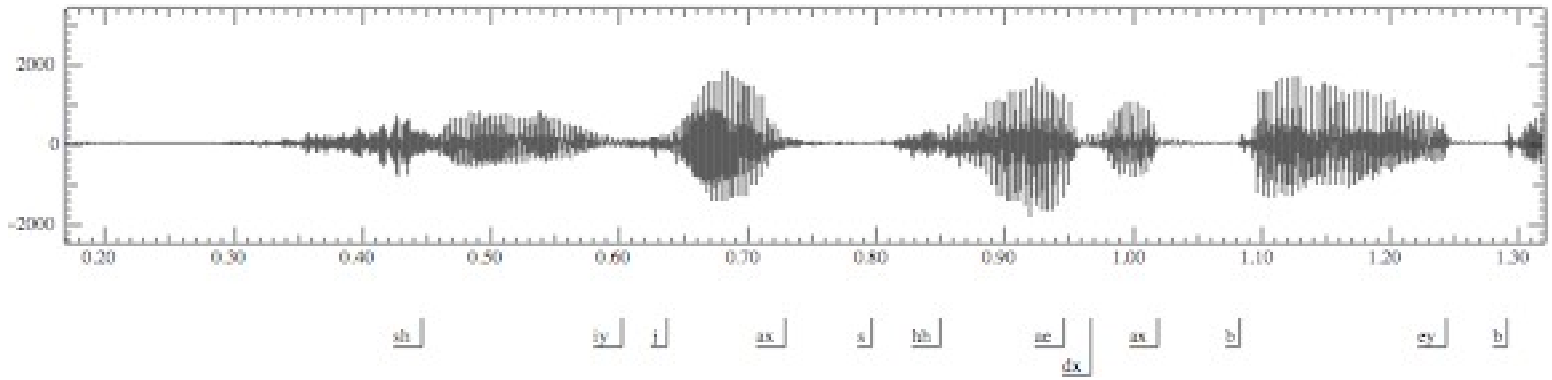


Clustering para la inicialización de HMM en RAH

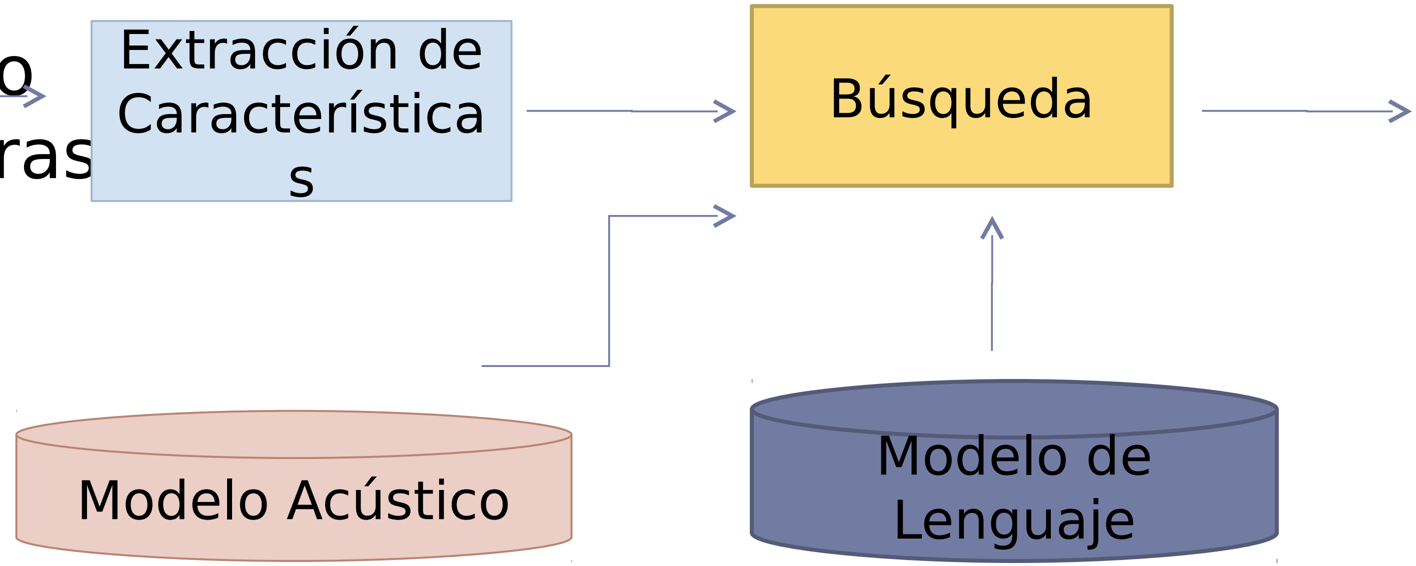
Jorge Luis Guevara Díaz

Introducción

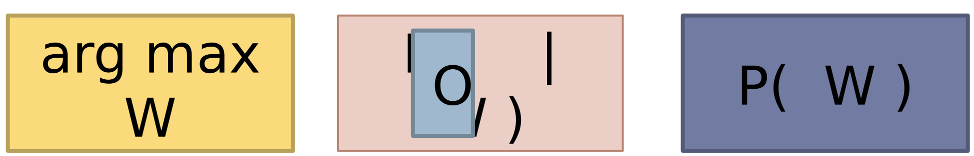
- Introducción
  - Definicion de RAH
    - Formulación
    - Arquitectura
      - Extracción de características
      - Modelo acústico
      - Modelo del lenguaje
      - Búsqueda
- Modelos Ocultos de Markov - HMM
  - Definición
  - Algoritmo forward-backward
  - Algoritmo de Viterbi
  - Algoritmo Baum-Welch
- Clustering para inicialización de HMM
  - k-means clustering
  - EM
  - SOM clustering
- Aplicación

Reconocimiento automatico del habla arquitectura

- audio palabras



- W =

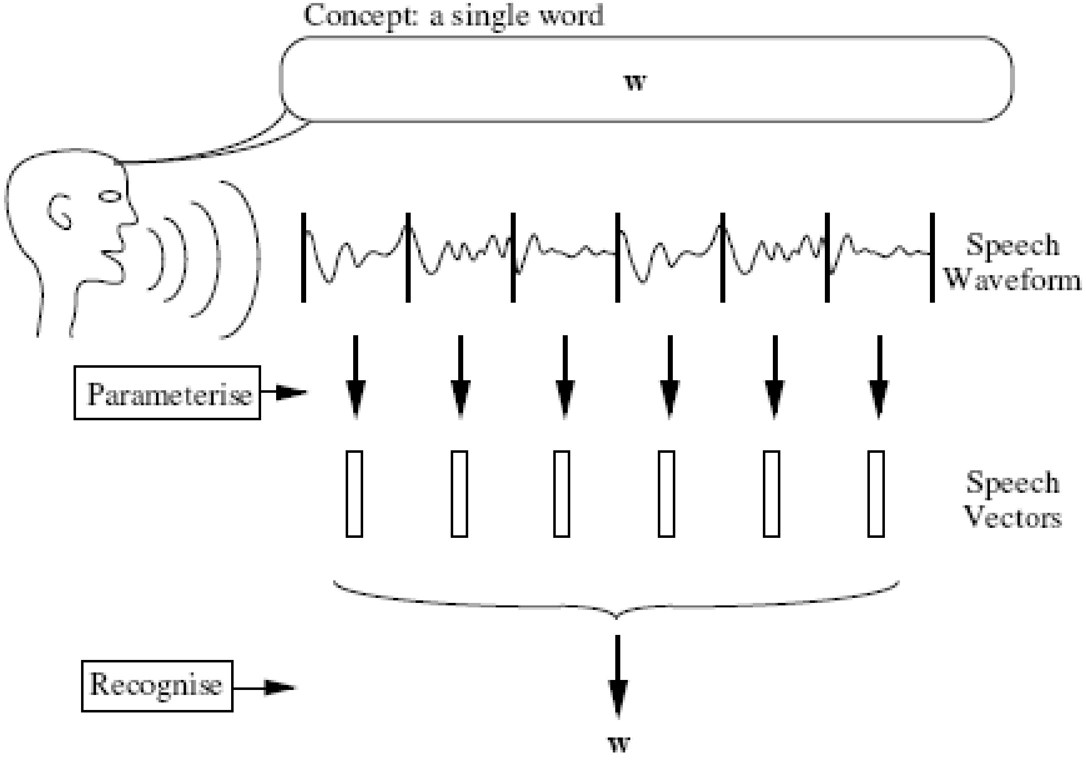


Formulacion

- Cada palabra hablada W es representada por una secuencia de vectores de habla ó observaciones O

$$O = o_1, o_2, \ldots, o_T$$

- Ot es el vector de habla en el tiempo t



Formulación

- El RAH puede ser formulado como

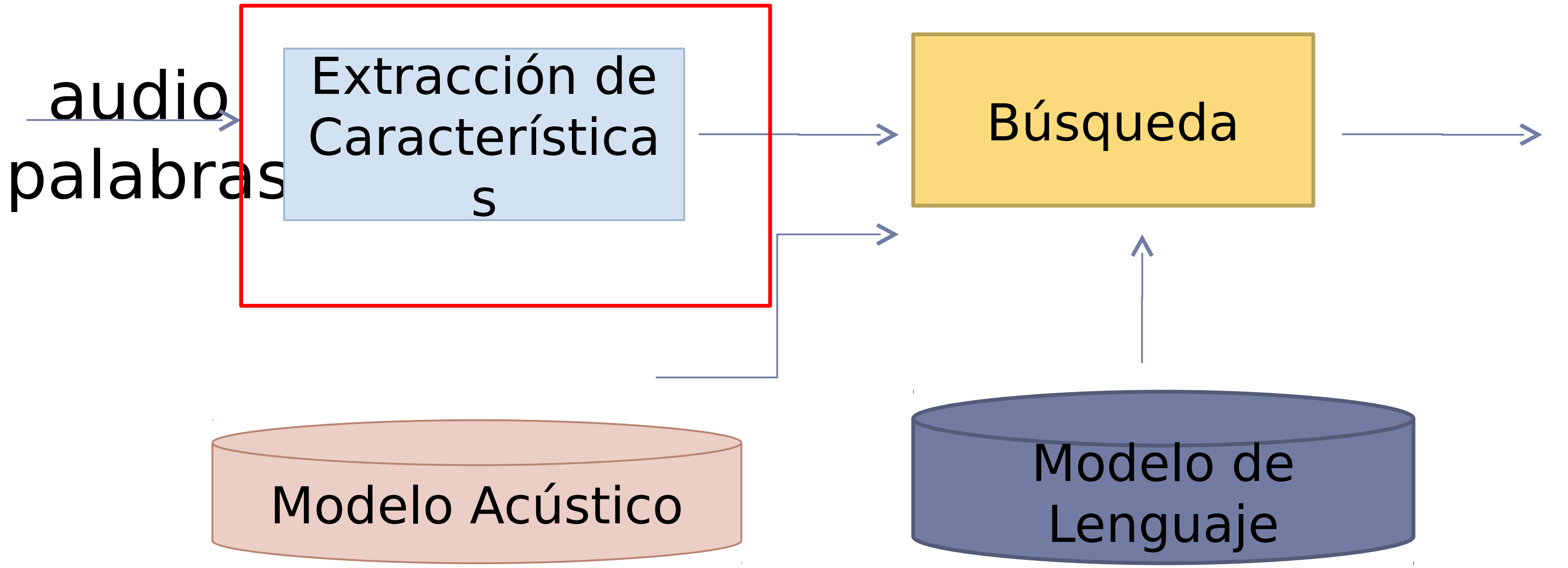
$$\arg\max_i \{P(w_i|\boldsymbol{O})\}$$

- wi es la i'ava palabra del vocabulario, esta probabilidad no es computable directamente entonces usando la regla de Bayes

$$P(w_i|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|w_i)P(w_i)}{P(\boldsymbol{O})}$$

la palabra más probable depende del likelihood

Reconocimiento automatico del habla arquitectura



- $W = \arg\max_{W} P(O|W)\,P(W)$

Extraccion de carácterísticas

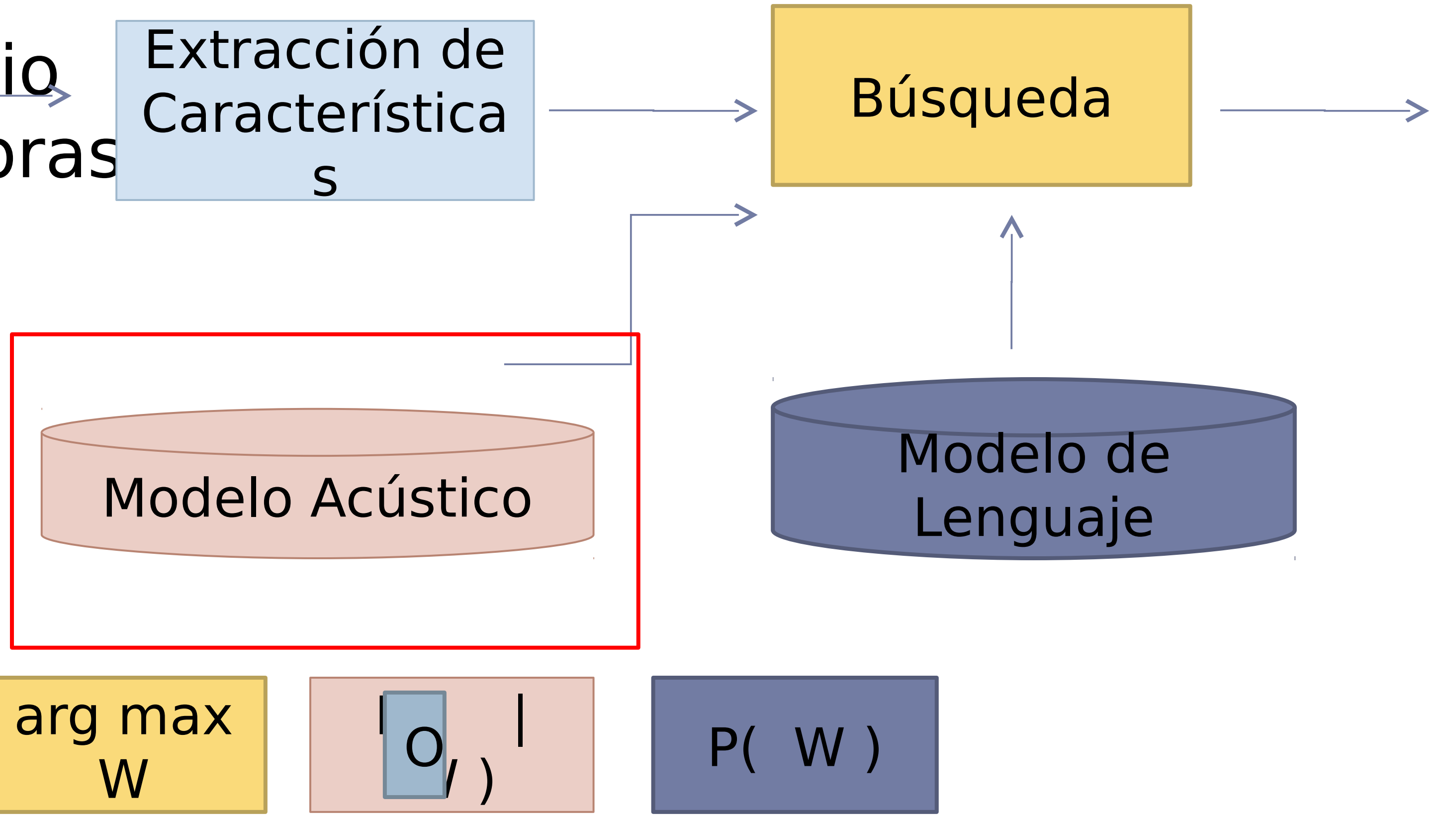
- Objetivo: dada una señal acústica de entrada obtener una codificación característica asociada para dicha señal

- Input: Señal de habla
- Output: O

- Principales algoritmos
  - MFCC
  - HFCC
  - PLP
  - LPC
  - LPC-Cepstrum
  - Basados en wavelets

Reconocimiento automatico del habla arquitectura

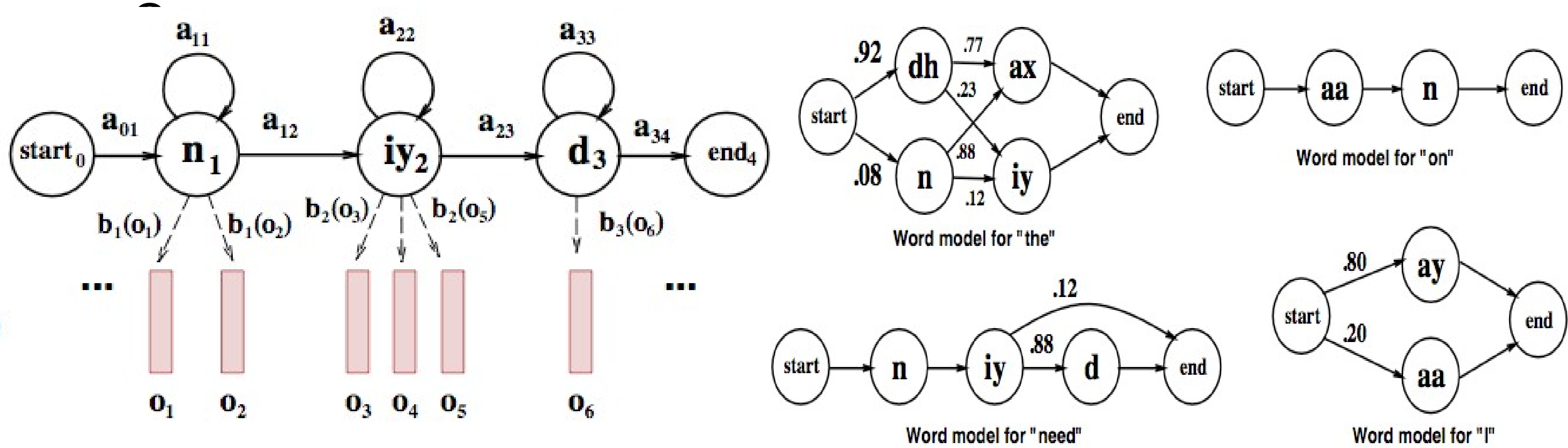
- audio palabras



- W =

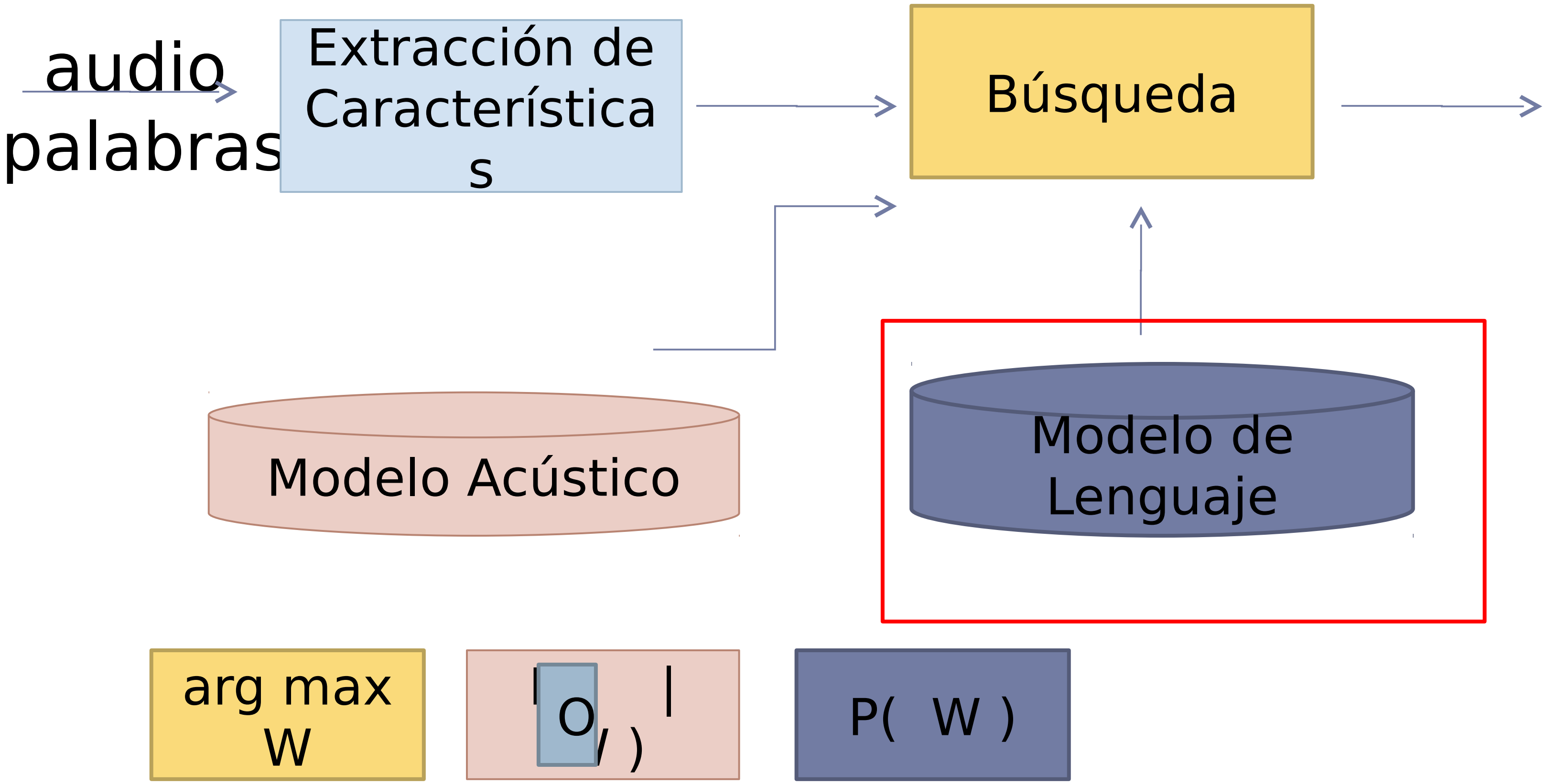
Modelo acustico

- Objetivo: construir modelos estadísticos de palabras que funcionen como generadores de las observaciones O



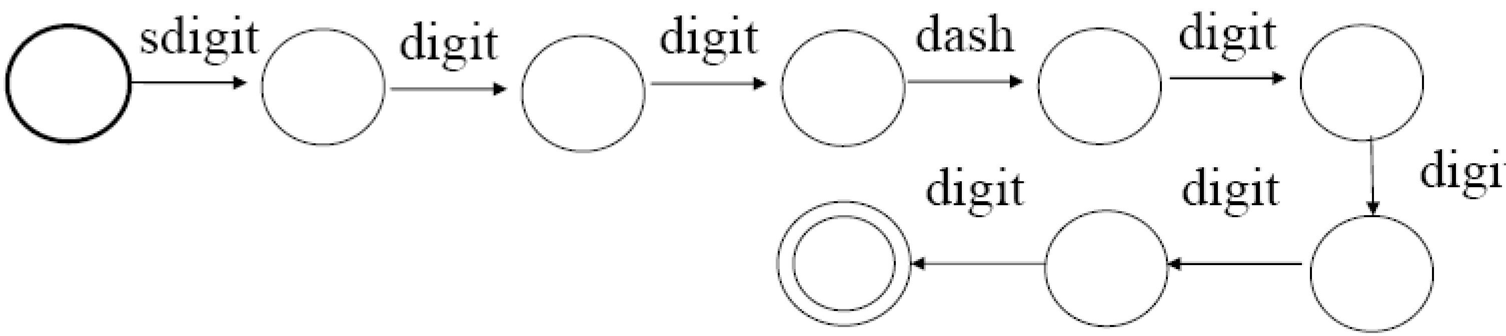
- Modelos ocultos de Markov
  - Modelos de palabras
  - Modelos de pronunciación

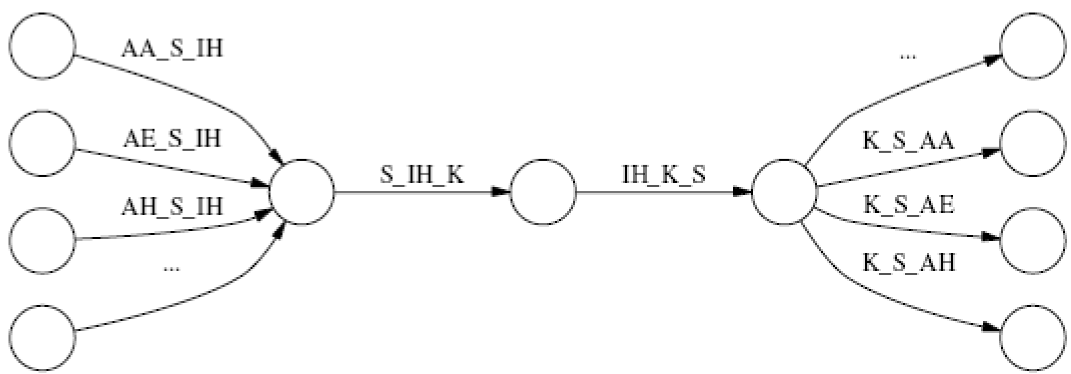
Reconocimiento automatico del habla arquitectura



- W =

Modelo de Lenguaje





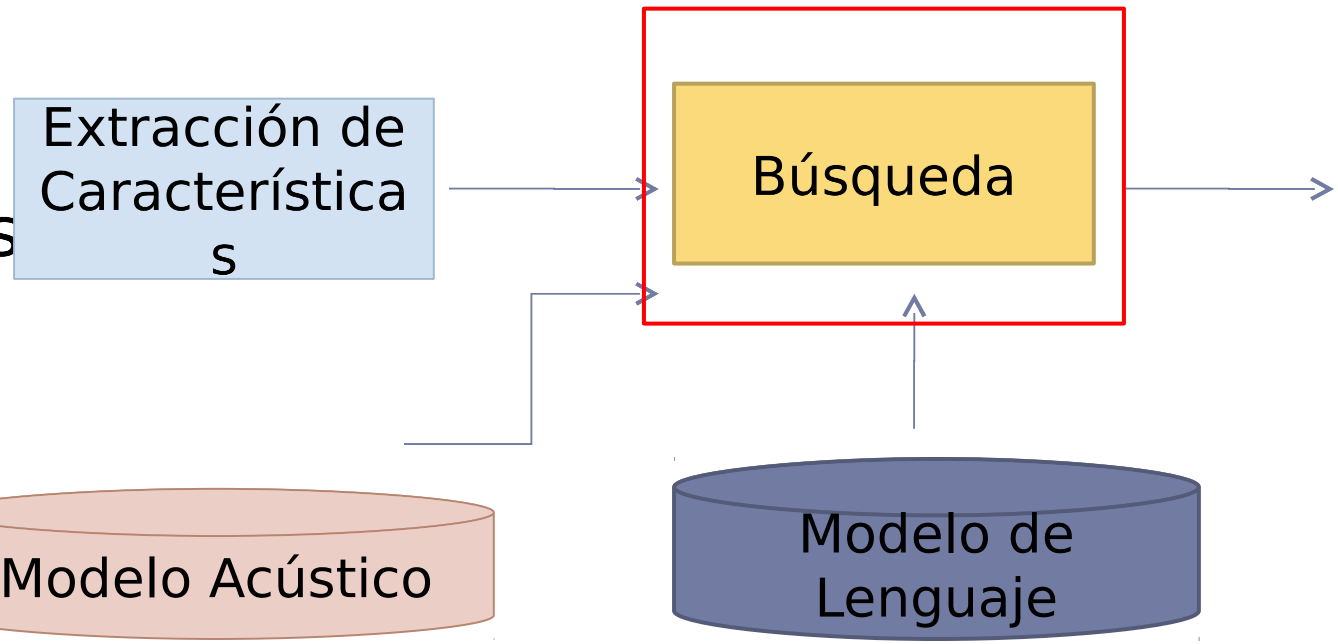
Modelo de Lenguaje

- Modela la probabilidad (likelihood) de una palabra dada la palabra(s) previa

- Conceptos:
  - Modelos n-gram:
  - Máquinas de estado finito
  - Gramáticas libres del contexto

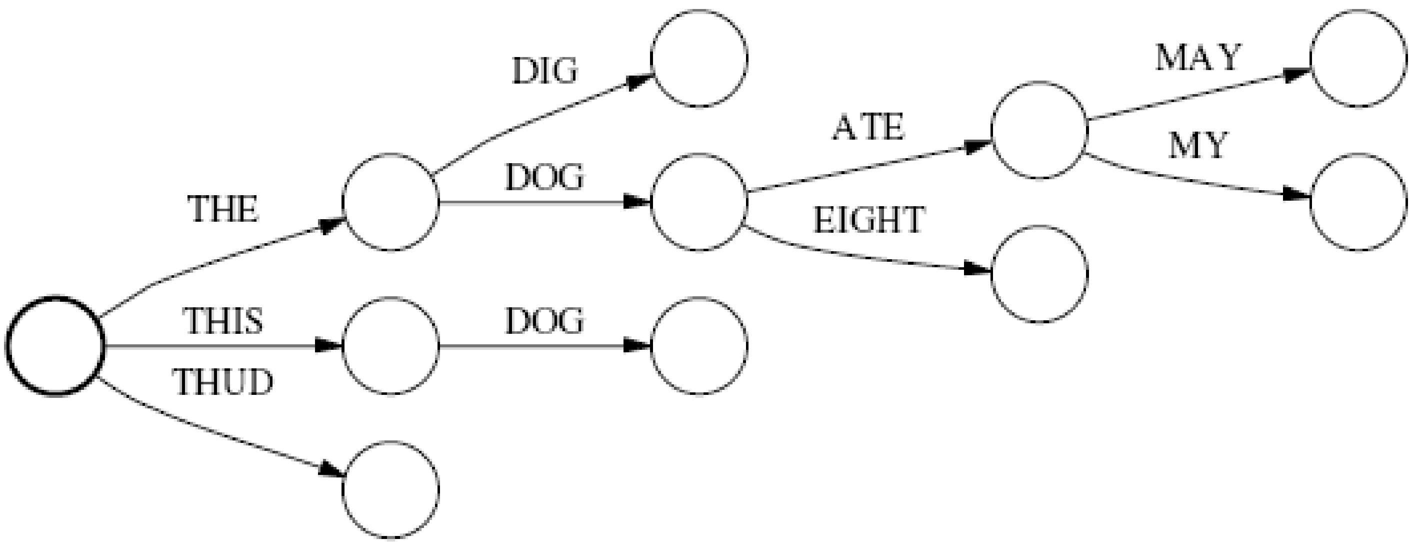
Reconocimiento automatico del habla arquitectura

- audio palabras



- $W = \arg\max_{W} P(O \mid W)\, P(W)$

Búsqueda

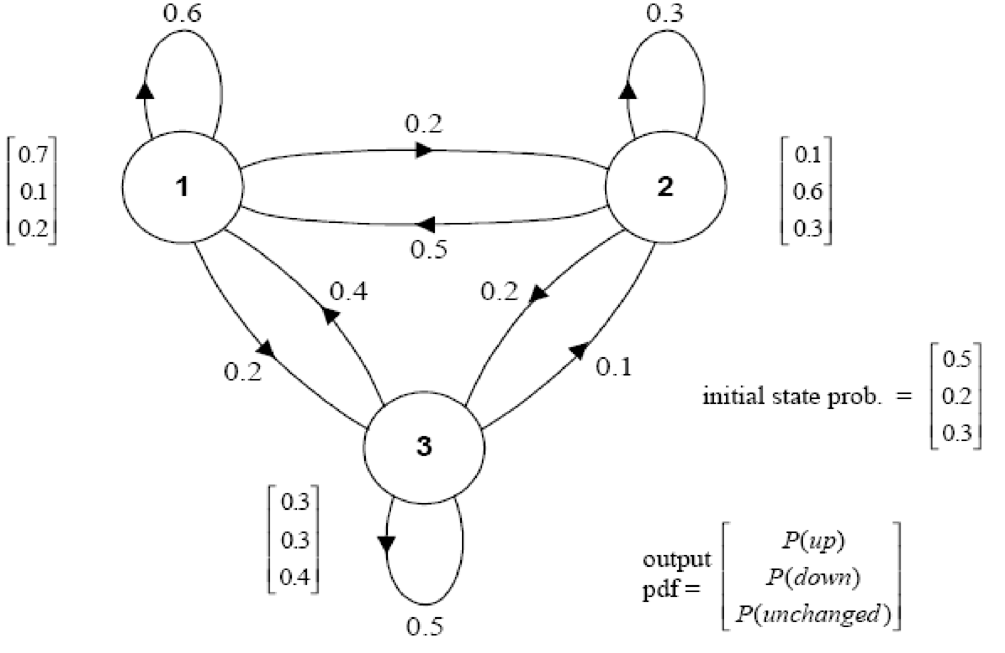


Búsqueda

- Buscar la mejor hipotesis P(O|W) P(W) dado
  - Una secuencia de vectores de caracteristicas acústicas (O)
  - Un HMM entrenado (AM)
  - Lexicon (PM)
  - Probabilidades de secuencias de palabras (LM)

- Algoritmos
  - Búsqueda local beam
  - Busqueda A*
  - etc

Modelos Ocultos de Markov - HMM



1. Problema 1 (evaluación)
2. Problema 2 (decodificación)
3. Problema 3 (aprendizaje)

Modelos Ocultos de Markov - HMM

1. **Problema 1 (evaluación)** Dada una secuencia de observación ***O = O1…OT*** y un modelo ***λ = (A,B,π)***, como calcular eficientemente ***P(O/ λ)***, la probabilidad de la observacion dado el modelo?- ***algoritmo forward***

2. **Problema 2 (decodificación)** Dada una secuencia de observación ***O = O1…OT*** y un modelo ***λ***, como escoger una correspondiente secuencia de estados ***Q = q1…qT*** que sea óptima, es decir que mejor explique la observación ***- algoritmo de Viterbi***

3. **Problema 3 (aprendizaje-entrenamiento)** Como ajustar los parámetros del modelo ***λ = (A,B,π)*** para maximizar ***P(O/ λ)*** ? ***-algoritmo forward-backward (Baum-Welch)***

- Para cada palabra del vocabulario realizar un entrenamiento para construir un modelo de palabra, esto se realiza solucionando el problema numero 3
  - ***algoritmo forward-backward (Baum-Welch)***

- Para saber cual es la secuencia de estados y hacer un refinamiento posterior, se solucionará el problema numero 2
  - ***algoritmo de Viterbi***

- El reconocimiento de una palabra se hará solucionando el problema 1 , una mejor manera es hacer el reconocimiento solucionando el problema número 2, utilizando el algoritmo de viterbi
  - ***algoritmo forward, algoritmo de Viterbi***

Solución al problema I Algoritmo *forward*

1. Inicialización

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

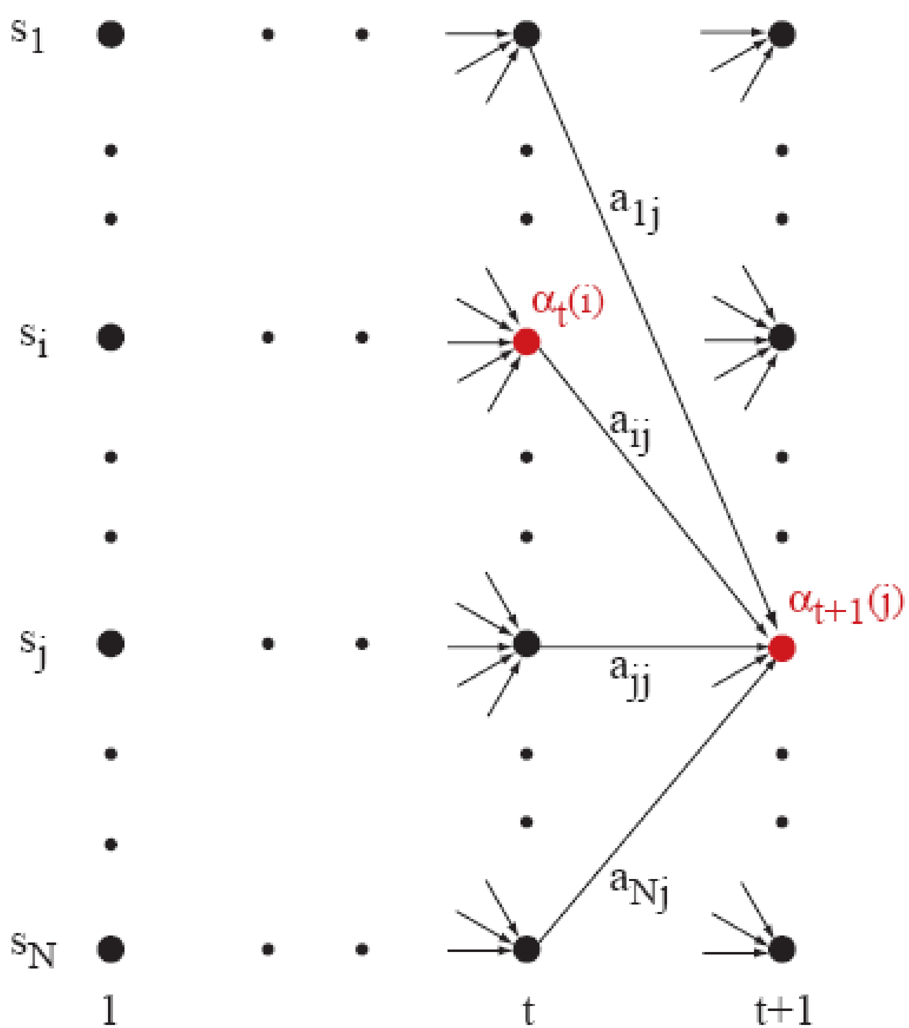
2. Inducción

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1$$

$$1 \leq j \leq N.$$

3. Terminación

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$



s1
si
sj
sN
1
t
t+1
T
a1j
aij
ajj
aNj
αt(i)
αt+1(j)

Solución al problema II Algoritmo *Viterbi*

1. Inicialización

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

$$\psi_1(i) = 0.$$

2. Inducción

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \qquad 2 \leq t \leq T$$
$$1 \leq j \leq N$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T$$
$$1 \leq j \leq N.$$

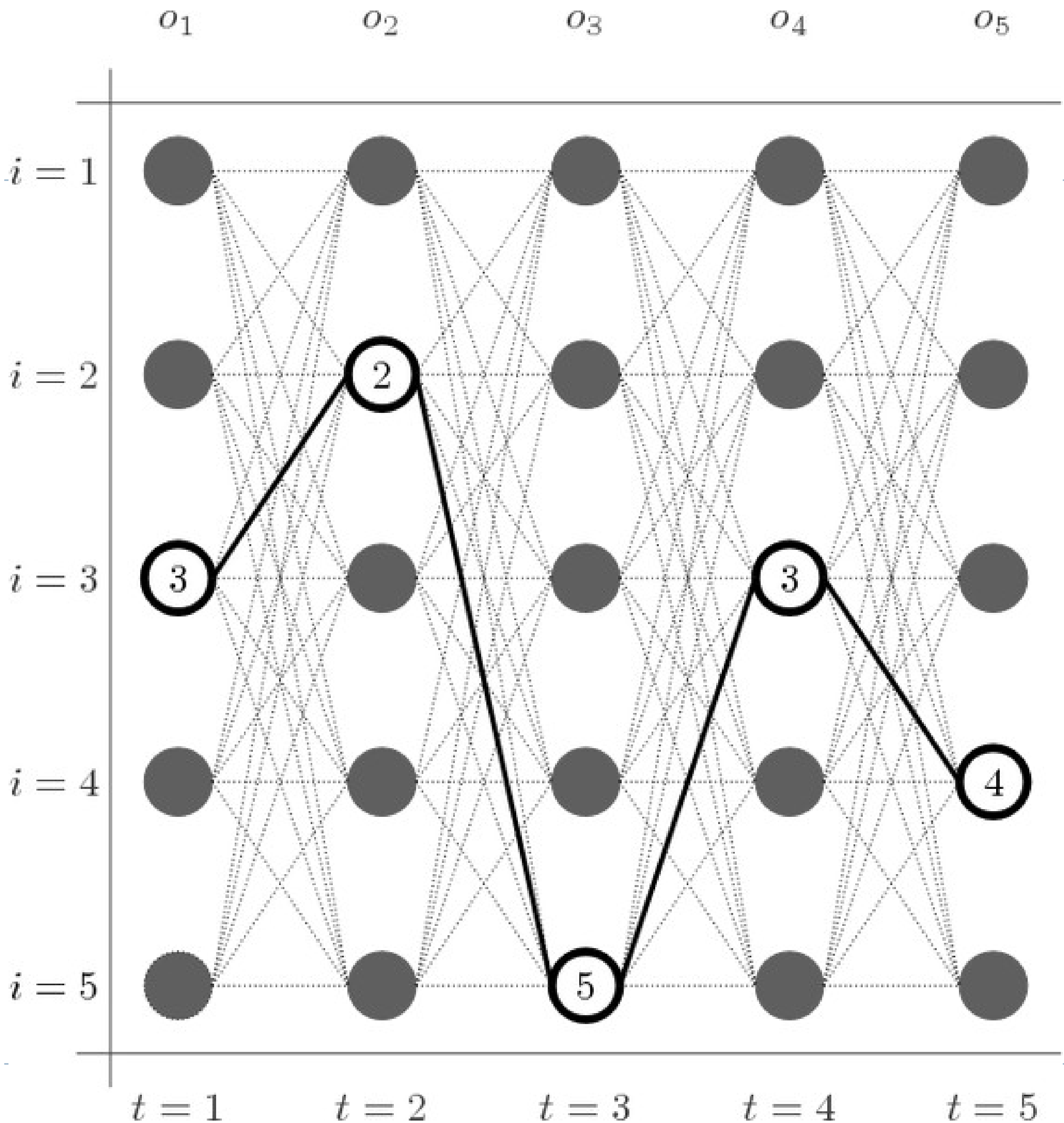
3. Terminación

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \operatorname*{argmax}_{1 \leq i \leq N} [\delta_T(i)].$$

4. Path backtraking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \cdots, 1.$$



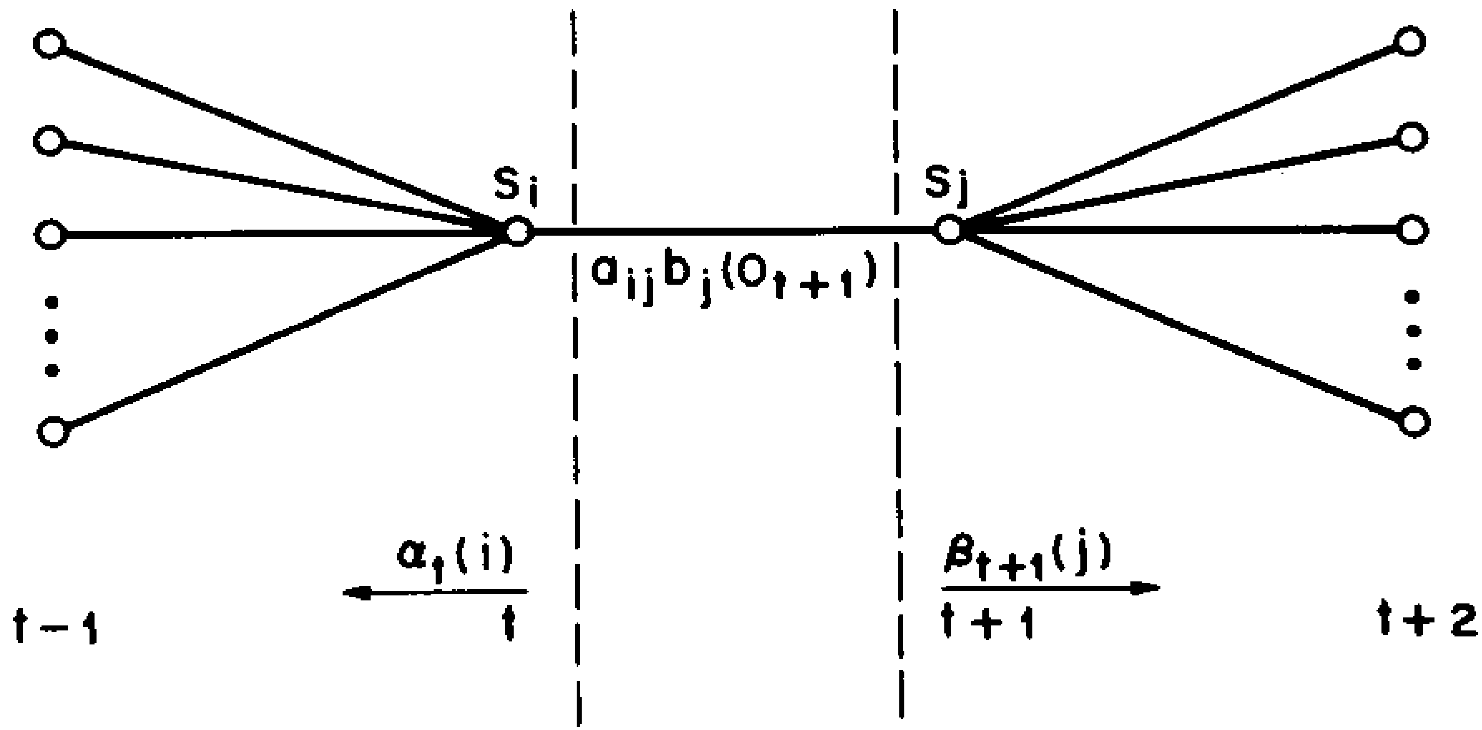
o_1 o_2 o_3 o_4 o_5
$i = 1$
$i = 2$
$i = 3$
$i = 4$
$i = 5$
3
2
5
3
4
$t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 5$

Solución al problema III Algoritmo *Baum Welch*

1. Inicializar λ=(A,B,π)
2. Calcular α, β, ξ
3. Estimar nuevo λ’=(A,B,π)
4. Remplazar λ con λ’
5. Si no converge ir a etapa 2
6. Fin

donde

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda).$$


S_i
S_j
$a_{ij}b_j(O_{t+1})$
$\alpha_t(i)$
$\beta_{t+1}(j)$
$t-1$
t
$t+1$
$t+2$

Clustering para la inicialización de un HMM

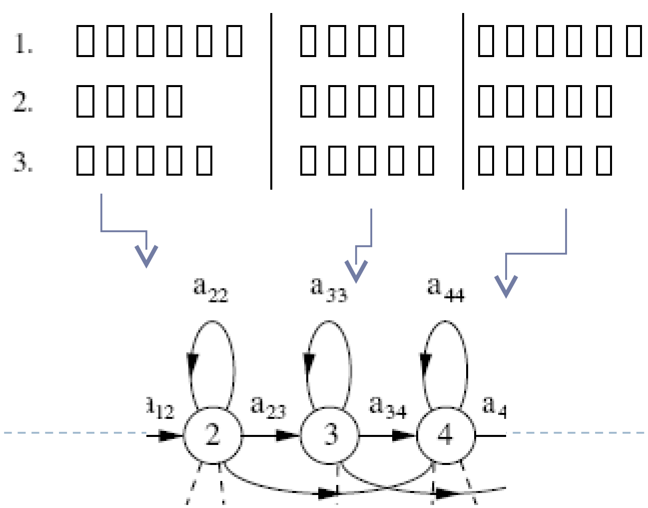
- Problema:
  - Como estimar los parámetros iniciales de un HMM, dado los vectores de características O?

- Hipótesis
  - Segmentando los vectores de características en número proporcional a los estados del HMM y aplicando posteriormente un algoritmo de clustering en cada segmento para estimar los parámetros de un modelo de mezclas de gaussianas

Clustering para la inicialización de un HMM

- Los HMM para RAH son generalmente modelos izquierda derecha para para capturar la información temporal del habla

- Una primera estimación se da segmentando cada vector de observacion en segmentos proporcionales al número de estados



Clustering para la inicialización de un HMM

- Para cada segmento aplicar un algoritmo de clusterizacion para estimar los parámetros de una distribución de mezclas de gaussianas GMM

- Algoritmos
  - K-means
  - EM (Expectation Maximization)
  - Mapa autorganizativo de Kohonen

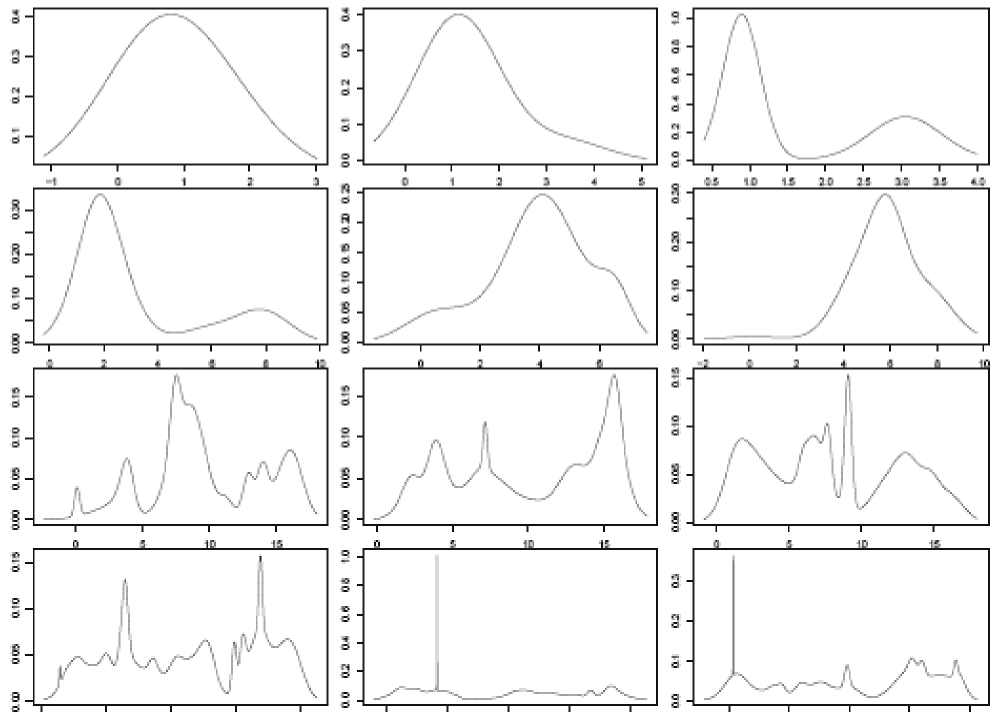
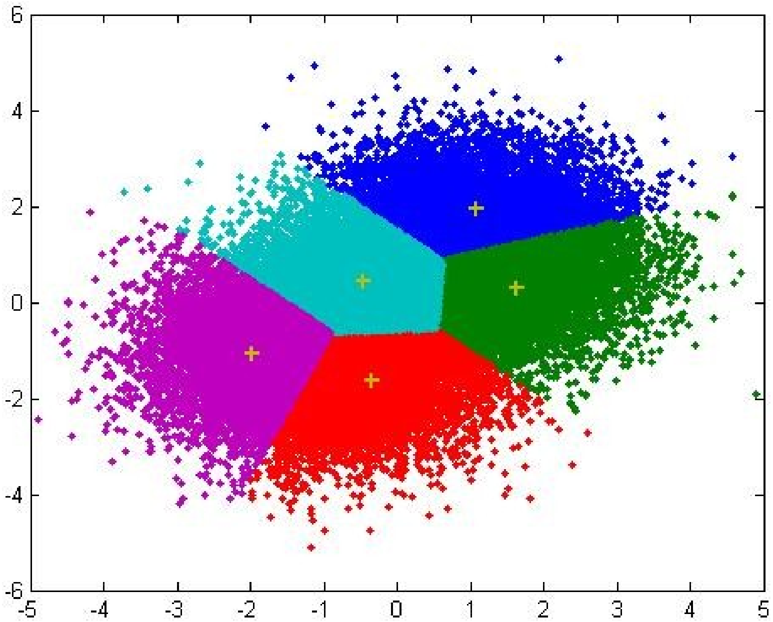


FIGURE 1. Some normal mixture densities for $K = 2$ *(first row)*, $K = 5$ *(second row)*, $K = 25$ *(third row)* and $K = 50$ *(last row)*.

Algoritmo K-means

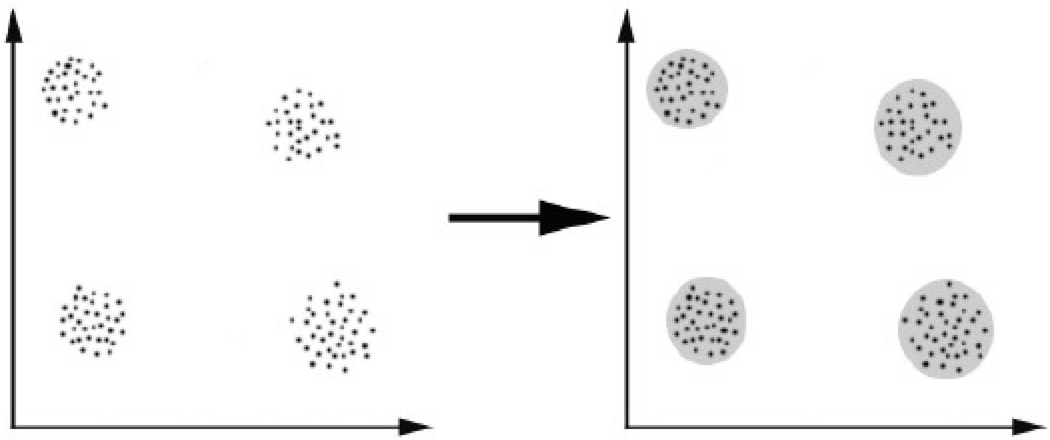


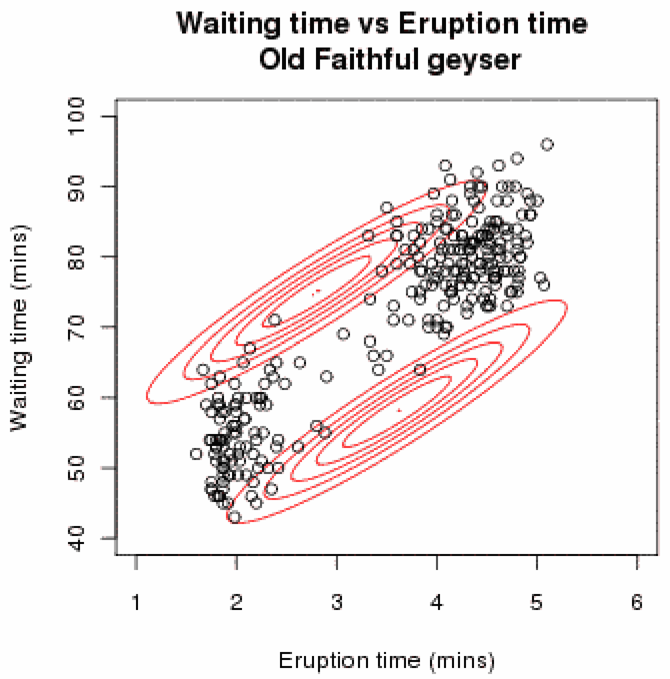
- Dado un conjunto de observaciones (**x**1, **x**2, …, **x***n*), donde cada xi es *d*-dimensional el algoritmo *k*-means clusteriza las *n* observaciones dentro de *k* conjuntos (*k* < *n*) **S**={*S*1,*S*2, …, *Sk*} teniendo algún criterio de minimización, somo sumar el cuadrado de las distancias euclidanas en los clusters

Algoritmo K-means

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

- La solución de este algoritmo es NP-hard, pero existen varios algoritmos con soluciones aproximadas



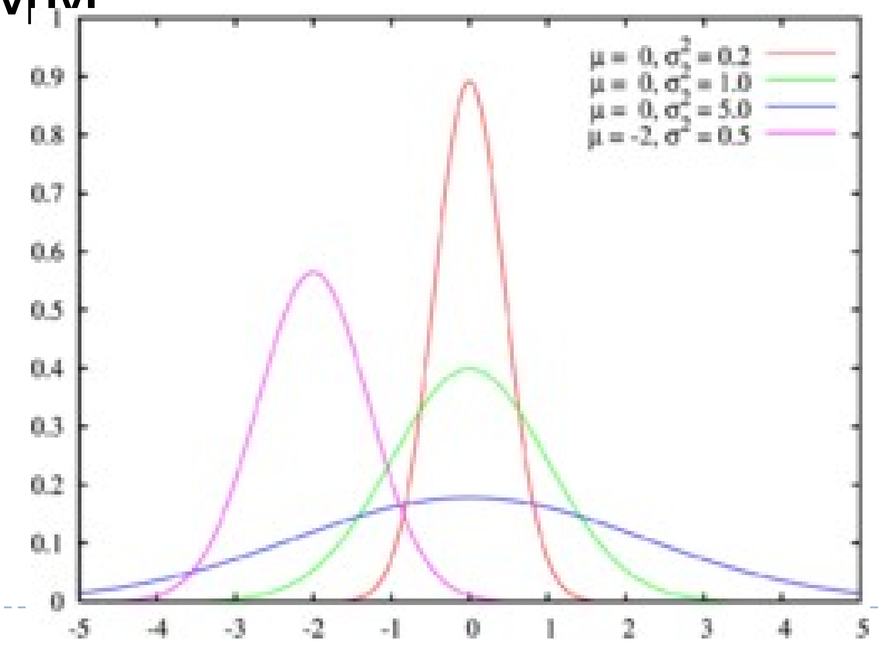


Algoritmo K-means

- Dar un conjunto inicial de *k* medias **m**1(1), …,**m***k*(1), luego alternar los dos pasos siguientes:
- Asignación
$$S_i^{(t)} = \left\{ \mathbf{x}_j : \left\| \mathbf{x}_j - \mathbf{m}_i^{(t)} \right\| \leq \left\| \mathbf{x}_j - \mathbf{m}_{i^*}^{(t)} \right\| \text{ for all } i^* = 1, \ldots, k \right\}$$
- Actualizaci´
$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

Algoritmo K-means

- Con el algoritmo K-means podemos clusterizar primeramente los segmentos, para luego obtener k medias, k matrices de covarianzas diagonales que permitan modelar k distribuciones gaussianas que forman parte de un GMM



Algoritmo K-means

- Cada gaussiana multidimesional se construye mediante la expresión

$$b_j(y) = \frac{1}{|\Sigma_j|^{1/2}(2\pi)^{K/2}} \exp\left(\frac{-(y-\mu_j)^T \Sigma_j^{-1}(y-\mu_j)}{2}\right),$$

- donde

$$\hat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} y_t ,$$

$$\hat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{\mu}_j)(y_t - \hat{\mu}_j)^T .$$

Algoritmo K-means

- Para todos los patrones de entrenamiento se tiene:

$$\overline{\boldsymbol{\mu}}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)\boldsymbol{y}_{te}}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)},$$

$$\overline{\boldsymbol{\Sigma}}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)\,(\boldsymbol{y}_{te}-\overline{\boldsymbol{\mu}}_j)(\boldsymbol{y}_{te}-\overline{\boldsymbol{\mu}}_j)^T}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)},$$

Algoritmo K-means

- Finalmente se construye un GMM (aca se tiene k=M)

$$b_j(y) = \sum_{m=1}^{M} c_{jm} b_{jm}(y) ,$$

donde

$$b_{jm}(y) = N\left(y ; \mu_{jm}, \Sigma_{jm}\right) .$$

$$\sum_{m=1}^{M} c_{jm} = 1 .$$

y

$$\bar{c}_{jm} = \frac{n_{jm}}{n_j} ,$$

Aplicación

- La aplicación se encuentra en fase experimental y es parte de un sistema de recuperación de información en textos hablados

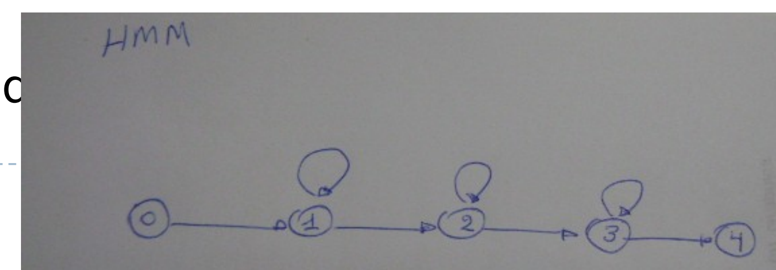
- Ejemplo
  - //observaciones
    - |1| |2| |3| |4| |5| |6| |7|

|1| |2| |3| |4| |5| |6| |7|

    - |10| |20| |30| |40| |50| |60| |70| |80| |90| |100|

|10| |20| |30| |40| |50| |60| |70| |80| |90| |100|

HMM de 5 estad ... termedios



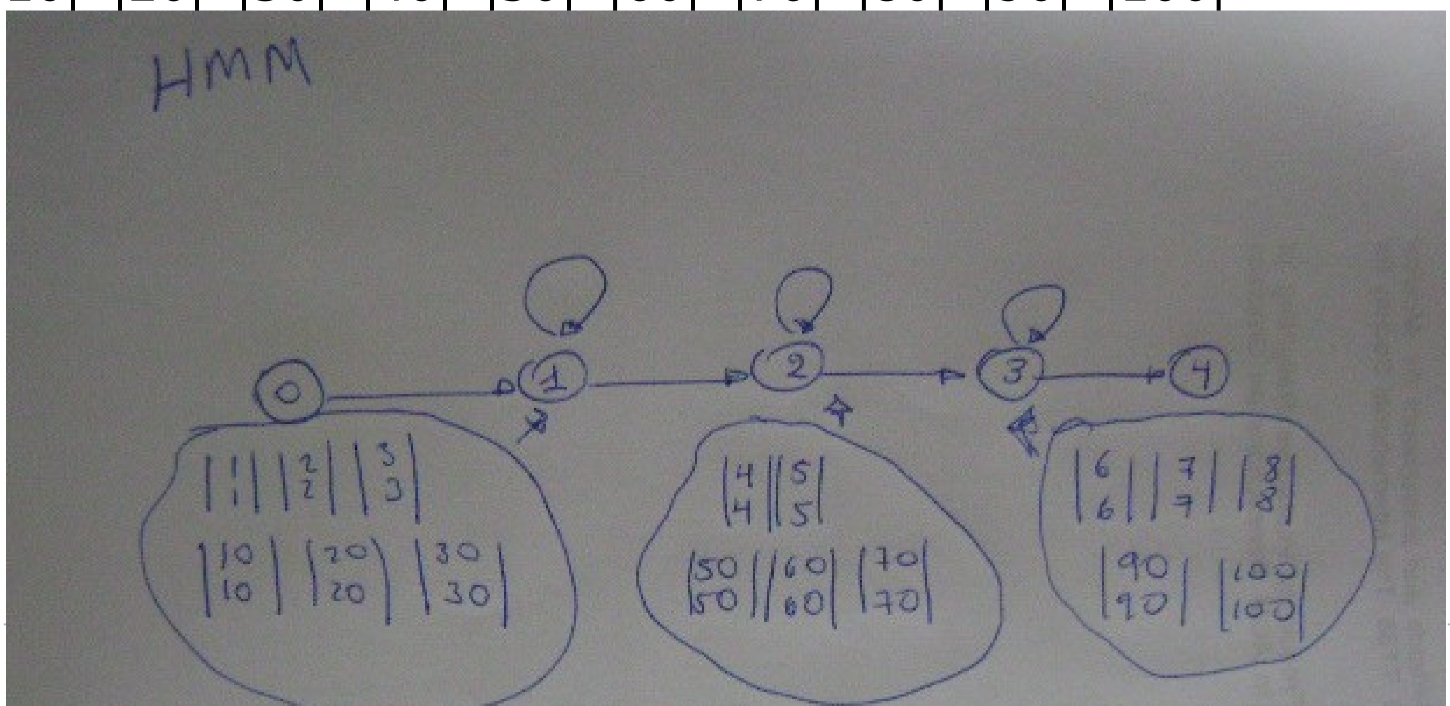
Resultados

- //observaciones
  - |1| |2| |3| |4| |5| |6| |7|

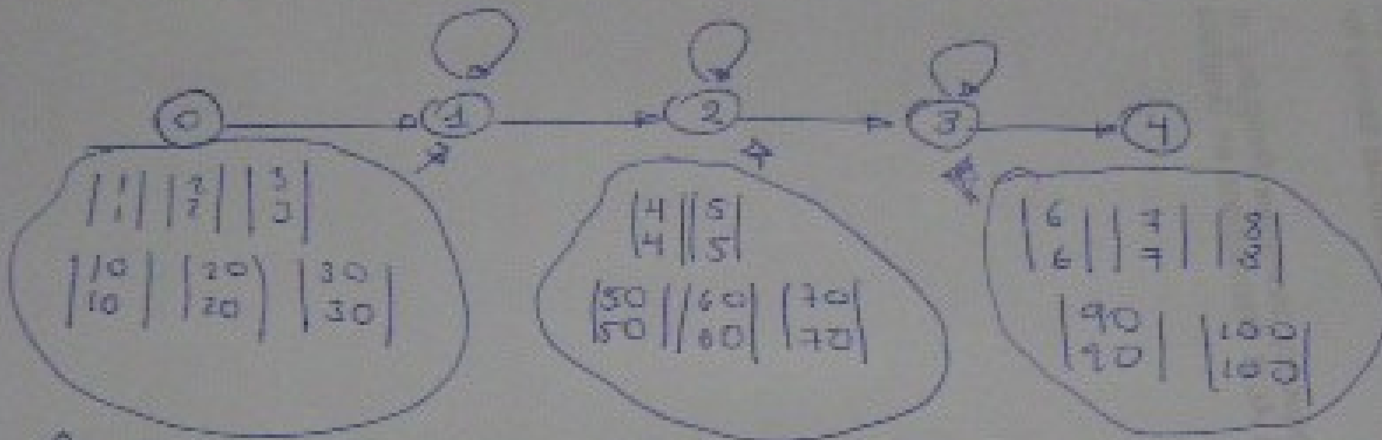
|1| |2| |3| |4| |5| |6| |7|

  - |10| |20| |30| |40| |50| |60| |70| |80| |90| |100|

|10| |20| |30| |40| |50| |60| |70| |80| |90| |100|



HMM



Gaussiana 0

$\mu = \begin{vmatrix} 30 \\ 30 \end{vmatrix}$

$\Sigma = \begin{vmatrix} 100 & 0 \\ 0 & 100 \end{vmatrix}$

$c_1 = 0.42$

Gaussiana 1

$\mu \begin{vmatrix} 4 \\ 4 \end{vmatrix}$

$\Sigma \begin{vmatrix} 16.7 & 0 \\ 0 & 16.7 \end{vmatrix}$

$c_2 = 0.57$

Gaussiana 0

$\mu = \begin{vmatrix} 60 \\ 60 \end{vmatrix}$

$\Sigma = \begin{vmatrix} 100 & 0 \\ 0 & 100 \end{vmatrix}$

$c_1 = 0.6$

Gaussiana 1

$\mu = \begin{vmatrix} 4.5 \\ 4.5 \end{vmatrix}$

$\Sigma = \begin{vmatrix} 0.5 & 0 \\ 0 & 0.5 \end{vmatrix}$

$c_2 = 0.4$

Gaussiana 0

$\mu = \begin{vmatrix} [illegible] \\ [illegible] \end{vmatrix}$

$\Sigma = \begin{vmatrix} 0.5 & 0 \\ 0 & 0.5 \end{vmatrix}$

$c_1 = 0.4$

Gaussiana 1

$\mu = \begin{vmatrix} 90 \\ 90 \end{vmatrix}$

$\Sigma = \begin{vmatrix} 100 & 0 \\ 0 & 100 \end{vmatrix}$

$c_2 = 0.6$

Trabajo futuro

- Implementar el algoritmo EM
- Implementar un modelo basado en redes neuronales (SOM)

Referencias

- Titterington, D., A. Smith, and U. Makov "Statistical Analysis of Finite Mixture Distributions," John Wiley & Sons (1985).
- McLachlan, G.J. and Peel, D. *Finite Mixture Models,* , Wiley (2000)
- Marin, J.M., Mengersen, K. and Robert, C.P. "Bayesian modelling and inference on mixtures of distributions". *Handbook of Statistics* 25, D. Dey and C.R. Rao (eds). Elsevier-Sciences.
- Lindsay B.G., Mixture Models: Theory, Geometry, and Applications. NSF-CBMS Regional Conference Series in Probability and Statistics Vol. 5, Institute of Mathematical Statistics, Hayward (1995).
- McLachlan, G.J. and Basford, K.E. "Mixture Models: Inference and Applications to Clustering", Marcel Dekker (1988)
- Everitt, B.S. and Hand D.J. "Finite mixture distributions", Chapman & Hall (1981)

alumnos 2007

- Proyectos realizados
  - *Extracción de características de la señal de voz utilizando LPC-Cepstrum - Jorge Velarde, Jhon Franko, Pretel Jesús, Alicia Isolina*
  - *Prediccion Lineal Perceptual PLP - Alan Alfredo Collantes Arana Dany Richard Sari Bustos*
  - *Audio files compression through wavelets - Fredy Carranza-Athó*
  - *Máquinas de Sopoerte Vectorial en el Reconocimiento Automático del Habla - Juan Carlos Federico Roeder Moreno*
  - *Efectos de las diferentes transformadas del coseno en RAH Márquez Fernández, Luz Victoria*

alumnos 2008

- **Proyectos realizados**
  - **Extracción de características de palabras aisladas usando MFCC y MFCC con pesos, Nils Murrugarra Llerena**
  - **Reconocimiento automático de palabras aisladas mediante el uso de los extractores de características: MFCC y MODGDF, Jorge Valverde Rebaza**
  - **Uso del método de extracción de características MFCC con formas arbitrarias a nivel de filtros para el reconocimiento de palabras aisladas, Pedro Shiguihara Juárez**
  - **Algoritmo N-Best: Eficiente procedimiento para la búsqueda de las N hipotesis de frases más probables, Luis Mostacero Zárate**
  - **Predicción y Entropia de Textos en Inglés, Juan Grados Vásquez**
  - **Aplicación del algoritmo MFCC-DTW en el reconocimiento de comandos activados por voz, Pedro Linares Kcomt**

GRACIAS!!!