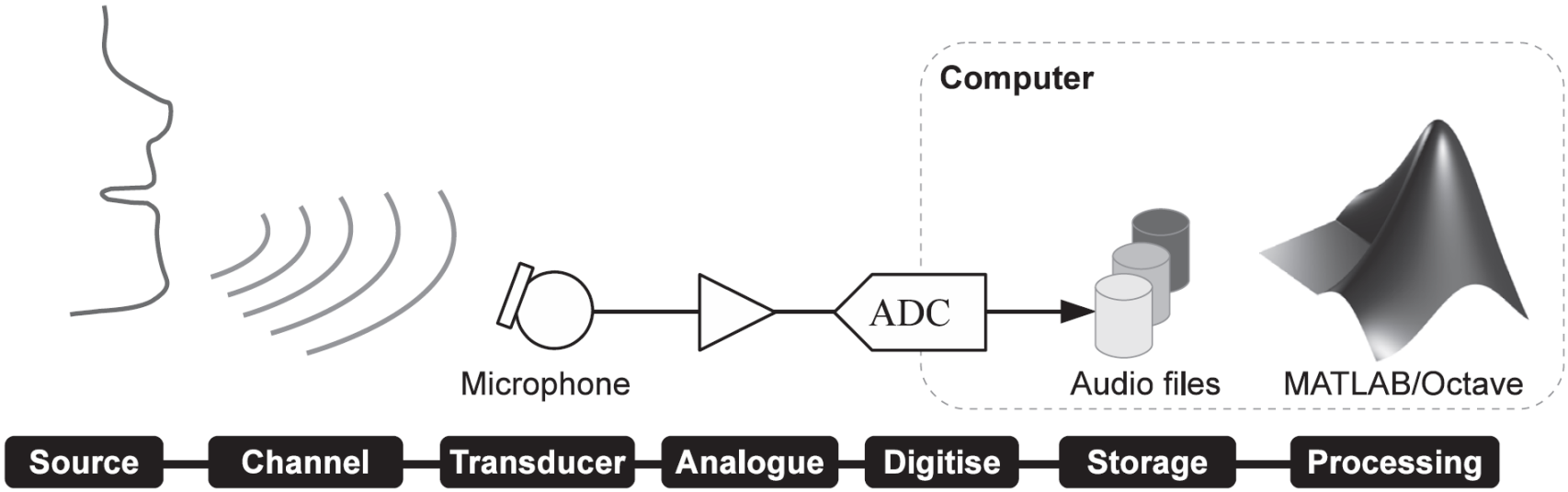


Figure 9.3 A diagram illustrating the various phases in the capture and processing of speech to perform ASR.

Transducer

- Un trasduttore capace di minimizzare il cammino audio del segnale fornirà le migliori prestazioni.
- Si ricordi che microfoni e altoparlanti hanno diversi parametri di sensibilità e direttività, da quelli omnidirezionali e quelli unidirezionali.
- L'uso di array microfonici consente alcune elaborazioni sofisticate come il *beam steering,* il puntamento elettronico del segnale catturato verso la bocca desiderata, cancellando allo stesso tempo le altre bocche o sorgenti di rumore interferenti, posizionando uno zero nella loro direzione.
- Lavorando con un sistema di riconoscimento accordato per una specifica voce che può raggiungere un accuratezza del 90%, un problema così banale come il cambiamento di posizione del microfono, l'uso di un diverso microfono, o la presenza di un lieve rumore di fondo può produrre una caduta dell'accuratezza del 20% o più.

Transducer

- La presenza o l'assenza di un rumore di fondo è un fattore d'operazione critico. Sistemi di riconoscimento progettati per lavorare con microfoni headset o simili, otterranno migliori risultati di quelli che catturano la voce da un microfono posizionato lontano dalla bocca del parlatore.
- Alcuni sistemi di riconoscimento fanno uso di chiavi/*cue* non vocali aggiuntive per migliorare le prestazioni, come
    - immagini video della bocca,
    - echi a ultrasuoni della bocca,
    - gesti del corpo,
    - espressioni facciali,
    - impulsi nervosi nel collo.

  Ciascuno richiede qualche tipo di trasduttore per catturare l'informazione. Hanno tutti diverse caratteristiche.

Some basic difficulties

- Ci sono diversi aspetti cui i sistemi di riconoscimento devono adattarsi:

- Voice activity detector: un dispositivo capace di rilevare la presenza della voce. Non ha senso spendere preziose risorse di calcolo per riconoscere cosa viene detto quando non è presente alcuna voce.
- Segmentazione della voce in più piccole unità. Viene spesso richiesta dai sistemi di elaborazione. Mentre nelle tecniche di elaborazione audio si usano dei frame di dimensione fissa, nei sistemi di ASR può essere richiesta una segmentazione in parole o fonemi. E' un compito non banale, non si tratta di trovare semplicemente i gap, le pause del parlato continuo. I gap tra le parole e le frasi sono talvolta più piccoli di quelli tra i fonemi.

Some basic difficulties

- Word stress, l'enfasi sulle parole: può essere molto importante per determinare il significato di una frase e sebbene raramente catturata da una parola scritta è ampiamente usata nelle comunicazioni vocali. Es:

He *said he did not eat this* : indicating that someone else said so
He **said** *he did not eat this* : indicating that you probably disbelieve him
He said **he** *did not eat this* : indicating that someone else ate it
He said he **did** *not eat this* : indicating that he is, or will be eating this
He said he did **not** *eat this* : indicating an emphatic negative
He said he did not **eat** *this* : indicating that he did something else with it
He said he did not eat **this** : indicating that he ate something, but not this

Some basic difficulties

- Il contesto di fonemi, parole, espressioni, o frasi, etc. Può essere usato per rafforzare il riconoscimento.
- Il contesto dei fonemi, ovvero la sequenza di 2-3 fonemi vicini è molto importante nei moderni riconoscitori vocali.

Voice activity detection (VAD) and segmentation

- Un VAD ha il compito di rilevare la presenza dalle voce nell'audio.
- E' utile in tante applicazioni.
- Ad esempio: un microfono nascosto in cui si vuole rilevare quando qualcuno sta parlando per registrare il suono solo in quei momenti.
- Altro esempio: la codifica voce nella telefonia mobile, in cui rilevando e codificando i soli frame voce diviene possibile ridurre del 50% la banda radio occupata e la dissipazione di potenza.
- Nel riconoscimento vocale: sappiamo che l'ASR è molto pesante da un punto di vista computazionale. Un VAD fa la differenza tra un sistema che cerca di riconoscere ogni singola frame audio e un sistema che viene attivato e consuma potenza solo quando è presente la voce.

Voice activity detection (VAD) and segmentation

- Nel progetto di un VAD, più la situazione è vincolata e migliori sono le prestazioni.
- Un VAD che operi con un rumore di fondo noto, con un microfono predeterminato, che rilevi voce continua da un range di parlatori noto è più semplice da progettare di uno progettato per catturare qualunque voce (anche piccole parole) dette da una persona qualsiasi, con qualunque situazione di rumore e qualunque microfono.

Metodi di Voice Activity Detection

- Ci sono miriadi di modi di progettare un VAD.
- Diverse tecniche traggono beneficio da diversi scenari di lavoro e portano a diversi compromessi tra prestazioni e complessità.
- In genere, le prestazioni di un VAD dipendono fortemente dalla qualità del segnale audio e degradano rapidamente all'aumentare del rumore di fondo.
- Le prestazioni dipendono anche dal tipo di rumore di fondo (anche per gli uomini certi tipi di rumore mascherano la voce meglio di altri). Alcuni rumori sono molto simili alla voce.
- I primi VAD erano dei *rivelatori di energia:*
  - La voce viene divisa in frame di analisi e viene calcolata l'energia in ogni frame. Questa viene confrontata con una soglia. Se eccede la soglia si giudica presente la voce.
  - Lavorano bene quando non c'è rumore o ce n'è poco e non assomiglia a voce.

Metodi di Voice Activity Detection

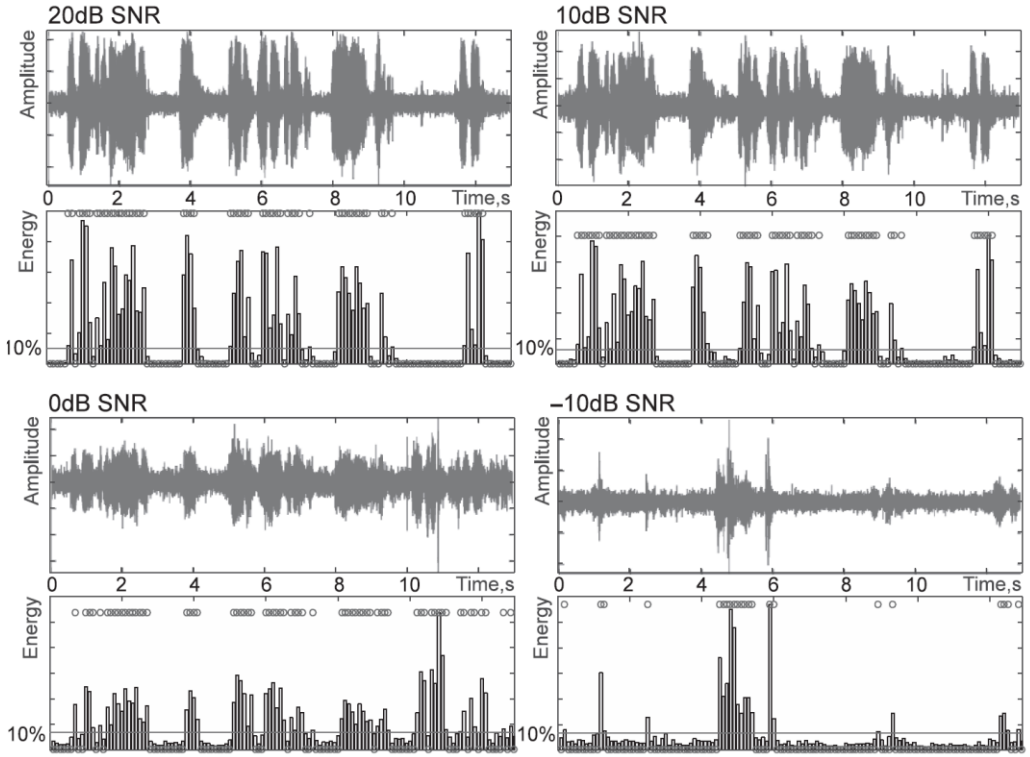


Figure 9.4 Illustration of using an energy detector to perform VAD, using a 10% level on a 13 s sample of broadcast speech mixed with office noise at levels of 20, 10, 0 and −10 dB SNR.

Metodi di Voice Activity Detection

```
%the noisy speech is in array nspeech
%fs is the sample rate
L=length(nspeech);
frame=0.1; %frame size in seconds
Ws=floor(fs*frame); %length
Nf=floor(L/Ws); %no. of frames
energy=[];
%plot the noisy speech waveform
subplot(2,1,1)
plot([0:L-1]/fs,nspeech);axis tight
xlabel('Time,s'); ylabel('Amplitude');
%divide into frames, get energy
for n=1:Nf
  seg=nspeech(1+(n-1)*Ws:n*Ws);
  energy(n)=sum(seg.^2);
end
```

Metodi di Voice Activity Detection

```
%plot the energy
subplot(2,1,2)
bar([1:Nf]*frame,energy,'y');
A=axis; A(2)=(Nf-1)*frame; axis(A)
xlabel('Time,s'); ylabel('Energy');
%find the maximum energy, and threshold
emax=max(energy);
emin=min(energy);
e10=emin+0.1*(emax-emin);
%draw the threshold on the graph
line([0 Nf-1]*frame,[e10 e10])
%plot the decision (frames > 10%)
hold on
plot([1:Nf]*frame,(energy>e10)*(emax),'ro')
hold off
```

Metodi di Voice Activity Detection

- La tecnica di rivelazione d'energia viene facilmente ingannata dal rumore.
- Potrebbe lavorare bene in uno scenario tipo studio di registrazione, ma in una casa, veicolo, ufficio, etc. sbaglierà facilmente, sia venendo attivato quando non c'è la voce (falso positivo) sia non essendo attivato quando è presente (falso negativo).
- Interpreta qualunque suono come una potenziale voce con il solo criterio di discriminazione dell'energia.
- Invece dovremmo cercare di rilevare se un particolare tipo di suono assomiglia a una voce e dovremmo rigettare quei suoni che non le assomigliano.
- La misura della discriminazione dovrebbe essere legata a feature note della voce, calcolate nel dominio del tempo o nel dominio della frequenza.

Frequency domain features

- Sappiamo che la voce è composta da una sequenza di fonemi voiced e unvoiced e che queste due classi hanno diverse caratteristiche dipendenti dal loro meccanismo di produzione.
- I fonemi vocalizzati contengono delle formanti che possono essere rivelate in uno spettro sul breve periodo sotto forma di picchi. Le prime tre formanti cadono in un range noto. Un rivelatore dei picchi dello spettro potrebbe identificare i fonemi nel range corretto e decidere la presenza della voce.
- Ciononostante, molti suoni in natura (e.g., strumenti musicali, versi di animali, etc.) contengono picchi spettrali con la stessa relazione armonica delle formanti umane.
- Nei fonemi non vocalizzati, alcuni fonemi come le fricative (/s/) contengono uno spettro a larga banda alle alte frequenze. Potremmo rilevare questi fonemi nel dominio della frequenza confrontando l'energia nelle varie bande.

Frequency domain features

- Ciononostante, una energia elevata alle alte frequenze non indica solo una /s/, si pensi al suono di gas che esce da un tubo, o al rumore del vento tra le foglie.
- Il pitch è pure importante nella voce. Sappiamo che f0 cade in un range limitato ed è presente nei suoni vocalizzati. Potremmo rilevare la presenza del pitch.
- Anche qui molti suoni e rumori di fondo hanno delle caratteristiche simili al pitch, specie le macchine rotanti.
- Nessuno di questi approcci da solo funziona bene, ma una migliore soluzione viene data da una combinazione di questi approcci.
- Se un detector può essere ingannato, due o tre detector combinati è difficile siano ingannati.
- Possiamo ottenere ancora migliori prestazioni combinando le misure nel dominio della frequenza con quelle nel dominio del tempo.

Time domain features

- L'energia da sola non è una buona metrica a causa della sua forte variabilità.
- I suoni vocalizzati hanno maggiore energia di quelli non vocalizzati.
- Possiamo rendere più affidabile la rilevazione tenendo conto di questa variabilità.
- Avere sole poche frame di alta energia non significa sia presente la voce, ma avere un pattern specifico di alta, media, bassa energia, simile alla voce è più informativo.
- Sappiamo che la voce viene pronunciata secondo certi pattern.
- Alcuni sono determinati dal nostro sistema (polmoni, glottide, gola), altri sono vincolati dal linguaggio, dal contenuto o dal nostro modo di parlare.
- Ci sono dei pattern specifici: rate sillabico, rate dei fonemi, il passaggio da suoni vocalizzati a non, pattern di silenzio, cambiamenti d'ampiezza, …
- Alcuni pattern possono essere rilevati studiando come varia l'energia, ma possiamo anche studiare come variano nel tempo le feature in frequenza.

Time domain features

- La rivelazione a questo punto diviene una questione di statistiche.
- Possiamo creare un modello delle variazioni di questi parametri nella normale voce. Dato il segnale audio sotto test possiamo ottenere le sue statistiche e confrontarle con il modello della voce per formare l'ipotesi che la voce risulti presente oppure no.
- A seconda di quanto differiscono dal modello, possiamo assegnare un valore di probabilità a ciascuna feature. Le probabilità delle feature sono sommate per ottenere la probabilità complessiva che la voce sia presente.
- Le statistiche dovrebbero contenere le medie, ma anche le variazioni attorno alla media (varianza o deviazione standard), e forse qualche misura di ordine superiore (lo spread delle varianze osservate attorno alla varianza media).

VAD performance

- Normalmente i VAD includono un «hang time» (o «hang over») e talvolta un «hang before» che prolungano il periodo di rivelazione in avanti o indietro rispetto al periodo effettivamente rilevato.
- Molte parole finiscono con fonemi a bassa energia difficili da rilevare o iniziano con suoni unvoiced bassi. I primi algoritmi VAD tagliavano questi suoni.
- Un hang time di 0.5 s - 1 s assicura che questi fonemi non siano persi, al costo di includere una breve sezione di silenzio all'inizio o alla fine della zona rivelata.

VAD performance

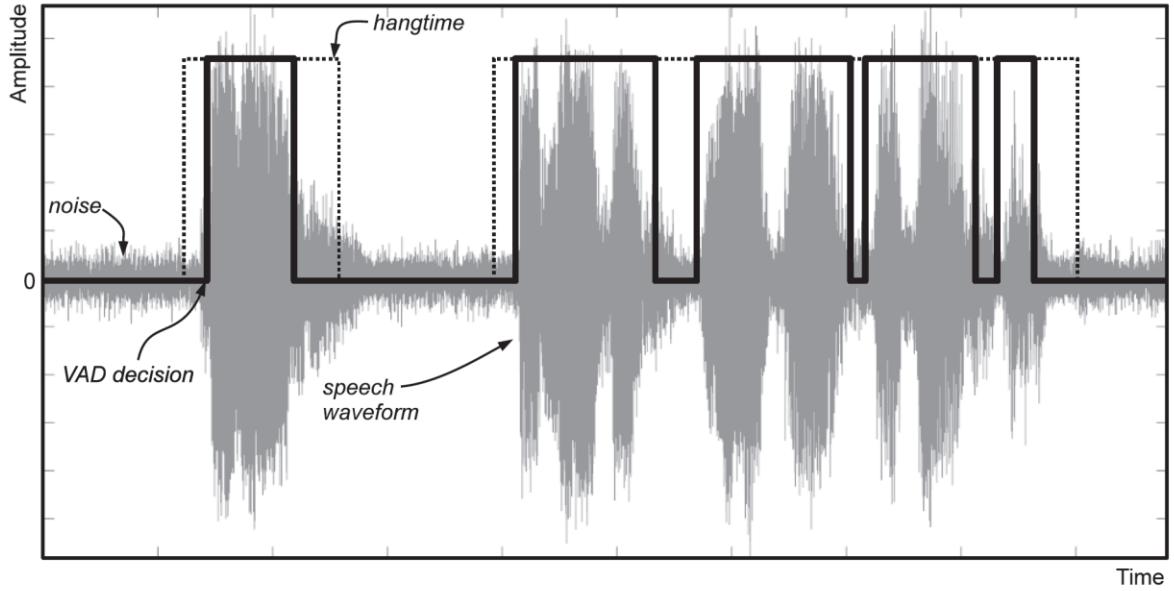


Figure 9.5 An illustration of VAD output (solid line) corresponding to the waveform of a speech recording. Often the VAD decision is extended in time to include a section of audio immediately before and immediately after a detected section, to ensure that the start and end portions of the utterance are captured.

VAD performance

- Rilevare le prestazioni di un VAD non è semplice.
- Un approccio semplicistico potrebbe contare la proporzione di volte il VAD rivela regioni in cui la voce è presente, ma non verrebbe a considerare quando il VAD manca completamente un segmento di voce, o quando si attiva all'interno della voce ma fallisce agli estremi.
- Un approccio altrettanto semplicistico potrebbe contare il numero di parole rilevate dal VAD: non verrebbe a tener conto dei falsi positivi (VAD attivato da rumore) né della parti di parole mancate.
- Un ulteriore difficoltà è rappresentata dal fatto che la voce contiene pause tra le frasi, i paragrafi, le parole, ma anche all'interno delle stesse parole.
- Sono questi gap parte della voce ?
- Una soluzione è quella di dividere la voce in frame di lunghezza conveniente alla *speech analysis,* frame che si spera siano più lunghi delle pause.

VAD performance

- Nei test, la così detta «*ground truth*» indica se ciascuna frame contiene voce o no.
- Un buon sistema VAD dovrebbe classificare ciascuna frame come la ground truth.
- Data una analisi frame per frame, possiamo identificare le prestazioni del VAD a seconda del tipo di errore:
  - Front-end clipping (FEC) percentuale di frame errati nella transizione da una regione non-speech a una speech.
  - Mid-speech clipping (MSC) proporzione di frame voce erroneamente classificati come non voce.
  - OVER o overrun percentuale di frame sbagliati quando si passa da una regione speech a una non-speech.
  - Noise detected as speech (NDS) proporzione di frame non-speech che sono classificati come speech.

VAD performance

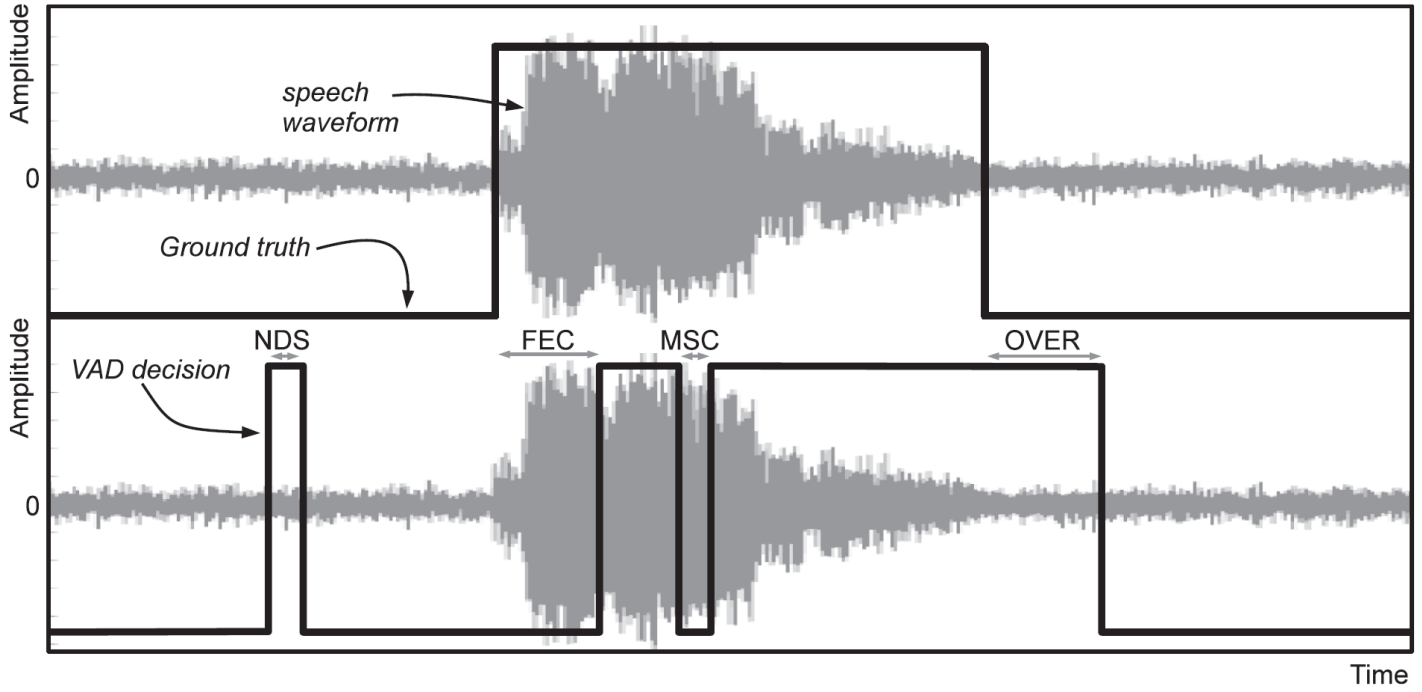


Figure 9.6 The lower plot illustrates four types of error used to score VAD performance (NDS, FEC, MSC, OVER). Ground truth is shown on the upper plot.

ASR training and testing

- I sistemi di machine learning hanno diverse fasi di operazione.
- Dapprima avremo una fase di training, che usa un dataset specificatamente per l'addestramento del modello. Ci potrebbe essere una fase di affinamento e c'è sempre una fase di test: operano entrambe su diversi dataset.
- Esempio: un sistema che riconosca i fonemi (o le parole). Il modello esaminerà le feature (e.g., MFCC, etc.) di un segmento di voce e verrà a stimare quale fonema è stato detto.
- L'addestramento viene effettuato con dei dati annotati - *ground truth* - che comprende la lista dei fonemi con l'informazione di inizio e fine. I dati annotati vengono confrontati con l'uscita del modello. L'addestramento viene effettuato rafforzando o aggiustando i pesi interni del modello sulla base dell'identificazione.

ASR training and testing

- Durante il test, un altro set di dati viene usato per fornire le feature al modello addestrato. L'uscita viene confrontata con la *ground truth* per determinare il tasso di errore.
- Durante la fase di operazione, i frame voce sono inviati in ingresso al sistema per determinare quale fonema è stato detto. Non c'è *ground truth* e quindi non c'è nessun confronto o aggiustamento del modello.
- Possiamo creare sistemi che riconoscono parole (e quindi addestrati su singole parole), fonemi (e quindi addestrati sui fonemi) o anche unità più piccole. Qualunque sia il riconoscimento effettuato, la *ground truth* deve corrispondere all'informazione d'addestramento.
- Le migliori prestazioni di riconoscimento si ottengono con una modellazione mediante tri-foni (*triphones*), in cui ciascuna combinazione di fonemi viene rappresentata mediante stati multipli.

ASR training and testing

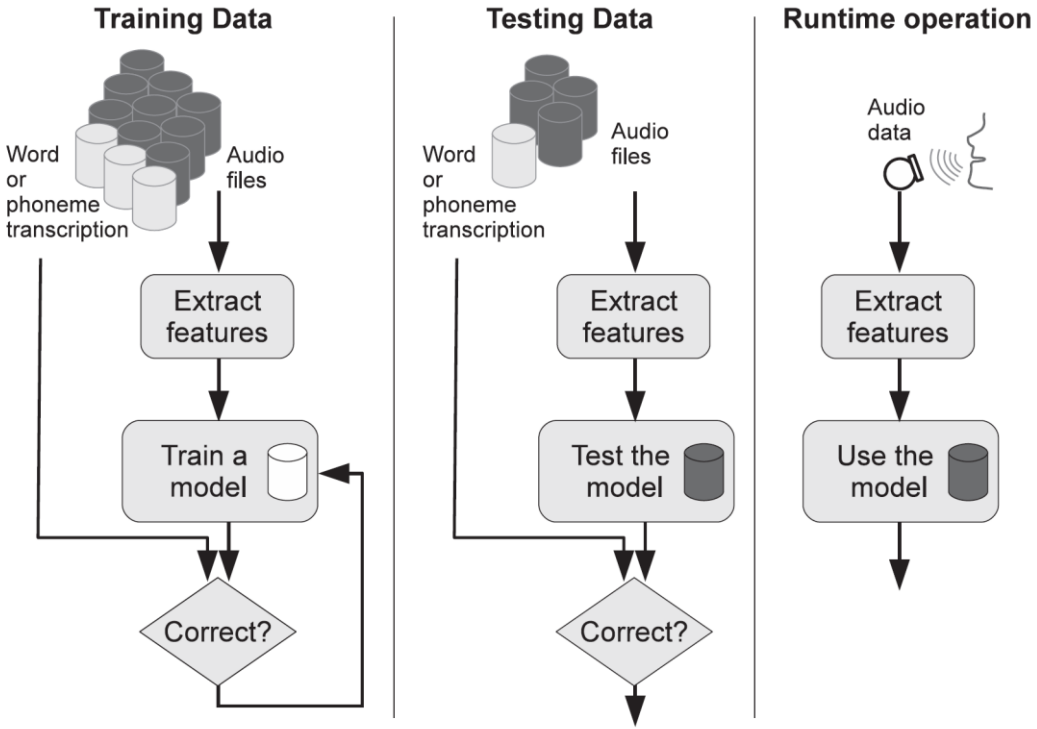


Figure 9.7 ASR phases.

Universal background model

- Le Gaussian Mixture Model (GMM) sono ampiamente usate nei moderni riconoscitori vocali, in particolare per formare quello che viene chiamato l'*Universal Background Model* (UBM), ottenendo un GMM-UBM.
- L'approccio modella ciascun segmento di voce con due componenti:
  - Un modello che include tutti gli effetti della voce e del canale, addestrato usando una elevata quantità di dati da diversi parlatori. Operazione fatta una volta sola con il modello addestrato usato poi per tanti sistemi voce. «Addestra una volta e usa tante volte»
  - Una seconda parte specifica per il parlatore o canale. Usa i dati del primo modello per un addestramento specifico per l'utente corrente o per lo scenario di operazione (microfono, rumore di fondo, caratteristiche uniche del parlatore, etc.).

Universal background model

- L'UBM viene addestrato usando dati diversi da quelli del modello del parlatore. Ci devono essere i dati di tanti parlatori per l'addestramento della UBM, mentre poco addestramento sarà richiesto per lo specifico parlatore.
- Il metodo UBM consente al parlatore di avere a disposizione con una piccola quantità di dati di training un sistema di ASR che opera altrettanto bene di un sistema da lui addestrato per centinaia di ore.

Universal background model

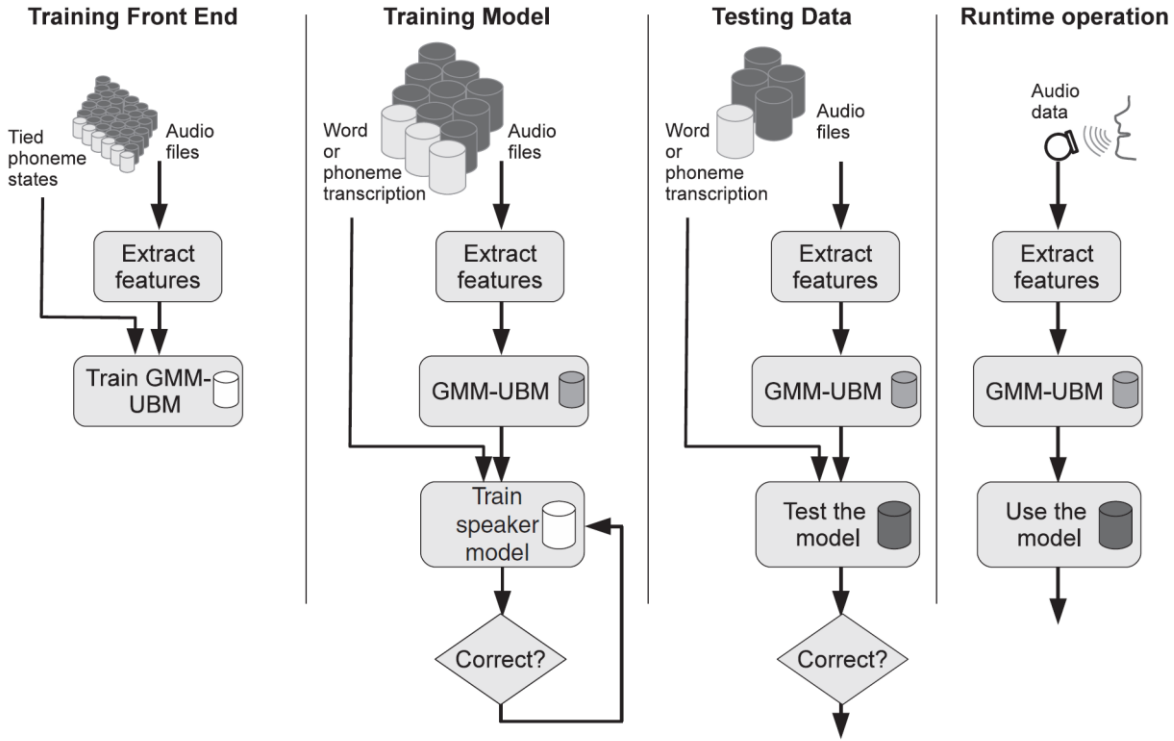


Figure 9.8 Many systems require four or more phases – where different parts of the system need to be trained using different data.

Feature

- Le feature più usate sono le MFCC, spesso insieme ai loro delta e delta-delta, cui è possibile aggiungere l'energia e la sua variazione.
- Spesso si usano 13 MFCC e quindi L={13+13+13}+{1+1}=41 elementi per frame.

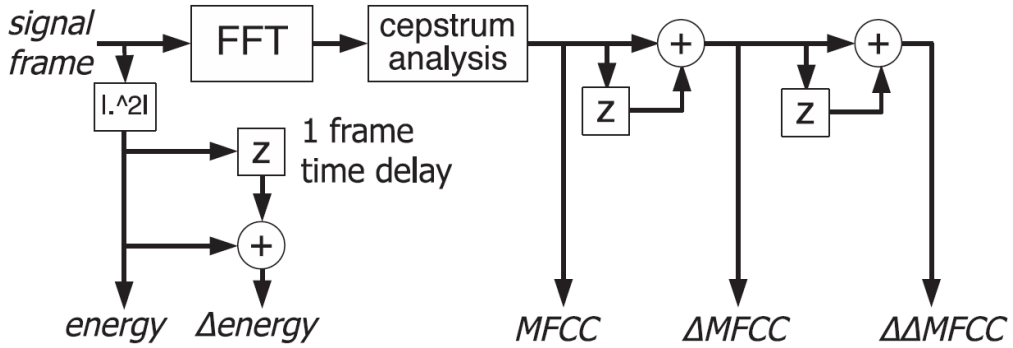


Figure 9.9 A block diagram showing the feature extraction process for energy and MFCC features, and their delays, prior to ASR input.

- Ma potremmo anche usare altre feature, come lo spettrogramma.

Feature

- In genere si lavora con frame corte con grande sovrapposizione, e.g., frame da 10ms con un overlap di 8ms.
- Siccome le frame sono corte, non c'è molta informazione per il riconoscimento e viene in genere aggiunto il **contesto**, ovvero le frame precedenti e quelle seguenti:

$$\left[\mathbf{v}^{i-c}, \mathbf{v}^{i-c+1}, \ldots, \mathbf{v}^{i-1}, \mathbf{v}^{i}, \mathbf{v}^{i+1}, \ldots, \mathbf{v}^{i+c-1}, \mathbf{v}^{i+c}\right]$$

- Il feature vector ha così (2c+1)L elementi e diviene molto grande, risultando in un sistema di ASR lento da addestrare.
- Pertanto i ricercatori usano un certo numero di metodi per ridurre la dimensione delle feature, come l'uso della Principal Component Analysis (PCA).

Variations on a theme: Transfer learning

- C’è una notevole varietà di approcci per formare i feature vector nei sistemi di ASR e c’è un ancor più ampia varietà di metodi per il riconoscimento.
- Vedremo qui brevemente il metodo di *transfer learning*.
- L’idea è di usare un insieme di buoni dati di addestramento per creare un sistema di machine learning ben addestrato e quindi trasferire questo sistema in un altro dominio.
- Questo è un problema importante: ci sono domini di applicazione che non hanno dati sufficienti per l’addestramento, mentre altri domini ne hanno vaste quantità.
- Il *transfer learning* consente al dominio debole di condividere i dati di training con il dominio più forte.
- L’approccio è molto utile per ottenere il riconoscimento vocale per lingue di minoranza, dove non si hanno molti dati per il training.

Variations on a theme: Transfer learning

- Viene fatto l'addestramento in una lingua in cui si possiedono grandi dati, e poi questo addestramento viene trasferito alla lingua di minoranza.
- Prendiamo una lingua poco parlata come quella della Cornovaglia (*cornish*): è poco probabile ci siano grandi quantitativi di dati disponibili per sviluppare un riconoscitore.
- I fonemi sono simili all'inglese. Potremmo usare un riconoscitore di fonemi addestrato sull'inglese, ma tutti quei fonemi importanti per il *cornish* e non presenti nell'inglese non verranno riconosciuti.
- Non possiamo allora usare direttamente il riconoscitore di fonemi inglesi, ma possiamo usare una tecnica tipo di *Deep Bottleneck Network* (BDN).
- La DBN agisce come un estrattore di feature. Viene ben addestrata usando un dataset di voci esteso.

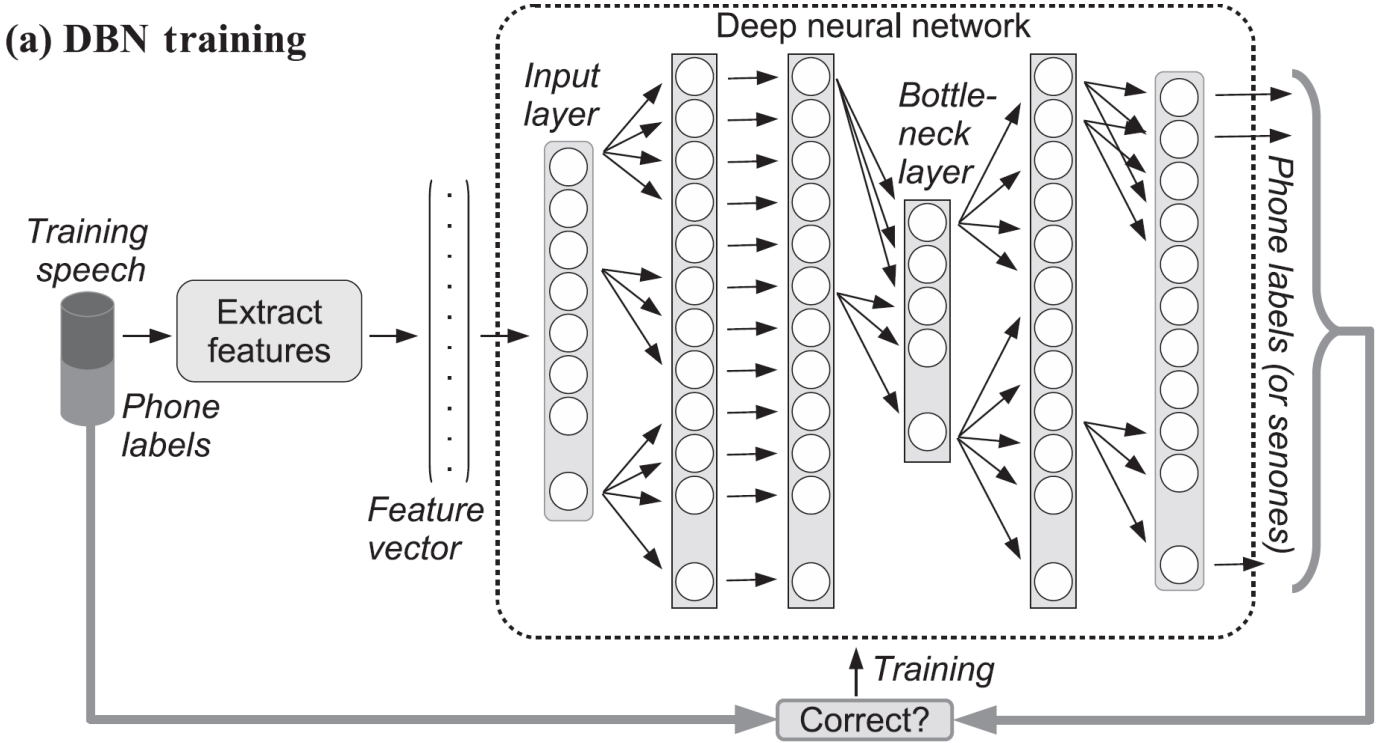
Variations on a theme: Transfer learning

- Le MFCC (o altre feature standard) più contesto vengono usate per addestrare una Deep Neural Network a molti strati per ottenere in uscita i fonemi o i tri-foni.
- La dimensione tipica dell'ingresso potrebbe avere 43 x 15 feature e in uscita potremmo avere 6000 classi.
- Una costrizione interna, detta *bottleneck* collo di bottiglia, forza tutta l'informazione fonetica a passare attraverso un vettore di più piccole dimensioni, ad esempio di 50 elementi.
- Dato un database di buona qualità, ben etichettato e in larga scala, la DNN può essere addestrata con grande accuratezza e può predire bene lo stato fonetico del feature vector.
- La natura a strati della DNN fa sì che tutta l'informazione utile alla classificazione sia concentrata nel collo di bottiglia.

Variations on a theme: Transfer learning

- Quel singolo vettore a bassa dimensionalità forma una rappresentazione completa e discriminativa dell'informazione fonetica presente nel feature vector più grande.
- Il bottleneck layer non è una rappresentazione fonetica e incorpora molta informazione acustica presente nel dominio di ingresso.
- Addestrata la DNN, gli strati dopo la bottleneck vengono rimossi e la restante parte della rete viene usata, senza cambiarne i pesi, come un estrattore di feature.
- Addestrata sulla lingua inglese sarà usata per il cornish.
- Frame di lingua cornish (rappresentati da MFCC, etc.) entreranno nella DBN.
- I vettori di uscita saranno quindi usati come ingresso per l'ASR cornish .
- La poca informazione disponibile cornish verrà usata per addestrare il back-end di ASR, addestramento che sarà facilitato dalle feature molto più discriminative ora disponibili.

Variations on a theme: Transfer learning



Variations on a theme: Transfer learning

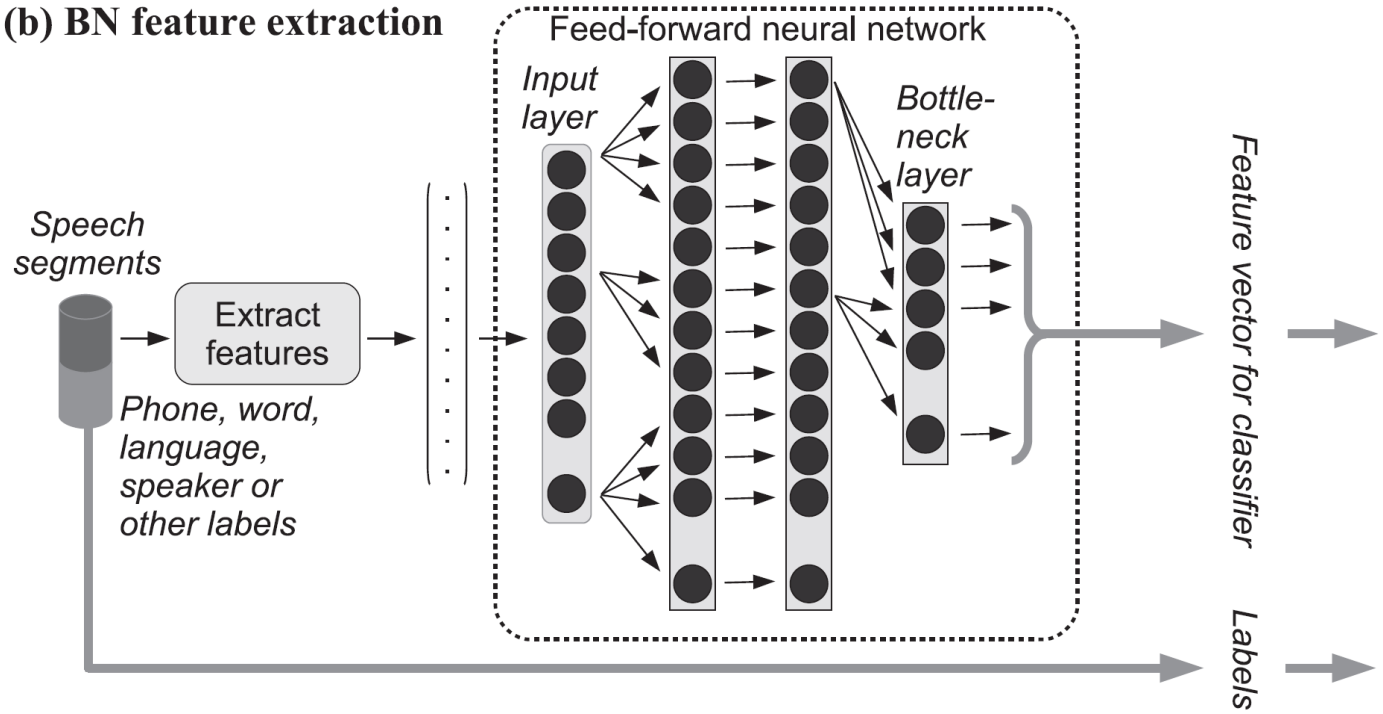


Figure 9.10 The initial training phase (above) and operating as a feature extractor (below) for a deep bottleneck network front end.

Hidden Markov Models (HMMs)

- Un modello di Markov è un tipo di macchina a stati in cui si passa attraverso una sequenza di stati, ognuno dei quali potrebbe causare qualche tipo di output.
- In ciascun istante discreto di tempo c'è la possibilità di cambiare di stato e produrre un'uscita.
- Quando si trova in un particolare stato, il modello non ricorda da quale stato è arrivato lì (è privo di memoria) e la scelta su quale stato passare è descritta da una probabilità.
- Da uno stato **i** il modello potrebbe essere in grado di passare a diversi stati (compreso lo stesso stato **i**). La somma di tutte le probabilità di transizione è 1.
- La HMM è hidden/nascosta perché un osservatore esterno non sa in quale stato si trova, ciò che conosce è solo l'uscita del modello chiamata *emission* / emissione.
- Ciascun stato è in grado di emettere qualcosa e ogni possibile emissione è descritta da una probabilità.

Hidden Markov Models (HMMs)

- Dato un particolare modello, possiamo osservare la sequenza delle uscite e possiamo cercare di determinare quale sequenza di stati è stata responsabile per quelle uscite. Normalmente non avremo certezza della sequenza degli stati, ma potremo conoscerla solo con una certa verosimiglianza.
- Oppure, se abbiamo una particolare sequenza di osservazioni (output) e un insieme di modelli, possiamo determinare quale modello è il più probabile abbia prodotto quella sequenza.
- Nell'ASR, le HMM sono usate per modellare fonemi, sillabe, parole o unità più grandi. Talvolta 3 insiemi di HMM vengono usati per modellare diverse parti del parlato:
  - Un insieme di HMM per gli stati dei tri-foni (insieme di tre fonemi),
  - Un insieme per le parole consentite,
  - Un insieme per la grammatica ad alto livello.
  - Implementano rispettivamente: il modello acustico, lessicale e sintattico.

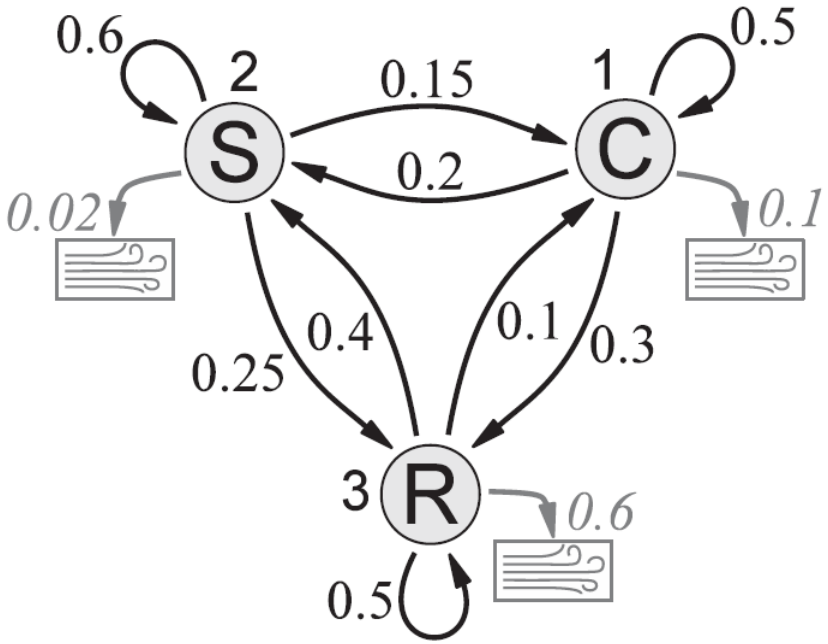
Hidden Markov Models (HMMs)

- Per implementare un riconoscitore di parole isolate, possiamo costruire una diversa HMM per ciascuna parola compresa nel training set.
- Se ci sono 10 parole costruiremo 10 HMM, una per parola.
- La sequenza osservata è costituita dalla sequenza di feature vector estratta dalla voce e il compito del riconoscitore è quello di decidere quale HMM ha più probabilmente prodotto questa osservazione.
- Siccome c'è una HMM per ciascuna parola, decidere qual è la HMM più verosimile equivale a decidere qual è la parola che più verosimilmente è stata pronunciata.

HMM example

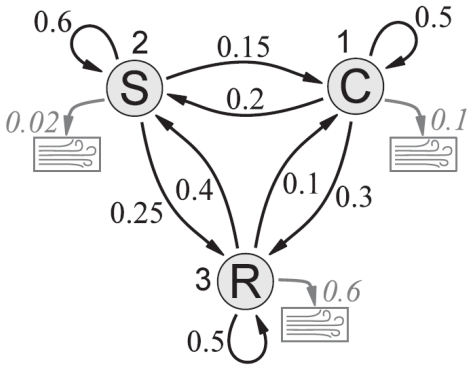
- Supponiamo ci siano tre tipi di meteo, sunny, rainy, e cloudy, e che ciascuno di questi tipi di meteo duri un giorno intero.
- Ci sono pertanto tre stati S, R, C.
- Assumiamo che il meteo ogni giorno possa essere solo in uno di questi stati ma che possa cambiare giorno per giorno e che sia più probabile il passaggio da C a R, che da S a R.
- Sappiamo che la possibilità di forti venti è collegata agli stati ed è meno probabile nello stato S e più probabile in C e R.
- Possiamo disegnare un diagramma a stati.
- Questo è un modello ergodico di Markov del primo ordine.

HMM example



HMM example

- Se oggi è *cloudy,* possiamo facilmente calcolare la probabilità che i prossimi 3 giorni siano solleggiati (ovvero C-S-S-S).
- E' il prodotto delle seguenti tre probabilità:



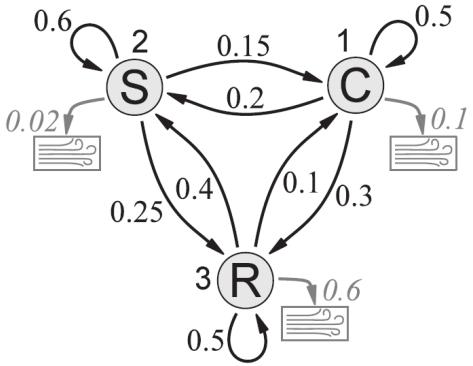
- P(day 1 is S | day 0 is C)
- P(day 2 is S | day 1 is S ∩ day 0 is C)
- P(day 3 is S | day 2 is S ∩ day 1 is S, day 0 is C)

$$0.2 \times 0.6 \times 0.6 = 0.0432, \text{ or a } 4\% \text{ chance}$$

- P(x|y) è la probabilità di x dato (o assunto) essere vero y.
- Facciamo il prodotto perché vogliamo che tutti e tre gli eventi si verifichino.

HMM example

- Le HMM codificano anche la probabilità di uscita (qui di vento forte).
- Possiamo calcolare la probabilità che ci sia vento forte oggi o domani, assumendo che oggi sia *cloudy*.
- Sarà la somma delle seguenti probabilità:



- P(wind on day 0 | day 0 is C)
- P(wind on day 1 | day 1 is C ∩ day 0 is C)
- P(wind on day 1 | day 1 is S ∩ day 0 is C)
- P(wind on day 1 | day 1 is R ∩ day 0 is C)

$$0.1 + 0.5 \times 0.1 + 0.2 \times 0.02 + 0.3 \times 0.6 = 0.334,$$

- Facciamo la somma perché ci basta sia vero uno degli eventi.

HMM example

- In realtà, dalle sole osservazioni delle uscite possiamo anche stimare in termini probabilistici lo stato del modello HMM.
- Ad esempio, sapendo che nell'ultima settimana c'è stato vento forte solo due giorni fa possiamo prevedere se oggi pioverà.

How HMMs work

- Una singola HMM è descritta da 5 elementi di informazione (S, V, π, A, B)
- N separate states denoted by $S = \{s_1, \ldots, s_N\}$.
- M possible output symbols (called a vocabulary), $V = \{v_1, \ldots, v_M\}$, each of which may be emitted by the HMM.
- N initial state probabilities π, $\{\pi_i, \ldots, \pi_N\}$, each bounded in the range $[0, 1]$, summing to 1 (to ensure that the model does actually start in some state, i.e. $\sum_{i=1}^{N} \pi_i = 1$).
- A transition probability matrix, A, comprising elements $a_{ij}, i \in S, j \in S$, which defines the probability of a transition from state j to state i in the next cycle, again bounded in the range $[0, 1]$. Since there *must* be a transition each cycle, $\sum_i a_{ij} = 1$ for each current state j.
- An emission probability matrix, B, comprising elements $b_{ij}, i \in V, j \in S$, which defines the probability of emitting symbol i when in state j.

How HMMs work

- In genere S e V sono definite strutturalmente sulla base del problema e del contenuto dei dati, mentre A, B, e π dobbiamo impararle dai dati.
- Durante la sua operazione, la HMM passa da uno stato al successivo ad ogni ciclo emettendo allo stesso tempo uno dei simboli di V.
- A differenza delle macchine a stati finiti classiche, sia le transizioni che le uscite sono definite da distribuzioni di probabilità: il sistema non segue una sequenza di stati deterministica, ma una sequenza determinata da probabilità, sequenza che possiamo conoscere solo verosimilmente.

I tre problemi delle HMMs

Problema 1:
Data una sequenza di osservazioni < x_1, x_2,… x_T> e un modello H=(A,B,π) come troviamo P(< x_1, x_2,… x_T> | H), ovvero la probabilità della sequenza dato il modello?

Problema 2:
Data una sequenza di osservazioni e il modello H=(A,B,π) come troviamo la sequenza di stati ottima secondo qualche criterio (ovvero quella che meglio spiega l'osservazione)?

Problema 3:
Come adattiamo i parametri del modello H=(A,B,π) per massimizzare la probabilità P(< x_1, x_2,… x_T> | H)?

Use of trained HMMs

- Le HMM possono essere usate per task come la classificazione o la decodifica.
- In entrambi i casi l'obiettivo è quello di spiegare una osservazione $X = < x_1 \dots x_T >$ che è una sequenza di lunghezza T di simboli di uscita appartenenti a V.
- Questa sequenza è qualcosa che abbiamo misurato o rilevato e che ora cerchiamo di capire usando una HMM. In pratica, facciamo l'ipotesi che il sistema responsabile per la sequenza *X* possa essere modellato con una HMM.
- Nella *classificazione* abbiamo diversi modelli addestrati $\mathcal{H}_1, \dots, \mathcal{H}_K$ dai quali dobbiamo trovare un modello $\mathcal{H}_k$ che ha la probabilità più grande di aver generato la sequenza.
- Da un punto di vista matematico, la probabilità di osservare la sequenza *X* a partire dal modello $\mathcal{H}$ è

$$P(< x_1 \dots x_T > | \mathcal{H})$$

Use of trained HMMs

- Vogliamo trovare la *maximum likelihood,* la probabilità massima che la sequenza derivi dal modello $\mathcal{H}_k$ esaminandoli tutti:

$$\text{class}(k) = \underset{K}{\text{argmax}}\, P(X \mid \mathcal{H}_k).$$

- La decodifica per contro usa una singola HMM e va alla ricerca della sequenza di stati $\hat{S} = < \hat{s}_1, \hat{s}_2 \ldots \hat{s}_T >$ che ha la massima probabilità di aver generato la sequenza d'uscita.

Use of trained HMMs

- Vediamo la classificazione.
- Assumiamo di aver già fatto il training di K HMMs $\mathcal{H}_k$ e che siano delle buone descrizioni del sistema che ha causato l'osservazione X.
- Lavoriamo a partire dalle osservazioni (gli stati interni sono nascosti).
- Dobbiamo scansionare la sequenza di simboli osservati e calcolare la probabilità che derivino dal modello attualmente esaminato.
- Ci sono solo due relazioni per ciascun simbolo, visto che gli stati sono privi di memoria: la relazione con il simbolo precedente e con quello successivo.
- Il calcolo delle probabilità di queste relazioni porta a due processi di scansione separati, in avanti e all'indietro.
- Lavoriamo con una scansione in avanti.

Use of trained HMMs

- Sia *s(t)* lo stato all'istante *t* e la sequenza osservata sino a questo stato sia

$$< X(1), X(2), \ldots, X(t) >$$

- La probabilità di trovarci nello stato *i* al tempo *t* osservando questa sequenza è

$$\alpha(t,i) = P\{X(1,\ldots,t) \cap s(t) = i \mid \mathcal{H}\},$$

- Possiamo calcolarla a partire dal primo stato

$$\alpha(1,i) = P\{X(1) \cap s(1) = i \mid \mathcal{H}\} = P(s(1) = i) \cdot P(X(1) \mid s(1) = i) =$$
$$= \pi_i \cdot b[X(1), i]$$

Use of trained HMMs

- Negli stati successivi possiamo calcolare la probabilità dello stato corrente assumendo che i precedenti siano noti:

$$P\{X(1,\ldots,t+1)\cap s(t+1)=i\}=P\{X(1,\ldots,t)\cap s(t+1)=i\cap X(t+1)\}$$

$$=P\{x(t+1)|s(t+1)=i\}\sum_{j=1}^{N}P\{s(t+1)=i|s(t)=j\}P\{X(1,\ldots t)|s(t)=j\}$$

$$\alpha(t+1,i)=b[X(t+1),i].\left\{\sum_{j=1}^{N}a_{ij}.\alpha(t,j)\right\}$$

- Dato il modello, la probabilità di osservare *X(1, … T)* è:

$$P(X(1,\ldots,T)\mid\mathcal{H})=\sum_{j=1}^{N}\alpha(T,j).$$

- Infine il modello con maggiore verosimiglianza sarà $\hat{H}_k=\underset{k}{\operatorname{argmax}}\,P(X\mid\mathcal{H}_k).$

Training

- Come viene fatto l'addestramento delle HMM.
- Sappiamo che una HMM è definita da 5 elementi di informazione (S, V, π, A, B).
- Ciascuno di questi deve essere accuratamente definito.
- Il numero degli stati viene scelto a priori dalla conoscenza del problema e il vocabolario è noto dalla analisi dei dati di training.
- L'addestramento della HMM deve trovare i valori opportuni di π, A, B.
- Sembra non ci sia alcun metodo analitico per trovare i valori ottimi globali, ma ci sono dei metodi che trovano iterativamente degli ottimi locali.
- L'approccio parte da una scelta iniziale ragionevole dei parametri e iterativamente si valuta l'effetto che hanno sul modello e li si aggiustano in qualche modo sino ad ottenere la convergenza.

Continuous observation densities

- L’uscita della HMM abbiamo detto è una sequenza di simboli del vocabolario V.
- Questi vanno bene per segnali d’uscita discreti, ma nella voce le uscite sono dei vettori di feature ad alta dimensione (le MFCC, Δ, ΔΔ, …)
- Siccome non si mappano bene in un vocabolario di stati finito, si lavora con uscite probabilistiche continue.
- La soluzione adottata è quella di modellare la sequenza di osservazioni con un *Gaussian mixture.*

Continuous observation densities

- Ciascun elemento in B è una combinazione di Gaussiane:

$$b(j, \mathbf{y}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{y}; \mu_{jm}, \Sigma_{jm}),$$

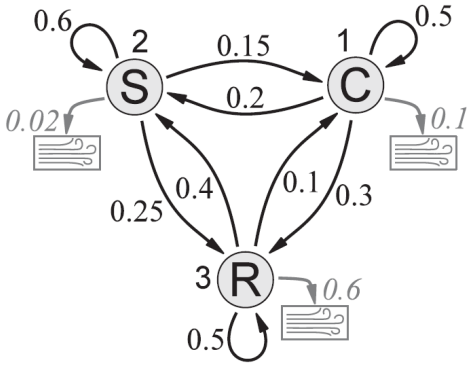
- Dove **y** è il vettore di osservazioni modellato dalla *mixture.*
- c_{jm} è la probabilità a priori dell'*m*-esima mixture nello stato *j* e

$$N(\boldsymbol{y}, \mu_{jm}, \Sigma_{jm}) = \frac{1}{\sqrt{(2\pi)^M |\Sigma_{jm}|}} e^{-\frac{1}{2}(\boldsymbol{y}-\mu_{jm})^T \Sigma_{jm}^{-1} (\boldsymbol{y}-\mu_{jm})}$$

- Si noti che i pesi devono sommare a 1: $\sum_{m=1}^{M} c_{jm} = 1$

Esempio HMM per meteo

- Nell'ultima settimana c'è stato vento forte solo due giorni fa.
- Stimiamo la probabilità del meteo odierno.



$$\pi = [0.7, 0.1, 0.2]\,, \quad \text{(C, S, R)}$$

$$A = \{a_{ij}\} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.15 & 0.6 & 0.25 \\ 0.1 & 0.4 & 0.5 \end{bmatrix},$$

$$B = \{b_{ij}\} = [0.1, 0.02, 0.6]\,.$$

$$X_T = < 0, 0, 0, 0, 1, 0, 0 >$$

$$P(X = < 0, 0, 0, 0, 1, 0, 0 > \ \cap \ \text{day 7 is R} \mid \mathcal{H}).$$

Esempio HMM per meteo

```
Pi=[0.7, 0.1, 0.2];
B=[0.1, 0.02, 0.6];
A=[0.5 0.2 0.3
  0.15 0.6 0.25
  0.1 0.4 0.5];
N=length(Pi);

X=[0 0 0 0 1 0 0];
T=length(X);
```

```
alpha=zeros(T,N);
%initial state
alpha(1,1:N)=B(:).*Pi(:);
```

Esempio HMM per meteo

```
for t=1:T-1
 for Pi=1:3
alpha(t+1,Pi)=B(Pi)*sum(A(Pi,:)*alpha(t,:)');
 end
end
```

$$\{\alpha_{ij}\} = \begin{bmatrix} 0.0700 & 0.0020 & 0.1200 \\ 0.0071 & 0.0008 & 0.0407 \\ 0.0016 & 0.0002 & 0.0128 \\ 0.0005 & 0.0001 & 0.0040 \\ 0.0001 & 0.0000 & 0.0012 \\ 0.0000 & 0.0000 & 0.0004 \\ 0.0000 & 0.0000 & 0.0001 \end{bmatrix} \rightarrow [1.4\times10^{-5}, 2.2\times10^{-6}, 1.2\times10^{-4}].$$

ASR in practice

- I sistemi di riconoscimento vocale di uso pratico tendono a condividere una stessa struttura generale di elaborazione, anche se i dettagli dell'implementazione possono variare molto.
- A partire dalla bocca del parlatore, abbiamo già considerato il canale, il trasduttore, sino ad arrivare a una sequenza di file audio salvati sul computer.
- Lavoriamo ora nel dominio digitale e consideriamo il diagramma a blocchi di un sistema generico di ASR.
- Il segnale di ingresso viene dapprima ripulito mediante un sistema di preprocessing, prima di estrarre il feature vector.
- Il preprocessing può assumere la forma di un filtraggio, probabilmente con una finestratura e una normalizzazione e qualche metodo di segmentazione. Spesso include la soppressione del rumore.

ASR in practice

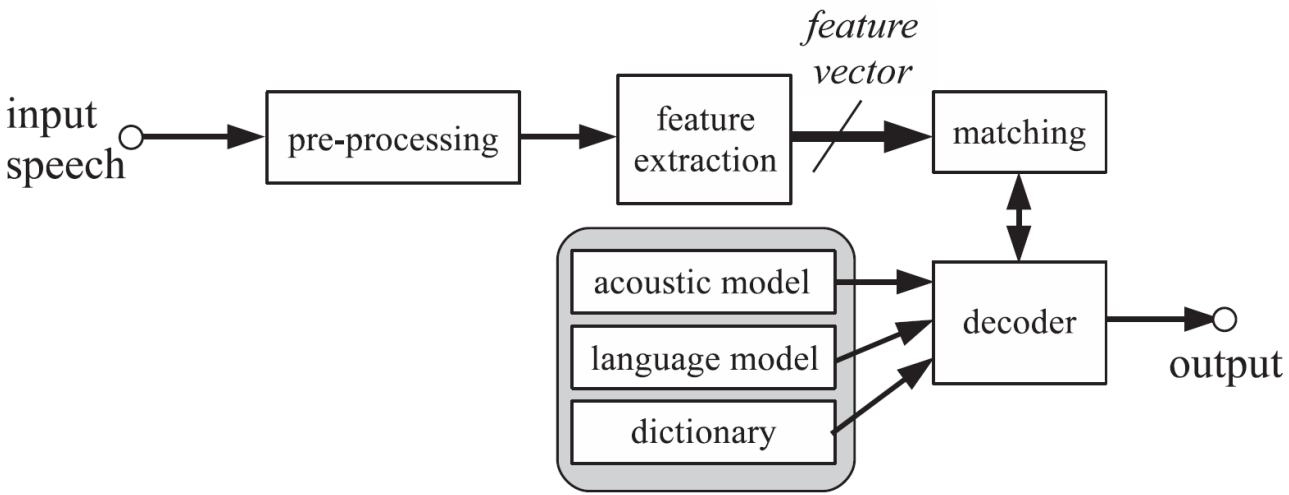


Figure 9.12 Block diagram of a generic speech recognition system, showing input speech cleaned up and filtered in a pre-processing block, feature extraction and then the matching and decoding processes driven from predefined models of the sounds, language and words being recognised.

ASR in practice

- Dopo il preprocessing, le feature sono estratte dalla voce.
- Ci sono tante feature che possono essere usate: i parametri LPC, LSP, i coefficienti cepstrali, etc., anche se i mel-frequency cepstral coefficient - MFCC sono probabilmente i più usati, insieme ai Δ, ΔΔ, all'energia e al pitch.
- Ciascuna «feature» è un vettore con decine di coefficienti calcolato su frame di 20-30 ms e aggiornato ogni 10 ms.
- Le feature sono quindi classificate da una HMM addestrata per calcolare le probabilità a posteriori dei fonemi (o più comunemente dei tri-foni).
- Un dizionario di tutte le parole e pronunce supportate dal sistema viene usato per vincolare le sequenze di fonemi in parole note. Un modello del linguaggio pesa le probabilità delle parole con lo score più alto sulla base della loro aderenza alle regole del linguaggio.

ASR in practice

- Per esempio, se si scopre che la parola con lo score più alto non è consentita sulla base della precedente parola, allora verrà rigettata in favore della seconda parola con più grande score.
- La complessità di questo processo è legata al numero di gradi di libertà e al vasto range di possibilità inerenti al linguaggio. Questo è uno dei motivi per cui il vocabolario ove possibile dovrebbe essere ristretto, ma anche il motivo per cui dovremmo cercare di minimizzare la dimensione del feature vector.
- Il modello del linguaggio considera le probabilità che la voce corrente sia correttamente riconosciuta data la conoscenza della precedente unità di voce riconosciuta. La storia può essere estesa all'indietro per più di una unità.
- Un modello di linguaggio *n-gramma* vede all'indietro le *n* unità di voce passate e le usa per calcolare la probabilità della prossima unità da un insieme pre-selezionato di un piccolo numero di migliori candidati provenienti dal modello acustico.

ASR in practice

- Questo procedimento aumenta di nuovo la complessità computazionale ma aumenta significativamente le prestazioni.
- Le unità considerate nel modello di linguaggio n-gramma potrebbero essere fonemi, parole, o simili.
- L'uscita di un sistema di ASR potrebbe essere data da una stringa di fonemi, ma è più utilmente una sequenza di parole riconosciute: la scelta dipende dalla particolare applicazione e configurazione del sistema.
- I modelli della voce, ovvero il modello acustico, di linguaggio e il dizionario sono particolarmente importanti: un suono mancante nel modello acustico, una caratteristica del linguaggio non coperta dal modello di linguaggio, e le parole che non sono nel dizionario non possono essere riconosciuti/e.

ASR in practice

- Sebbene il dizionario sia spesso creato da una lista di parole predefinita, i due altri modelli sono generalmente ottenuti con un addestramento.
- Teoricamente è possibile definirli «a mano», ma è molto più facile e accurato addestrare un sistema usando delle voci rappresentative e definire i modelli statisticamente.
- Per un sistema che opera con soggetti diversi ma noti è possibile rilevare chi sta parlando e quindi passare a un modello acustico individuale per quel parlatore (che potrà ottenere migliori risultati di un modello generale).
- Comunque i sistemi di riconoscimento continuo a grande vocabolario di oggi sono principalmente statistici e usano un grande modello per tutti i parlatori, magari con qualche forma di adattamento al parlatore (come la normalizzazione del tratto vocale – utile per lavorare con parlatori di diversa età e genere).

ASR in practice

- In modo simile, per sistemi che lavorano con diversi linguaggi o dialetti può essere utile rilevarli e cambiare in modo opportuno il vocabolario e il modello del linguaggio.

Vedere:

- Ian Vince McLoughlin, “Speech and Audio Processing”- Cambridge Univesity Press (2016)
  - Cap. 9 (da 9.1 a 9.5, no 9.4.2.3)