

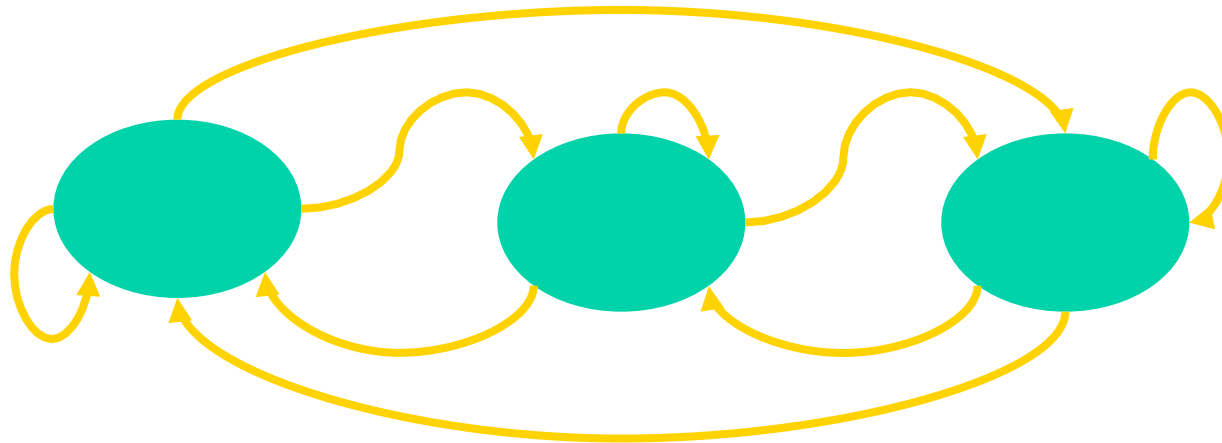
Modelos Gráficos Probabilistas

L. Enrique Sucar

INAOE

Sesión 7:

Modelos Ocultos de Markov



Modelos Ocultos de Markov

- Cadenas de Markov
 - Preguntas básicas
 - Aplicación: orden en Google
- Modelos Ocultos de Markov
- Problemas básicos
 - Evaluación
 - Secuencia óptima
 - Aprendizaje
- Aplicaciones
 - Reconocimiento de voz
 - Reconocimiento de gestos

Máquina de estados

- Es un modelo para procesamiento de información especificado por:
 - Un conjunto de estados,
 - Un estado inicial,
 - Un conjunto finito de entradas,
 - Un conjunto finito de salidas,
 - Una función de transición de
 - Una función de salida de

Cadena de Markov (CM)

- Máquina de estados finitos en la que:
 - Las transiciones de un estado a otro no son determinísticas
 - La probabilidad de transición de un estado a otro sólo depende del estado actual (y no de los anteriores) – propiedad de Markov

Ejemplo

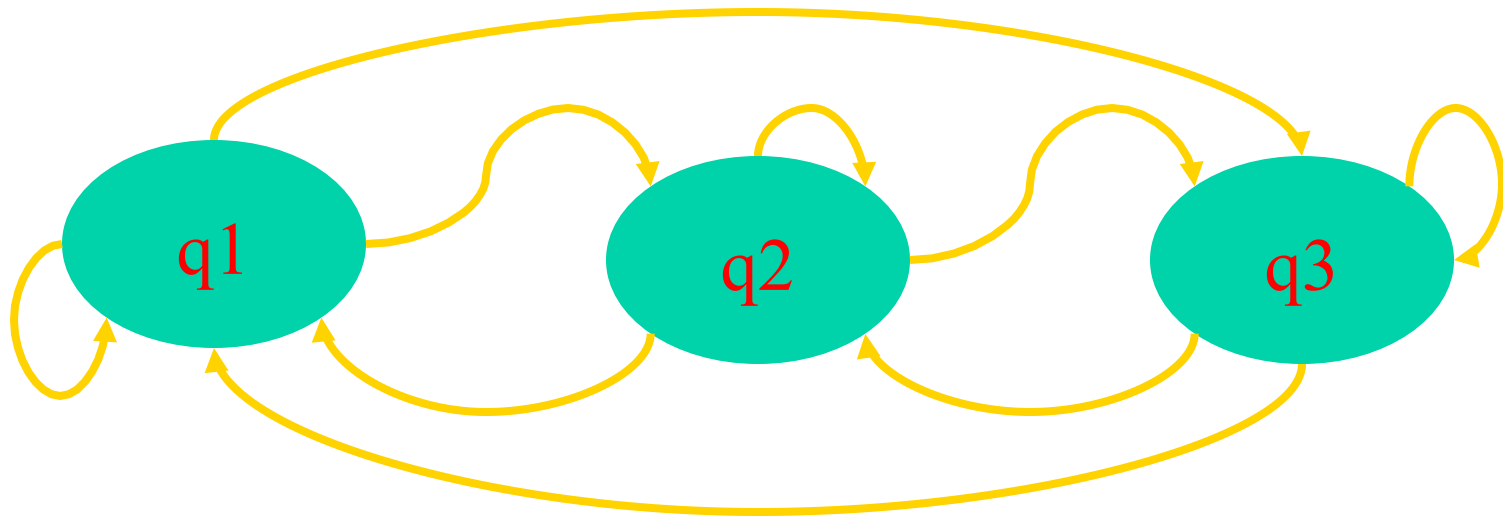
- Estados del clima:
 - Lluvioso (q1)
 - Nublado (q2)
 - Soleado (q3)
- Probabilidades de transición:

	L1	Nub	Sol
L1	0.4	0.3	0.3
Nub	0.2	0.6	0.2
Sol	0.1	0.1	0.8

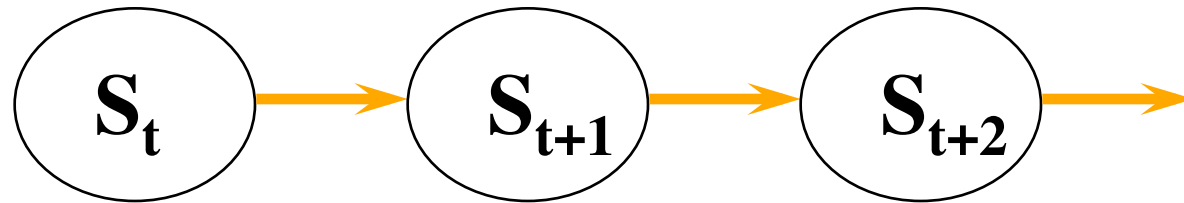
- Probabilidades iniciales:

	L1	Nub	Sol
	0.3	0.5	0.2

Ejemplo – diagrama de estados



CM – modelo gráfico



Especificación de una CM

- Conjunto de estados $Q = \{q_1 \dots q_n\}$
- Una vector de probabilidades iniciales,
 $\Pi = \{\pi_1 \dots \pi_n\}$, $\pi_i = P(S_0 = q_i)$
- Un matriz de probabilidades de transición,
 $A = \{a_{ij}\}$, donde $a_{ij} = P(S_t = q_j \mid S_{t-1} = q_i)$
- En forma compacta:
 $\lambda = \{A, \Pi\}$

Propiedades

1. Axiomas de probabilidad:

$$\sum_j a_{ij} = 1$$

2. Markoviano:

$$\begin{aligned} P(S_t=q_j \mid S_{t-1}=q_i, S_{t-2}=q_k, \dots) \\ = P(S_t=q_j \mid S_{t-1}=q_i) \end{aligned}$$

Salidas

- A cada estado le corresponde una salida, O_i
- Una secuencia de observaciones de $t = 1$ a $t = T$ se denota por:

$$O = \{o_1 \dots o_T\}$$

Preguntas básicas

- Probabilidad de pertenencia: probabilidad de cierta secuencia de estados
- Probabilidad de permanencia: probabilidad de que permanezca en cierto estado por cierto tiempo
- Permanencia promedio: tiempo esperado de permanencia en un estado

Preguntas

- Probabilidad de pertenencia:

$$P(q_j q_k q_l \dots) = a_{0j} a_{jk} a_{kl} \dots$$

- Probabilidad de permanencia:

$$P(d_i) = a_{ii}^{d-1} (1 - a_{ii})$$

- Permanencia promedio:

$$E\{d\} = \sum d_i P(d_i)$$

$$E\{d\} = \sum d_i a_{ii}^{d-1} (1 - a_{ii}) = 1/(1 - a_{ii})$$

Ejemplos de Preguntas

Dado el modelo del clima -

- Probabilidad de pertenencia:
 - Dado que el tiempo inicial es soleado, cual es la P de que sea sol, sol, lluv, lluv, sol, nub, sol?
- Probabilidad de permanencia:
 - Probabilidad de que este nublado por 3 días seguidos?
- Permanencia promedio:
 - Tiempo esperado que permanezca nublado?

Estimación de parámetros

- Dada una secuencia de observaciones, se pueden determinar los parámetros (A, Π) del modelo:
 - Probabilidades iniciales: $\pi_i \sim \gamma_{0i} / N$
 - Probabilidades de transición: $a_{ij} \sim \gamma_{ij} / \gamma_i$
- Donde:
 - γ_{0i} = número de veces que el estado i es el inicial
 - γ_i = número de veces que pasa por el estado i
 - γ_{ij} = número de transiciones del estado i al j
 - N = número de secuencias

Ejemplo de estimación

- Obtener los parámetros del modelo dadas las secuencias:

q2q2q3q3q3q3q1

q1q3q2q3q3q3q3

q3q3q2q2

q2q1q2q2q1q1q3

Convergencia

- Otra pregunta interesante es: ¿Si se “transita” la cadena un gran número de veces, a la larga cuál es la probabilidad de cada estado (en el límite)?
- Dada una probabilidad inicial, π , la probabilidad después de N iteraciones se obtiene multiplicando π por $A \times A \times A \dots$:

$$p = \pi A^N$$

- Después de un cierto número, normalmente el valor de p ya prácticamente no cambia

Convergencia

Ejemplo:

$$A = \begin{pmatrix} 0 & 1.0000 & 0 \\ 0 & 0.1000 & 0.9000 \\ 0.6000 & 0.4000 & 0 \end{pmatrix}$$
$$\pi = 0.5000 \quad 0.2000 \quad 0.3000$$

Si multiplicamos $\pi * A$:

$$\begin{aligned} 1. & \quad 0.1800 \quad 0.6400 \quad 0.1800 \\ 2. & \quad 0.1080 \quad 0.3160 \quad 0.5760 \\ & \quad \dots \\ 10. & \quad 0.2358 \quad 0.4190 \quad 0.3452 \end{aligned}$$

En el límite:

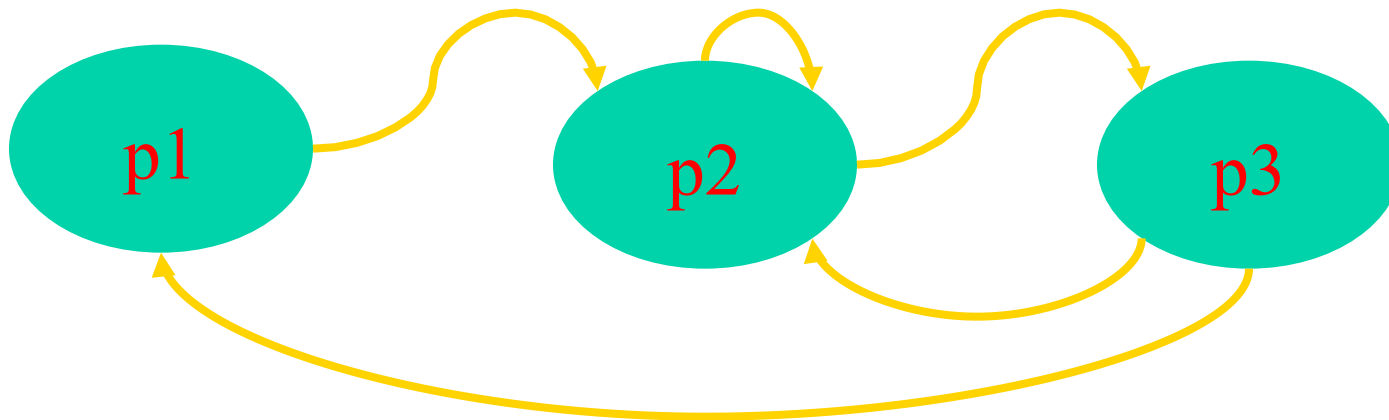
$$p = 0.2 \quad 0.4 \quad 0.4$$

Teorema de Perron-Frobenius

- Dadas ciertas condiciones, la cadena converge a un distribución invariante p , tal que: $p A = p$
- Condiciones:
 1. Irreducible: de cualquier estado hay cierta probabilidad de visitar los demás estados
 2. Aperiódica: la cadena no cae en *ciclos*
- La rapidez de convergencia está determinada por el *segundo eigen-valor* de A

Aplicación: orden (*rank*) de Google

- Podemos representar la Web como una CM, donde cada estado es un página y los arcos representan las ligas que apuntan a cada página
- Las probabilidades se “reparten” en función de las ligas salientes de cada página



Aplicación: orden (rank) de Google

- La probabilidad a la que converge la cadena provee una estimación de que tan probable es que una persona visite una página en cierto momento
- Google basa el orden (importancia) de las páginas que encuentra (para cierta búsqueda) en éstas probabilidades

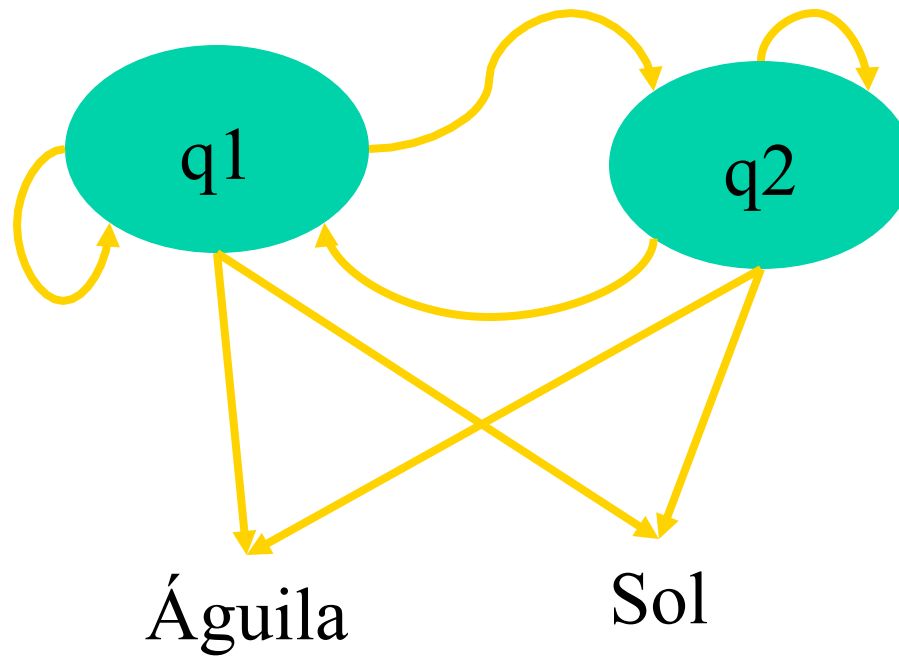
Modelos Ocultos de Markov (HMM)

- Es un modelo de Markov en que los estados no son directamente observables
- Se puede ver como un doble proceso estocástico:
 - Un proceso estocástico “escondido” que es no observable
 - Otro proceso estocástico que produce la secuencia de observaciones

Ejemplo

- Se tienen dos monedas (M1 y M2) que se seleccionan en forma aleatoria
- Cada moneda esta cargada:
 - M1 – $P=0.8$ de águila
 - M2 – $P=0.8$ de sol
- Se tiran en secuencia las monedas (N veces) y sólo se observa la salida (A o S)

Ejemplo



Especificación de un HMM

- Conjunto de estados $Q = \{q_1 \dots q_n\}$ y de posibles observaciones $O = \{o_1 \dots o_m\}$

- Una vector de probabilidades iniciales,

$$\Pi = \{\pi_1 \dots \pi_n\}, \pi_i = P(S_0 = q_i)$$

- Un matriz de probabilidades de transición,

$$A = \{a_{ij}\}, \text{ donde } a_{ij} = P(S_t = q_j \mid S_{t-1} = q_i)$$

- Un vector de probabilidades de salida por cada estado (matriz),

$$B = \{b_{ik}\}, \text{ donde } b_{ik} = P(O_t = o_k \mid S_t = q_i)$$

- En forma compacta:

$$\lambda = \{A, B, \Pi\}$$

Ejemplo - especificación

- Π :

0.5 0.5

- A :

0.5 0.5

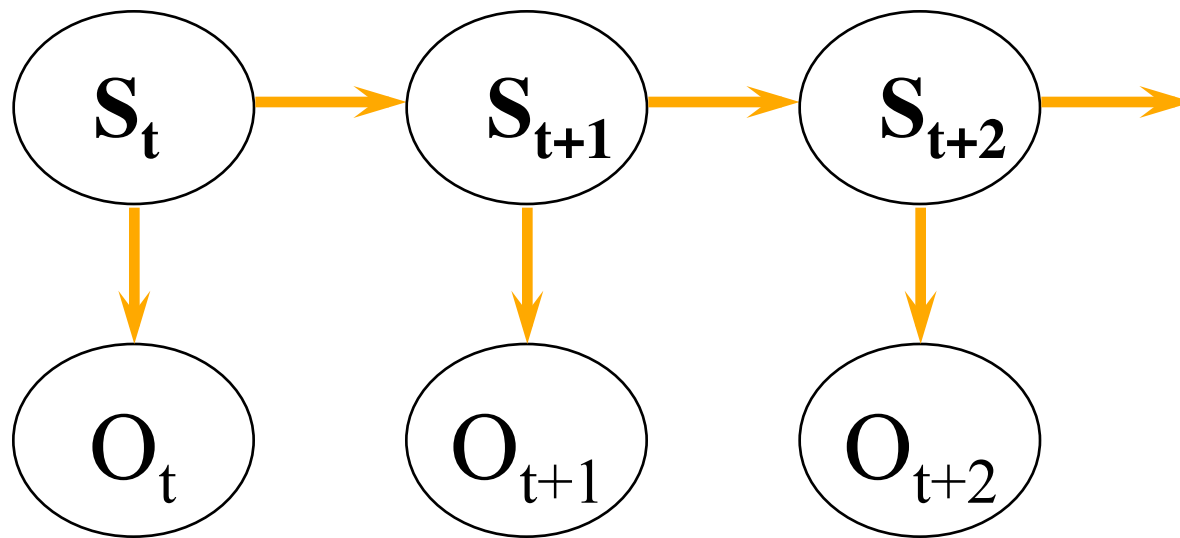
0.5 0.5

- B :

0.8 0.2

0.2 0.8

HMM – modelo gráfico



Consideraciones

- Proceso markoviano: el estado actual sólo depende del estado anterior
- Estacionario: las probabilidades de transición y observación no cambian con el tiempo
- Independencia observaciones: las observaciones sólo dependen del estado actual

Preguntas básicas

- Dado el modelo, calcular la probabilidad de una secuencia de observaciones (evaluación)
- Dado el modelo, obtener la secuencia de estados más probable correspondiente a una secuencia de observaciones (secuencia óptima)
- Dada una secuencia de observaciones, ajustar los parámetros del modelo (aprendizaje)

Evaluación – método directo

- Dada la secuencia de observaciones:

$O_1 O_2 O_3 O_4 \dots$

- Pueden ser generados por diferentes secuencias de estados, considerando una:

$S_1 S_2 S_3 S_4 \dots$

- Entonces la probabilidad de las observaciones y dicha secuencia de estado es:

$$P(O, Q_i) = \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{(T-1)T}} b_{q_T}(O_T)$$

Evaluación – método directo

- Considerando todas las secuencias:

$$P(O) = \sum_Q P(O, Q_i)$$

- Que es lo mismo que:

$$P(O) =$$

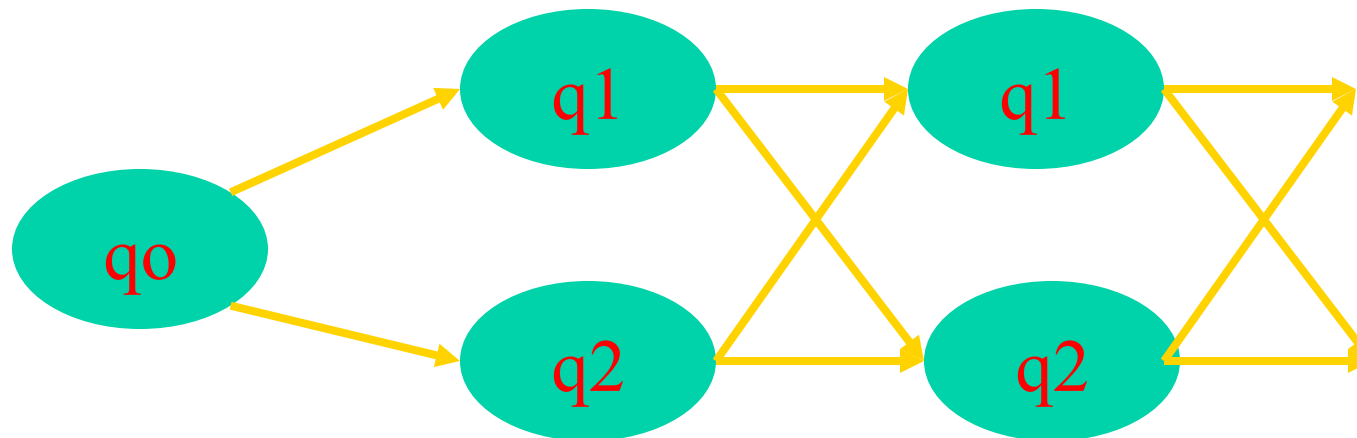
$$\sum_Q [\pi_{q1} b_{q1}(O_1) a_{q12} b_{q2}(O_2) \dots a_{q(T-1)T} b_{qt}(O_T)]$$

Evaluación – método directo

- Número de operaciones
 - Para cada término:
 $2T$
 - Número de posibles secuencias (sumatoria)
 N^T
 - Total:
 $2T \times N^T$
- Por ejemplo, $N=5$ y $T=100 \rightarrow 10^{72}$ operaciones!
- Se requiere de un método más eficiente!

Evaluación – método iterativo

- Se basa en la idea de ir evaluando en paralelo la probabilidad de estados/observaciones para cada tiempo



Evaluación – método iterativo

- Se define la variable “forward”:

$$\alpha_t(i) = P(O_1 O_2 O_3 O_4 \dots O_t, S_t = q_i)$$

- Es decir, la probabilidad de una secuencia parcial de observaciones y que llegue a cierto estado

Algoritmo

1. Inicialización

$$\alpha_1(i) = P(O_1, S_1 = q_i) = \pi_i b_i(O_1)$$

2. Inducción

$$\alpha_{t+1}(j) = [\sum_i \alpha_t(i) a_{ij}] b_j(O_{t+1})$$

3. Terminación

$$P(O) = \sum_i \alpha_T(i)$$

Complejidad

- En cada iteración se tiene del orden de N multiplicaciones y N sumas
- Para las T iteraciones:

$$N^2 \times T$$

- Para $N=5$ y $T=100 \rightarrow 2,500$ operaciones

Secuencia óptima

- Encontrar la secuencia de estados *óptima* dada la secuencia de observaciones
- Óptimo se puede definir de diferentes maneras:
 - Estados más probables
 - Secuencia total más probable

Definiciones

- Variable “*backward*”:

$$\beta_t(i) = P (O_{t+1} O_{t+2} \dots O_T , S_t = q_i)$$

- En forma iterativa:

$$\beta_t(i) = \sum_j \beta_{t+1}(j) a_{ij} b_j (O_{t+1})$$

- Definiendo:

$$\beta_T(j) = 1$$

Definiciones

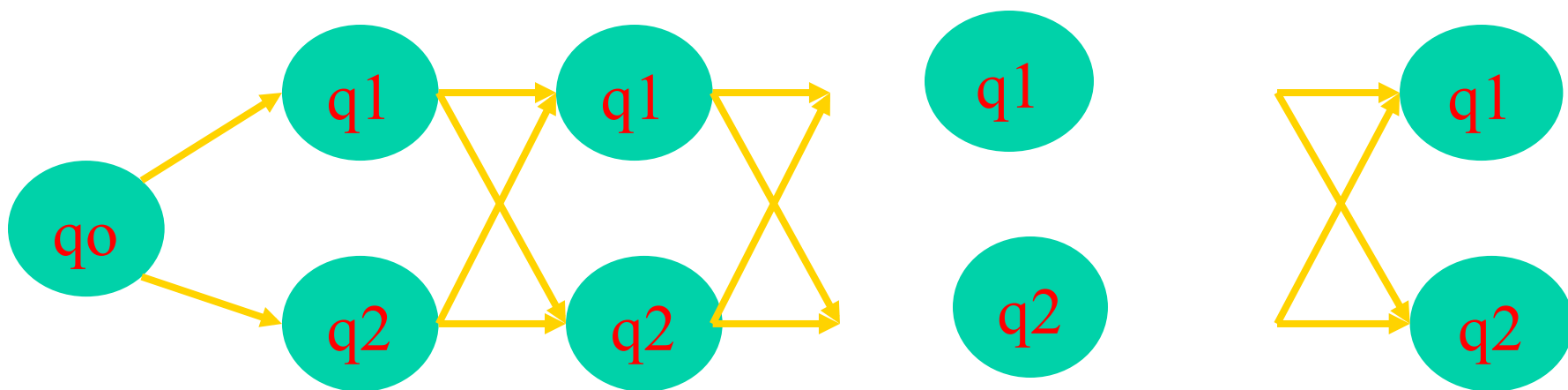
- Por lo tanto, combinando ambas definiciones:

$$P(O, S_t = q_i) = \alpha_t(i) \beta_t(i)$$

- Y entonces:

$$P(O) = \sum_i \alpha_t(i) \beta_t(i)$$

Cálculo iterativo



Más definiciones

- Probabilidad condicional:

$$\gamma_t(\mathbf{i}) = P(S_t = q_i | O) = P(S_t = q_i, O) / P(O)$$

- En términos de α y β :

$$\gamma_t(\mathbf{i}) = \alpha_t(\mathbf{i}) \beta_t(\mathbf{i}) / P(O)$$

$$\gamma_t(\mathbf{i}) = \alpha_t(\mathbf{i}) \beta_t(\mathbf{i}) / \sum_i \alpha_t(\mathbf{i}) \beta_t(\mathbf{i})$$

Estados más probable

- El estado individual más probable para el tiempo t es:

$$\text{ARG MAX}_i \gamma_t(i)$$

- El problema es que la concatenación de los estados más probables no necesariamente corresponde a la secuencia más probable

Secuencia más probable

- Secuencia total más probable es:

$$\text{MAX } P(Q | O)$$

- Dado que $P(Q|O) = P(Q,O) / P(O)$, entonces es equivalente a:

$$\text{MAX } P(Q , O)$$

Algoritmo de Viterbi

- Antes de ver el algoritmo es necesario definir otra variable
- La subsecuencia de estados óptimos hasta el tiempo t:

$$\delta_t(i) = \text{MAX} P(S_1 S_2 \dots S_t = q_i, O_1 O_2 \dots O_t)$$

- En forma iterativa:

$$\delta_{t+1}(i) = [\text{MAX}_j \delta_t(j) a_{ij}] b_j (O_{t+1})$$

Algoritmo

1. Inicialización:

$$\delta_1(i) = \pi_i b_i(O_1)$$

$$\psi_1(i) = 0$$

2. Recursión

$$\delta_t(j) = \text{MAX}_i [\delta_{t-1}(i) a_{ij}] b_j(O_t)$$

$$\psi_t(j) = \text{ARGMAX}_i [\delta_{t-1}(i) a_{ij}]$$

3. Terminación

$$P^* = \text{MAX}_i [\delta_T(i)]$$

$$q_T^* = \text{ARGMAX}_i [\delta_T(i)]$$

4. Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

Aprendizaje

- Consiste en determinar los parámetros del modelo, $\lambda = \{A, B, \Pi\}$, dada una secuencia de observaciones
- Para ello, se buscan los parámetros que maximicen $P(O | \lambda)$
- Número de parámetros (N estados, M obs.):

$$N + N^2 + N \times M$$

Algoritmo de Baum-Welch

- Otra variable auxiliar – probabilidad de estar en el estado i en t y pasar a j en $t+1$ dada la secuencia de observaciones:

$$\xi_t(i,j) = \frac{P(S_t = q_i, S_{t+1} = q_j | O)}{P(S_t = q_i, S_{t+1} = q_j, O) / P(O)}$$

- En términos de α y β :

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

Algoritmo de Baum-Welch

- La variable $\gamma_t(i)$ se puede calcular como:

$$\gamma_t(i) = \sum_j \xi_t(i,j)$$

- Esta variable sumada sobre todos los tiempos da una estimación del número de veces que se pasa por el estado “i”

$$\sum_t \gamma_t(i)$$

- Mientras que la suma sobre t de $\xi_t(i,j)$ da una estimación del número de transiciones de “i -> j”:

$$\sum_t \xi_t(i,j)$$

Re-estimación de los parámetros

1. Probabilidades iniciales – número de veces en el estado “i” en $t=1$:

$$\pi_i = \gamma_1(i)$$

2. Probabilidades de transición – número de transiciones de “i \rightarrow j” entre el número de veces en “i”

$$a_{ij} = \sum_t \xi_t(i,j) / \sum_t \gamma_t(i)$$

3. Probabilidades de salidas – número de veces en estado “j” y observar “k” entre el número de veces en dicho estado:

$$b_{jk} = \sum_{t, O=k} \gamma_t(i) / \sum_t \gamma_t(i)$$

Re-estimación de los parámetros

- Se inicia con ciertos valores (al azar) y se van mejorando iterativamente (se repite el proceso varias veces)
- Se obtiene un estimador de *máxima verosimilitud*
- No se garantiza el óptimo global
- Este algoritmo pertenece a la familia de métodos EM (maximización de la expectativa)

Aplicaciones

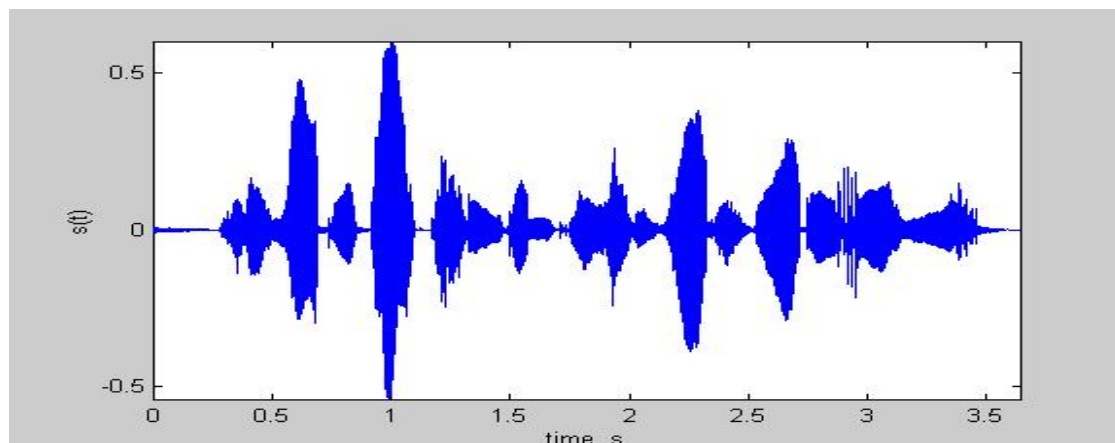
- Modelado de procesos dinámicos, como:
 - Reconocimiento de voz
 - Reconocimiento de gestos

Reconocimiento de voz

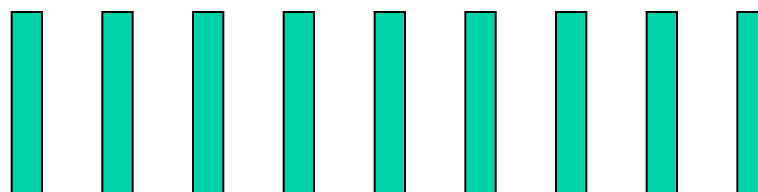
- Se modela a nivel palabra o fonema utilizando HMMs
- Las observaciones consisten de vectores de características obtenidas del procesamiento de la señal de voz
- Se utilizan secuencias de voz para el entrenamiento y, posteriormente durante reconocimiento, se obtiene la probabilidad de cada modelo (palabra o fonema), seleccionando la de mayor probabilidad

Rec. de Voz

Señal
de voz

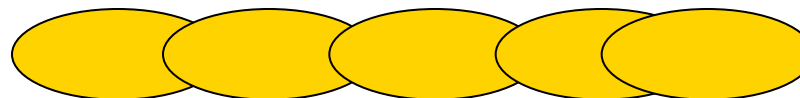


Vector de
características
espectrales



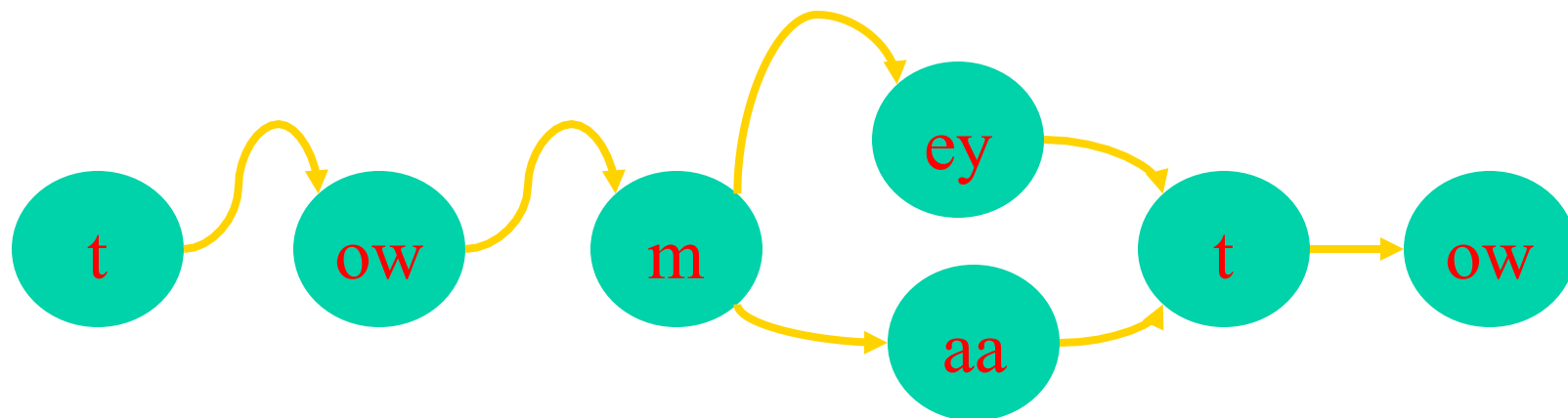
Estimación de
probabilidades
fonemas

HMMs

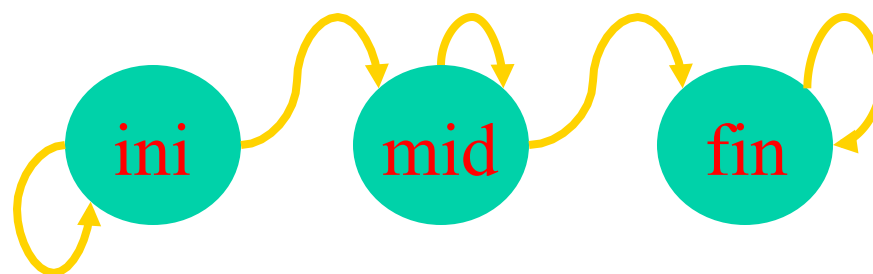


Reconocimiento de voz

Palabra: "tomato"



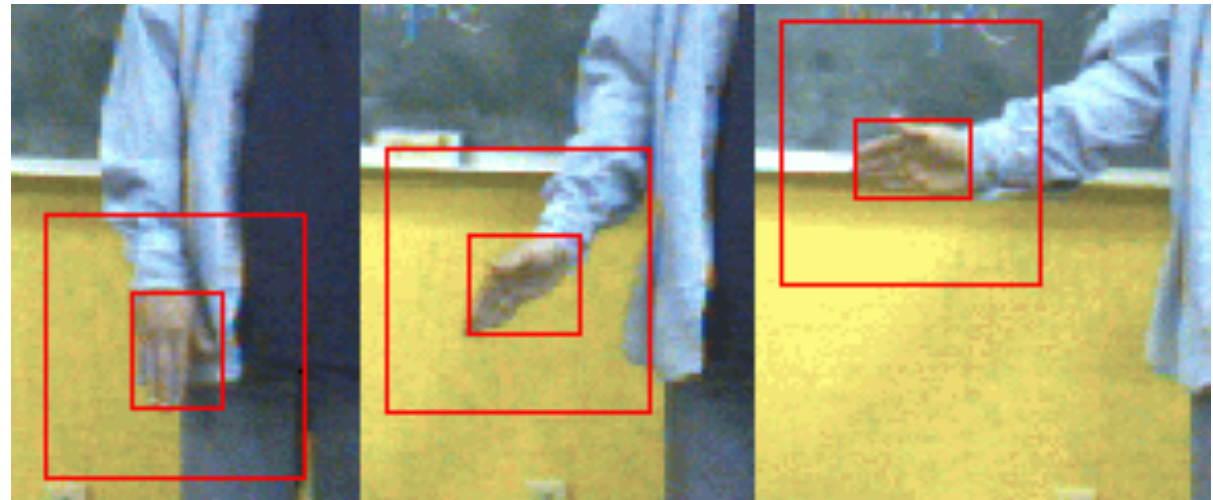
Fonema



Reconoci- miento de Gestos dinámicos



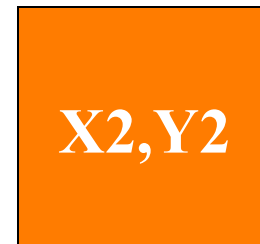
Reconocimiento de gestos



Seguimiento de la mano en una secuencia imágenes

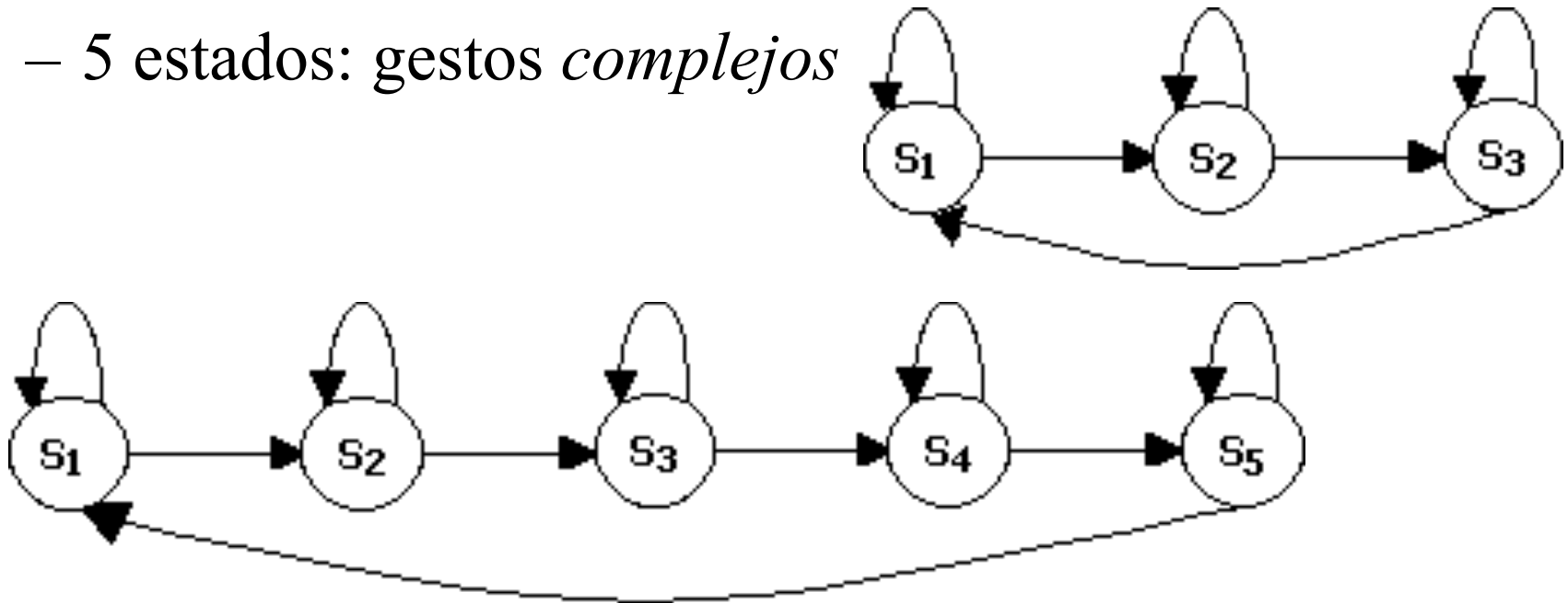
Características

- Observaciones:
 - cambio en X (ΔX)
 - cambio en Y (ΔY)
 - cambio en área (ΔA)
 - cambio en razón X-Y (ΔR)
- Cada una se codifica en 3 valores: (+, 0, -), lo que da un total de 81 posibles observaciones



HMM

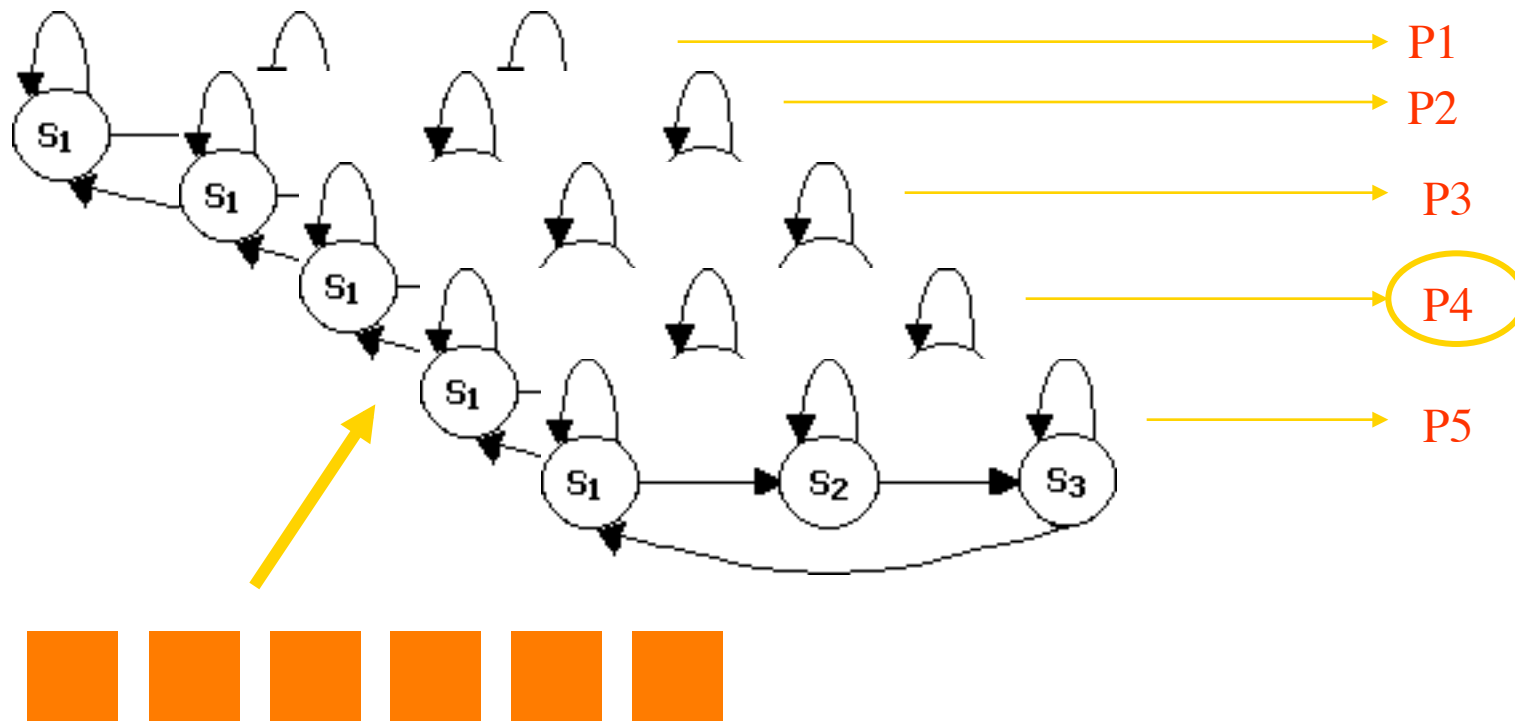
- Se utiliza un HMM para cada gesto (5 gestos):
 - 3 estados: gestos *simples*
 - 5 estados: gestos *complejos*



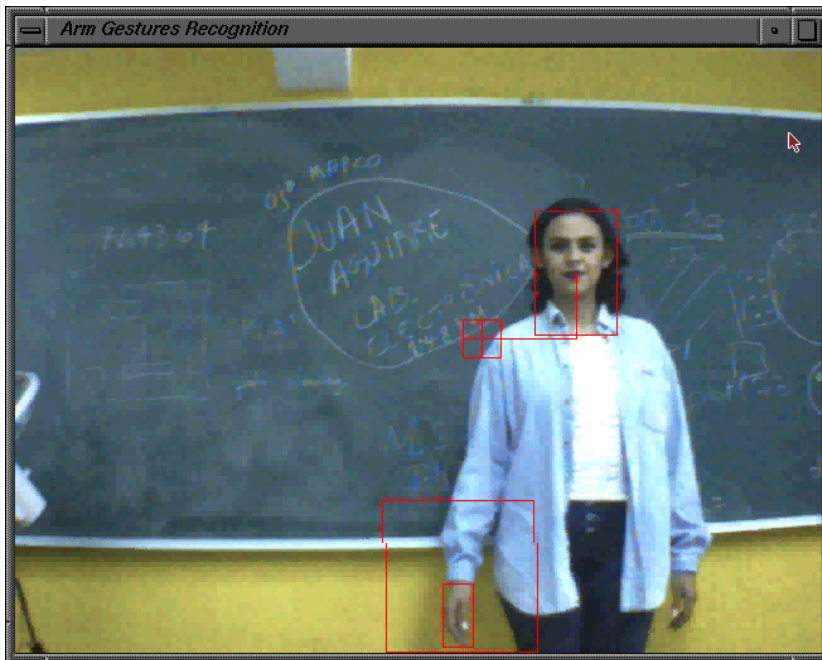
Entrenamiento y Reconocimiento

- Se tiene un HMM por gesto que se entrena (*algoritmo de Baum-Welch*) con ejemplos de secuencias del gesto
- Para reconocer gestos, se obtiene la probabilidad de cada modelo dadas las observaciones (*algoritmo Forward*) y se selecciona el modelo con mayor probabilidad

Reconocimiento



Ejemplos



Seguimiento de la mano



Control con gestos

Referencias

- L. R. Rabiner, B. H. Juang, “An introduction to hidden Markov models”, IEEE ASSP, Enero 1986.
- L. R. Rabiner, “A tutorial on hidden Markov Models and selected applications in speech recognition”, Readings in speech recognition, pp. 267-296, 1990.
- J. K. Kemeny, J. L. Snell, “Finite Markov Chains”, Van Nostrand, 1965.
- D. Jurafsky, J. Martin, “Speech and language processing”, Prentice-Hall, 2000 – Capítulo 7

Referencias

- [Koller & Friedman] Cap. 6
- L. Page et al., “The PageRank citation ranking: Bringing order to the Web”, Stanford Digital Libraries Working Paper, 1998.
- H. Avilés, L. E. Sucar, “Visual recognition of similar gestures”, ICPR’06
- A. Montero, L.E. Sucar, “A decision theoretic video conference system based on gesture recognition”, F&G’06

Actividades

- Hacer ejercicios de HMM
- Leer artículo HMMs de Rabiner
- Leer artículo de PageRank