4. MODELADO CON CADENAS DE MARKOV.

Una vez caracterizado el canal por el que se efectua las comunicaciones en sistemas móviles, está clara la necesidad de elaborar un modelo del canal con el que sea más fácil trabajar.Esto se configue con las cadenas ocultas de Markov, las cuales representan modelos que tratan de acercarse lo más posible al comportamiento real de un sistema.

Con las cadenas ocultas de Markov podemos simular el comportamiento de un sistema físico describiendo todos los diferentes estados en que puede encontrarse el sistema e indicando cómo este se mueve por los estados con el transcurrir del tiempo. Ejemplos del uso de los procesos de Markov pueden ser encontrados de forma amplia en ciencias biológicas, físicas, y sociales así como en negocios y evidentemente en ingeniería.

4.1. INTRODUCCIÓN.

Una cadena de Markov es un modelo probabilístico compuesto por un número de estado interconectados entre los cuales nos podemos ir moviendo, y donde cada uno de ellos emite una salida observable. Cada estado tiene dos clases de parámetros. La probabilidad de emisión de símbolo que describe la probabilidad de que se den las posibles salidas en cada estado, y la probabilidad de transición entre cada estado que especifica la probabilidad de movernos a un nuevo estado desde el actual o permanecer en el que ya se ha alcanzado. La parte visible de una cadena oculta de Markov, los símbolos observables, son generados comenzando en un estado inicial y moviéndonos probabilísticamente de estado en estado hasta que un estado final es alcanzado, emitiendo símbolos observables desde cada estado por el que hemos pasado. Una secuencia de estados es una cadena de Markov de primer orden, pero esta secuencia de estados esta oculta, sólo la secuencia de símbolos que se emiten son observables; por esto el término de cadena oculta de Markov.

Ya hemos dicho que asociado a cada proceso de Markov hay un conjunto de estados. Se asume que el sistema al ser modelado por el proceso ocupa uno y sólo uno de estos estados en cada instante del tiempo. La evolución del sistema está representado por la transición del proceso entre un estado y otro. Estas transiciones ocurren instantáneamente, o dicho en otras palabras el tiempo consumido para movernos de un estado a otro se supone cero. La propiedad fundamental de los sistemas de Markov es que la evolución futura del sistema depende sólo del estado actual y no de la historia pasada.

Como ejemplo ilustrativo, consideremos el comportamiento de una rana que salta de hoja en hoja en un lago. Las hojas constituyen los estados del sistema. El lugar a donde la rana saltará depende sólo de la información que ella pueda deducir de su hoja actual ya que aquella no tiene memoria y no recuerda nada de los estados que ha visitado anteriormente al estado en el que se encuentra, incluso la longitud de tiempo que ha estado moviéndose por las hojas.

4.2. NOTACIÓN.

En los experimentos que realizaremos solo consideraremos dos posibles símbolos observables desde cada estado ya que nos encontramos con el estudio de sistemas que son fácilmente sintetizables bajo la suposición de que en cada estado solo podemos observar dos tipos de objetos, salidas binarias. Sin embargo y para ser más estricitos vamos a considerar un conjunto de N urnas cada una con un número determinado de canicas para

poder sintetizar la notación que se va a utilizar. Dentro de cada urna las canicas son de distintos colores. Nuestro experimento consiste en sacar canicas de las urnas en una secuencia; solo la secuencia de canicas que se extrae nos es mostrada. Las canicas en este caso corresponderán a los bits que tendremos en el receptor, a la cadena con ráfagas de errores que tendremos que modelar a través de un modelo de cadenas de Markov. No nos será posible saber de que urna se están sacando canicas en cada instante. Las urnas corresponderán a los estados ocultos de la cadena de Markov.

Definimos la siguiente notación para nuestro modelo:

N= número de estados (urnas) en el modelo.

M= número total de símbolos posibles observables (canicas de M colores distintos).

1,2,...,N denotara las N urnas respectivamente.

i_t designa el estado en el cual nosotros estamos en el tiempo t.

$V=\{v_1,...,v_M\}$ el conjunto discreto de posibles símbolos observables.

$\pi=\{\pi_i\}$, $\pi_i=P(i_1=i)$, la probabilidad de estar en el estado i al principio del experimento en t=1.

$A=\{a_{ij}\}$ donde $a_{ij}=P(i_{t+1}=j|i_t=i)$, la probabilidad de estar en el estado j en el instante t+1 suponiendo que estabamos en el estado i en el instante t. Si suponemos que a_{ij} son independientes del tiempo nos encontramos en el caso de una cadena de Markov de carácter estacionario.

$B=\{b_j(k),b_i(k)\}=P(v_k \text{ en } t|i_t=j)$, la probabilidad de observar el símbolo v_k dado que estamos en el estado j.

O_t designará el símbolo observado en el instante t.

$\lambda=(A,B,\pi)$ será usado para designar a la cadena o modelo oculto de Markov (HMM).

Una vez que tenemos el modelo que define la cadena oculta de Markov, una secuencia $O = O_1,O_{2,...,}O_T$ es generada como sigue: Empezamos nuestro experimento eligiendo una de las urnas (de acuerdo con la distribución de probabilidad inicial π que tiene cada estado), y a continuación elegimos una canica (símbolo observado) de esta urna. Este instante inicial es tomado como t=1 y el estado y símbolo observado en este instante t=1 son designado por i_1 y O_1 respectivamente. Tras esto elegimos otra urna (puede ser la misma u otra diferente a la que ha habido en el instante t=1) y de acuerdo a la distribución de probabilidad de transición A que existe entre la urna inicial y las demás incluida ella misma y considerando ahora que estamos en el instante t=2 sacamos una canica (designada por O_2) desde esta urna dependiendo de la probabilidad de distribución de los símbolos

$b_j(k)$ para esta urna (estado). Continuando así hasta que el tiempo t=T, generamos la secuencia de observación $O=O_1,O_2,...,O_T$.

4.3. MAPEADO DE SISTEMAS CON MODELOS OCULTOS DE MARKOV.

Una vez que hemos decidido utilizar un modelo de Markov para emular el comportamiento de un sistema particular se nos plantea una serie de problemas. Estos ponen de relieve cuales han de ser los métodos que nos proporcionen el modelo, ya que o bien es generado de forma teórica a partir del estudio de sistema, algo complicado en sistemas de comunicaciones móviles dada la complejidad del sistema, o bien es generado de forma experimental procurando ajustar lo más exactamente posible el modelo al sistema real. En el primer caso, la generación del modelo de forma teórica conlleva el conocimiento de los estados y sus características de probabilidad de transición entre estados y emisión de símbolos en cada estado. Esto particulariza el modelo de forma única, cosa que en algunos casos resulta imposible, ya que lo que vemos del sistema modelable es solamente su comportamiento externo y no su funcionamiento interno. Sin embargo, y esto es el segundo caso, si es posible en la mayoría de los casos la deducción del modelo a partir de técnicas que tratan de aproximar de forma algorítmica el comportamiento del sistema. Utilizando métodos de máxima verosimilitud podemos calcular la distancia que en cada paso de iteración existe entre el sistema modelable y el modelo, viendo lo lejos que estamos de la solución. Así pues la mayoría de las aplicaciones de modelos ocultos de Markov se concentran en resolver principalmente tres problemas. Estos son:

Problema 1: Dado un modelo $\lambda=(A,B,\pi)$, cómo podemos computar $P(O|\lambda)$, la probabilidad de ocurrencia de la secuencia observada $O=O_1,O_2,...,O_T$ (única información que tenemos del sistema a modelar) con el modelo $\lambda=(A,B,\pi)$ que tenemos ya sea el modelo final o uno que sirve de camino a la solución final, y que nos proporcionará una medida de como de probable es que se O a partir del modelo $\lambda=(A,B,\pi)$.

Problema 2: Dado el modelo $\lambda=(A,B,\pi)$, cómo elegimos la secuencia de estados $I=i_1,i_2,...,i_T$ de tal forma que $P(O,I|\lambda)$, la probabilidad conjunta de la secuencia observada y la secuencia de estado con el modelo propuesto, sea máxima.

Problema 3. Como podemos ajustar los parámetros del modelo oculto de Markov $\lambda=(A,B,\pi)$ tal que $P(O|\lambda)$ ($P(O,I|\lambda)$) sea máxima. Este es el problema principal ya que nos generará el modelo, aplicandolo en el desarrollo de algoritmos que conduzcan al modelo más próximo o con probabilidad de semejanza mayor.

Los problemas 1 y 2 pueden ser vistos como problemas de análisis mientras que tercer problema es un típico problema de síntesis (o modelo de identificación o entrenamiento).

4.4. TÉCNICAS

4.4.1. Probabilidad de ocurrencia de símbolos.

El camino más sencillo para determinar P(O|λ) es encontrar P(O,I|λ) para una secuencia de estado fijada $I=i_1,i_2,...,i_T$ y multiplicarla por P(O|λ) sumando para todos los posibles secuencias de estados I. Tenemos

$$P(O \mid I, l) = b_{i_1}(O_1) b_{i_2}(O_2) \cdot\cdot b_{i_T}(O_T)$$

$$P(I \mid l) = p_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdot\cdot a_{i_{T-1} i_T}$$

De aquí tenemos:

$$P(O \mid l) = \sum_I P(O \mid I, l) P(I \mid l)$$

$$\sum_I p_{i_1} b_{i_1}(O_1) a_{i_1 i_2} b_{i_2}(O_2) \cdot\cdot a_{i_{T-1} i_T} b_{i_T}(O_T)$$

donde $I=i_1,i_2,...,i_T$.

En la última ecuación vemos que el sumatorio de ésta involucra 2T-1 multiplicaciones y existen N^T posibles secuencias de estados distintas. Por lo tanto observamos que una computación directa de esta ecuación involucrara del orden de $2T\ N^T$ multiplicaciones. Incluso para valores pequeños, N=5 y T=100 esto significa aproximadamente 10^{72} multiplicaciones las cuales conllevaría un tiempo impracticable incluso para una supercomputadora. Así vemos que se necesita un procedimiento más eficiente para resolver el problema 1. Este procedimiento existe y es llamado procedimiento de ida y vuelta (forward-backward).

- Procedimiento forward-backward.

Consideramos la variable de ida (forward) α(i) definida como:

$$a_t(i) = P(O_1, O_2, ..., O_t, i_t = i \mid l)$$

la probabilidad de la secuencia parcial observada hasta el instante t y el estado i en el instante t, dado el modelo λ; α(i) puede ser computada de forma inductiva de la forma:

1.
$$a_1(i) = p_i b_i(O_1), 1 \le i \le N$$

2. para t=1,2,...,T-1, 1<j<N
$$a_{t+1}(j) = \left[\sum_{i=1}^{N} a_t(i) a_{ij} \right] \cdot b_j(O_{t+1})$$

3.y tenemos:

$$P(O \mid l) = \sum_{i=1}^{N} a_T(i)$$

En el paso 2 nosotros queremos computar la probabilidad de la secuencia parcial observada hasta el instante t+1 y el estado j en el instante t+1; el estado j puede ser alcanzado (con probabilidad a_{ij}) independientemente desde cualquiera de los N estados en el instante t. El sumatorio de la ecuación del paso segundo refleja este hecho. Además el sumatorio da la secuencia observada hasta el instante t; es por eso que $b_j(O_{t+1})$ este fuera de los corchetes. En el paso 3 solo sumamos todos los posibles (independientes) caminos en los que puede realizarse la secuencia observada. Esto se muestra tomando el caso de

$$a_2 = P(O_1, O_2, i_2 = j)$$

La secuencia $O_1,O_2,i_2=j$ se da de la forma: primero O_1, luego el estado i_2, luego el símbolo O_2. Pero O_1 puede ocurrir a través de cualquiera de las formas: estado 1 y O_1, estado 2 y O_1, y así hasta el estado N los cuales se excluyen ya que si se da uno imposibilita que puedan darse los otros. Sabemos que si $\{S_i\}$ es un conjunto de eventos exclusivos entre si entonces para cualquier evento E tenemos que:

$$P(E) = \sum_i P(E \mid S_i) P(S_i)$$

De aquí podemos escribir:

$$a_2(j) = P(O_1, O_2, i_2 = j)$$

$$= \sum_i P(O_2, i_2 = j \mid O_1 del\_estado\_i) P(O_1 del\_estado\_i)$$

$$= \sum_i \left(\left[P(O_2 \mid i_2 = j, O_1 del\_estado\_i) P(i_2 = j \mid O_1 del\_estado\_i) \right] \left[P(O_1 \mid i_1 = i) P(i_1 = i) \right] \right)$$

$$= \sum_i \left(\left[P(O_2 \mid i_2 = j) P(i_2 = j \mid i_1 = i) \right] \left[P(O_1 \mid i_1 = i) P(i_1 = i) \right] \right)$$

$$= \sum_i \left(\left[b_j(O_2) a_{ij} \right] \left[b_i(O_1) p_i \right] \right)$$

$$= \left[\sum_i \left(p_i b_i(O_1) \right) a_{ij} \right] b_j(O_2)$$

$$= \left[\sum_i \left(a_1(i) \right) a_{ij} \right] b_j(O_2)$$

que es la misma que la del paso 2.

A partir de la ecuación del paso 3 y siguientes podemos escribir

$$P(O) = \sum_i P(O \mid i_T = i) P(i_T = i)$$

$$= \sum_i P(O, i_T = i)$$

$$= \sum_i a_T(i)$$

El paso 1 involucra N multiplicaciones. En el paso 2 el sumatorio conlleva N multiplicaciones más una por el termino externo a los corchetes $b_j(O_{t+1})$. Esto ha sido hecho desde j=1 hasta N y desde t=1 hasta T-1, siendo el número total de multiplicaciones en el paso 2 (N+1)N(T+1). En el paso 3 no hay multiplicaciones. El número total de multiplicaciones es N+N(N+1)(T-1) del orden de N^2T. Para N=5 y T=100 necesitamos 3000 cálculos que comparado con los 10^{72} que se necesitaban por el método directo provocan una diferencia de 69 ordenes de magnitud.

Consideremos ahora la variable de vuelta (backward). De forma similar podemos definir las variables hacia atrás $\beta_t(i)$ como:

$$b_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T \mid i_t = i, l)$$

la probabilidad de que la secuencia observada desde t+1 hasta T dándose el estado i en el instante t y bajo el modelo λ. Hay que apreciar aquí que i_t=i ya ha sido supuesto (esto no pasa en el caso de las variables hacia delante). Esta distinción se hace para que sea posible combinar las variables hacia delante y hacia atrás para producir resultados útiles, como veremos. Podemos resolver fácilmente $\beta_t(i)$ como hicimos para $\alpha_t(i)$:

1.
$b_t(i) = 1, 1 \le i \le N$

2. para t=T-1,T-2,...,1, $1 \le i \le N$
$$b_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) b_{t+1}(j)$$

3.
$$P(O \mid l) = \sum_{i=1}^{N} p_i b_i(O_1) b_1(i)$$

La demostración de estas ecuaciones es similar a la dada para las ecuaciones del procedimiento de ida.

El calculo de $P(O|\lambda)$ usando $\beta_t(i)$ conlleva del orden de N^2T operaciones. Es por esto que tanto el método para el calculo de las variables de ida (hacia delante) y de vuelta (hacia atrás) es eficiente para la obtención final de $P(O|\lambda)$.

4.4.2. Generación de secuencia de estados. Algoritmo de Viterbi.

Tenemos que encontrar una secuencia de estados $I = i_{1,}i_{2,...,}\ i_T$ tal que la probabilidad de ocurrencia de la secuencia observada $O = O_1, O_2,..., O_T$ desde la secuencia de estados sea mayor que desde otra secuencia de estados. En otras palabras, nuestro problema es encontrar I tal que maximice $P(O,I|\lambda)$. Hay un algoritmo famoso para conseguir esto llamado el algoritmo de Viterbi. Es un algoritmo inductivo en el cual en cada instante se mantiene la mejor secuencia de estados posible para cada uno de los N estados como estado intermedia para la secuencia de observación deseada $O = O_1, O_2,..., O_T$. En este camino finalmente se tiene la mejor trayectoria para cada una de los N estados así como el último estado para la secuencia de observación deseada. Finalmente seleccionamos la que tiene mayor probabilidad. El algoritmo de Viterbi trata de determinar entre todos los caminos cual es el optimo dada una definición de distancia. Este algoritmo será utilizado y así los es también en la realidad cuando nos veamos inmersos en la decodificación de códigos convolucionales y códigos cíclicos binarios o no binarios los cuales son empleados ampliamente hoy en día en sistemas de comunicaciones digitales. Este algoritmo fue propuesto en 1967 por Viterbi y constituye un algoritmo de máxima verosimilitud y nos proporciona un resultado optimo en cuanto a que minimiza la probabilidad de error.

Para hacernos una idea del algoritmo de Viterbi cuando se aplica al problema de estimación del estado óptimo podemos reformular el problema de la siguiente forma:

Consideremos la expresión para $P(O,I|\lambda)$; de

$$P(O \mid I, l) = b_{i_1}(O_1) b_{i_2}(O_2) \cdot\cdot b_{i_T}(O_T)$$

$$P(I \mid l) = p_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdot\cdot a_{i_{T-1} i_T}$$

tenemos:

$$P(O, I \mid l) = P(O \mid I, l) P(I \mid l)$$

$$= p_{i_1} b_{i_1}(O_1) a_{i_1 i_2} b_{i_2}(O_2) \cdot\cdot a_{i_{T-1} i_T} b_{i_T}(O_T)$$

Ahora aplicando logaritmos tenemos

$$U(i_1, i_2, ..., i_T) = -\left[\ln\left(p_{i_1} b_{i_1}(O_1)\right) + \sum_{t=2}^{T} \ln\left(a_{i_{t-1} i_t} b_{i_t}(O_t)\right) \right]$$

y fácilmente podemos ver que

$$P(O, I \mid l) = \exp(-U(i_1, i_2, ..., i_T))$$

como consecuencia el problema de estimación del estado óptimo que se formulaba como

$$\max_{\{i_t\}_{t=1}^{T}} P(O, i_1 \cdots i_T \mid l)$$

se convierte en el equivalente

$$\min_{\{i_t\}_{t=1}^{T}} U(i_1, i_2, ..., i_T)$$

Esta reformulación nos permite ahora ver términos como $-\ln(a_{i_j i_k} b_{i_k}(O_t))$, coste asociado en ir del estado i_j al estado i_k en el instante t.

El algoritmo de Viterbi para encontrar la secuencia de estados óptima puede ser descrita como sigue:

Supongamos que nos encontramos en el estado i que estamos y considerando visitar el estado j. Diremos que el peso de camino del estado i al j es $-\ln\left(a_{ij} b_j(O_t)\right)$ (esto es menos el logaritmo de ir del estado i al j y seleccionar el símbolo de observación O_t en el estado j) donde O_t es el símbolo observado seleccionado antes de visitar estado j- este es el mismo símbolo que aparece en la secuencia de observación $O = O_1, O_2, ..., O_T$. Cuando el estado inicial es seleccionado como i el peso correspondiente es $-\ln\left(p_i b_i(O_1)\right)$ - podemos llamar a este el peso inicial. Hay que hacer notar que este corresponde a multiplicar las correspondientes probabilidades. Ahora encontrar la secuencia optima es simplemente encontrar el camino de pesos mínimos a través del cual se da la secuencia de observación.

El algoritmo de Viterbi es esencialmente una aproximación de programación dinámica para minimizar $U(i_1, i_2, ..., i_T)$.

La implementaron del algoritmo de Viterbi es:

1. Iniciación.
   Para $1 \le i \le N$

   $$d_1(i) = -\ln(p_i) - \ln(b_i(O_1))$$

   $$y_1(i) = 0$$

2. Cálculos recursivos.
   Para $2 \le t \le T$ y $1 \le j \le N$

   $$d_t(i) = \min_{1 \le i \le N} \left[d_{t-1}(i) - \ln(a_{ij}) \right] - \ln(b_j(O_t))$$

   $$y_t(j) = \arg \min_{1 \le i \le N} \left[d_{t-1}(i) - \ln(a_{ij}) \right]$$

3. Terminación.

   $$P^* = \min_{1 \le i \le N} \left[d_T(i) \right]$$

$$q_T^* = \arg \min_{1 \le i \le N} [d_T(i)]$$

4. Remontándonos al estado optimo.
   Para $t = T-1, T-2, ..., 1$

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

$\ell^{(-P^*)}$ nos da la probabilidad de estado optimizada, y $Q^* = \{q_1^*, q_2^*, ...q_T^*\}$ es la secuencia de estados optima.

Una pequeña reflexión sobre los pasos anteriores nos mostrará que computacionalmente el algoritmo de Viterbi es similar al procedimiento que nos dan las variables de ida (hacia delante) y vuelta (hacia detrás), excepto por la comparación que se hacen para encontrar el valor máximo. Además su complejidad es también del orden de N^2T.

En este trabajo se utilizara el procedimiento de las variables de ida y vuelta para la generación del modelo de Markov con el método de las Fórmulas de re-estimación de Baum-Welch como a continuación se comenta, mientras que utilizaremos el algoritmo de Viterbi para la decodificación de códigos convolucionales y códigos cíclicos binarios que se utilizan en GSM sistema que se utilizará para probar el modelo de canal generado y en que podremos apreciar la ganancia en lo que se refiere a BER que se produce con la aplicación del algoritmo de Viterbi.

4.4.3. Generación del modelo de Markov.

En este apartado consideramos el problema de obtener el modelo oculto de Markov que hace que la secuencia observada utilizada como secuencia de entrenamiento sea propia o con características similares a la que generaría el modelo hallado. Hay dos métodos que trataremos dependiendo de cual sea la probabilidad elegida para la identificación del modelo:

- **Algoritmo de media K por segmentos (Segmental K-means algorithm):** en este algoritmo los parámetros del modelo $\lambda=(A,B,\pi)$ se ajustan para maximizar $P(O,I|\lambda)$ donde I aquí es la secuencia óptima dada por la solución del problema 2.
- **Fórmulas de re-estimación de Baum-Welch:** Aquí los parámetros del modelo $\lambda=(A,B,\pi)$ son ajustados mientras se este incrementando $P(O|\lambda)$ hasta que se alcance el máximo valor. El cálculo de $P(O|\lambda)$ conlleva la suma de $P(O,I|\lambda)$ sobre todas las posibles secuencias de estados I.

4.4.3.1. Algoritmo de media K por segmentos.

Este método nos lleva de λ^k a λ^{k+1} de tal forma que $P(O, I_k^* \mid l^k) \le P(O, I_{k+1}^* \mid l^{k+1})$ donde, I_k^* es el óptimo para la secuencia $O = O_1, O_2, ..., O_T$ y λ^k, es el obtenido de acuerdo con la solución que se ha dado al problema 2. Este criterio de optimización es llamado el criterio de estado óptimo con máxima probabilidad. Esta función

$P(O, I_k^* \mid l^k) = max_I P(O, I \mid l)$ es llamada la función de probabilidad del estado óptimo. En este método para educar al modelo se requieren un número de secuencias de observación. Supongamos que existen un número ω de tales secuencias. Cada secuencia $O=O_1,O_2,...O_T$ que están formadas por T símbolos. En vez de un número ω de esas secuencias nosotros podríamos dar solo una secuencia muy larga; en este caso nosotros la dividimos en segmentos de forma que tengamos un número conveniente de secuencias ω cada una de ellas de longitud T. Cada símbolo observado (O_i) es considerado como un vector de dimensión D (≥ 1). El algoritmo consta pues de los siguientes pasos:

1. Elección aleatoria de N símbolos (vectores de dimensión D) y asignación a cada uno de los ωT vectores de observación uno de los N vectores a los cuales su distancia Euclídea sea mínima. Ahora tenemos formados N grupos cada uno llamado estado (de 1 hasta N). Esta elección principal de vectores agrupados no deciden el modelo oculto de Markov final que nosotros obtendremos pero si incidirán sobre el número de iteraciones que se necesitan para la obtención del modelo de Markov. En nuestro caso tomamos N secuencias igualmente espaciadas de vectores característicos y tomamos un vector de cada una de las secuencias. El primer vector es tomado como el primer vector de la primera de las secuencias, el segundo como el segundo de la segunda de las secuencias y así. Naturalmente esto es hacer una elección inicial de grupos tan ampliamente distribuidos como sea posible y uno tiene la libertad de coger los vectores según sus necesidades.

2. Calcular las probabilidades iniciales y las probabilidades de transición: Para $1 \leq i \leq N$:

$$\hat{p}_i = \frac{Número\_de\_ocurrencias\_de\_\{O_1 \in i\}}{Número\_total\_de\_ocurrencias\_de\_O_1}$$

Para $1 \leq i \leq N$ y $1 \leq j \leq N$:

$$\hat{a}_{ij} = \frac{Número\_de\_ocurrencias\_de\_\{O_1 \in i, O_{t+1} \in j\} para\_todo\_t}{Número\_de\_ocurrencias\_de\_\{O_1 \in i\} para\_todo\_t}$$

3.Calcular el vector de medias y la matriz de covarianza para cada estado: Para $1 \leq i \leq N$

$$\hat{m}_i = \frac{1}{N_i} \sum_{O_t \in i} O_t$$

$$\hat{V}_i = \frac{1}{N_i} \sum_{O_t \in i} (O_t - \hat{m}_i)^T (O_t - \hat{m}_i)$$

4. Calcular la distribución de probabilidad de símbolos para cada vector de aprendizaje para cada estado (asumimos distribución gaussiana): Para $1 \leq i \leq N$

$$\hat{b}_i(O_t) = \frac{1}{(2p)^{D/2}\left|\hat{V}_i\right|^{1/2}} \exp\left[-1/2(O_t - \hat{m}_i)\hat{V}_i^{-1}(O_t - \hat{m}_i)^T\right]$$

5. Encontrar la secuencia de estados óptima I^* para cada secuencia de entrenamiento usando $\hat{l}_i = (\hat{A}_i, \hat{B}_i, \hat{p}_i)$ calculadas en los pasos 2 al 4 anteriormente. A un vector se le reasigna un estado si su asignación original es diferente del correspondiente estado estimado óptimo; por ejemplo si suponemos O_2 de la 5ª secuencia de entrenamiento fue asignado al estado 3 (al 3º grupo) y ahora nosotros encontramos que en la secuencia de estado óptima I^* correspondiente a la 5ª secuencia de entrenamiento, i^*_2 no es 3 sino 4. De aquí reasignamos O_2 de la 5ª secuencia al estado 4. Asignamos O_t al estado i si i_t^* (correspondiente a la k-esima secuencia de entrenamiento) es el estado i y esto es hecho para todas las secuencias de entrenamiento (desde k=1 hasta ω).

6. Si algún vector es reasignado a un nuevo estado en el paso 5, usar la nueva reasignación y repetir desde el paso 2 hasta el 6: de cualquier otra forma parar el algoritmo.

Se puede demostrar que este algoritmo converge a la función óptima de estado de probabilidad para un amplio margen de funciones de densidad de observación incluida la función de densidad gaussiana que nosotros hemos asumido.

4.4.3.2. Fórmulas de re-estimación de Baum-Welch.

Este método asume un modelo oculto de Markov inicial el cual es mejorado usando las formulas (dadas abajo) de tal forma que se maximice P(O|λ). Un modelo inicial de modelo oculto de Markov puede ser construido de cualquier forma pero podemos usar los cinco primeros pasos del algoritmo de medias K por segmentos antes descrito para obtener una estimación inicial razonable del modelo. De aquí en adelante asumiremos que el modelo inicial de modelo oculto de Markov es conocido. Este algoritmo maximiza P(O|λ) ajustando los parámetros de λ. Este criterio de optimización es llamado criterio de máxima verosimilitud. La función P(O|λ) es llamada función de verosimilitud. Antes de mostrar las formulas que proporcionan este método vamos a introducir algunos conceptos y notaciones que se requerirán en las formulas finales. Considera $\gamma_t(i)$ definida de la siguiente forma:

$$g_t(i) = P(i_t = i \mid O, l)$$

esto es la probabilidad de estar en un estado i en el instante t dándose como secuencia de observación $O=O_1,O_2,...,O_T$ y el modelo $\lambda=(A,B,\pi)$. Por la ley de Bayes tenemos:

$$g_t(i) = \frac{P(i = i, O \mid l)}{P(O \mid l)}$$

$$= \frac{a_t(i)b_t(i)}{P(O \mid l)}$$

donde $\alpha_t(i)$ y $\beta_t(i)$ han sido definidas anteriormente, $\alpha_t(i)$ esta estimada por $O_1,O_2,...,O_t$ y el estado i en el instante t, y $\beta_t(i)$ por $O_{t+1},...,O_T$ dado que estamos en el estado i en el instante t.

Podemos definir por otra parte $\xi_t(i,j)$ como:

$$x_t(i,j) = P(i_{t.} = i, i_{t+1} = j \mid O, l)$$

esto es la probabilidad de estar en el estado i en el instante t y haciendo una transición al estado j en el instante t+1, dado que la secuencia observada es $O_1,O_2,...,O_T$ y el modelo este determinado por $\lambda=(A,B,\pi)$. Usando la ley de Bayes se puede ver que

$$x\ (i,j) = \frac{P(i_{t.} = i, i_{t+1} = j, O \mid l)}{P(O \mid l)}$$

Pero $O=O_1,O_2,...,O_T$. Entonces

$$Numerador = P(i_t = i, O_1,...,O_t, O_{t+1},...,O_T, i_{t+1} = j \mid l)$$

$$= P(i_t = i_1 O_1,...,O_t \mid l) P(O_{t+1},...,O_T, i_{t+1} = j \mid l)$$

(ya que la cadena de Markov es causal)

Consideremos el segundo termino. El símbolo observado en el instante t+1 es O_{t+1} en el cual nosotros exigimos estar en el estado j; esto significa que el símbolo observado O_{t+1} es adquirido del estado j. De aquí tenemos

$$P(O_{t+1},...,O_T, i_{t+1} = j \mid l) = P(i_{t+1} = j, O_{t+1} \mid l) P(O_{t+1},...,O_T, i_{t+1} = j \mid l)$$

$$= a_{ij} b_j(O_{t+1}) b_{t+1}(j)$$

Combinando las formulas anteriores tenemos

$$x_t(i,j) = \frac{a_t(i) a_{ij} b_j(O_{t+1}) b_{t+1}(j)}{P(O \mid l)}$$

Aquí $\alpha_t(i)$ se estima de $O_1,...,O_t$, a_{ij} de la transición al estado j, $b_j(O_{t+1})$ del símbolo O_{t+1} en el estado i (cuando no hay transición en t=T). De forma similar $\xi_t(i,j)$ se sumara desde t=1 hasta T-1, obteniéndose de esta forma el número esperado de transiciones desde el estado i al estado j. Por tanto

$$\sum_{t=1}^{T-1} g_t(i) = Número\_esperado\_de\_trasiciones\_desde\_el\_estado\_i$$

$$\sum_{t=1}^{T-1} x_t(i,j) = Número\_esperado\_de\_trasiciones\_desde\_el\_estado\_i\_al\_estado\_j$$

Ya estamos preparados para presentar las formulas de re-estimación de Baum-Welch:

$$\hat{p}_i = g_t(i), \quad 1 \le i \le N$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} x_t(i,j)}{\sum_{t=1}^{T-1} g_t(i)}$$

$$\hat{b}_j(k) = \frac{\sum_{t=1}^{T} g_t(j)}{\sum_{t=1}^{T} g_t(j)}_{O_t=k}$$

La formula de restimación de π_i es simplemente la probabilidad de estar en el estado i en el instante t. La formula de a_{ij} es la razón de el número esperado de veces que se hace una transición desde el estado i al estado j y el número esperado de veces que se realiza una transición desde i. La formula para $b_k(i)$ es la razón entre el número de veces en que estando en el estado j se observa el símbolo O_k y el número esperado de veces que vamos a encontrarnos en el estado j. Hacer notar que el sumatorio en la última formula llega hasta t=T, algo que no sucede en la formula que nos da la estimación de a_{ij} ya que para hacer el calculo $\xi_t(i,j)$ necesitamos $\beta_{t+1}(i)$.

Si designamos al modelo inicial por l y a la estimación del modelo $\hat{l}$ constituido por los parámetros estimados por las formulas anteriores de Baum_Welch, entonces se puede ver que:

1. El modelo inicial l es un punto crítico de la función de verosimilitud en cuyo caso $\hat{l} = l$ o bien,

2. $P(O \mid \hat{l}) > P(O \mid l)$, esto es hemos encontrado un modelo mejor ya que la secuencia de observación $O=O_1,O_2,...,O_T$ es mejor ajustada.

Así seguimos de forma iterativa computando hasta que $P(O|\lambda)$ sea máxima. Así se resuelve el problema 3.

Hemos visto que el algoritmo de Baum-Welch maximiza $P(O|\lambda)$, dada por (4). Cada término en el sumatorio de esta ecuación esta entre 0 y 1 y por lo tanto el sumatorio (que es un producto de muchos términos de estos) será muy pequeño y habrá tantos términos pequeños como el número de posibles secuencias de estado haya – para un ordenador capaz de computar tales términos de forma individual, debería ser capaz de almacenarlos con la precisión más pequeña que haya en todos ellos. Si T=100 debería haber alrededor de 200 números (entre 0 y 1) para ser multiplicados y por lo tanto cada término puede irse por

debajo de la precisión del computadora utilizado. El tomar logaritmos no es ninguna ayuda ya que $P(O|\lambda)$ esta dado por un sumatorio de pequeños términos. Quizás un escalado apropiado podría ser usado para resolver este problema.

En el otro lado el algoritmo de k medias por segmentos maximiza $P(O|\lambda)$, que es calculada usando el algoritmo de Viterbi. Aquí podemos usar logaritmos para protegernos del producto de números pequeños ya que no se ven involucrados en ningún sumatorio. El algoritmo de k medias por segmentos es además preferido porque la mayoría del tiempo nosotros estamos interesados en las ocurrencias de la secuencia de observación a partir de la mejor (la óptima) secuencia de estado y considerando que la ocurrencia de la secuencia de observación a través de todos las posibles secuencias de estados no es la representación deseada del problema en la mayoría de los casos. Además el algoritmo de k medias por segmentos requiere mucho menos computación cuando es comparado con el método de Baum-Welch.