PART IV
Automatic Speech Recognition (ASR)

ASR: What for?

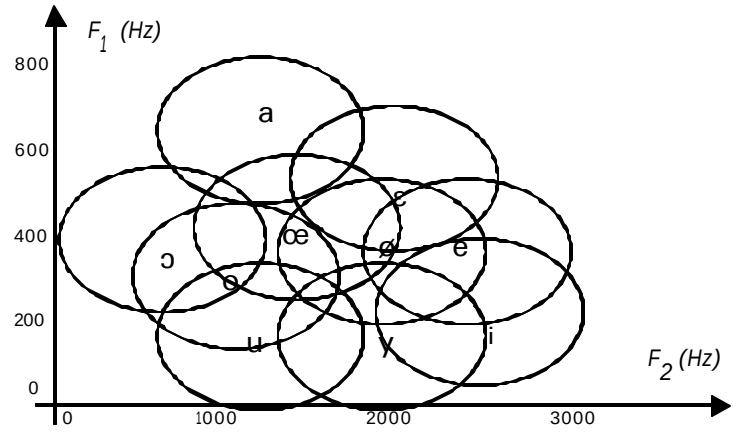
- Office/desktop:
  - voice control of PC/Workstations, of programs, dictation systems
- Manufacturing/Business:
  - aid in manufacturing process, quality control, stock control and management
- Medical/Legal:
  - creation of medical/legal reports, briefs, diagnostics...
- Others:
  - games, aid to handicapped, interactive kiosk information systems

ASR: What for?

- Telecommunications:
  - access to data or services over the telephone (e.g. AT&T's Maxwell personal telephone attendant)

Challenges

- Coarticulation!



Challenges

- Inter-**speaker** variability
  - Vocal tract, gender, dialects
- **Language** variability
  - From isolated words to continuous speech
  - Out-of-vocabulary words
- **Noise**
  - Convolutive: recording/transmission conditions, reverberation
  - Additive: recording environment, transmission SNR
  - Intra-speaker variability: stress, age, humor, Lombard effect, ...

Levels of complexity

	Isolated		*Connected*		*Continuous*	
Speaker dependent	small large	**1** **4**	small large	**4** **5**	small large	**5** **6**
Multi speaker	small large	**2** **4**	small large	**4** **5**	small large	**6** **7**
Speaker independent	small large	**3** **5**	small large	**4** **8**	small large	**5** **10**

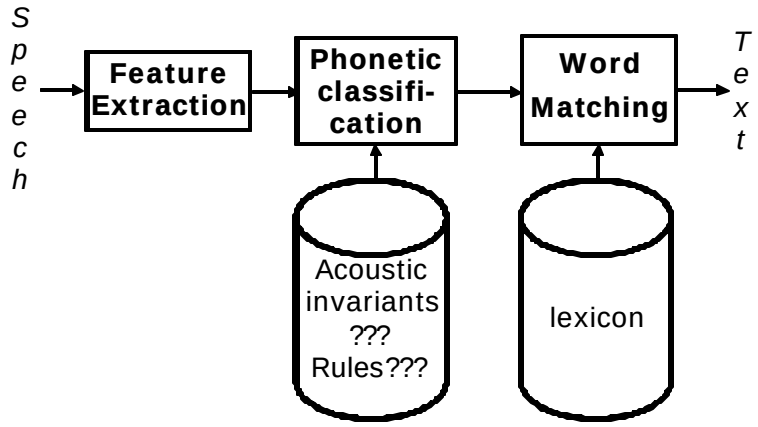
Typology of ASR systems

- **Speaker**-dependent vs. -independent
- **Language** constraints:
  - isolated word recognition
  - connected word recognition
  - keywordspotting
  - continuous speech recognition

  } + **vocabulary** : small (100), medium (5000), large (50000)

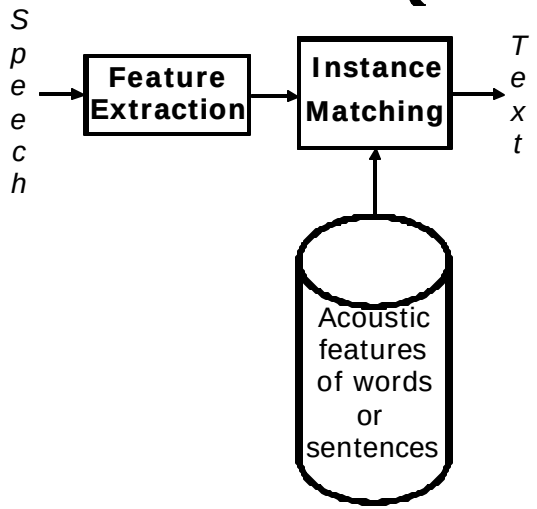
  + **perplexity**
- **Robustness** constraints
  - laboratory (office) conditions: imposed microphone, no ambiant noise
  - (quiet) telephonesystem (Δ mic., Δ noise in a given range)
  - real-life (human-like) ASR...

ASR flow-chart (60 's)



The « analytical » way of doings ASR
Very poor efficiency

ASR flow-chart (70 's)

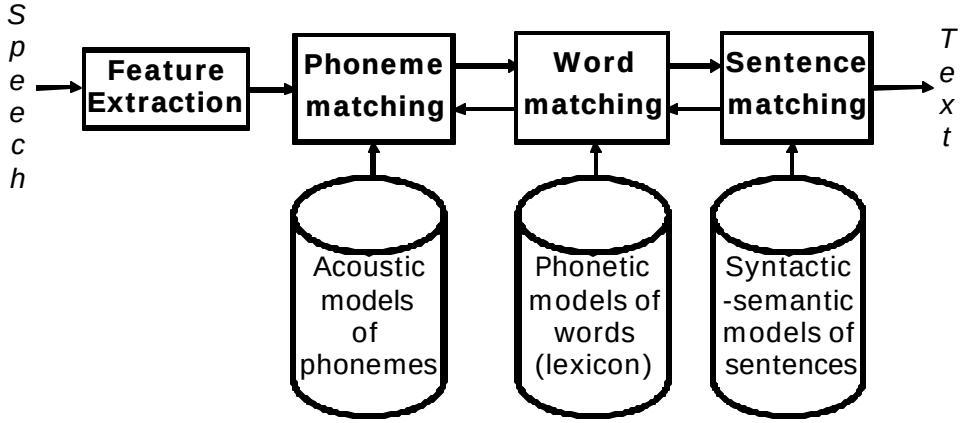


The **instance-based approach** (DTW)
OK for small vocabulary, speaker depdt

Contents

- Introduction
- **Feature extraction**
- Instance-based approach (DTW)
- Model-based approach (HMM, HMM/ANN)
  - Acoustic model
  - Phonetic model
  - Language model

Today 's ASR flow-chart

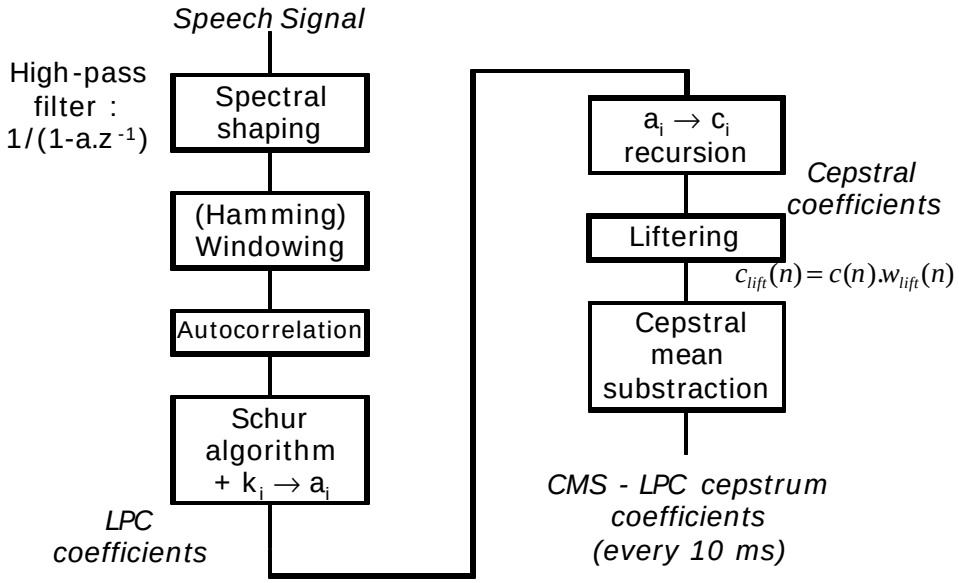


Phoneme-based approach using statistical models (HMM or HMM/ANN) for acoustics and linguistics: Large vocabulary, speaker indepdt

Speech models for ASR

- Ideal properties of parameters:
  - Invariant across the speakers for the same sounds
  - Good discrimination between speech sounds
  - Robust to noise
- Types of features used in practice:
  - LPC-based features : **cepstrum coefficients**
  - Frequency warped spectral features: idem + use a non-linear frequency axis to mimic the human auditory system. (e.g. **PLP analysis, MEL-cepstrum**).
  - Auditory features : outputs of auditory models of the cochlea and auditory nerves

CMS-LPC coefficients



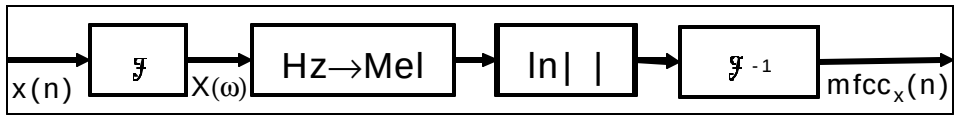
Enriching the feature set

- $c_x(n)$ can be anything for silence frames
  $\Rightarrow$ add **energy** σ_x of x(n)
- Add F_0 ? tried, without success
- Add **delta features** for better accounting for (frame-to-frame) spectral dynamics:

$$\Delta c_x^i(n) = c_x^i(n) - c_x^{i-1}(n)$$
$$\Delta \sigma_x^i(n) = \sigma_x^i(n) - \sigma_x^{i-1}(n)$$

MFCC - PLP

- Mel-Frequency Cepstrum Coefficients



- Perceptual Linear Prediction coefficients
  - apply frequency warping on the spectrum of x(n)
  - Bark scale : based on auditory critical bands (non linearly spaced in Hz; linearly spaced in Barks)
  - Apply LP model on the result

Contents

- Introduction
- Feature extraction
- **Instance-based approach (DTW)**
- Model-based approach (HMM, HMM/ANN)
  - Acoustic model
  - Phonetic model
  - Language model

Instance-based ASR

- Unknown utterance $\boldsymbol{X}=\{x_1,x_2,\ldots,x_N\}$
  (*with* $x_l=[x_{l1},x_{l2},\ldots,x_{ld}]^T$)
  Known utterances $\boldsymbol{Y^1}=\{y^1_1,y^1_2,\ldots,y^1_{J(1)}\}$
  $\boldsymbol{Y^2}=\{y^1_1,y^1_2,\ldots,y^1_{J(2)}\}$
  …
  $\boldsymbol{Y^K}=\{y^1_1,y^1_2,\ldots,y^1_{J(M)}\}$
- Compute $D(\boldsymbol{X},\boldsymbol{Y^k})$ for $k=1\ldots M$
- *Recognize:*

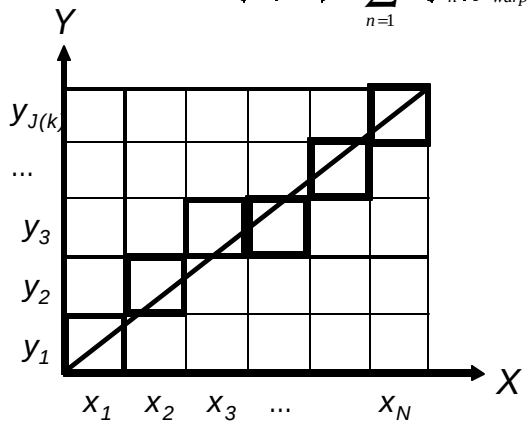
$$\boldsymbol{X}=\boldsymbol{Y^{best}}$$
with $D(\boldsymbol{X},\boldsymbol{Y^{best}})\pounds D(\boldsymbol{X},\boldsymbol{Y^k})$ for $k=1\ldots M$

- OK for **spkr-dpndt isolated word reco.**

Global distance $D(X, Y^k)$?

- Linear time warping

$$D(X,Y^k)=\sum_{n=1}^{N} d(x_n,y^k_{warp(n)})$$



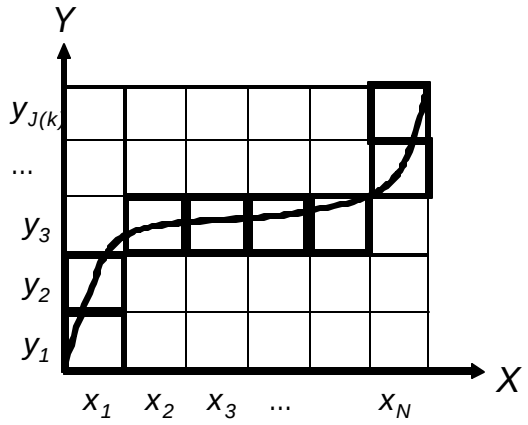
- Not realistic for long words or phrases

Global distance $D(X,Y^k)$?

- Local distance: $d(x_n,y^k_j)$?
  - Euclidian distance: $d(x_n,y^k_j)=(x_n-y^k_j)^T(x_n,y^k_j)=\sqrt{\sum_{i=1}^{d}(x_{ni}-y^k_{ji})^2}$
  - Mahalanobis distance: $d(x_n,y^k_j)=(x_n-y^k_j)^T\Sigma^{-1}(x_n,y^k_j)$
  - Itakura distance (LPC-based)
  - …

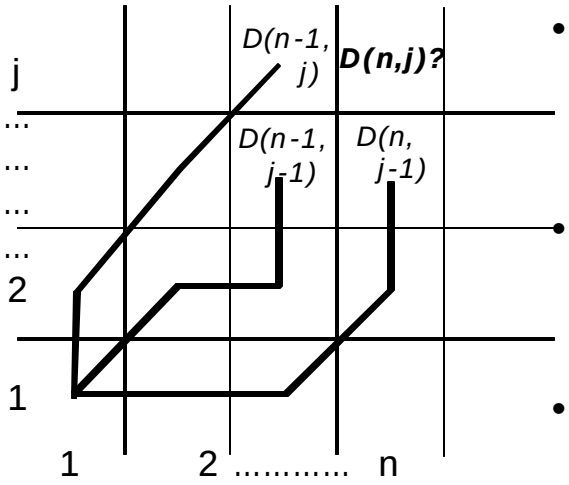
Global distance $D(X, Y^k)$?

- **Non linear** time warping



- Best path?

Dynamic Time Warping (DTW)

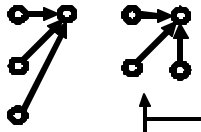


- $D(n,j)$ = accumulated distance from $(1,1)$ to (n,j)
- $D(n,j) = min\ of$
  - $d(n,j)+D(n-1,j-1)$
  - $d(n,j)+D(n,j-1)$
  - $d(n,j)+D(n-1,j)$
- $D(\boldsymbol{X},\boldsymbol{Y}^k) = D(N,J(k))$

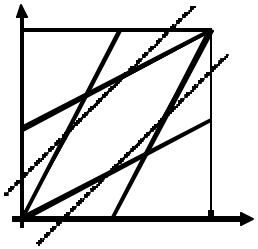
Contents

Dynamic Time Warping (DTW)

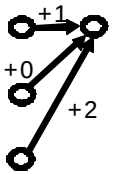
- Possible paths are constrained
  - by allowable steps; ex:



  - by additional global constraints



- Penalties can be added to steps, so as to avoid too much deviation from diagonal



Model-based ASR

- Unknown utterance

$$\boldsymbol{X} = \{x_1, x_2, \ldots, x_N\}$$

$$(with\ x_l = [x_{l1}, x_{l2}, \ldots, x_{ld}]^T)$$

Models of known utterances:

$$\boldsymbol{M_1}, \boldsymbol{M_2}, \ldots, \boldsymbol{M_J}$$

- **<u>BAYESIAN (MAP) CLASSIFICATION</u>:**
  - Compute $P(\boldsymbol{M_j}|\boldsymbol{X})$ for $j=1\ldots J$
  - *Recognize:*

$$\boldsymbol{X} = \boldsymbol{M_{best}}$$

with $P(\boldsymbol{M_{best}}|\boldsymbol{X}) \geq P(\boldsymbol{M_i}|\boldsymbol{X})$ for $j=1\ldots J$

Model-based ASR

- $P(M_j|X)$ = « **posterior** probability of M_j » not easy to compute
- Bayes rule: $$P(M_j|X) = \frac{P(X|M_j).P(M_j)}{P(X)}$$
- $P(X|M_j)$ = « **likelihood** of X »
- $P(M_j)$ = « **prior** probability of M_j »
- $P(X)$ = constant

$$\max P(M_j|X) = \max [P(X|M_j) \cdot P(M_j)]$$

Contents

Model-based ASR

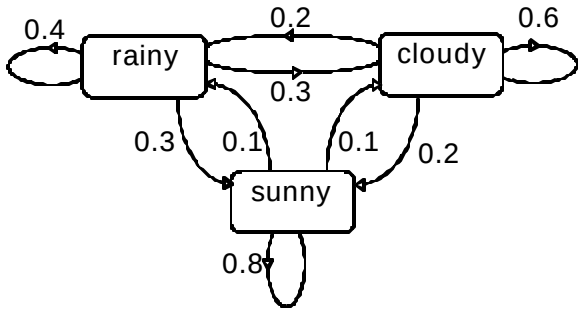
$$\max P(M|X) = \max [P(X|M) \cdot P(M)]$$

- M is a sequence of words $(W_1, W_2, .., W_K)$
- $(W_1, W_2, .., W_K)$ may have several phonetic transcriptions P_l (l=1...L)
- $P(X|M) = P(X|P_1)P(P_1|M) + P(X|P_2)P(P_2|M) + ... + P(X|P_L)P(P_L|M)$

$P(X|P_l)$ ← **Acoustic model**
$P(P_l|M)$ ← **Phonetic model**
$P(M)$ ← **Language model**

Markov chain

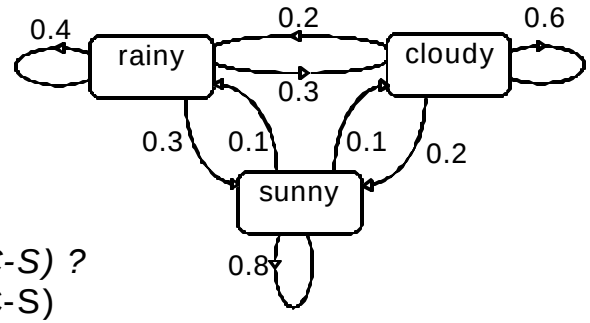
- = stochastic finite state automaton
  - ex:



- Defined by its topology (states, interconnections), and by transition probs.

Markov chain

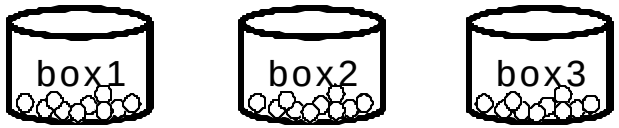
- Use?



– *P(S-S-S-R-R-S-C-S) ?*
P(S-S-S-R-R-S-C-S)
=P(S).P(S|S).P(S|S).P(R|S).P(R|R)...
=(1.0)(0.8)(0.8)(0.1)(0.4)...

- **1state « 1 observation**
Try state=phoneme and observation=x_n?
x_n is observable with several phonemes

Hidden Markov Model (HMM)

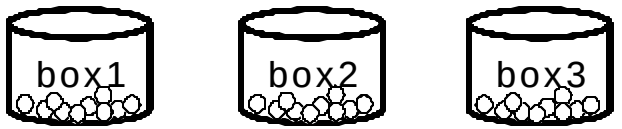


$$P(r,b,r|Model) = P(r,b,r|B1,B1,B1)\,P(B1,B1,B1) + P(r,b,r|B1,B1,B2)\,P(B1,B1,B2) + P(r,b,r|B1,B1,B3)\,P(B1,B1,B3) + P(r,b,r|B1,B2,B1)\,P(B1,B2,B1) + P(r,b,r|B1,B2,B2)\,P(B1,B2,B2) \ldots + P(r,b,r|B3,B3,B3)\,P(B3,B3,B3)$$

$$with \quad P(r,b,r|Bi,Bj,Bk) = P(r|Bi)\,P(b|Bj)\,P(r|Bk)$$

$$P(Bi,Bj,Bk) = P(Bi)\,P(Bj|Bi)\,P(Bk|Bj)$$

Hidden Markov Model (HMM)



box1	box2	box3	
P(r|B1)	*P(r|B3)*	*P(r|B3)*	**Emission probabilities**
P(b|B1)	*P(b|B3)*	*P(b|B3)*	
P(g|B1)	*P(g|B3)*	*P(g|B3)*	

P(B1|B1), P(B2|B1), P(B3|B1)
P(B1|B2), P(B2|B2), P(B3|B2)
P(B1|B3), P(B2|B3), P(B3|B3)
Transition probabilities

Double, embedded stochastic process:
-choose box using transition probs.
-choose ball using emission probs.

Hidden Markov Model (HMM)

- **Estimation** problem : *P(**X|M**)?*
O(Nstates^{Nobs}) computations???
Dynamic programming

Baum-Welch (all paths) ***Viterbi*** (best path only)

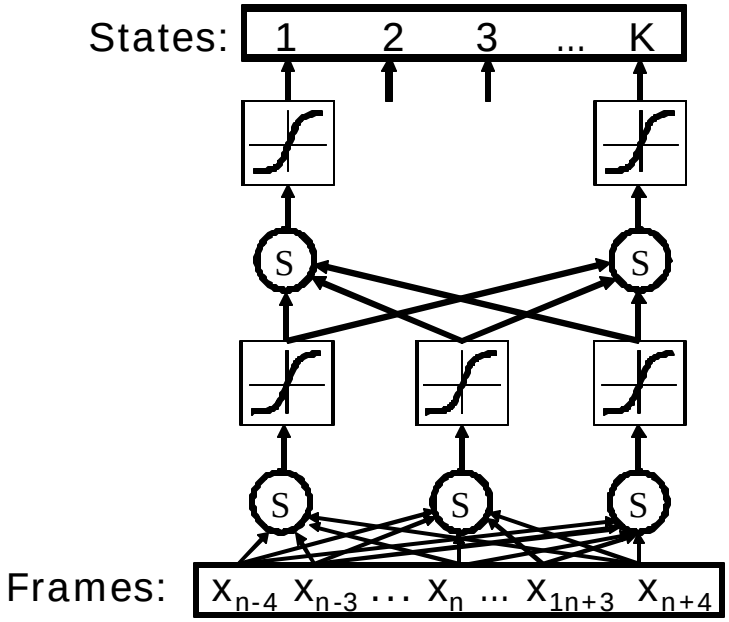
- **Training** problem: best emission and transition probs, given model topology and data
- **Decoding** problem: best sequence of states
➔ ***Viterbi*** again

Training HMMs

- Data: « brbrrbgbrbbgbggbrbgbrggbrbgg… »
- Emission and transition probabilities?
  If states were known: counting
- **EM** (expectation-maximization) **algorithm**
  - Initialize Probs. (first guess if possible): $\mathbf{M^0}$
  - Decode the data with $\mathbf{M^0}$ ➔ states
  - Re-estimate Probs. by counting: $\mathbf{M^1}$

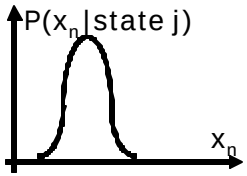
  until $\mathbf{M^j} \approx \mathbf{M^{j+1}}$

HMM/ANN hybrids for ASR

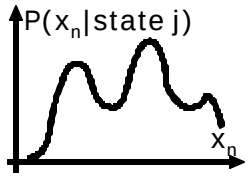


HMMs for ASR

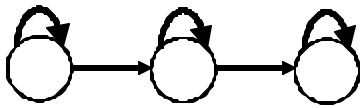
- Observation = x_n ⇨ $P(x_n|state)$ = ***continuous***
  Cannot be estimated by counting
  Estimated via the p.d.f. of a distribution
  - ex: Gaussian



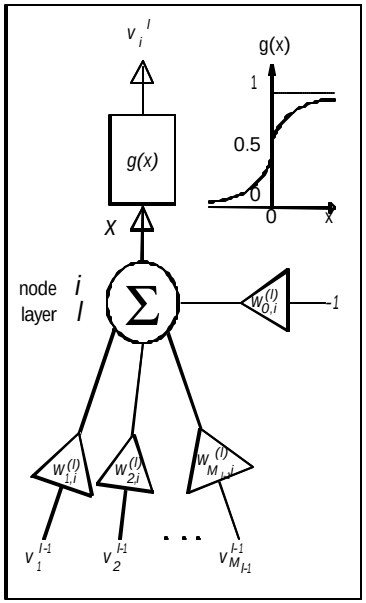
  Multi-Gaussian



- In practice, each phoneme is modeled as 3 states (*Bakis* model)



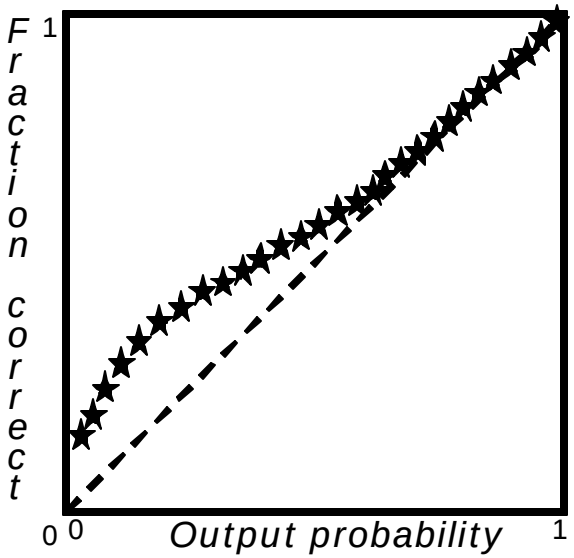
(The perceptron)



HMM/ANN hybrids for ASR

- Multi-Layer Perceptrons (MLP) can be used for estimating ***P(x$_n$\|state)***

[Bourlard & Wellekens 90]



Contents

HMM/ANN hybrids for ASR

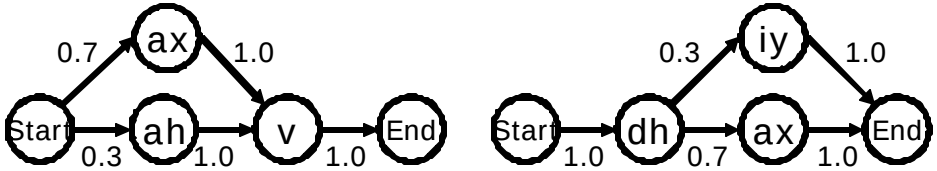
- ***P(x$_n$\|state)*** is estimated ***without any hypothesis on its p.d.f.*** (as opposed to even multi-gaussians)
- MLPs are ***discriminant*** : at training time, not only the correct output (state) is maximally selected, but also the other outputs are maximally rejected
- Less parameters to train (the same MLP is used for prediction all states)

Phonetic model

- $P(\boldsymbol{X}|\boldsymbol{M}) = P(\boldsymbol{X}|\boldsymbol{P_1})P(\boldsymbol{P_1}|\boldsymbol{M}) + P(\boldsymbol{X}|\boldsymbol{P_2})P(\boldsymbol{P_2}|\boldsymbol{M}) + \ldots + P(\boldsymbol{X}|\boldsymbol{P_L})\,P(\boldsymbol{P_L}|\boldsymbol{M})$

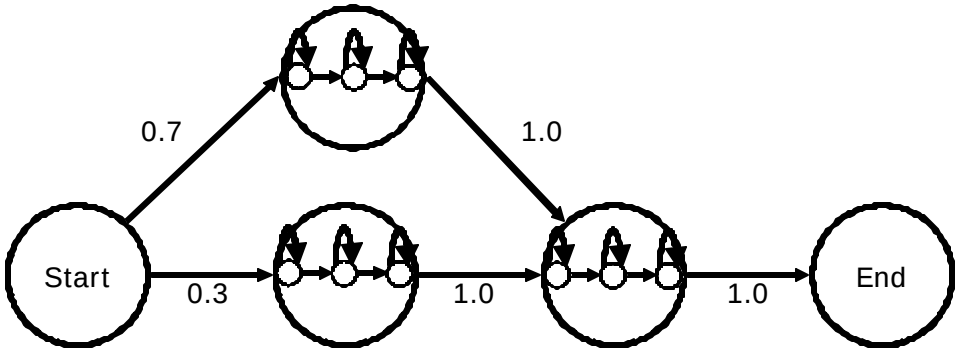
$P(\boldsymbol{P_i}|\boldsymbol{M})$**?**

- Using Markov chains: « of », « the »



Phonetic model

- Each phoneme is itself an HMM:



- Phonetic transition probabilities can be trained simultaeously with acoustic probs
- *P(**X**|**M**)* can be estimated in one shot

Contents

Language model

- *P(**M**)* is actually *P(**M**|*some language model*)*
- 3 problems (cf. acoustic models)
  - **Training** of the language model
  - **Estimation** of *P(**M**|*language model*)*
  - **Decoding**: how to integrate *P(**M**)* in the recognition process?
- Notice that:

$$P(\mathbf{M}) = P(\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_K)$$
$$= \prod_{k=1}^{K} P(\mathbf{W}_k \mid \mathbf{W}_{k-1}, \mathbf{W}_{k-2}, ..., \mathbf{W}_1)$$

word-pair model

- A sentence is admitted only if all word pairs have been encountered at least one in a very large text corpus:

$$P(\mathbf{M}) = 1 \quad \text{if} \quad \exists(\mathbf{W}_k, \mathbf{W}_{k-1}) \quad \text{for all } k$$
$$= 0 \quad \text{otherwise}$$

- Estimation and training are trivial
- Decoding: see *n*-grams
- Much too simple; banishes lots of well-formed sentences

n-gram models

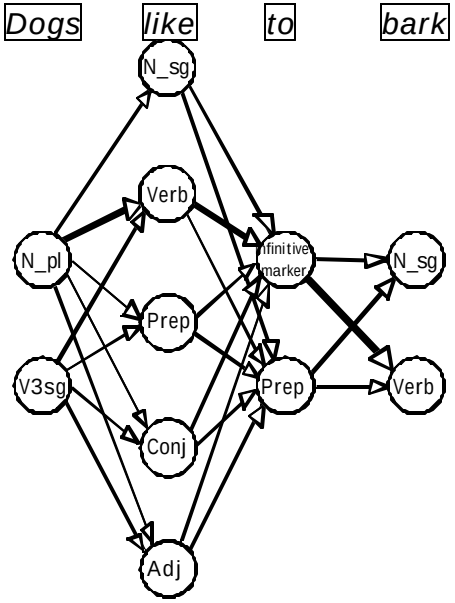
- ***Hyp:*** the probability of having a word in a sentence does not depend more on all the words of the sentence than on the *n* previous words:

$$P(\mathbf{M}) = \prod_{k=1}^{K} P(\mathbf{W}_k \mid \mathbf{W}_{k-1}, \mathbf{W}_{k-2}, \ldots, \mathbf{W}_1)$$

$$= \prod_{k=1}^{K} P(\mathbf{W}_k \mid \mathbf{W}_{k-1}, \mathbf{W}_{k-2}, \ldots, \mathbf{W}_{k-n})$$

  - ex: **estimation with a trigram** (*n*=2)***:***
    P(the weather is nice)= P(the|_,_)
    P(weather|the,_) P(is|weather,the)
    P(nice|is,weather)

Seeing words as pos



n-gram training

- $P(\mathbf{W_k}|\mathbf{W_{k-1}},..,\mathbf{W_{k-n}})$?
- Count frequency of occurrence in a very, very, very large corpus
- If vocabulary>> or *n*>1, use *smoothing techniques:*
  - **Back-off**: if $P(\mathbf{W_k}|\mathbf{W_{k-1}},..,\mathbf{W_{k-n}})$ cannot be estimated, try to use a combination of *n-i*-grams (*i*=1..*n*)
  - Use **part-of-speech** instead of words for context : $P(\mathbf{W_k}|\mathbf{W_{k-1}},..,\mathbf{W_{k-n}}) \approx P(\mathbf{W_k}|\text{pos}(\mathbf{W_{k-1}}),..,\text{pos}(\mathbf{W_{k-n}}))$
- *In practice : n<=2 :* ***trigrams***

n-gram decoding

- $\max P(\mathbf{M_j}|\mathbf{X}) = \max [P(\mathbf{X}|\mathbf{M_j}) \cdot P(\mathbf{M_j})]$
- It has been assumed that $\mathbf{M_j}$ was known, while $\mathbf{M_j}$ should obviously depend on $\mathbf{X}$ itslef! (try to recognize the most probable sentence first)
- *Depth-first* search:
  - at each frame *i*, the decoder stores on a *stack* the list of most likely word sequences up to frame *i*
  - these word sequences are tested first for the estimation of the acoustic score for frame *i+1*

n-gram results

"That this simple approach is so successful is a source of considerable irritation to me and to some of my colleagues. We have evidence that better language models are obtainable, we think we know many weaknesses of the trigram model, and yet, when we devise more or less subtle methods of improvement, we come up short."

F. Jelinek, « Up from trigrams », 1993

Today 's error rates

- Importance of the language model:
  - Ressource Management: without LM: 85% words; with LM: 97%

  ⇒ The last 3% might still be a *language* problem
- These are laboratory systems, working on read speech!
  Real life systems, spontaneous speech: -30% :-(
- Current issues :

  Robustness Spkr adaptation Language models

Conclusion

Type	*Task*	*Mode*	*Vocabulary*	*errorrate*
Isolated words	Equiprobable words	Sp. Depdt	10 digits	0%
		Sp. Indepdt	39 ascii	4.5%
		Sp. Indepdt	1109 basic English	4.3%
		Sp. Indepdt	10 digits	0.1%
		Sp. Indepdt	39 ascii	7.0%
		Sp. Indepdt	1218 names	4.7%
Connected words	Sequence of digits	Sp. Depdt	10 digits	0.1%
	id.	Sp. Indepdt	11 digits	0.2%
	Flight reservation	Sp. Depdt	129 words	0.1%
Continuous speech	Ressource management (perplexity 60)	Sp. Indepdt	991 words	3.0%
	Airline travel information system (perplexity 25)	Sp. Indepdt	1800 words	3.0%
	Wall street journal (perplexity 145)	Sp. Indepdt	20000 words	12.0%

Recognition* is not *understanding