Índice de contenido

1. Introducción

1.1 Reconocimiento de habla aislada

El reconocimiento del habla es un problema que ha sido estudiado de forma intensiva durante los últimos cuarenta años, tiempo en el cual se han desarrollado diferentes estrategias de aproximación al mismo. Hacia mediados de la década de los sesenta se las primeras tentativas ponían de manifiesto la alta variabilidad de los patrones, incluso para una misma palabra pronunciada por una misma persona. La clasificación estadística usando modelos clásicos, representados por densidades de probabilidad estáticas, se demostraron incapaces de alcanzar tasas de error razonable. Con el tiempo se difundieron técnicas derivadas de la programación dinámica y la cuantización vectorial, como el Dinamic Time Warping, que permitían contrastar contra un mismo patrón secuencias de distintas longitudes y que presentaban distintas velocidades de variación respecto a distintas partes del patrón de referencia. Paralelamente, estudios biológicos y psicológicos abrían un abanico de mejoras para el modelado de la fuente de voz y la extracción de características relevantes de la señal.

Los modelos ocultos de Markov irrumpieron en el campo del reconocimiento de habla en los años ochenta, y se erigieron en protagonistas en poco tiempo. Enlazaban por una parte con lo mejor de la programación dinámica, en el sentido de que aportaban un marco de modelo de estados que tenía en cuenta de un modo inherente el carácter temporal secuencial de las observaciones y su variabilidad, cubriendo así las carencias que presentaban los métodos puramente estadísticos. Por otra parte, estos modelos están entroncados en el ámbito de la estadística, en el seno del cual existen métodos de optimización como el

algoritmo EM que permiten reestimar de forma consistente los parámetros del modelo para adecuarlo a las observaciones. Inicialmente utilizados en el contexto del habla aislada, pronto se empezaron a utilizar también en el reconocimiento de habla continua, donde aún hoy día mantienen cierta vigencia.

Pero es en el primer problema, el reconocimiento de habla aislada donde se ponen de manifiesto de un modo más simple sus características definitorias, y es por tanto el enfoque adoptado en el proyecto.

1.2 Motivación y Objetivos

Una vez planteado el problema, el presente proyecto persigue poner de manifiesto el funcionamiento de los modelos ocultos de Markov, y mostrarlos en su lugar en relación con la clasificación estadística. Se persigue también poner de manifiesto en el campo de estos modelos la importancia del algoritmo EM, y realizar un acercamiento ordenado y suficiente a las soluciones de la casuística que rodea el uso de los modelos de Markov como clasificadores, todo ello respaldado por una implementación de los algoritmos que pueda ser realmente utilizada.

1.3 Estructura del proyecto

El proyecto se estructura básicamente en tres bloques principales: una aproximación a la clasificación estadística, teoría de los modelos de markov y su funcionamiento y finalmente aplicación a voz. Estos bloques de contenido se distribuyen en los siete capítulos subsiguientes

- En el capitulo dos haremos una aproximación a la clasificación estadística y a la construcción de clasificadors. Daremos un repaso a los criterios de optimización más conocidos para entrenamiento supervisado y no supervisado, y en el contexto de este último presentaremos el algoritmo EM.
- En el capítulo tres exponemos los modelos ocultos de Markov a partir de una explicación previa de las cadenas de Markov. En este mar-

co previo se da resolución a problemas como la estimación de parámetros o el cálculo de la verosimilitud que serán más complejos cuando refieran a modelos ocultos de Markov.

- El capítulo cuatro se centra en la resolución del primero y más sencillo de los problemas, el cálculo de la verosimilitud de una secuencia de muestras respecto al modelo, y se introducen las piezas clave para el trabajo con modelos ocultos de markov, las varibale hacia delante y hacia atrás.
- En el capítulo cinco resolvemos el problema del calculo del camino óptimo, la secuencia de estados por la que con más probabilidad pasa un modelo para generar una observación dada
- En el capítulo seis presentamos el último y más importante de los problemas, la reestimación de parámetros, y entroncamos con el algoritmo EM expuesto en el capítulo dos.
- El capítulos siete explica la problemática propia del reconocimiento de habla, y cómo los modelos de markov enlazan con la misma.
- Finalmente el capítulo octavo muestra algunas conclusiones y posibles lineas futuras.

Las simulaciones están al final de cada capítulo, y se desarrollan según lo hace la teoría poder obtener una mejor comprensión.

2. Clasificación estadística

2.1 Introducción

En este capítulo expondremos las bases de la Teoría de la Decisión, y haremos un recorrido por los métodos más importantes para la construcción de estimadores.

Expondremos en primer lugar cómo obtener clasificadores usando métodos de entrenamiento supervisado, que requieren de la información de clase asociada a cada muestra de la población. En este contexto encontraremos los estimadores de máxima verosimulitud (MLE, *Maximum Likelyhood Estimator*), máxima probabilidad a posteriori (MAP, *Maximum A Posteriori*) y los de máxima información mutua (MMIE, *Maximum Mutual Information Estimator*).

En segundo lugar nos acercaremos a los métodos denominados de aprendizaje no supervisado, como el algoritmo *K-means* para cuantización vectorial, entre los que se encuentra el algoritmo EM (*Expectation Maximization*). Veremos como el algoritmo EM es fundamental para el modelado de f.d.p multimodales, y sentaremos las bases que en capítulos posteriores nos permitirán reestimar los parámetros de los modelos ocultos de Markov.

2.2 El problema de la clasificación

La clasificación es el acto de separar observaciones unas de otras, asignándolas a distintos grupos o clases, en base a un conjunto de criterios. La clasificación se basa en la creencia de que existen patrones diferenciados y discernibles dentro de un conjunto de observaciones, y su objetivo es encontrar cuáles son, y con cuál se corresponde cada muestra. La clasificación tiene por pilares fundamentales dos ideas fundamentales:

i. existen patrones, lo que cristaliza en el concepto de clase.

ii. se pueden distinguir unas observaciones de otras, que da lugar al concepto de discriminador.

En general las clases y los criterios de clasificación están muy relacionados. Cuando conocemos a la perfección las clases es que ya hemos clasificado todo el espacio. Cuando clasificamos muestras según un criterio creamos clases.

La clasificación estadística permite trabajar cómodamente con los conceptos de clase y discriminador, separándolos en cierto modo. La clasificación estadística define una clase como una distribución o densidad de probabilidad sobre el espacio de observaciones. Los discriminadores se basarán en cantidades del campo probabilístico, como la verosimilitud, la información mutua, etc...

La clasificación estadística se adapta bien a un gran número de situaciones reales. No en todos los problemas de clasificación que se nos presenta disponemos de la misma cantidad de información acerca del proceso que genera la salida que observamos. Si nos atenemos a este baremo de desconocimiento, la clasificación estadística ocupa un amplio rango donde su aplicación es posible y o acertada.

Vamos a presentar los conceptos básicos de la clasificación estadística. Supongamos que tenemos una e observación x, que pertenece necesariamente a alguna clase del conjunto $\omega=\{\omega_1 ... \omega_N\}$. Definimos:

i. $p(x/\omega_i)$ Verosimilitud. Probabilidad de que la clase i genere una observación x. No es sino el valor que toma para x la densidad o distribución de probabilidad asociada a esa clase.

ii. $p(\omega_i)$ Probabilidad a prior. Es la probabilidad de que una clase genere la observación, antes de saber cual será el valor de la observación generada.

iii. $p(\omega_i/x)$ Probabilidad a posteriori. Es la probabilidad de que una clase haya generado una observación x conocida.

iv. $p(x)$ Probabilidad de observar x. Como la muestra tiene que pertenecer a alguna de las clases, se puede escribir como

$$p(x) = \sum_{\forall k} p(x/\omega_k)\, p(\omega_k) \tag{2.1}$$

Las cuatro se relacionan a través del teorema de Bayes

$$p(\omega_i/x) = \frac{p(x/\omega_i)\, p(\omega_i)}{p(x)} \tag{2.2}$$

2.3 La regla de decisión de Bayes.

La Teoría de la Decisión de Bayes es central en el campo de la clasificación estadística. Se basa en la premisa de que el problema de la decisión puede tratarse en términos estadísticos, y en que todos los parámetros relevantes asociados a las clases son conocidos. Esto significa que por el momento no nos vamos a preocupar de cómo hemos conseguido estimar los parámetros de las clases. Dicho esto traemos de nuevo a colación la expresión de la probabilidad a posteriori (2.2).

La probabilidad a priori de una clase $p(\omega_i)$ no nos dice gran cosa. Si intentamos adivinar a qué clase pertenece una observación antes de conocer dicha observación, la única elección sensata es pensar que, probablemente, la observación pertenecerá a aquella clase con mayor probabilidad a priori. Esta elección siempre se decantará por la misma clase, para cualquier observación, lo cual nos dice que no es muy útil. Parece más lógico basar las posibles reglas de decisión en el valor de la probabilidad a posteriori, la probabilidad de que una obser-

vación pertenezca a una clase, ¡una vez que conocemos dicha observación! El sentido común nos dice que la mejor regla basada en $p(\omega_i/x)$ será la que entrone como elegida a la clase con mayor probabilidad a posteriori de haber generado la muestra. formalmente

$$\delta(x) = \underset{i}{argmax}\, p(\omega_i/x) \tag{2.3}$$

$\delta(x)$ es por lo tanto una regla de decisión que a cada observación le asigna el índice de una clase, en concreto la de aquella para la cual la probabilidad a posteriori es máxima. Como durante la fase de clasificación las distribuciones asociadas a las clases, así como las probabilidades a priori, no varían[1] podemos considerar la probabilidad de p(x) como constante. Entonces podemos simplificar la regla de decisión, expresándola en función de la verosimilitud y de la probabilidad a priori. Formalmente

$$p(x) = \sum_{\forall k} p(x/\omega_k)\, p(\omega_k) \equiv cte$$
$$\delta(x) = \underset{i}{argmax}\, p(x/\omega_i)\, p(\omega_i) \tag{2.4}$$

El sentido común no se equivoca en esta ocasión, ya que podemos demostrar que esta regla de decisión minimiza el riesgo en la elección. Para demostrarlo, vamos a suponer que tenemos un conjunto de clases $\omega=\{\omega_1 \omega_{1...} \omega_N\}$, por lo que el espacio de salida de la regla de decisión se define como $\delta(x) \in [1,N]$. Vamos a definir como $l(\delta(x)=k/\omega_i)$ la función de pérdidas, que representa las pérdidas que se derivan de elegir como correcta la clase k cuando realmente la muestra pertenecía a la clase *i*. Por lo tanto, el riesgo condicionado se expresa como

$$r(\delta(x)=k/x) = \sum_{\forall i} l(\delta(x)=k/\omega_i)\, p(\omega_i/x)$$

1 Esto es cierto durante la fase de clasificación. Si consideramos p(x) tal como aparece en la expresión (2.1), tenemos que aceptar que durante la fase de entrenamiento este valor podría cambiar.

El riesgo condicionado representa el riesgo que se corre al asignar una observación x a una clase k, siguiendo una regla de decisión $\delta(x)$ sobre un conjunto ω de clases. El riesgo total vendrá dado por

$$R = \sum_{\forall k} \int r(\delta(x)=k/x)\, p(x)\, dx$$

El riesgo total representa el riesgo esperado al aplicar una regla de decisión dada. Llegados a este punto, trataremos de minimizar este riesgo total. El riesgo total puede ser optimizado optimizando el riesgo condicionado $r(\delta(x)=k/x)$ para todo valor de x. Si escogemos una función de pérdidas tal que

$$l(\delta(x)=k/\omega_i) = \begin{cases} 0 & k=i \\ 1 & k\neq i \end{cases}$$

Esta función de pérdidas asigna la misma pérdida a cualquier error de clasificación: todos cuestan lo mismo. Se conoce como función de pérdidas simétrica o cero-uno. Usando esta función de pérdidas, el riesgo condicionado toma la forma

$$r(\delta(x)=k/i) = \sum_{\forall i} l(\delta(x)=k/\omega_i)\, p(\omega_i/x) = \sum_{\forall i\neq k} p(\omega_i/x) = 1 - p(\omega_k/x)$$

Por lo tanto para minimizar el riesgo condicionado tenemos que escoger como válida aquella clase k para la cual la probabilidad a posteriori $p(\omega_k/x)$ sea máxima, que es precisamente el criterio que sigue la regla de decisión de Bayes.

2.4 Funciones discriminantes

Una forma de ver la tarea de clasificadión es la siguiente. Tenemos una serie de observaciones a clasificar, y un conjunto de N clases en las que clasificarlas. Podemos representar la clasificación como un proceso en el cual usamos N funciones discriminantes $d_i(x)$. Una funcion discriminante asociada a una clase calcula el parecido de una muestra con la misma. Entonces la elección de clase se reduce a escoger la clase asociada a la función discriminante con mayor valor para la muestra x. formalmente

$$d_k(x) > d_i(x) \quad \forall i \neq k \tag{2.5}$$

Para el caso de un clasificador bayesiano, la regla de decisión es tal que minimiza el riesgo condicional. Como en el caso de la función discriminante lo que buscamos es que sea máxima para la clase correcta, definimos la función discriminante para asociada a clasificadores bayesianos como

$$d_k(x) = -r(\delta(x) = k / x)$$

Que es equivalente en términos de máximos y mínimos a

$$d_k(x) = p(\omega_k / x) \tag{2.6}$$

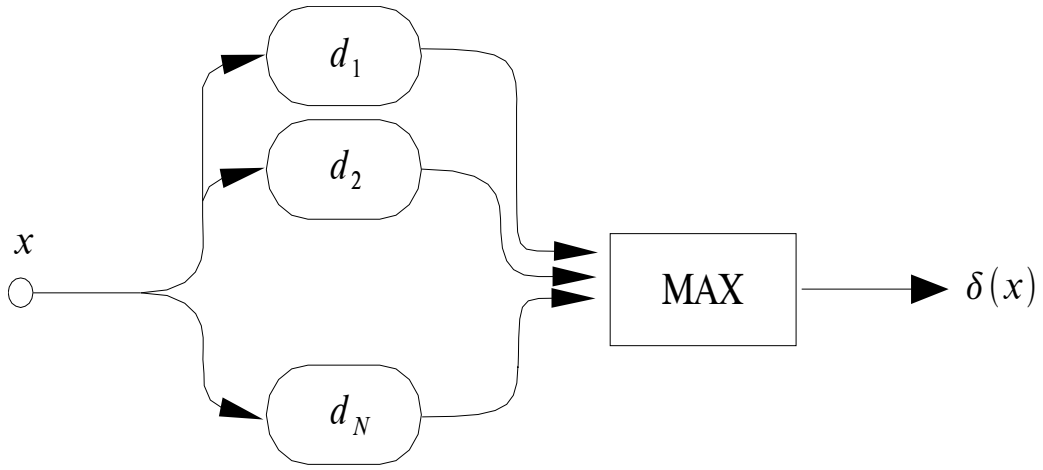


Ilustración 2.1 Diagrama de bloques de un clasificador basado en funciones discriminantes.

2.5 Estimadores y Métodos de aprendizaje.

Las distribuciones que representan clases han tenido que salir de algún sitio. En general las distribuciones se obtienen de las mismas observaciones (o de un subconjunto de ellas, llamado conjunto de entrenamiento) usando estimadores.

Las distribuciones pueden ser paramétricas (por ejemplo una gaussiana tiene dos parámetros, media y varianza) o no paramétricas (se construyen por frecuencia relativa de aparición de valores). En realidad estas últimas tienen tantos parámetros como muestras contenga el conjunto de entrenamiento.

Los métodos de aprendizaje se dividen en dos grandes grupos: supervisados y no supervisados. Los supervisados son aquellos en que, al menos para un conjunto de observaciones, disponemos información sobre a qué clase pertenecen. Formalmente, disponemos de muestras etiquetadas $[x, \omega_k]$. En los no supervisados, podemos prescindir de esa información. No es necesario saber a qué clase pertenecen las muestras.

2.6 Métodos supervisados

2.6.1 Estimación de Máxima Verosimilitud (MLE)

Es un método de estimación supervisado, es decir, requiere de muestras etiquetadas para poder ser aplicado. MLE (*Maximum Likelihood Estimation*) es el método más usado para estimar distribuciones por su gran eficiencia y suele utilizarse con distribuciones parametrizadas. Su meta es encontrar el conjunto de parámetros que maximizan la probabilidad una clase de haber generado los datos que le son propios, es decir, la verosimilitud de las observaciones que pertenezcan a una clase $p(x/\omega_k)$.

Sea $x=\{x_1 x_2 ... x_T\}$ un conjunto de observaciones estadísticamente independientes que sabemos pertenecen a la clase *k*. Si llamamos θ_k al conjunto de parámetros que determinan la distribución asociada a la clase k, la MLE viene dada por

$$\theta_{MLE}^{k} = \underset{\theta_k}{argmax} \prod_{\forall t} p(x_t/\theta_k) = \underset{\theta_k}{argmax} \sum_{\forall t} \ln p(x_t/\theta_k) \qquad (2.7)$$

Para aplicarlo a una distribución concreta sólo hay que sustituir $p(x/\theta_k)$ por su valor y derivar respecto a algún parámetro.

2.6.2 Estimación de Máxima probabilidad A Posteriori

Método supervisado igual que MLE, permite hacer estimaciones razonables de los parámetros cuando el conjunto de entrenamiento es tan limitado que no es suficiente para una estimación directa de máxima verosimilitud. A cambio es necesario realizar suposiciones acertadas o tener conocimientos previos sobre los parámetros iniciales de la distribución, es decir, hay que asignar una probabilidad a priori al vector de parámetros. Por eso se llama a posteriori, porque lo que realmente vamos a optimizar para cada clase es

$$p(\theta_k/x) = \frac{p(x/\theta_k)\,p(\theta_k)}{p(x)}$$

Supuesto que la probabilidad de x no depende de los parámetros de la distribución de esa clase, el estimador queda como

$$\theta^k_{MAP} = \underset{\theta_k}{argmax}\left[p(\theta_k/x)\right] = \underset{\theta_k}{argmax}\left[\ln p(\theta_k) + \sum_{\forall t} p(x_t/\theta_k)\right] \qquad (2.8)$$

Podemos observar que cuando disponemos de muchas muestras (para valores altos de T) la estimación MAP (*Maximum a Posteriori*) converge a una estimación MLE.

2.6.3 Métodos Discriminativos

Hasta ahora los métodos expuesto tratan de adecuar hacer que los parámetros de cada clase reflejen lo mejor posible la distribución que siguen las observaciones que pertenecen a esa clase. Estamos preocupándonos de que la distribución asociada a cada clase se adecue bien a los datos de esa clase, pero ¿cómo se adecua a los datos de otras clases? Podría darse el caso de que datos que pertenecen a otra clase arrojaran valores de verosimilitud *mayores* que datos de la propia clase, con lo que estaríamos dando pie a errores de clasificación. Los métodos discriminativos utilizan datos etiquetados pertenecientes a *todas* las clases para entrenar cada clase, consiguiendo así una máxima discriminación entre modelos[2] para mejorar el rendimiento en el reconocimiento de patrones.

Hay distintos métodos que se rigen por este objetivo, entre ellos el MMIE y las redes neuronales.

2 Es más que posible que los modelo así estimados arrojen, de facto, valores de verosimilitud respecto a las muestras que les son propias menores que los que se obtendrían con modelos estimados según MLE o MAP. Y, aún así, clasificarán con menor error las observaciones.

2.6.3.1 Estimación de Máxima Información Mutua (MMIE)

Partimos de la regla de Bayes, e intentaremos encontrar parámetros tal que maximicen $p(\omega_i/x)$. Esto nos lleva a una estimación denominada como de máxima verosimilitud condicionada (CMLE, *Conditional Maximum Likelyhood Estimation*)

$$\theta_{CMLE} = \underset{\theta}{argmax}\, p(\omega_i/x) \tag{2.9}$$

Nótese que ahora no nos centramos en encontrar los parámetros θ_k de una clase en concreto, sino de todas las clases (θ). Cabe preguntar por qué se denomina de máxima información mutua. Como las probabilidades a priori para cada clase no dependen de los parámetros de que intentamos estimar, podemos obviarlas al optimizar, con lo que maximizar $p(\omega_i/x)$ será equivalente a maximizar la información mutua instantánea entre x y ω_i .

$$p(\omega_i/x) = \frac{p(x/\omega_i)\,p(\omega_i)}{p(x)} \qquad I(x,\omega_i) = \ln\left(\frac{p(x,\omega_i)}{p(x)\,p(\omega_i)}\right)$$

$$si \quad p(\omega_i/\theta)=cte \;\; \forall i \quad \Rightarrow \quad \underset{\theta}{argmax}\, p(\omega_i/x) = \underset{\theta}{argmax}\, I(x,\omega_i) = \theta_{MMIE}$$

Si tenemos en cuenta que p(x) ya no se puede considerar constante (ya que puede escribirse de manera que dependa de los parámetros que buscamos estimar), la expresión a optimizar es

$$p(\omega_i/x) = \frac{p(x/\omega_i)\,p(\omega_i)}{p(x)} = \left\langle p(x)=\sum_{\forall k} p(x/\omega_k)\,p(\omega_k)\right\rangle =$$

$$= \frac{p(x/\omega_i)\,p(\omega_i)}{p(x/\omega_i)\,p(\omega_i) + \sum_{\forall k\neq i} p(x/\omega_k)} = \frac{1}{1 + \dfrac{\sum_{\forall k\neq i} p(x/\omega_k)}{p(x/\omega_i)\,p(\omega_i)}}$$

De modo que maximizar $p(\omega_i/x)$ es equivalente a

$$\theta_{MMIE} = \underset{\theta}{argmax}\left[p(\omega_i/x)\right] = \underset{\theta}{argmin}\left[\frac{\sum_{\forall k \neq i} p(x/\omega_k)}{p(x/\omega_i)\, p(\omega_i)} \right]$$

Donde la minimización se resuelve por métodos de gradiente.

2.7 Métodos no supervisados

Hasta ahora hemos dado un repaso general a los conceptos básicos de clasificación estadística y hemos visto los métodos más importantes de aprendizaje supervisado. Los métodos no supervisados tienen que lidiar con datos incompletos, ya que la información de clase es desconocida. Los más importantes son la cuantización vectorial (VQ) y el algoritmo EM.

2.7.1 Cuantización vectorial

La cuantización es el proceso de aproximar señales de amplitud continua por símbolos discretos. Si cuantizamos una sola señal, hablamos de cuantización escalar. Si cuantizamos varias señales a la vez, trataremos entonces de cuantización vectorial. Al conjunto de símbolos que usamos para aproximar la señal lo llamamos libro de claves (*codebook*), y a cada símbolo palabra clave (codeword). Cualquier tipo de cuantización lleva asociada una distorsión, que es el error cometido al aproximar los valores por símbolos. Por lo tanto los pilares fundamentales del proceso de cuantización vectorial serán

i. Una medida de la distorsión, para poder evaluarla y minimizarla.

ii. Un método para generar cada palabra clave del libro.

Denominamos $q(\boldsymbol{x})$ la función de cuantización, que mapea un vector de valores contínuos $\boldsymbol{x}$ en uno de los vectores prototipo o palabra clave $\boldsymbol{z} \in \boldsymbol{Z}=\{\boldsymbol{z}_1 \boldsymbol{z}_2 ... \boldsymbol{z}_M\}$, donde Z es el libro de claves de tamaño M. Si definimos una función distancia $d(\boldsymbol{x}, \boldsymbol{z}_k)$ entonces la regla de decisión a seguir será la que minimice la distorsión, es decir

$$q(\boldsymbol{x})=\boldsymbol{z}_k \quad si\ y\ solo\ si \quad k=\underset{i}{argmin}\left[d(\boldsymbol{x}, \boldsymbol{z}_i)\right]$$

La medida de distorsión o distancia puede ser de muchos tipos: cuadrática, de Mahalanobis, perceptual, etc... dependiendo de la aplicación. Usando una distorsión y una regla de decisión, el espacio queda dividido en regiones o células cuyos puntos cumplen $\boldsymbol{x} \in C_i \Leftrightarrow p(\boldsymbol{x})=\boldsymbol{z}_i$.

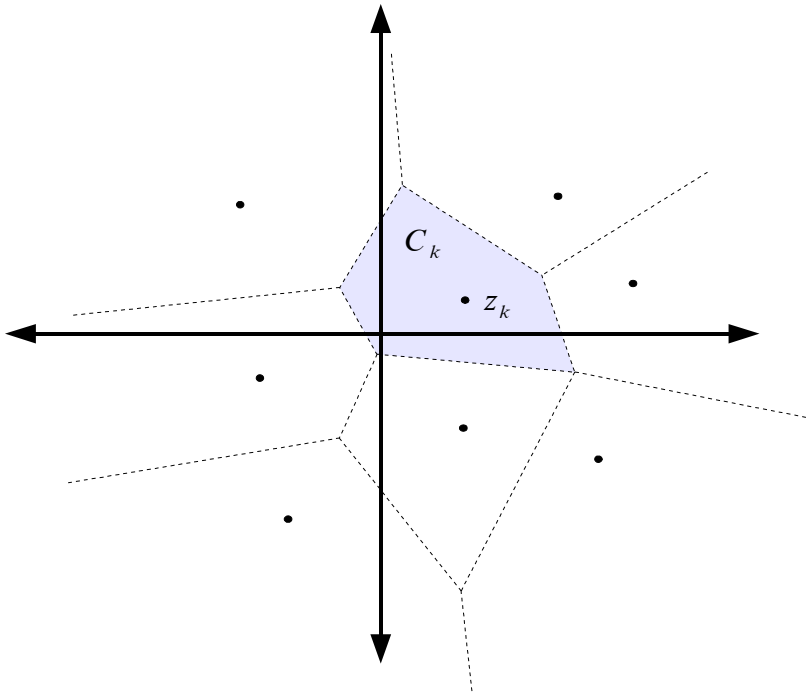


figura 2.2 División de un espacio bidimensional mediante cuantización vectorial usando como métrica la distancia euclidea.

2.7.2 Algoritmo k-means

Este algoritmo se aplica en cuantización vectorial y persigue minimizar la distorsión global, buscando por tanto un libro de claves que sea óptimo en ese sentido. Lo logrará de manera recursiva. La distorsión media puede expresarse como

$$D = E[d(\boldsymbol{x},\boldsymbol{z})] = \sum_{\forall i} p(\boldsymbol{x}\in C_i)E\left[d(\boldsymbol{x},\boldsymbol{z}_i)/\boldsymbol{x}\in C_i\right]$$

Para optimizar esta función lo haremos en dos pasos. Primero, usando un codebook inicial, calcularemos que regiones del espacio ocupa cada célula (o en otras palabras, usaremos el libro de claves inicial para determinar a partir de p(x) que palabra clave asignamos cada valor de entrada). Aparecen por lo tanto dos condiciones necesarias para alcanzar un óptimo. Después intentaremos hallar unos nuevos vectores prototipos tal que minimicen la distorsión de las observaciones que pertenecen a cada una de las muestras. Y así, repetidas veces, hasta que la distorsión se estabilice.

Para que el primer paso permita que vayamos optimizando la distorsión, la regla para decidir que clave se le asigna a un vector de entrada x (o lo que es lo mismo, a qué célula pertenece) ha de ser tal que minimice la distorsión $d(\boldsymbol{x},\boldsymbol{z})$ (algo que ya veníamos haciendo por sentido común).

Para escoger los nuevos centroides o vectores prototipos, los escogeremos tal que minimicen la distorsión dentro de la célula correspondiente. La distorsión media D_i en la célula C_i viene dada por

$$D_i = \frac{1}{T}\sum_{1}^{T} d(\boldsymbol{x}_t,\boldsymbol{z}_i)$$

El mínimo de esta expresión depende del tipo de función distancia. Para una función de distancia cuadrática $d(\boldsymbol{x},\boldsymbol{z}) = (\boldsymbol{x}-\boldsymbol{z})^t(\boldsymbol{x}-\boldsymbol{z})$ el mínimo se aparece para

$$\hat{\boldsymbol{z}}_i = \frac{1}{K_i}\sum_{\forall k} x_k$$

donde K_i es el número de observaciones dentro de la la célula C_i, y $\boldsymbol{x}=\left(\boldsymbol{x}_1\boldsymbol{x}_2...\boldsymbol{x}_{K_i}\right)$

2.7.3 El algoritmo EM

2.7.3.1 Introducción

El algoritmo EM (Expectation- Maximization) se engloba dentro de los métodos de aprendizaje no supervisados. Esto quiere decir que los datos que poseemos son insuficientes para realizar una estimación directa de verosimilitud, usando MLE, ya que desconocemos a que clase pertenece cada dato. Lo que hace este algoritmo es dividir en dos pasos el proceso. En primer lugar, partiendo de un número de clases dado, cada una con sus valores para los parámetros, calculamos la verosimilitud de las observaciones respecto a cada clase. En segundo lugar, usando el valor de esa verosimilitud para ponderar la pertenencia a cada clase de las observaciones, reestimamos los parámetros de las clases según MLE. Y vuelta a empezar, hasta que la verosimilitud deje de variar.

Puede verse en cierta forma similar al algoritmo *k-means*, usado para VQ. En el algoritmo k-means escogemos unos centroides razonables iniciales, y mediante algún tipo de disttancia asignamos cada observación al símbolo más cercano, para luego recalcular los centroides con las observaciones que han sido clasificadas como pertenecientes a un símbolo, obteniendo un nuevo *codebook*. El algoritmo EM es similar, salvo que la medida de distancia es la verosimilitud $p(x/\omega_i)$ y depende por tanto de la distribución usada para modelar cada clase, y la asignación de las observaciones a las distintas clases es blanda, no del tipo todo o nada que se utiliza en *k-means*.

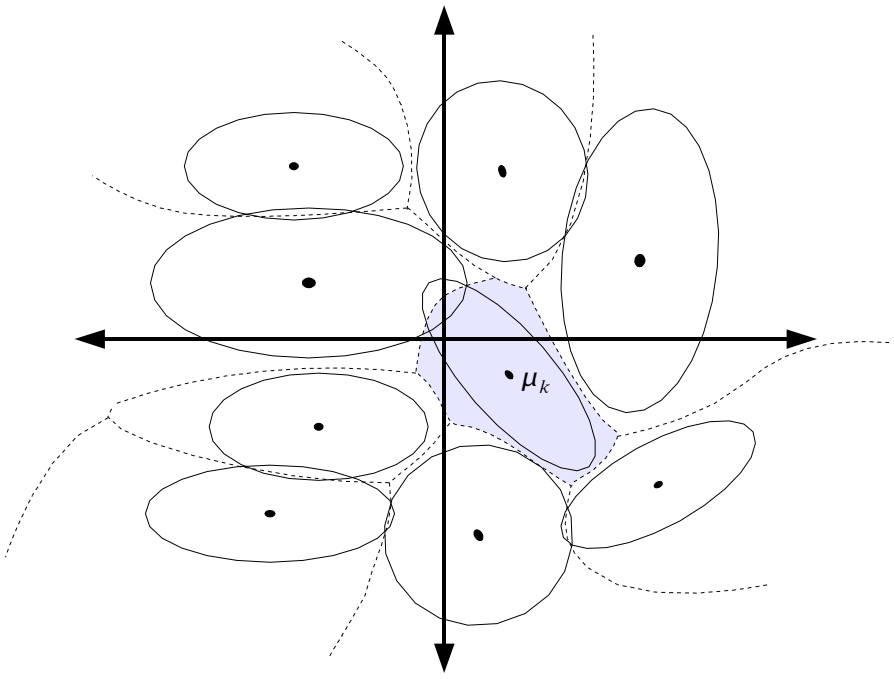


figura 2.3 Partición de un espacio bidimensional con nueve f.d.p gaussianas

2.7.3.2 Demostración y exposición

El algoritmo EM persigue maximizar la verosimilitud de las observaciones respecto a los parámetros del modelo. Podemos definimos por tanto como función a minimizar respecto a los parámetros θ la verosimilitud logaritmica negativa.

$$f(\theta) = -\ln p(y/\theta) \tag{2.10}$$

Una visión distinta al enfoque clásico expuesto en [1] de la demostración de algoritmo es la siguiente. Para explicar la optimización de la verosimilitud nos basaremos en el concepto de *función mayorizante (majorizing function)*.

Función mayorizante

Una función se dice mayorizante de otra si cumple dos condicio-

nes. La primera de ellas consiste en que ambas funciones tomen el mismo valor en un punto dado. La segunda que en el resto de los puntos, el valor de la función mayorizante esté por encima del de la original. Si denotamos como $f(\theta)$ la función original y como $g(\theta)$ la función mayorizante, ésta última ha de cumplir

$$g_i(\theta^{\,i}) = f(\theta^{\,i}) \tag{2.11}$$

$$g_i(\theta) \geq f(\theta) \tag{2.12}$$

El cumplir estas dos condiciones conlleva que si somos capaces de hallar un nuevo valor para los parámetros tal que minimizen la función mayorizante, estaremos forzando la minimización de la función original, y que, de algun modo, estamos "bajando el techo".

$$\theta^{\,i+1} = \underset{\theta}{argmin}\, g_i(\theta)$$

$$f(\theta^{\,i+1}) \leq g_i(\theta^{\,i+1}) \leq g_i(\theta^{\,i}) = f(\theta^{\,i}) \tag{2.13}$$

El quid de la cuestión reside en que la función mayorizante sea más fácil de optimizar que la función original.

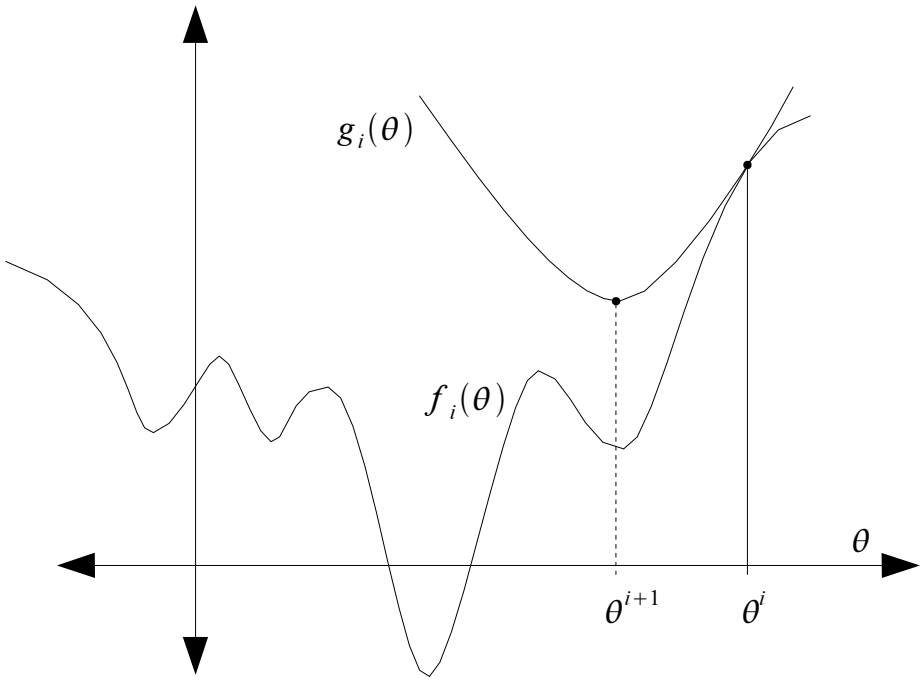


figura 2.4 La función mayorizante siempre es mayor o igual que la función objetivo, por lo que si encontramos un valor de los parámetros que disminuya el valor de la función mayorizante, estaremos disminuyendo el valor de la función original

Buscando una función mayorizante para la verosimilitud

Buscaremos ahora una función mayorizante que lo sea de la verosimilitud. En principio podría pensarse que buscar una función mayorizante no es tarea fácil, pero un par de ideas nos ayudarán a encauzar la búsqueda. En adelante denotaremos como $E_z\{\cdot\}$ la esperanza respecto a la distribución asociada a un vector aleatorio z. En caso de que la distribución de z esté condicionada a otro vector aleatorio y escribiremos $E_{z/y}\{\cdot\}$. Para hacer patente la dependencia respecto a los parámetros del modelo, se usará la expresión $E_{z/y,\theta}\{\cdot\}$, que incluye explícitamente θ .

La desigualdad de Jensen afirma que para cualquier función cóncava *h(x)*, siendo *x* un vector aleatorio, se cumple que

$$E\{h(x)\} \leq h(E\{x\}) \tag{2.14}$$

La demostración se basa en que si llamamos *d(x)* al plano tangente a la curva *h(x)* en el punto *E{x}*, por ser *h(x)* cóncava y por tanto menor o igual que *d(x)* se cumple que

$$h(E\{x\}) = d(E\{x\}) = E\{d(x)\} \geq E\{h(x)\}$$

Con los conceptos anteriores en mente, podemos definir una función tal que

$$g_i(\theta) = f(\theta^i) - E_{z/y,\theta^i}\left\{\ln\left[\frac{p(y,z/\theta)}{p(y,z/\theta^i)}\right]\right\}$$

Donde z es un vector aleatorio tal que su densidad de probabilidad condicionada respecto a y depende exclusivamente de los parámetros θ del modelo. Veamos si esta función es mayorizante de la verosimilitud logarítmica. Sin más que sustituir los θ por θ^i, vemos que en ese punto el logaritmo se anula y por tanto se cumple la igualdad (2.11)

$$g_i(\theta^i) = f(\theta^i) - E_{z/y,\theta^i}\left\{\underbrace{\ln\left[\overbrace{\frac{p(y,z/\theta^i)}{p(y,z/\theta^i)}}^{1}\right]}_{0}\right\} = f(\theta^i)$$

Queda demostrar que para todos los demás puntos, la función mayorizante escogida está por encima de la original. Utilizando en primer lugar la desigualdad de Jensen (2.14) a raiz de la concavidad de la función logaritmo, y posteriormente teorema de Bayes (2.2) se sigue que

$$g_i(\theta) \geq f(\theta^i) - \ln\left[E_{z/y,\theta^i}\left\{\frac{p(y,z/\theta)}{p(y,z/\theta^i)}\right\}\right] =$$

$$= f(\theta^i) - \ln\left[E_{z/y,\theta^i}\left\{\frac{p(y,z/\theta)}{p(z/y,\theta^i)\cdot p(y/\theta^i)}\right\}\right]$$

$$= f(\theta^i) - \ln\left[\frac{1}{p(y/\theta^i)} \underbrace{\int p(y,z/\theta)dz}_{p(y/\theta)}\right]$$

$$= -\ln p(y/\theta^i) - \ln\left[\frac{p(y/\theta)}{p(y/\theta^i)}\right] = -\ln p(y/\theta) = f(\theta)$$

Cumpliendo por ende la función escogida la desigualdad que le faltaba (2.12) para ser considerada función mayorizante de la verosimilitud. Demostrado que optimizando $g_i(\theta)$ optimizamos $f(\theta)$, sólo nos interesa la parte de la función mayorizante que varíe con los parámetros θ del modelo. Esta parte optimizable la llamaremos $\bar{g}_i(\theta)$, y en el caso del algoritmo EM toma el valor

$$\bar{g}_i(\theta) = E_{z/y,\theta^i}\left[\ln p(y,z/\theta)\right]$$

¿Será esta función más fácil de optimizar que la original? ¿Qué representa z? Como ya dijimos, z es un vector aleatorio tal que su densidad de probabilidad condicionada respecto a y depende exclusivamente de los parámetros θ del modelo. Esto significa que z no depende de la salida, sino más bien al contrario: la salida depende de z. z es útil porque

explicita una dependencia de y, porque pone de manifiesto que existen resultados intermedios que son un escalón previo a la hora de que el modelo genere una salida.

En el caso de una mezcla de gaussianas, este punto intermedio sería la elección de que gaussiana generará la salida; en el caso de un modelo oculto de Markov, será el camino de estados que sigue el modelo para generar la salida.

El elegir qué representa z es determinante para adecuar el algoritmo EM a un problema concreto. Parece probable que siempre que esta elección evoque a un resultado intermedio, refleje un proceso oculto pero necesario para generar la salida, se conseguirá que el cálculo de la función mayorizante sea más simple que el de la original.

2.7.3.3 Resumen del algoritmo

De todo lo expuesto se sigue que el algoritmo EM reducirá monotónicamente la verosimilitud logaritmica negativa en cada iteración:

i. Inicialización

$$\theta^0 = dado$$

ii. Recursión

- Paso de Promediado (Expectation Step)

$$\bar{g}_i(\theta) = E_{z/y,\theta^i}\left[\ln p(y, z/\theta)\right]$$

- Paso de Maximización (Maximization Step)

$$\theta^{i+1} = argmin_\theta \bar{g}_i(\theta)$$

2.7.3.4 El algoritmo EM aplicado a una mezcla de Gaussianas.

Lo veremos más claro con un ejemplo. Supongamos que las salidas $y=\{y_1 y_2 .. y_T\}$ que observamos proceden de una mezcla de N gaussianas con densidad $N(x,\mu_k,\sigma_k)$. Cada vez que la mezcla genera una salida, sucede en dos pasos:

i. Una variable multinomial que determina que gaussiana generará la salida con probabilidades

$$\{c_1 c_2 ... c_k ... c_N\}$$

ii. La gaussiana escogida generará una muestra según su densidad de probabilidad

$$N(x,\mu_k,\sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2\right)$$

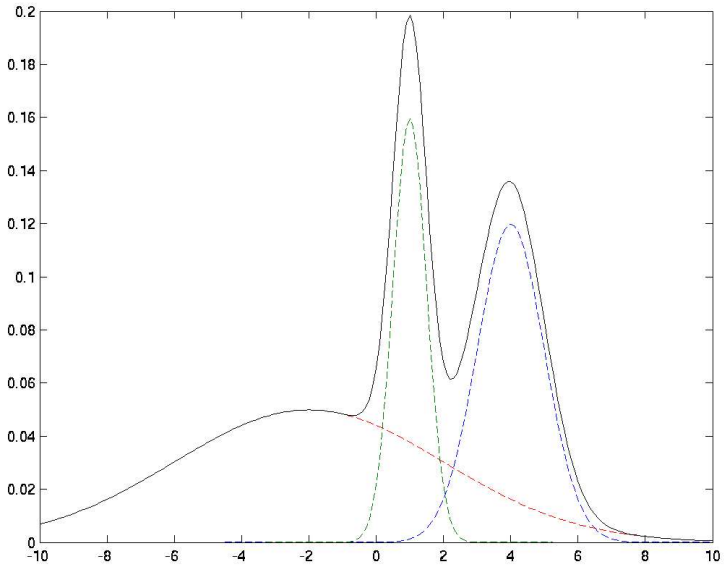


figura 2.5 Densidad de probabilidad de una mezcla de tres gaussianas. Las f.d.p de las gaussianas componentes de la mezcla aparecen en trazo discontínuo.

Veamos. En este caso los parámetros θ del modelo serían las probabilidades $\{c_1 c_2 .. c_N\}$, así como las medias y varianzas de las variables de la mezcla. Si asignamos z a la variable oculta que nos dice qué componente ha sido elegida, escribiremos

$$\bar{g}_i(\theta) = E_{z/y,\theta^i}\{\ln p(y,z/\theta)\} = \sum_{\forall t}\sum_{\forall k} \ln p(y_t, z=k/\theta)\cdot p(z=k/y_t,\theta_i)$$

Esta es la expresión a optimizar. Para que nos sea más sencillo, descompondremos usando Bayes el término que depende de los nuevos parámetros. Con esto conseguiremos separar la ecuación en dos términos cada uno de los cuales depende exclusivamente de un subconjunto de los parámetros.

$$\sum_{\forall t}\sum_{\forall k} \ln p(y_t, z=k/\theta)\cdot p(z=k/y_t,\theta_i) =$$

$$\sum_{\forall t}\sum_{\forall k} \ln (p(y_t/z=k,\theta)\cdot p(z=k/\theta))\cdot p(z=k/y_t,\theta_i) =$$

$$\sum_{\forall t}\sum_{\forall k} \ln p(y_t/z=k,\theta)\cdot p(z=k/y_t,\theta_i) + \sum_{\forall t}\sum_{\forall k} \ln p(z=k/\theta)\cdot p(z=k/y_t,\theta_i)$$

Llegados a este punto, empezaremos por optimizar respecto a los coeficientes de la mezcla, c_k . Claramente el primer término no depende de estos parámetros (puesto que es una distribución condicionada precisamente los mismos), por lo que desaparecerá al derivar. Dado que la probabilidad de obtener un z dado respecto a los parámetros es fija e igual a los coeficientes de la mezcla

$$p(z=k/\theta) = c_k$$

Si derivásemos $g_i(\theta)$ e igualáramos a cero directamente obtendríamos que el óptimo se alcanza para $c_k=0 \ \forall k$.

$$\frac{\partial \bar{g}_i(\theta)}{\partial c_k} = \sum_{\forall t} \frac{\partial \ln p(z=k/\theta)}{\partial c_k} p(z=k/y_t,\theta_i) = \sum_{\forall t} \frac{1}{c_k} p(z=k/y_t,\theta_i) = 0$$

Hay que complementar la ecuación original con un término lagrangiano que asegure que los coeficientes suman la unidad. haciendo esto obtenemos

$$p(z=k/\theta) = c_k$$

$$\frac{\partial}{\partial c_k}\left(\bar{g}_i(\theta) + \lambda\left[\sum_{\forall k} p(z=k/\theta) - 1\right]\right) =$$

$$\frac{\partial}{\partial c_k}\left(\sum_{\forall t}\sum_{\forall k} \ln p(z=k/\theta)\cdot p(z=k/y_t,\theta_i)\right) + \lambda \frac{\partial}{\partial c_k}\left[\sum_{\forall k} p(z=k/\theta) - 1\right] =$$

$$\sum_{\forall t} \frac{1}{c_k} p(z=k/y_t,\theta_i) + \lambda = 0$$

Si sumamos para todos los valores de k, obtenemos $\lambda=-T$, por lo que el óptimo se obtiene para

$$c_k = \frac{1}{T}\sum_{\forall t} p(z=k/y_t,\theta_i)$$

El siguiente paso es optimizar respecto a los parámetros de las gaussianas. En este caso el único término que cambia es el primero. Notando que $p(y_t/z=k,\theta_i)$ no es sino la verosimilitud de y_t respecto a la k-ésima gaussiana.

$$p(y_t/z=k,\theta_i) = N(y_t,\mu_k,\sigma_k)$$

Derivando con respecto a las medias

$$\frac{\partial \bar{g}_i(\theta)}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\left(\sum_{\forall t}\sum_{\forall k} \ln p(y_t / z=k,\theta)\cdot p(z=k/y_t,\theta_i)\right) =$$

$$= \frac{\partial}{\partial \mu_k}\left(\sum_{\forall t}\sum_{\forall k}\left[\ln \sigma_k - \frac{(y_t-\mu_k)^2}{2\sigma_k^2} + c\right] p(z=k/y_t,\theta_i)\right) =$$

$$= \sum_{\forall t}\left[\frac{y_t-\mu_k}{\sigma_k^2}\right] p(z=k/y_t,\theta_i) = 0$$

$$\mu_k = \frac{\sum_{\forall t} p(z=k/y_t,\theta_i)\, y_t}{\sum_{\forall t} p(z=k/y_t,\theta_i)}$$

Derivando con respecto a las varianzas

$$\frac{\partial \bar{g}_i(\theta)}{\partial \sigma_k^2} = \frac{\partial}{\partial \sigma_k^2}\left(\sum_{\forall t}\sum_{\forall k} \ln p(y_t / z=k,\theta)\cdot p(z=k/y_t,\theta_i)\right) =$$

$$= \frac{\partial}{\partial \sigma_k^2}\left(\sum_{\forall t}\sum_{\forall k}\left[-0.5\ln \sigma_k^2 - \frac{(y_t-\mu_k)^2}{2\sigma_k^2} + c\right] p(z=k/y_t,\theta_i)\right) =$$

$$= \sum_{\forall t}\left[-\frac{1}{2\sigma_k^2} + \frac{(y_t-\mu_k)^2}{2\sigma_k^2}\right] p(z=k/y_t,\theta_i) = 0$$

$$\sigma_k^2 = \frac{\sum_{\forall t} p(z=k/y_t,\theta_i)(y_t-\mu_k)^2}{\sum_{\forall t} p(z=k/y_t,\theta_i)}$$

El único término que necesitamos calcular para utilizar estas expresiones es la probabilidad a posteriori $p(z=k/y_t,\theta_i)$. Por el teorema de Bayes (2.2) resolvemos

$$p(z=k/y_t,\theta) = \frac{p(y_t/z=k,\theta_i)\,p(z=k/\theta_i)}{p(y_t/\theta_i)} = \frac{p(y_t/z=k,\theta_i)\,p(z=k/\theta_i)}{\sum_{\forall j} p(y_t/z=j,\theta_i)\,p(z=j/\theta_i)}$$

2.7.3.5 Simulaciones

Simulación 1

Para las simulaciónes del algoritmo EM hemos usado como fuente una mezcla de tres gaussianas. Los parámetros de la mezcla que se usan para generar las observaciones son

$$c=\begin{bmatrix}0.2 & 0.3 & 0.5\end{bmatrix} \quad \begin{cases} \mu_1=1 & \sigma_1^2=0.25 \\ \mu_2=4 & \sigma_2^2=1 \\ \mu_3=-2 & \sigma_3^2=16 \end{cases}$$

Usando esta fuente se ha generado una secuencia de observaciones de longitud N=1000, con la que se entrenará un segundo modelo utilizando el algoritmo EM.

Este segundo modelo tiene todos sus parámetros (incluso el número de componentes de la mezcla) escogidos al azar dentro de un rango razonable. Es de notar que como el algoritmo alcanza óptimos locales el punto inicial es importante de cara a lograr un buen resultado (veremos un contraejemplo en la siguiente simulación).

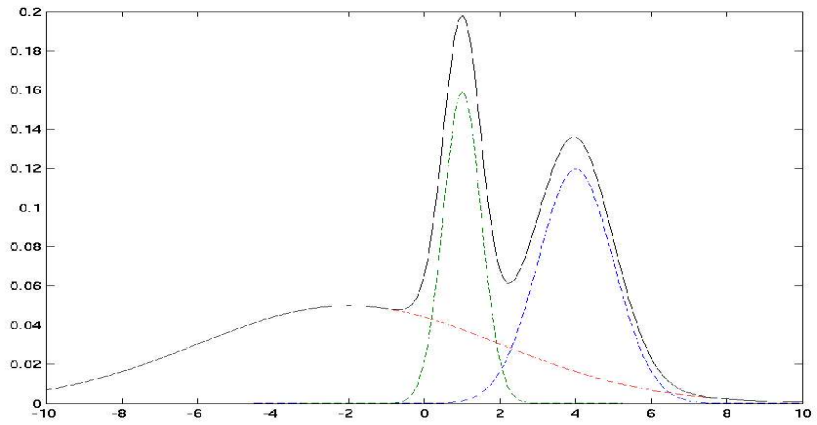


figura 2.6 Densidad de probabilidad real de la fuente, una mezcla de tres gaussianas.

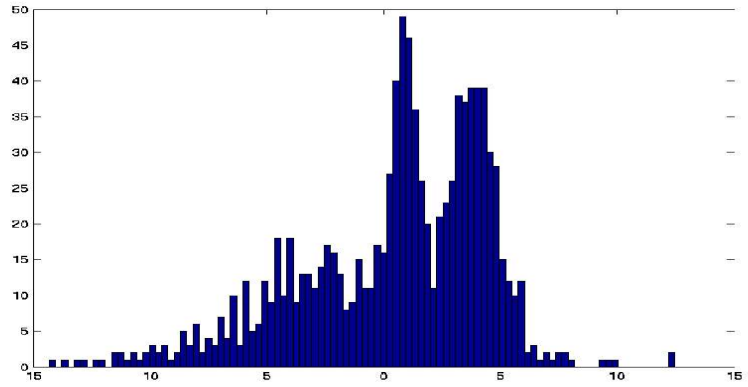


Figura 2.7 Histograma de las observaciones generadas por la fuente (N=1000)

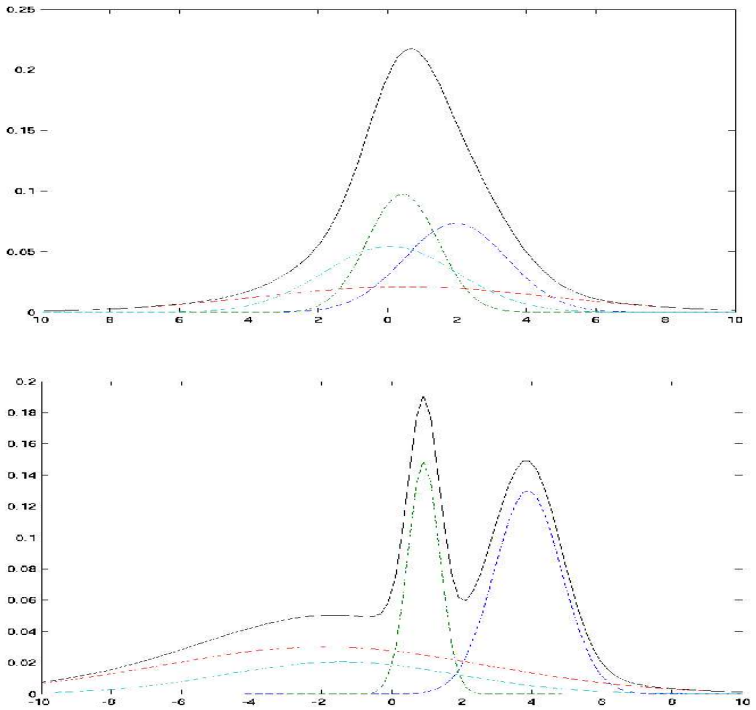


Figura 2.8 Arriba, mezcla de cuatro gaussianas a entrenar, con sus parámetros iniciales. Debajo, resultado del entrenamiento tras 20 iteraciones. Nótese como la densidad de probabilidad ha convergido a la original, alcanzando el máximo global.

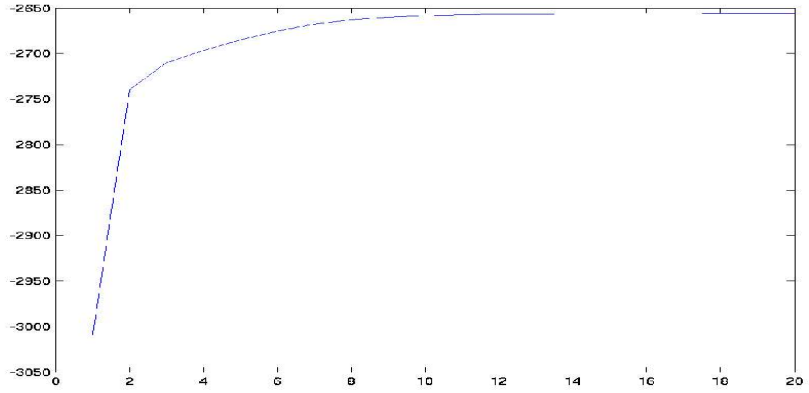


figura 2.9 Verosimilitud logarítmica de las observaciones respecto al segundo modelo durante el entrenamiento. Podemos observar como para cada iteración, el valor es cada vez mejor (monotónicamente creciente).

2.8.2 Algoritmo EM. Simulación 2

En este caso, a causa de la elección de los parámetros iniciales, sólo se alcanza un máximo local.

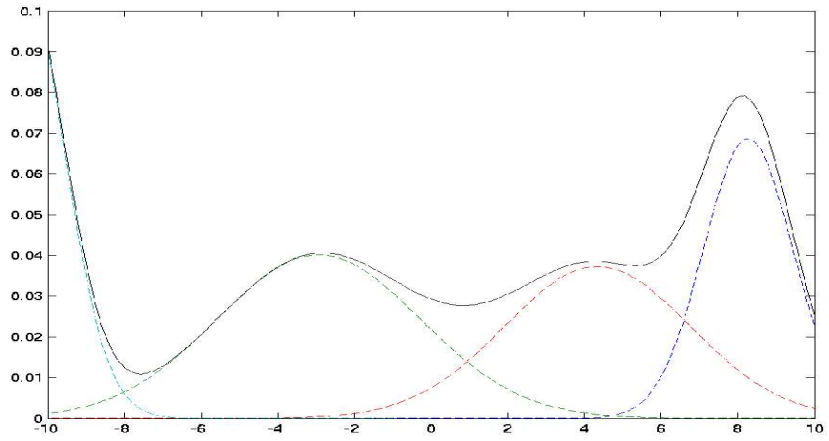


figura 2.10 Parámetros iniciales del modelo de la segunda simulación, con cuatro gaussianas.

3. Introducción a los modelos ocultos de Markov

3.1 Procesos de Markov

Un proceso de markov de orden n será aquel proceso estocástico en que, una vez conocidas n salidas consecutivas del proceso, la distribuciónes de las siguientes salidas quedan completamente determinadas. Así, un proceso aleatorio será un proceso de Markov de primer orden si la distribución de x_n , dado su pasado infinito, sólo depende de de la muestra precedente, x_{n-1} . Un ejemplo de proceso de Markov de primer orden sería el descrito por las ecuaciones

$$x_n = \rho \cdot x_{n-1} + w_n$$

donde w_n sería una secuencia de gausianas, de media cero, independientes e idénticamente distribuidas con densidad

$$f_w(w) = \frac{1}{\sqrt{(2 \cdot \pi \cdot \sigma_0^2)}} \exp(-w^2 / 2\sigma_0^2)$$

La densidad condicional de x_n dado x_{n-1} vendría dada por

$$f_{x_n / x_{n-1}}(x_n / x_{n-1}) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_0^2}} \exp\left(\frac{-(x_n - \rho \cdot x_{n-1})^2}{2\sigma_0^2} \right)$$

Donde podemos comprobar que la probabilidad condicionada rspecto a una secuencia de observaciones pasadas, sólo depende de la observación inmediatamente anterior.

$$f_{x_n / x_{n-1}, x_{n-2}, \ldots}(x_n / x_{n-1}, x_{n-2}, \ldots) = f_{x_n / x_{n-1}}(x_n / x_{n-1}) \qquad (3.1)$$

El resultado más importante que se deriva de la definición es la de que la densidad conjunta de cualquier conjunto de muestras de un proceso de markov de primer orden, queda completamente determinada por las densidades de primer orden f_{x_n} y las densidades condicionales $f_{x_n/x_{n-1}}$.

Para dejar patente este hecho, considérese el conjunto de muestras $\{x_0, x_1, \ldots x_n\}$. Para cualquier proceso estocástico podemos escribir (omitiendo los argumentos para simplificar la notación) que:

$$f_{x_0, x_1, \ldots x_n} = f_{x_n/x_{n-1}, \ldots x_0} \cdot f_{x_{n-1}/x_{n-2} \ldots x_0} \cdot \ldots \cdot f_{x_1/x_0} \cdot f_{x_0}$$

Si el proceso de Markov la expresión se reduce a

$$f_{x_0, x_1, \ldots x_n} = f_{x_n/x_{n-1}} \cdot f_{x_{n-1}/x_{n-2}} \cdot \ldots \cdot f_{x_1/x_0} \cdot f_{x_0} \tag{3.2}$$

la densidad conjunta para las muestras a partir de un instante dado *n*, dependerá de x_n pero no lo hará de ningún valor de la secuencia previo a *n*.

3.2 Cadenas de Markov

3.2.1 Introducción

Cuando un la salida de un proceso de Markov toma sólo un conjunto de valores discretos, hablamos de cadenas de Markov. Basándonos en las definiciones dadas hasta ahora, un proceso aleatorio que tome sólo valores discretos será una cadena de Markov si satisface

$$p(x_n = S_i / x_{n-1} = S_j, x_{n-2} = S_k, \ldots) = p(x_n = S_i / x_{n-1} = S_j)$$

Hay cierta terminología que se asocia comunmente a la descripción de cadenas de Markov. Cuando $x_n = S_i$ se dice que la cadena de Markov está en el estado i. En adelante, denotaremos estos instantes en que se produce el cambio de estado con el índice t (en lugar de n), y el estado para un instante t determinado como q_t (en lugar de x_n). Así de acuerdo con la nueva notación escribiríamos:

$$p(q_t = j / q_{t-1} = i, q_{t-2} = k, \ldots) = p(q_t = j / q_{t-1} = i)$$

es *la probabilidad concreta* de que se produzca una transición desde el estado i (en el que el sistema estaba en el instante t-1) hacia el estado j (en el instante t). Es decir, la probabilidad condicionada de que la salida del proceso tome el valor *j* dado que la anterior muestra tomó el valor *i*.

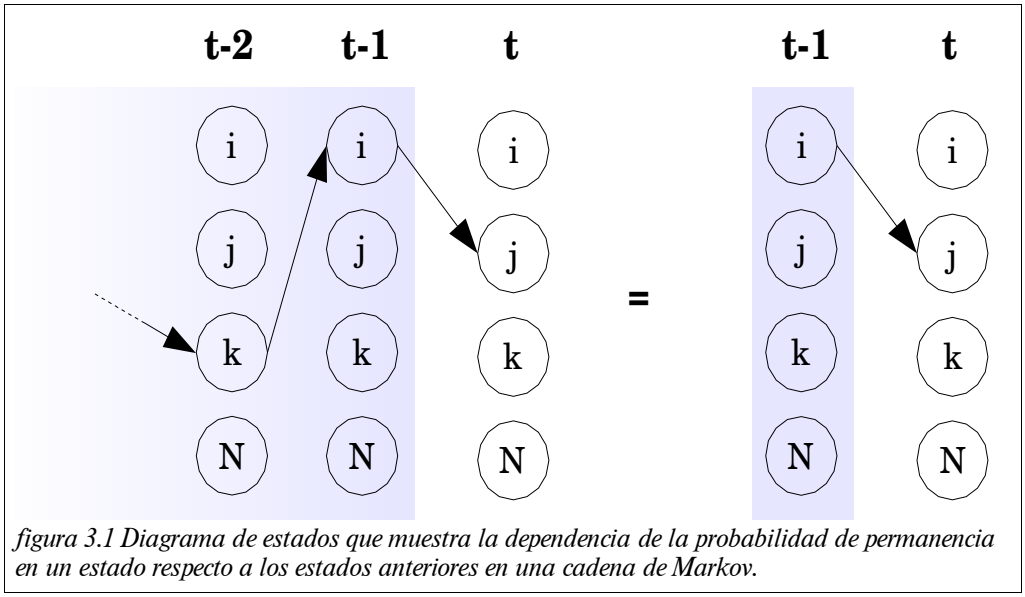


figura 3.1 Diagrama de estados que muestra la dependencia de la probabilidad de permanencia en un estado respecto a los estados anteriores en una cadena de Markov.

3.2.2 La matriz estocástica

Si consideramos aquellas cadenas de primer orden en que las probabilidades de transición entre estados son independientes del tiempo, podemos definir una matriz $\boldsymbol{A}$ constante que contenga dichas probabilidades. Siendo N el número de estados en la cadena de Markov, los elementos de $\boldsymbol{A}$ se definen como

$$a_{ij} = p(q_t = j / q_{t-1} = i) \quad , \qquad 1 \leq i, j \leq N \tag{3.3}$$

Los coeficientes de la matriz cumplen por definición las restricciones estocásticas

$$a_{ij} \geq 0 \qquad\qquad \sum_{j=1}^{N} a_{ij} = 1$$

Esta matriz se conoce como *matriz estocástica*. Permite utilizar cómodamente la representación en términos de estados, ampliamente difundida y es muy útil ya que pone de manifiesto de forma intuitiva las propiedades del proceso. La matriz de estocástica nos permite obtener facilmente las probabilidades de alcanzar un segundo estado desde un primero, sin más que multiplicar. Para exponer más claramente las propiedades de $\boldsymbol{A}$ lo haremos con un ejemplo. Supongamos que vamos a modelar el clima de una ciudad. En primer lugar asociaremos un estado a cada característica meteorológica.

Estado	Situación meteorológica
1	Despejado
2	Nublado
3	Lluvioso

La matriz de transición nos dará la probabilidad de pasar de una situación a otra. Para ser coherentes haremos que sea algo más fácil alcanzar un estado Lluvioso pasando por Nublado que directamente desde

Despejado, y viceversa. Entonces $\boldsymbol{A}$ podría ser algo como

$$A = \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

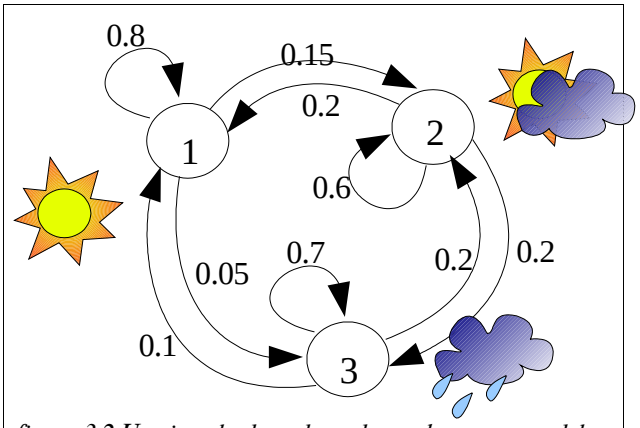


figura 3.2 Un ejemplo de cadena de markov para modelar el clima

De manera matricial podemos calcular la probabilidad de acabar en un estado, dadas unas probabilidades iniciales de ocupación de estado en el instante anterior usando la expresión

$$\boldsymbol{q}_{t+1} = \boldsymbol{A}^T \cdot \boldsymbol{q}_t \tag{3.4}$$

En nuestro ejemplo si un día está nublado, $\boldsymbol{q}_t = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. La probabilidad para cada uno de los estados climatológicos al día siguiente se obtiene aplicando (3.4).

$$\boldsymbol{q}_{t+1} = \boldsymbol{A}^T \cdot \boldsymbol{q}_t = \begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.15 & 0.6 & 0.2 \\ 0.05 & 0.2 & 0.7 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.2 \end{bmatrix}$$

3.2.3 Tiempo medio de permanencia en un estado

La probabilidad de cambio para un estado i dado se describe mediante una variable de Bernouilli con probabilidad de éxito

$p = \sum_{\forall j \neq i} a_{ij}$. Así, el tiempo medio de permanencia en un estado o duración media del estado coincidirá con la media de la distribución geométrica asociada a la anterior. A esta cantidad habría que añadirle en nuestro caso una unidad, ya que partimos de un estado inicial i dado, y para eso hay que estar en ese estado

$$\textit{duracion media observada} \;=\; 1+\mu_{geom} \;=\; 1+\frac{(1-p)}{p} \;=\; \frac{1}{p} \;=\; \frac{1}{\sum_{\forall j \neq i} a_{ij}}$$

Cuando la duración promedio de un fenómeno puede estimarse, se convierte en una ayuda valiosa a la hora de acotar los parámetros de diseño del modelo, ya que nos determinaría la diagonal de la matriz estocástica. Aplicado a nuestro ejemplo, el número medio de días consecutivos en que el tiempo permanece nublado, lluvioso o soleado se corresponde con

Situación atmosférica	Probabilidad de cambio de estado	Nº medio de dias consecutivos
Soleado	0.2	5
Nublado	0.4	2.5
Lluvioso	0.3	3.333...

3.2.4 Probabilidades de transición de orden superior

¿Y las probabilidades *N* días después? En ocasiones es necesario conocer las probabilidades de transición de orden k . Estas probabilidades se definen como

$$p_{ij}^{k} \;=\; p(q_t = j / q_{t-k} = i)$$

y son fácilmente computables en base a las de primer orden. En principio podríamos pensar en hallar todos las posibles secuencias de estados que uniesen el estado inicial con el final en k pasos y sumar sus respec-

tivas probabilidades de ocurrencia. El calculo con la matriz de transición nos ahorra trabajo. Si llamamos $\boldsymbol{q}_t$ al vector que contiene las probabilidades de encontrarse en los distintos estados para un instante t dado, tenemos que

$$\begin{aligned} \boldsymbol{q}_{t+1} &= \Pi \times \boldsymbol{q}_t \\ \boldsymbol{q}_{t+2} &= \Pi \times \boldsymbol{q}_{t+1} = \Pi \times \left(\Pi \times \boldsymbol{q}_t\right) \\ &\vdots \\ \boldsymbol{q}_{t+k} &= \Pi^k \times \boldsymbol{q}_t \end{aligned}$$

donde se ha introducido la matriz de transición $\Pi = A^T$. Las probabilidades de transición de orden k serán por tanto las componentes de la matriz Π^k .

3.2.5 Probabilidades asociadas a los estados límite

Aunque no será de especial interés en posteriores desarrollos hacer notar que bajo ciertas condiciones, aparece un comportamiento asintótico de las probabilidades de ocupación de estados. Si esto sucede las mismas se denominan asociadas a los estados límite.

$$\pi = \lim_{k \to \infty} \boldsymbol{p}_k = \lim_{k \to \infty} \Pi^{k-k_0} \times \boldsymbol{p}_{k_0} \tag{3.5}$$

Estas probabilidades límite, si existen, se usan normalmente como condición inicial del sistema, equivaliendo al supuesto de que éste lleva mucho tiempo funcionando. Alcanzado este estado estable se cumple que

$$\Pi \times \pi = \pi \quad \Rightarrow \quad (\Pi - I) \times \pi = 0$$

Como las columnas de $\Pi - I$ suman cero, su determinante será nulo y el sistema tiene solución. Luego *siempre* existirá un autovector asociado al autovalor uno. Puede demostrarse (aunque no es trivial)

que la matriz de transición tiene todos sus autovalores (que pueden ser complejos) con módulo menor o igual que uno. Para que exista un *único vector límite independiente de las condiciones iniciales*, será necesario que:

i. No tenga autovalores complejos con módulo uno, ya que rotarían siempre sin alcanzar nunca un valor estable.

ii. No exista más de un autovalor con valor uno, ya que entonces el vector límite dependería de las condiciones iniciales.

En nuestro ejemplo se cumplen las condiciones, y existen esos estados límite.

$$autovalores(\Pi) = \begin{bmatrix} 1 & 0.6823 & 0.4117 \end{bmatrix}$$

$$autovectores(\Pi) = \begin{bmatrix} -0.7326 & -0.7651 & 0.2852 \\ -0.5037 & 0.1355 & -0.8052 \\ -0.4579 & 0.6295 & 0.5199 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.4324 & 0.2973 & 0.2703 \end{bmatrix}^t$$

Los valores obtenidos se pueden hallar tambien iterando, aumentándo el valor de k para acercarnos al límite (3.5). Si existen estados límite, todas las columnas de Π^k convergen a π, independizando así el vector $\boldsymbol{p}^k$ de las condiciones iniciales del sistema $\boldsymbol{p}^0$. Las simulaciones siguientes muestran la convergencia para los valores de nuestro ejemplo.

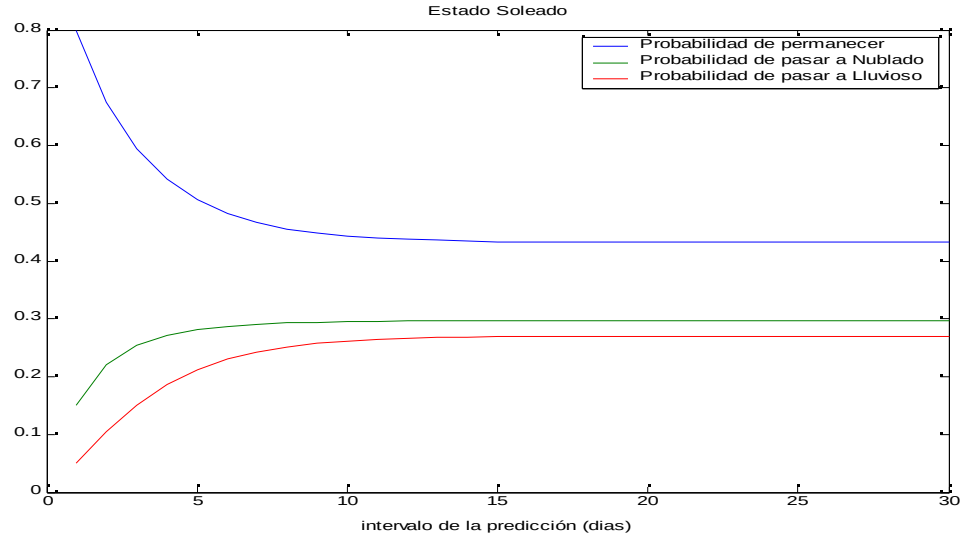
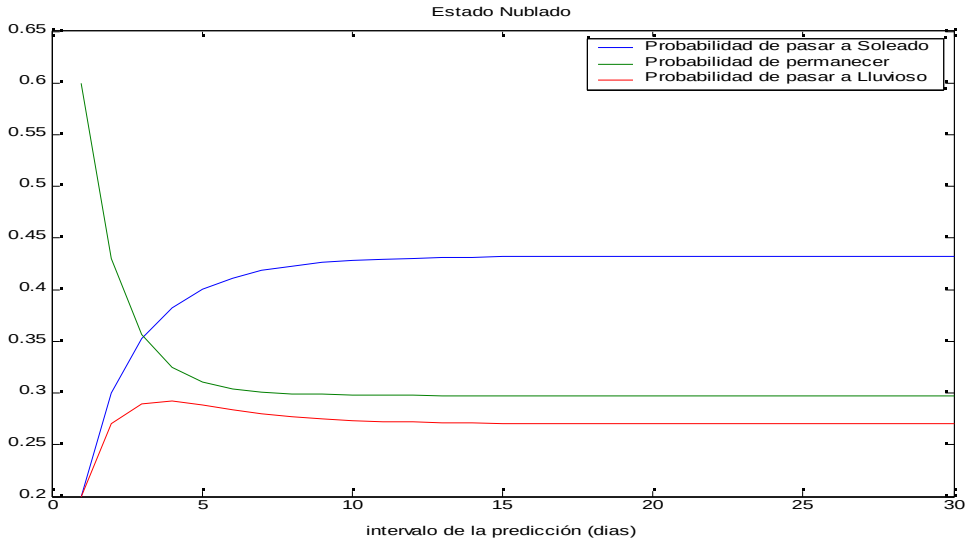


Figura 3.3 Probabilidades de transición desde soleado (arriba) y nublado (abajo) a todo los estados, respecto al numero de transiciones. Cuando el orden de las transiciones aumenta, las probabilidades de pasar a otros estados tienden a un límite. Si las mismas probabilidades límite rigen para todos los estados, no importa que clima haga hoy: la probabilidad de que dentro de un periodo largo de dias llueva es la misma.



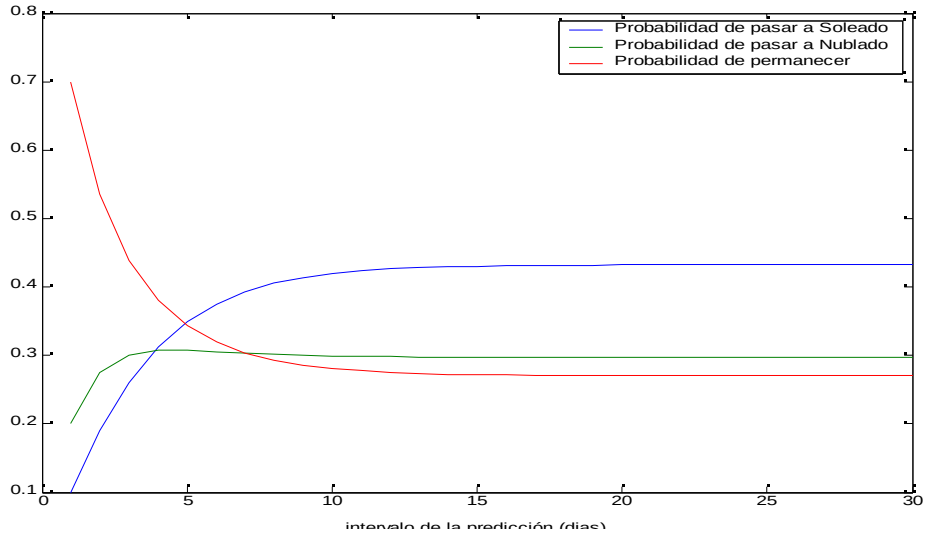


Figura 3.4 Probabilidad de transición desde el estado Lluvioso a los demás estados, frente al orden de la transicion.

3.2.6 Verosimilitud de una observación

De acuerdo con la definición de ***A*** y con (3.1) podemos calcular la sin mayor dificultad la probabilidad de que una secuencia de observaciones dada haya sido generada por nuestro modelo. En lo sucesivo representaremos por $o=\{o_1, o_2, \ldots, o_T\}$ la secuencia observada, y por $\lambda(\theta)$ al modelo, donde θ representa los parámetros del modelo en cuestión (en el caso que nos ocupa, la matriz estocástica ***A***). Es de reseñar que cada secuencia de observaciones tiene *una y sólo una* secuencia de estados asociada $q=\{q_1, q_2, \ldots, q_T\}$. Supongamos una primera secuencia de observaciones para nuestro ejemplo

$$o=\{S, S, S, S, S, N, N, L, L, L, L, N, S\}$$

Donde por claridad se han sustituido los índices del estado por las iniciales del clima asociado. Como la probabilidad de pasar de un estado a otro viene dada por la matriz de transición, y la secuencia de estados es conocida, la verosimilitud de la observación respecto anuestro modelo

vendrá dada por

$$o=\{S,S,S,S,S,N,N,L,L,L,L,N,S\}$$

$$p(o/\lambda) = p(q/\lambda) = \pi_{q_1}\prod_{t=1}^{T-1} a_{q_t q_{t+1}}$$

$$= \pi_1 \cdot a_{11} \cdot a_{11} \cdot a_{11} \cdot a_{11} \cdot a_{12} \cdot a_{22} \cdot a_{23} \cdot a_{33} \cdot a_{33} \cdot a_{32} \cdot a_{22} \cdot a_{21} \simeq 4.4 \cdot 10^{-5}$$

Si en lugar de modelar el problema con una cadena de Markov nos hubiéramos ceñido sólamente a la probabilidad de aparición de los símbolos (probabilidad de los estados límite), ignorando completamente las probabilidades de transición tendríamos

$$o=\{S,S,S,S,S,N,N,L,L,L,L,N,S\}$$

$$p(o/\vec{\pi}) = p(q/\vec{\pi}) = \prod_{t=1}^{T} \pi_{q_t} = (\pi_1)^6 \cdot (\pi_2)^3 \cdot (\pi_3)^4 \simeq 9.2 \cdot 10^{-7}$$

Una verosimilitud un par de órdenes de magnitud menor. Parece por tanto que la primea secuencia propuesta le era bastante "propia" a nuestro modelo (es bastante probable que haya sido generada por algo parecido a nuestro modelo). Probemos con una segunda secuencia de observaciones.

$$o=\{S,L,S,L,S,L,S,L,S,N,N,N,S\}$$

$$p(o/\lambda) = p(q/\lambda) = \pi_{q_1}\prod_{t=1}^{T-1} a_{q_t q_{t+1}}$$

$$= \pi_1 \cdot a_{13} \cdot a_{31} \cdot a_{13} \cdot a_{31} \cdot a_{13} \cdot a_{31} \cdot a_{13} \cdot a_{31} \cdot a_{12} \cdot a_{22} \cdot a_{22} \cdot a_{23} \simeq 2.9 \cdot 10^{-8}$$

Aún apareciendo cada estado el mismo número de veces que en la primera, tenemos una diferencia de cuatro órdenes de magnitud en

la probabilidad de la secuencia. Normalizando respecto a la longitud, una comparación de la primera respecto a la segunda nos deja un factor de 1.75 (para la probabilidad media por símbolo). Estas diferencias notables son debidas a la no uniformidad de las probabilidades de transición entre estados, a la estructura interna de la fuente.

En estos ejemplos queda patente la importancia de lo que constituye la esencia de las cadenas de Markov, las probabilidades de transición, y como necesariamente improntan las salidas del modelo.

2.3.7 Estimación de los parámetros del modelo

Si queremos estimar los parámetros de una cadena de Markov para que se ajusten los más fielmente posible a un conjunto de secuencias de observaciones de entrenamiento

$$\theta^* = argmax_\theta\, p(o/\lambda(\theta))$$

La estimación se reduce a una estimación de máxima verosimilitud. No hay más que contar directamente el número de transiciones. Así, la probabilidad de transición desde i hacia j se hará corresponder con la frecuencia observada de tal transición en el cuerpo de entrenamiento.

3.3 Cadenas o modelos ocultos de Markov

3.3.1 Definición

En los modelos vistos hasta ahora, la salida del proceso en cada instante está constituida por un valor unívocamente asociado al estado. La salida queda determinada completamente por el estado del sistema, y se sabe en qué estado está el sistema sin más que observar la salida. La secuencia de estados que se recorre es, por lo tanto, directamente observable, y las probabilidades que están asociadas a esta secuencia, fácilmente computables.

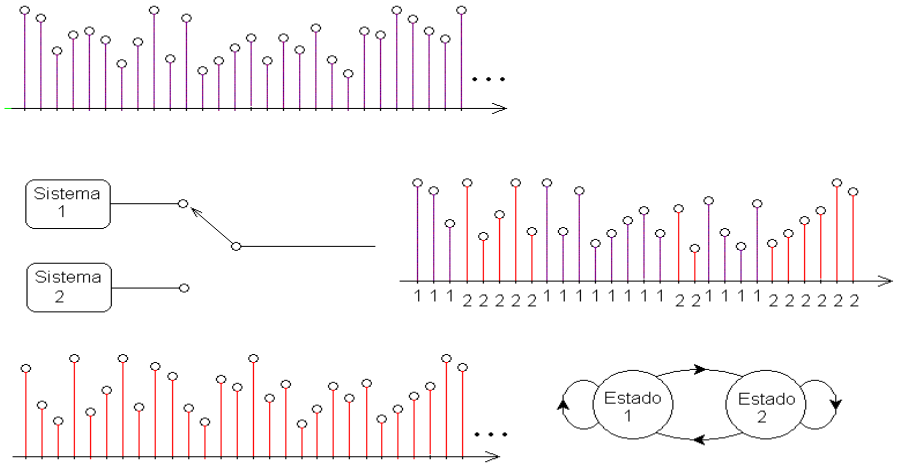


figura 3.5 Ejemplo que ilustra la relación entre estado y observación en los modelo oculto de markov

En una cadena oculta o modelo oculto de Markov (Hidden Markov Model, HMM en adelante) esto no sucede. Un ejemplo claro de un HMM de valores contínuos se muestra en la figura 3.5. En ella, dos sistemas o procesos estocásticos cualesquiera generan sus respectivas secuencias. Un tercer sistema, el conmutador, alterna entre ellos para crear la salida. El estado de este tercer sistema *no determina los valores que tomará la salida*, pero sí qué otro proceso se manifestará en la misma.

Así en un HMM, la salida asociada a un estado no es un valor, sino la salidad de un proceso estocástico o una variable aleatoria. Por eso dice que un HMM es doblemente estocástico. Esto tiene implicaciones a la hora de tratar con los dos problemas ya descritos para cadenas de Markov, ya que la secuencia de estados que sigue el sistema no es directamente observable en la salida.

3.3.2 Un ejemplo clásico

Un escenario clasico para poner de manifiesto un oculto de markov de valores discretos es el que plantea el siguiente experimento. Imaginemos que alguien tras unas cortinas oscuras alguien tiene va-

rias urnas iguales, llenas en distinta proporción de una mezcla de bolas rojas, azules y amarillas.

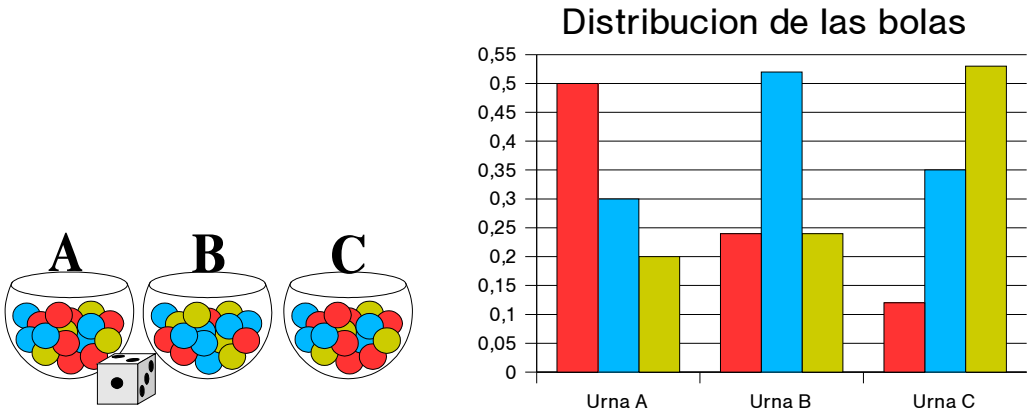


figura 3.6 Tres urnas con distinta proporción de bolas se irán seleccionando para generar la salida, que consiste en la extracción de una bola de la urna escogida.

El ejecutor se dedica a sacar una bola de la urna y nos la enseña tímidamente por la raja de la cortina, para luego devolverla a su urna de origen. El ejecutor escoge la urna usando un dado y (quizás) algun tipo de regla para la elección basada en la última urna escogida. Por ejemplo podría decir si he sacado de la urna A, vuelvo a sacar de A si en el dado sale de uno a cuatro, saco de B si sale cinco o seis, y en ningún caso saco de C. Y todo lo que nosotros podremos observar es una secuencia de bolas rojas, azules, amarillas... pero no sabemos de qué urna la están sacando, ni según qué criterios.

Podemos formular un modelo en el que a cada urna le corresponde un estado del sistema. Entonces, la salida en cada estado estaría gobernada por una distribución multinomial, sobre las bolas rojas, azules y amarillas. La matriz estocástica A se encargaría de modelar el sistema de elección de urnas, tanto el dado como las posibles reglas.

Hay que hacer notar que el HMM propuesto será intuitivamente "mejor" si el número de estados coincide con el de urnas, las probabili-

dades de transición modelan los criterios de transición y las multinomiales son fieles a las proporciones de cada urna. Todas estas consideraciones se pondrán de manifiesto en sucesivos capítulos.

Implícitamente en el enunciado estamos desvelando gran parte del misterio, y conocemos muchas cosas: existen varias urnas, existe un dado que escoge entre ellas y las bolas que nosotros vemos salen de alguna de esas urnas. Con estas pistas sabemos que un HMM puede modelar a la perfección la situación. Así, podemos escoger por ejemplo un modelo con dos estados y unos valores iniciales para las variables asociadas a ellos e intentar que se asemeje lo más posible a lo observado (más adelante veremos cómo se hace esto). Si no estamos ssatisfechos podemos probar con otro numero de estados (cinco, por ejemplo) y otros valores iniciales, con la esperanza de que, muy probablemente, en alguno de los intentos conseguiremos un modelo tan bueno que nos sirva para lo que sea que queremos hacer con él. Hasta aquí todo es correcto. Pero supongamos que no sabiamos esas cosas, o que realmente no sabemos que es lo que ocurre detrás de la cortina. Las bolas podrían estar siendo escogidas de una u otra urna según criterios que implicaran el color de la bola anterior, o por el capricho de un niño, o podrían representar una imagen televisada, digitalizada y transmitida sobre una linea con una modulación ternaria. En estos casos no importa cuánto probemos: *puede* que nuestro modelo *jamás* refleje los procesos subyacentes lo bastante bien, porque puede que estemos cometiendo un error al modelarlos como HMMs.

En el mundo real, estas pistas provienen en general de disciplinas complementarias o del conocimiento obtenido en otras áreas. Así, sabemos que la voz puede ser modelada usando un modelo oculto de Markov porque sabemos que en el proceso de habla los organos fonadores adoptan configuraciones estables durante cortos periodos de tiempo, que este numero de configuraciones es finito dentro de una lengua y que la ejecución de cualquier palabra tratará de seguir un camino a través de esa constelacion.

3.3.3 Parámetros del modelo

Los parámetros de un HMM incluyen al los que hemos visto que definían el comportamiento de las cadenas de Markov. Las probabilidades de trenasición y probabilidades iniciales siguen estando presentes, pero ahora, en lugar de un símbolo, tenemos asociada una variable aleatoria a cada estados del modelo. Cualquier HMM vendrá especificado por:

i. ***A*** La **matriz estocástica**. Determina la probabilidad de transición entre estados, igual que en las cadenas de Markov. Siendo N el numero de estados del modelo tenemos que

$$\sum_{j=1}^{N} a_{ij}=1 \qquad 1\le i\le N$$

ii. π **Probabilidades iniciales.** A pesar de su la nomenclatura no tienen por qué coincidir con las probabilidadesd de los estados límite. Tambien han de cumplir la restricción estocástica.

iii. $\boldsymbol{b_j}$ **Distribución de probabilidad o densidad de probabilidad** (en el caso contínuo) asociada al estado j. Si denominamos como o_t el valor observado para un instante *t* tenemos que...

- *Caso discreto*. ***M***=numero de simbolos del alfabeto $\{v_1, v_2, \ldots\ v_k\ \ldots, v_M\}$ de salida

$$b_j(v_k)=p(o_t=v_k/q_t=j) \quad 1\le j\le N\ ,\ 1\le k\le M$$

- *Caso contínuo*

$$b_j(o_t) \ = \ f_{x/q_t=j}(o_t)$$

donde $f_{x/q_t=j}(x)$ es una distribución contínua, por ejemplo una gaussiana o una mezcla de gaussianas. Así, una vez escogido el numero de estados que tendrá, el HMM queda definido por

$$\lambda=(A,B,\pi)$$

En adelante, cuando nos refiramos a los parámetros de un HMM en su conjunto usaremos la letra griega $\theta = \{A,B,\pi\}$, por lo que si nos encontramos con $\lambda(\theta)$ estaremos poniendo de manifiesto la dependencia del modelo con respecto a sus parámetros.

3.3.4 Los tres problemas fundamentales

Trabajar con HMMs plantea tres problemas base. Estos problemas, que eran triviales cuando trabajamos con cadenas de Markov, se vuelven más complicados ahora que la secuencia de estados está más débilmente unida con la salida observable.

i. El primer problema problema lo llamaremos *Cálculo de la verosimilitud respecto a un modelo*. Consiste en, conocida una secuencia de observaciones $O = \{O_1, O_2 ... O_T\}$, determinar con qué probabilidad podría haber sido generada por un modelo $\lambda=(A,B,\pi)$ dado. El cálculo de esta cantidad es necesario cuando, teniendo un conjunto de modelos en mutua competición, intentamos determinar a cual de ellos es más probable que pertenezca la secuencia de simbolos observada. De manera formal, lo podemos expresar como

$$p(O|\lambda) \quad .$$

ii. El segundo problema lo llamaremos *Cálculo del camino óptimo*. Conocida una secuencia de observaciones $O = \{O_1, O_2 ... O_T\}$ y un mo-

delo $\lambda=(A,B,\pi)$, consiste en encontrar la secuencia de estados $Q=\{q_1, q_2, \ldots, q_T\}$ por la que debe pasar el modelo que pueda ser considerada óptima, en el sentido de que sea la que *mejor explica* que se haya producido esa observación. El problema emerge del hecho de que la secuencia de estados ya no es observable y de que en general habrá una miriada de caminos que puedan dar pie a la misma secuencia de observación. Formalmente buscamos

$$argmax_Q\, p(O/Q,\lambda)$$

iii. El tercer problema lo llamaremos *Reestimación de parámetros.* Conocida una o múltiples secuencias de observaciones O , cómo variar los parámetros de nuestro modelo $\lambda=(A,B,\pi)$ de modo que maximicen la probabilidad de que ´la secuencia o secuencias pertenezcan a nuestro modelo. Este problema se plantea cuando intentamos entrenar un modelo usando un conjunto de obsevaciones de entrenamiento, y es de lejos el más dificil de los tres. Formalmente intentaremos encontrar

$$argmax_\theta\, p(O/\lambda(\theta))$$

Cada uno de estos problemas fundamentales serán resueltos de uno en uno en los siguientes capítulos.

4. Cálculo de la verosimilitud respecto a un modelo

4.1 Planteamiento

Este problema se nos plantea cuando tenemos que decidir, dada una observación, cual de nuestros modelos es con mayor probabilidad el que la ha generado, en cuál de nuestros modelos cuadra mejor la observación.

4.2 Aproximación directa

Una forma sencilla de enfocar el problema consiste en considerar que al generar una observación de longitud T, el sistema recorre necesariamente alguna secuencia de estados $q=q_{1\ldots}q_T$. Si queremos calcular la probabilidad de que una secuencia se corresponda con una observación, tenemos que recurrir a la densidad de probabilidad (o a las probabilidades de símbolo en el caso discreto) de cada uno de los estados que atravesemos en cada instante. Formalmente, la probabilidad de que una secuencia concreta de estados de lugar a una observación dada se expresa como

$$p(o/q,\lambda) = \prod_{t=1}^{T} p(o_t/q_t,\lambda) = b_{q_1}(o_1)b_{q_2}(o_2)\cdots b_{q_T}(o_T)$$

Donde se ha asumido que las observaciones son estadísticamente independientes. Ahora bien, también necesitamos conocer la probabilidad de que el sistema pase por esa secuencia de estados en concreto. La probabilidad dependerá de los coeficientes de la matriz de transición, y viene dada por la expresión

$$p(q/\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

Con estos datos en la mano ya podemos averiguar la probabilidad de que la observación halla sido generada por el modelo y de que además el modelo recorra la secuencia de estados que estamos considerando. Aplicando Bayes tenemos

$$p(o,q/\lambda) = p(o/q,\lambda)\cdot p(q/\lambda)$$

Al buscar la probabilidad de obtener O desde nuestro modelo estamos buscando sino la suma de la probabilidad conjunta sobre todos los posibles caminos. Sin más que tomar la marginal de la probabilidad conjunta obtenemos la solución que buscábamos.

$$p(o/\lambda) = \sum_{\forall q} p(o/q,\lambda)\cdot p(q/\lambda) = \sum_{\forall q} \pi_{q_1} b_{q_1}(o_1)\cdot a_{q_1 q_2} b_{q_2}(o_2)\cdots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

Una manera de calcular la expresión anterior es la siguiente: empezamos en t = 1 computando la probabilidad de que el modelo se encuentre en el estado q_1 . Multiplicamos por la probabilidad de que, estando en ese estado, genere la observación o_1 correspondiente; el siguiente producto implica la probabilidad de que se produzca el salto desde q_1 a q_2 , y así vamos recorriendo el sendero a través del espacio de estados. Y repetimos la operación para todos los posibles caminos.

Si ya hemos resuelto el problema, ¿por qué hay más líneas por debajo de esta? Esta aproximación directa es, sencillamente, computacionalmente inviable: de un instante al siguiente nacen N nuevos caminos, luego en total existirán N^T caminos distintos. Y para cada camino necesitamos alrededor de 2T operaciones (2 operaciones de multiplicación por estado). Para hallar la verosimilitud de una observación con cien muestras respecto a un modelo que tenga cinco estados necesitaríamos alrededor de

$$2\cdot T\cdot N^T = 2\cdot 100\cdot 5^{100} \simeq 10^{72}$$

4.3 Procedimiento Backward-Forward

4.3.1 Fundamentos

El algoritmo mostrado en el acercamiento directo al problema es de orden exponencial, debido a las bifurcaciones (N-furcaciones en realidad) del sendero de estados. Podemos conseguir algo mejor si nos damos cuenta de que los múltiples caminos no sólo se dividen en cada paso, sino que también en cada paso confluyen. Esta idea se formaliza en el procedimiento Adelante-Atrás (Backward-Forward)

Se basa en dos hechos:

i. Sólo existe un numero finito de estados: todos los caminos discurren, y desembocan necesariamente a cada paso en un estado de los posibles.

ii. El verdadero interés reside en obtener la suma de la probabilidad de todos los caminos, y no en la probabilidad de un camino en concreto. Esto nos permitirá olvidar el camino recorrido hasta un punto dado.

4.3.2 Variable hacia delante (*Forward variable*)

La variable hacia delante nos ayudará a calcular la suma de las probabilidades de todos los caminos. Se define como la probabilidad de que el modelo genere la salida hasta $t=t_0$ y que en t_0 esté en el estado i. Formalmente,

$$\alpha_t(i) = p(o_1 o_2 \ldots o_t,\ q_t = i \mid \lambda) \qquad 1 \le t \le T \qquad (4.1)$$

Su misma definición parece prometer que podremos calcularla iterativamente. Supongamos que tenemos un modelo con solo dos estados. De un modo intuitivo, si en el instante t llegamos a al estado j, he-

mos llegado ahí porque venimos de algún otro estado en el instante anterior (t-1).

Cómo hayamos llegado a ese estado anterior nos importa poco, siempre y cuando conozcamos la probabilidad de haber ocupado el estado anterior hace un instante. La probabilidad de haber estado en un estado cualquiera i en el instante t-1 (dadas las observaciones pasadas) es precisamente $\alpha_{t-1}(i)$. Luego la probabilidad de llegar al estado j en el instante t+1 vendrá determinada por estas probabilidades y la matriz de transición:

$$p(o_1 o_2 \cdots o_{t-1},\, q_t = j \mid \lambda) \;=\; \sum_{\forall i} \alpha_{t-1}(i)\, a_{ij}$$

Ahora que conocemos la probabilidad de estar en el estado j en el instante t (dadas las observaciones pasadas), podemos calcular la probabilidad de estar en el estado j en el instante t y además estar observando o_t .

$$p(o_1 o_2 \cdots o_t,\, q_t = j \mid \lambda) \;=\; b_j(o_t) \sum_{\forall i} \alpha_{t-1}(i)\, a_{ij} \;=\; \alpha_t(j)$$

Que es por definición $\alpha_t(j)$, pudiendo por tanto calcularla para todo instante de manera recursiva.

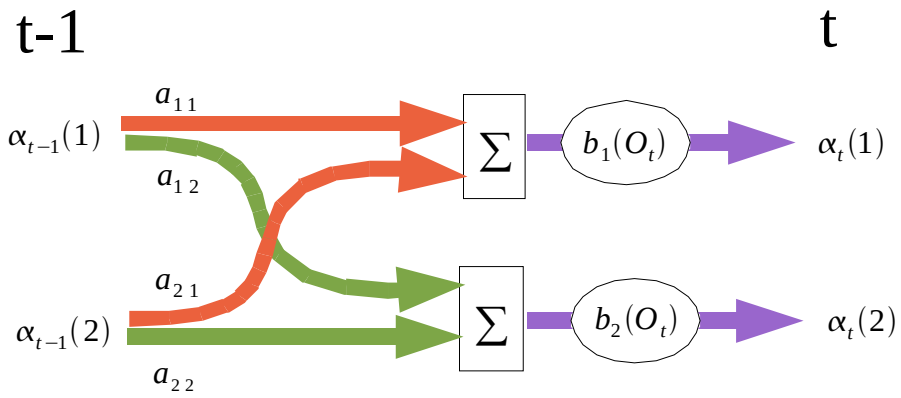


Figura 4.1 Diagrama que representa una iteración en el cálculo recursivo de la variable hacia delante

Si calculamos la variable hacia delante, en cada iteración tendremos que hacer N*N productos (probabilidades de transición), N*N sumas y N productos (probabilidad de generar la observación) Luego el nuevo algoritmo requiere sólo 2N(N-1)T operaciones, es lineal con el tiempo y cuadrático con el número de estados. Para el mismo ejemplo anterior, calcular $\alpha_T(i) \quad \forall i$ dada una observación con cien muestras respecto a un modelo que tenga cinco estados necesitaríamos en torno a 10^4 operaciones.

¿Cómo se relaciona la verosimilitud con la variable hacia delante? Por definición la variable hacia delante es la probabilidad de que dada la observaciones hasta t = t, nos encontremos en ese instante en el estado j. Si el instante considerado es el último, la verosimilitud de la secuancia completa de observaciones respecto al modelo se reduce a la suma para todos los estados posibles de $\alpha_T(j)$.

$$p(o/\lambda) = \sum_{\forall j} p(o_1 o_2 \cdots o_T, q_T = j \mid \lambda) = \sum_{\forall j} \alpha_T(j)$$

Con lo cual hemos resuelto el problema de un modo factible. Resumiendo el procedimiento hacia delante queda como:

i. Inicialización

$$\alpha_1(i)=\pi_i b_i(o_1) \quad 1\le i\le N$$

ii. Inducción

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(o_{t+1}) \quad 1\le t\le T-1$$
$$1\le j\le N$$

iii.Terminación

$$p(o/\lambda)=\sum_{i=1}^{N}\alpha_T(i)$$

4.3.3 Variable hacia detrás (*Backward variable*).

La variable hacia detrás es la contrapartida de la variable hacia delante. Nos permite igual que la anterior calcular la verosimilitud respecto a un modelo, aunque en este caso la aproximación es distinta. Presentamos la variable hacia atrás no para resolver el problema que nos ocupa (para el cual bastaría la variable hacia delante), sino porque será necesaria la utilización de ambas en la resolución de los problemas por venir.

La variable hacia se define como la probabilidad de que las observaciones para para instantes posteriores a t hayan sido generadas por el modelo, *si el modelo estaba en el estado i en el instante t* . Formalmente,

$$\beta_t(i)=p(o_T,o_{T-1}\cdots o_{t+1}/\ q_t=i\ ,\lambda) \qquad 1\le t\le T-1 \qquad (4.2)$$

Al igual que en el caso de la α , podremos utilizar un algoritmo recursivo y de bajo coste para calcularla. Supongamos que estamos en el instante t en un estado i, e intentamos calcular la probabilidad de que el modelo, en el futuro, genere una secuencia de muestras $o=\{o_{t+1}...o_T\}$. Esa no es sino la definición de la variable hacia delante. Podemos plantearlo en dos partes:

Primero calculamos la probabilidad de llegar a un estado cualquiera i en el instante posterior (t+1), que no es otra que a_{ij} . Formalmente

$$p(q_{t+1}=j/\, q_t=i\ ,\lambda)\ =\ a_{ij}$$

Una vez en el estado j, la probabilidad de observar $\{o_{t+1}...o_T\}$ podemos descomponerla en dos partes:

$$p(o_T\,,o_{T-1}\cdots o_{t+1}/\, q_{t+1}=j\ ,\lambda)\ =$$

$$=\ p(o_T\,,o_{T-1}\cdots o_{t+2}/\, q_{t+1}=j\ ,\lambda)\cdot p(o_{t+1}/\, q_{t+1}=j\ ,\lambda)$$

$$=\ \beta_{t+1}(j)\cdot b_j(o_{t+1})$$

Si unimos las dos expresiones podemos calcular la probabilidad de que, estando en el estado i en el instante t, nos encontremos en un estado j en el siguiente y en el futuro el modelo genere las observaciones $\{o_{t+1}...o_T\}$. Ésta viene dada por

$$p(o_T\,,o_{T-1}\cdots o_{t+1}\,,q_{t+1}=j/\, q_t=i\,,\lambda)\ =$$

$$=\ p(o_T\,,o_{T-1}\cdots o_{t+1}/\, q_{t+1}=j\ ,\lambda)\cdot p(q_{t+1}=j/\, q_t=i\ ,\lambda)$$

$$=\ a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$$

Para obtener $\beta_t(i)$ sólo tenemos que tomar la marginal de esta probabilidad conjunta, o en otras palabras, sumar la expresión para todos los posibles estados j.

$$\beta_t(i)=\sum_{\forall j}\left[a_{ij}b_j(o_{t+1})\beta_{t+1}(j)\right]$$

Esta expresión es la forma recursiva de calcular la variable hacia detrás. Para que al inicio del algoritmo (en el instante t = T-1) la expresión devuelva un valor correcto, ha de utilizarse $\beta_T(i)=1 \ \forall i$. La representación gráfica de un paso del algoritmo para un modelo con dos estados es la mostrada en la figura 4.2.

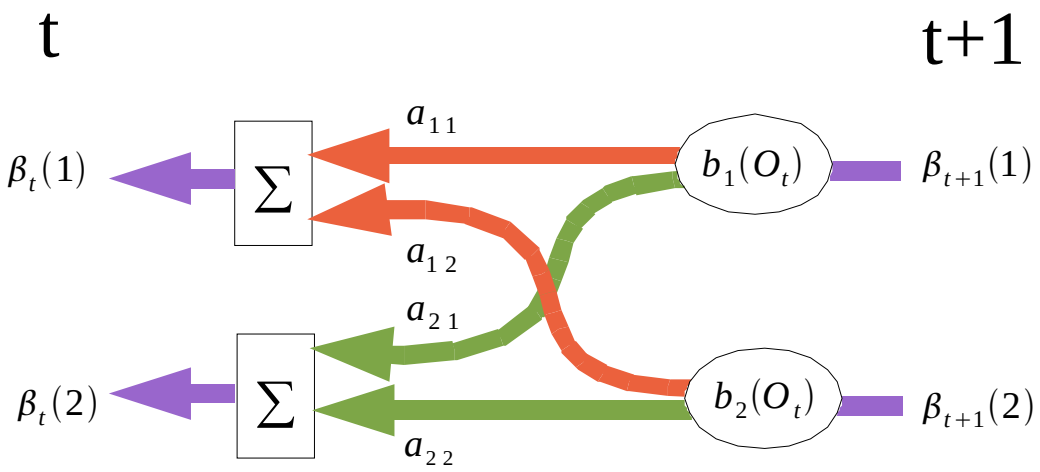


figura 4.2 Diagrama que representa el cálculo iterativo de la variable hacia atrás

El cálculo de beta de nuevo requiere del orden de N^2T operaciones, y también con ella se puede calcular la verosimilitud $p(o/\lambda)$ a través de la expresión

$$p(o/\lambda)=\sum_{\forall i}\pi_i b_i(o_1)\beta_1(i)$$

Resumiendo el algoritmo...

i. Inicialización

$$\beta_T(i)=1 \qquad 1\leq i\leq N$$

ii. Inducción

$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\beta_{t+1}(j) \qquad 1\leq t<T$$

$$1\leq j\leq N$$

iii.Terminación

$$p(o/\lambda) = \sum_{j=1}^{N} \pi_j \beta_1(j)$$

4.3.4 Notas de implementación: escalado

Los algoritmos expuestos han conseguido reducir drásticamente el número de operaciones necesarias para calcular la verosimilitud. Estando por esa parte resueltos los problemas, es necesaria en cualquier caso otra modificación.

¿En que rango de valores se mueve la verosimilitud? Si nos fijamos, en cada iteración, ya sea para el cálculo de alfa o beta, estamos multiplicando por un par de coeficientes: los coeficientes de transición a_{ij} (siempre menores que la unidad) y la probabilidad de la observación dado el estado $b_j(o_t)$ (con un rango amplísimo, dependiendo de la función densidad). Cuando la longitud de la secuencia observada crece la realidad es que el rango dinámico de las variables se hace insostenible.

Para evitar esta contrariedad, se utilizan siempre los que se de-

nominan variables escaladas. Las alfas escaladas se calculan igual que las normales, con la salvedad de que en lugar de utilizar para el cálculo iterativo α_{t-1} , utilizan α_{t-1} normalizada, que denotaremos como $\bar{\alpha}_{t-1}$. Formalmente

$$\hat{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \bar{\alpha}_t(i) a_{ij}\right] b_j(o_{t+1})$$

$$\bar{\alpha}_t = c_t \hat{\alpha}_t \qquad c_t = \frac{1}{\sum_{\forall j} \hat{\alpha}_t(j)} \tag{4.3a 4.3b}$$

Donde las $\bar{\alpha}_t(j)$ son las variables normalizadas (suman uno), y se ha añadido un circunflejo para denotar a las alfas escaladas.

El cálculo de la verosimilitud total se ve alterado pero es inmediato al notar que el coeficiente de escalado puede sacarse fuera de la fórmula, de modo que la equivalencia entre la variable original y la escalada es directa

$$\hat{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \bar{\alpha}_t(i) a_{ij}\right] b_j(o_{t+1}) = c_t \left[\sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij}\right] b_j(o_{t+1})$$

$$\alpha_{t+1}(j) = \hat{\alpha}_{t+1}(j) \left(\prod_{k=1}^{t} \frac{1}{c_k}\right)$$

$$p(o/\lambda) = \sum_{\forall j} \alpha_T(j) = \sum_{\forall j} \hat{\alpha}_T(j) \left(\prod_{k=1}^{T-1} \frac{1}{c_k}\right) = \prod_{k=1}^{T} \frac{1}{c_k} \tag{4.4}$$

El mismo proceso se sigue con la variable hacia detrás. En ese caso los coeficientes de escalado c_t toman un valor distinto al de los de la variable hacia delante, pero suelen ser del mismo orden de magnitud, por lo que en realidad puede usarse el de una de las variables para las dos (esto tendrá utilidad en capítulos posteriores).

4.3.5 Simulaciones

Simulación 4.A

La primera simulación consiste en el cálculo mediante la variable hacia delante y hacia atrás de la verosimilitud de una observación respecto a dos modelos. Para darle un trasfondo real, podemos suponer que las observaciones que se recogen provienen de un sónar que se utiliza en flotas pesqueras. Tras la reconstrucción de la señal procedente de un banco de peces, supondremos que los peces se mueven en un espacio bidimensional. Para caracterizar el banco tomaremos como parámetros de entrada el módulo de la velocidad del centro del banco y algo que refleje la forma del banco, como por ejemplo el radio mayor y el cociente entre radio mayor y menor. Como hemos supuesto que el banco se distribuye y evoluciona en dos coordenadas espaciales, en el instante t tendremos una entrada de tres componentes del tipo

$$\boldsymbol{o}_t = \left(v \quad r_{max} \quad \frac{r_{max}}{r_{min}}\right)$$

Los distintos modelos de markov representarán distintas especies de pez, y cada estado puede pensarse como correspondiente a una actividad que el animal realice, como alimentarse, desplazarse o huir.

Para este ejemplo vamos a suponer que lo único que hacen estos peces es comer (estado uno) y huir (estado dos). Y contrastaremos dos especies de comportamientos antagónicos, digamos que la pusilánime sardina (modelo a, de poco comer y mucho huir) y el atemperado boquerón (modelo b, comensal casi imperturbable).

Modelo a

$$\boldsymbol{A}^a = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix} \qquad \pi^a = \begin{bmatrix} 0.33 & 0.67 \end{bmatrix}$$

$$b_1(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_1, \boldsymbol{U}_1); \quad \mu_1^a = \begin{bmatrix} 0.1 & 7 & 1.3 \end{bmatrix} \quad \boldsymbol{U}_1^a = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

$$b_2(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_2, \boldsymbol{U}_2); \quad \mu_2^a = \begin{bmatrix} 2.2 & 13 & 5.2 \end{bmatrix} \quad \boldsymbol{U}_2^a = \begin{bmatrix} 1.9 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1.6 \end{bmatrix}$$

Modelo b

$$\boldsymbol{A}^b = \begin{bmatrix} 0.75 & 0.25 \\ 0.45 & 0.55 \end{bmatrix} \quad \pi^b = \begin{bmatrix} 0.6429 & 0.3571 \end{bmatrix}$$

$$b_1(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_1, \boldsymbol{U}_1); \quad \mu_1^b = \begin{bmatrix} 0.4 & 9 & 1.4 \end{bmatrix} \quad \boldsymbol{U}_1^b = \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$$

$$b_2(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_2, \boldsymbol{U}_2); \quad \mu_2^b = \begin{bmatrix} 1.7 & 12 & 3.4 \end{bmatrix} \quad \boldsymbol{U}_2^b = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

La sardina es, en promedio, algo más rápida huyendo (2.2m/s) que el atún (1.7m/s), suele desplazarse más lentamente mientras se alimenta (0.1m/s frente a 0.4m/s) y sus bancos son algo menores (~7m de radio frente a los ~9m del banco de boquerones[3]). Los valores serán lo bastante distintos (vistas las varianzas) como para evitar que las probabilidades de transición de cada especie se impongan de manera excesiva en el cálculo de la verosimilitud.

Es cierto que si modelamos con gaussianas estaremos cometiendo irremediablemente un error de modelado (ya que ninguna de las características toma valores en el eje negativo, siendo más adecuada una distribución de Rayleigh), pero lo dejaremos estar. De las mediciones extraemos la secuencia de observaciones mostrada en la figura 4.3.

3 Los datos del ejemplo como es evidente a estas alturas son todos ficticios.

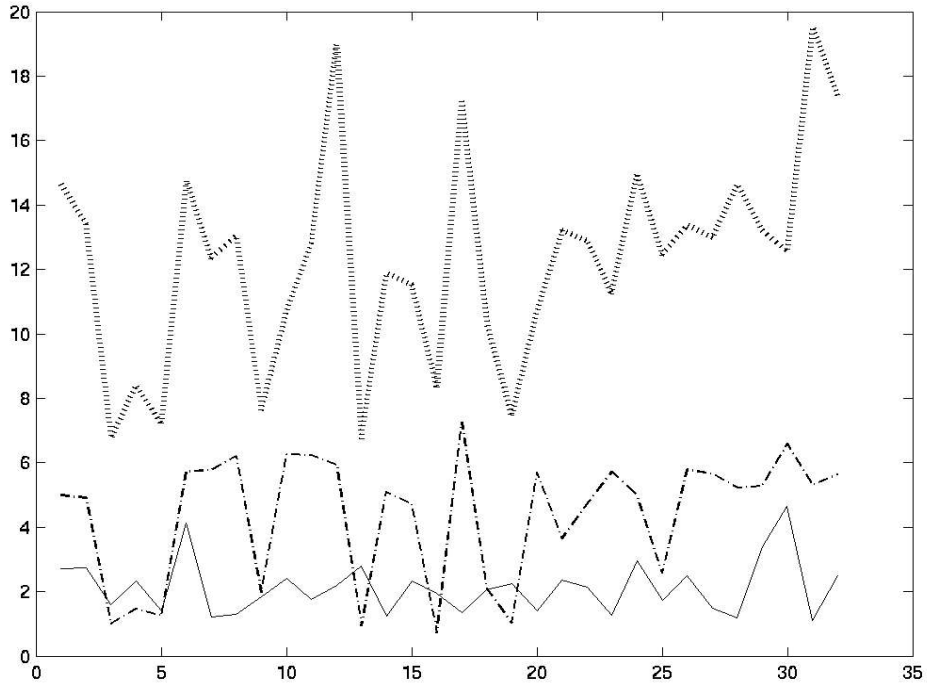


figura 4.3 Secuencia de observaciones que muestra la velocidad escalar (linea punteada y rayada), el eje mayor (linea punteada) y la excentricidad (linea contínua) de un banco de peces.

A simple vista podemos observar que en los instantes cercanos al cuatro el banco de peces se desplaza a baja velocidad y mantiene una forma bastante redondeada, con lo que es posible que se estén alimentando o reposando. En los instantes inmediatamente posteriores el banco se ahúsa y su velocidad aumenta de forma drástica, y después (t = 8), desplazándose todavía a gran velocidad, vuelve a redondearse pero con un radio mayor (los peces están más separados). Este segundo intervalo de observaciones se asemeja bastante a una huida.

Veamos los resultados que arroja el primer modelo (figuras 4.4 y 4.5). Los valores mostrados se corresponden con las variables normalizadas o escaladas que mencionamos en la sección 4.3.4. Podemos observar que, dado que cada una rinde cuentas acerca de una parte las observaciones, para un instante dado las variables hacia delante y hacia atrás toman valores distintos.

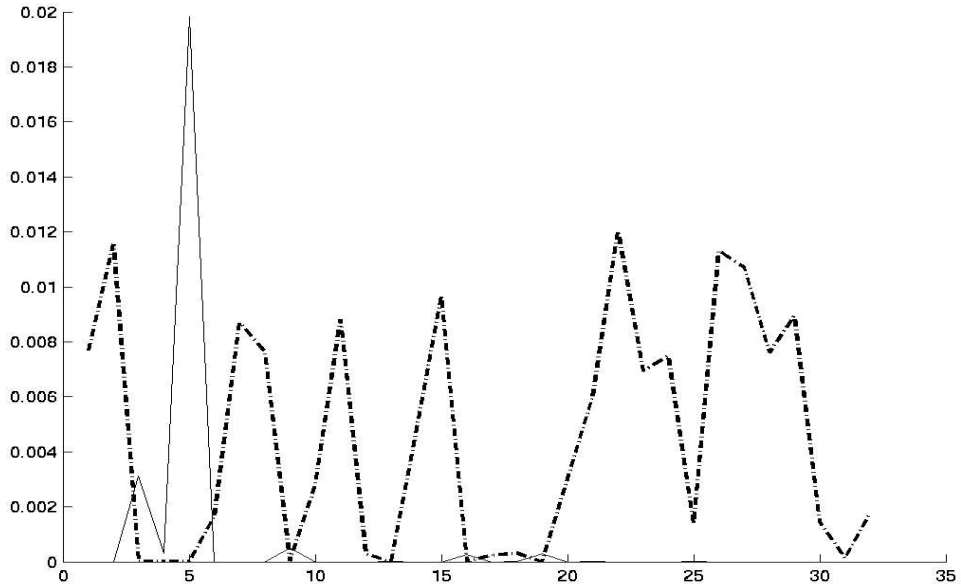


figura 4.4 Valor de la variable hacia delante usando el modelo a (sardina). El trazo fino continuo es el valor de alfa para el estado uno ("comiendo"); el discontinuo se corresponde con el estado dos

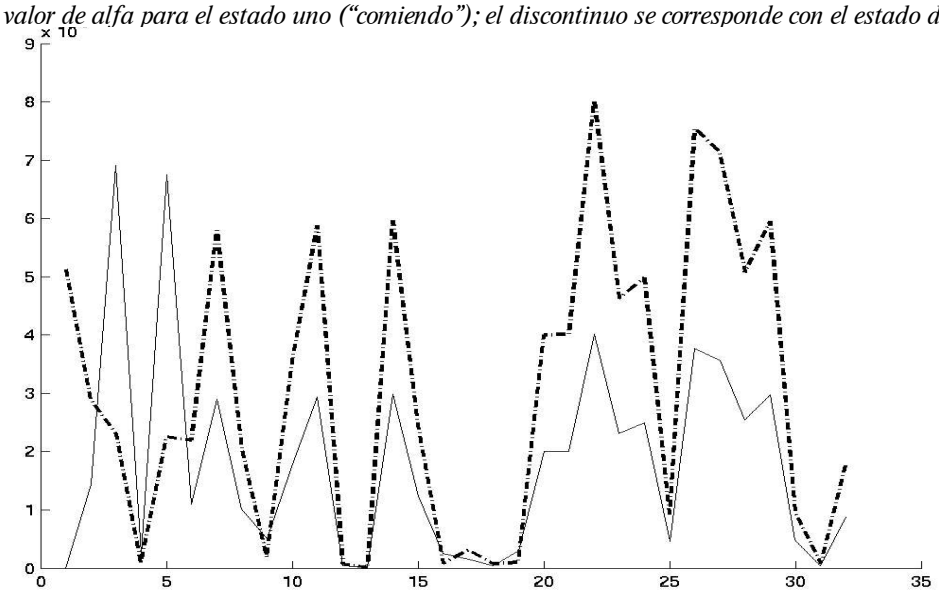


figura 4.5 Valor de la variable hacia atrás usando el modelo a (sardina). El trazo fino continuo es el valor de beta para el estado uno ("comiendo"); el discontinuo se corresponde con el estado dos ("huyendo"). En cada instante, beta representa la probabilidad de ocupar un estado, dadas todas las observaciones futuras.

Aun así en la mayoría de los casos dan como más probable el mismo estado. En cualquier caso lo que nos interesa es la verosimilitud de la

observación respecto al modelo.

Como las variables están escaladas ($\hat{\alpha}_t, \hat{\beta}_t$), y vimos que la verosimilitud podía calcularse en función de los coeficientes de escalado (4.4), que no son sino la suma de las variables escaladas, podemos hacer

$$p(o/\lambda) = \prod_{t=1}^{T}\left[\sum_{\forall j}\hat{\alpha}_t(j)\right] = \prod_{t=1}^{T}\left[\sum_{\forall j}\hat{\beta}_t(j)\right]$$

En lugar de este valor suele tomarse la verosimilitud logarítmica, de modo que no nos quedemos sin rango dinámico en secuencias largas de observaciones. En nuestro caso, obtenemos

$$\ln p(o/\lambda_a) = \sum_{t=1}^{T}\ln\left(\sum_{\forall j}\hat{\alpha}_t(j)\right) = \sum_{t=1}^{T}\ln\left(\sum_{\forall j}\hat{\beta}_t(j)\right) = -195.1620$$

Puede parecer un valor muy pequeño, pero sólo adquiere significado en relación con la verosimilitud de los otros modelos en competencia. Las variables hacia delante y hacia atrás para el modelo b pueden observarse en las figuras 4.6 y 4.7. La verosimilitud de la observación respecto a este segundo modelo se calcula igual, y en este caso el valor es aún menor

$$\ln p(o/\lambda_b) = \sum_{t=1}^{T}\ln\left(\sum_{\forall j}\hat{\alpha}_t(j)\right) = \sum_{t=1}^{T}\ln\left(\sum_{\forall j}\hat{\beta}_t(j)\right) = -244.0545$$

Concluimos pues que, según la regla de Bayes, lo que estamos observando son sardinas.

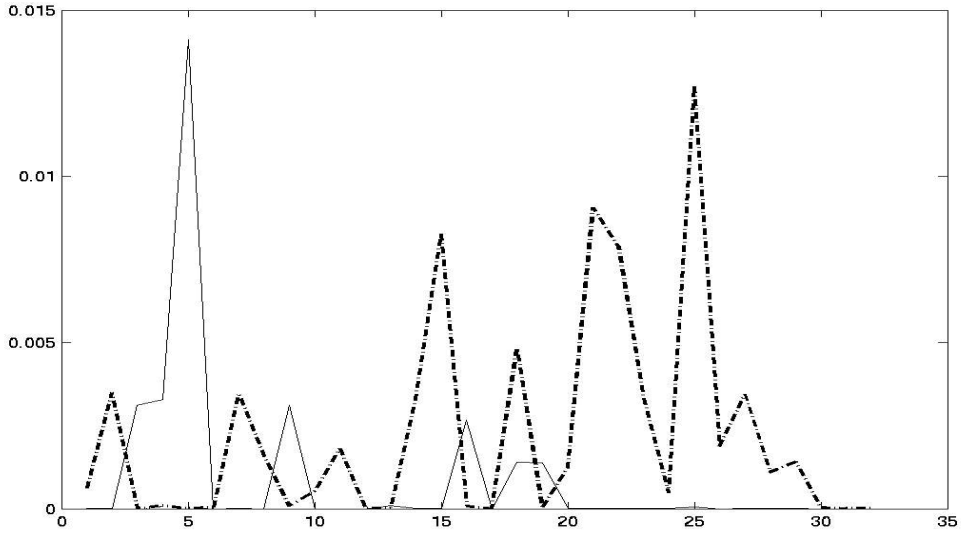


figura 4.6 Valor de la variable hacia delante usando el modelo b (boquerón). El trazo fino continuo es el valor de alfa para el estado uno ("comiendo"); el discontinuo se corresponde con el estado dos ("huyendo"). En cada instante, alfa representa la probabilidad de ocupar un estado, dadas todas las observaciones anteriores.

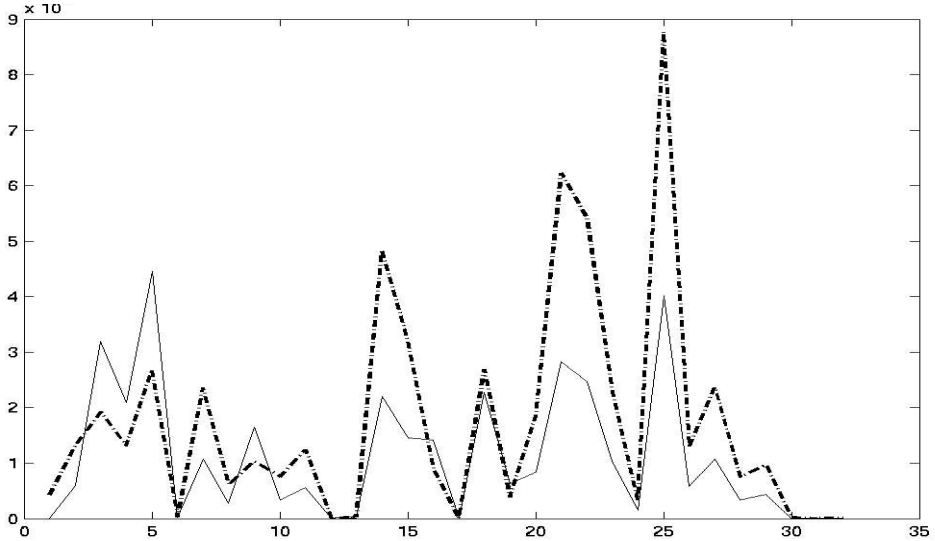


figura 4.7 Valor de la variable hacia atrás usando el modelo b (boqueron). El trazo fino continuo es el valor de beta para el estado uno ("comiendo"); el discontinuo se corresponde con el estado dos ("huyendo"). En cada instante, beta representa la probabilidad de ocupar un estado, dadas todas las observaciones anteriores.

Simulación 4.B

En esta segunda simulación vamos a dejar patente el efecto de las probabilidades de transición, el núcleo de la caracterización temporal que un proceso de Markov hace de las observaciones, en el marco de los modelos ocultos de Markov. Para ello simplemente veremos (como hicimos en el capitulo tres con las cadenas de Markov) cómo difieren la verosimilitud con una matriz A dada y la verosimilitud cuando la probabilidad de cambio de estado es uniforme (por lo que no entra en juego). Basándonos en el ejemplo anterior, tomamos en este caso dos modelos: el modelo del boquerón, y el modelo de boquerón homogéneo (en sus costumbres tanto da comer que huir).

Los coeficientes de transición añaden un sesgo a la verosimilitud de la señal observada en un instante dado, en función de cuál era la probabilidad de ocupación de estados en el *instante anterior*. Vemos que aunque sutiles, existen diferencias entre las variables hacia adelante calculadas para los dos modelos planteados (fig 4.9).

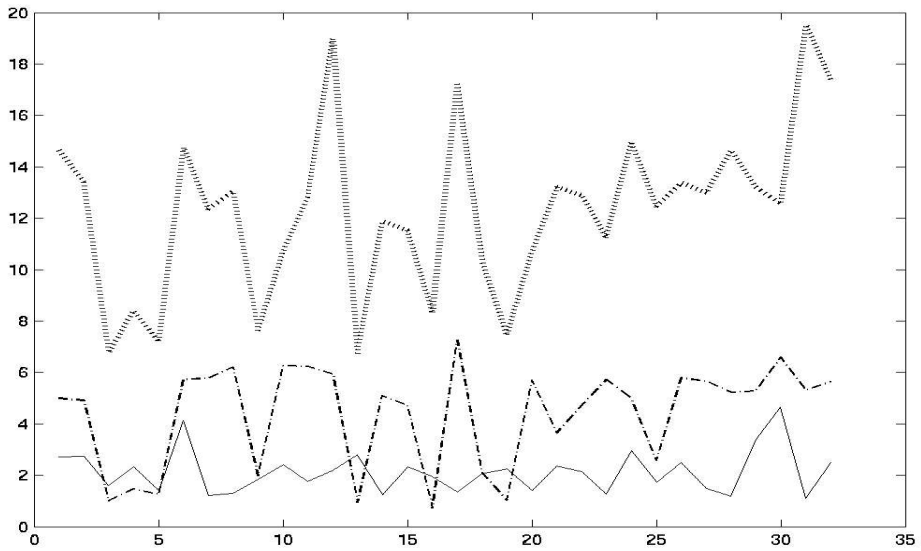


figura 4.8 Primera secuencia de observaciones. La linea punteada es el radio mayor, la linea rayada y punteada la excentricidad y la línea contínua representa la celeridad.

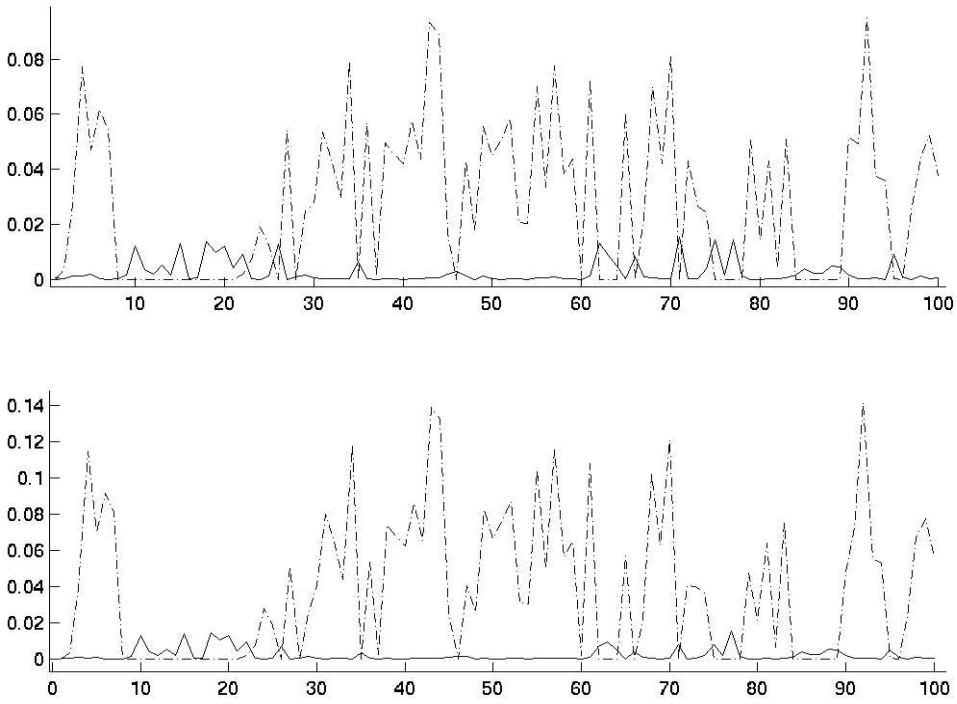


figura 4.9 Arriba, variable hacia delante correspondiente a un modelo del boquerón que obvia las probabilidades de transición. Debajo, variable hacia delante modelo de boquerón visto en la simulación anterior. Linea discontinua, comiendo, linea continua, huyendo.

Si pensamos en maximizar la verosimilitud, y nos fijamos en cómo se calcula, vemos que en cada instante nos conviene que la suma de las probabilidades para todos los estados de observar el dato sea lo mayor posible. Lo que hacen las distintas probabilidades de transición es mejorar una de estas probabilidades (la que se corresponde con un estado dado) y empeorar las otras. ¿Cuál debiera ser la candidata a la mejora,?

$$\sum_{\forall t} \ln(k_1 b_1(o_t) + k_2 b_2(o_t))$$

Disminuir escalando por un factor fijo las probabilidades más pequeñas no disminuirá mucho la suma, y aumentar[4] escalando por uno

4 Si el uno menos el factor es mayor que un medio, ya estamos mejorando (aumentando) respecto a una ponderación homogénea de las probabilidades.

menos ese factor la probabilidad más grande puede mejorar la misma notablemente. Ese sería el enfoque con la vista puesta a aumentar la verosimilitud, pero la elección de los coeficientes que producen el mejor sesgo en cada instante no depende de nuestro capricho, sino que viene impuesta la misma estructura del modelo, en que el estar en un estado u otro (y aplicar por tanto el sesgo de manera beneficiosa o perjudicial) depende *exclusivamente* de las probabilidades de ocupación de estado *en el instante previo*. Esta es como ya sabemos la esencia de cualquier proceso markoviano.

Si la probabilidades de transición son homogéneas, ni se penaliza ni se premia las probabilidades de las variables de los distintos estados, ni se premia o penaliza el cambio de un estado a otro cualquiera estado.

Si, en el caso contrario, la probabilidad de permanecer en un estado es mucho mayor que la probabilidad de pasar a otros estados

$$a_{ii} \gg \sum_{\forall j \neq i} a_{ij}$$

esta característica desembocará en una de las mayores mejoras posibles para la probabilidad de la variable asociada a ese estado i (limitada por cierto esta mejora a, como máximo, duplicar el valor que obtendríamos usando a una distribución homogénea de las probabilidades de transición) . Al mismo tiempo incurriremos en un gran empeoramiento para las probabilidades asociadas a los otros estados j.

Mientras tengamos tiradas largas en las que la probabilidad de que las observaciones provengan de la variable asociada al estado i sea la mayor, estaremos mejorando la verosimilitud (como ya dijimos, de forma limitada). Pero ¿y si en un momento dado esa probabilidad $b_i(o_t)$ resulta ser mucho menor que la asociada a otro estado digamos $b_j(o_t)$? (lo que ocurre siempre que cuando la fuente cambia de estado).

Como la probabilidad de pasar al nuevo estado es minúscula, en este caso tendremos apechugar con un empeoramiento de la probabili-

dad del nuevo estado, que al ser la mayor, arrastrará consigo la verosimilitud del modelo entero en ese instante. Y este empeoramiento, a diferencia de la mejora, no está limitado, ya que

$$\ln(a_{ii}) \rightarrow 0 \quad \Rightarrow \quad \ln(a_{ij}) \rightarrow -\infty$$

El equilibrio, el máximo, se alcanza como siempre cuando la longitud promedio de las secuencias de observaciones en las que $b_i(o_t) > b_j(o_t) \ \forall j \neq i$ se corresponde con la duración media de estado. Si esta longitud promedio es distinta a la un modelo, la verosimilitud respecto al modelo así definido decaerá.

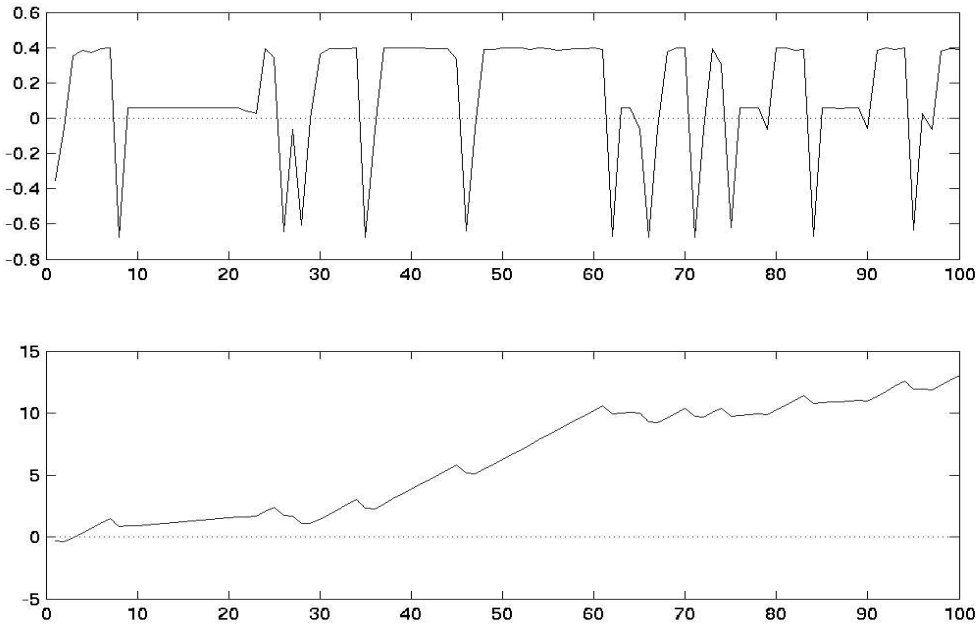


Figura 4. 10 Arriba, diferencia entre el logaritmo de la suma de las alfas del modelo original y el del modelo con matriz de transición homogénea. Debajo, su suma acumulada (representación en el tiempo de diferencia entre ambas verosimilitudes). Como puede observarse, el modelo original para el boquerón (matriz de transición no homogénea), resulta más verosímil.

En la figura 4.10 se muestra para la primera observación, el logaritmo de la "mejora" para cada instante, expresada como

$$\ln\left(\sum_{\forall i} \alpha_t(i)\right) \; - \; \ln\left(\sum_{\forall i} \alpha_t^{homo}(i)\right)$$

La suma de esta cantidad para todos los instantes nos proporciona la diferencia de verosimilitud de la secuencia de observaciones respecto a ambos modelos, tomando como base el modelo de transiciones homogéneas. Para esta primera secuencia, la diferencia es de 13.102 belios a favor del modelo original. Para la segunda secuencia (figura 4.11) la diferencia se salda con 4.982 belios a favor del modelo homogéneo que, en este otro caso, casa mejor con la observación (figura 4.13).

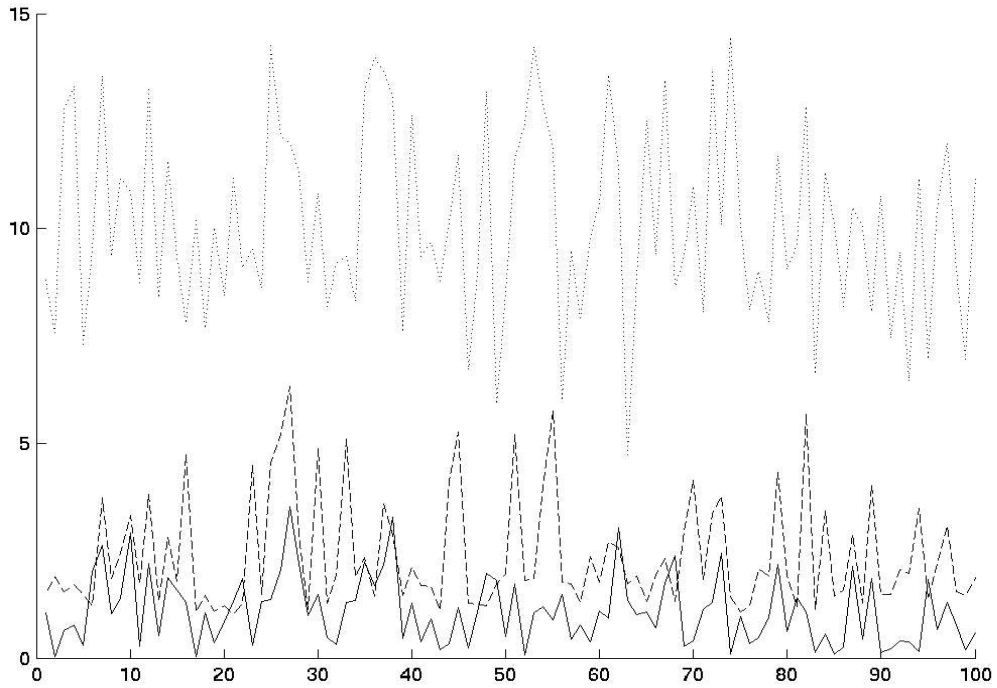


figura 4.11 Segunda secuencia de observaciones radar, que casa mejor con el modelo de transiciones homogéneas.

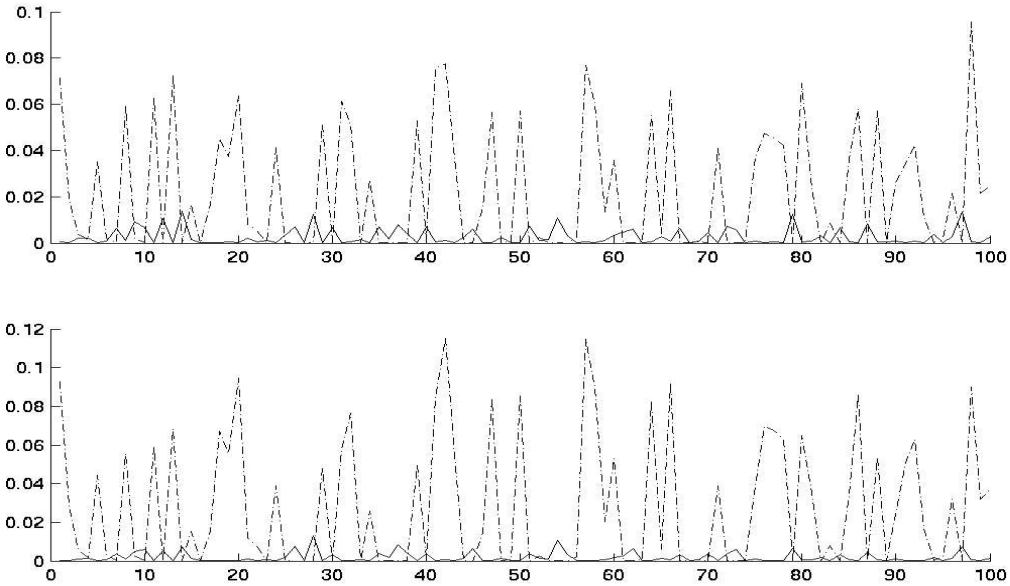


figura 4.12 Arriba, variable hacia delante correspondiente a un modelo del boquerón que obvia las probabilidades de transición. Debajo, variable hacia delante modelo de boquerón visto en la simulación anterior. Linea discontinua, comiendo, linea continua, huyendo.

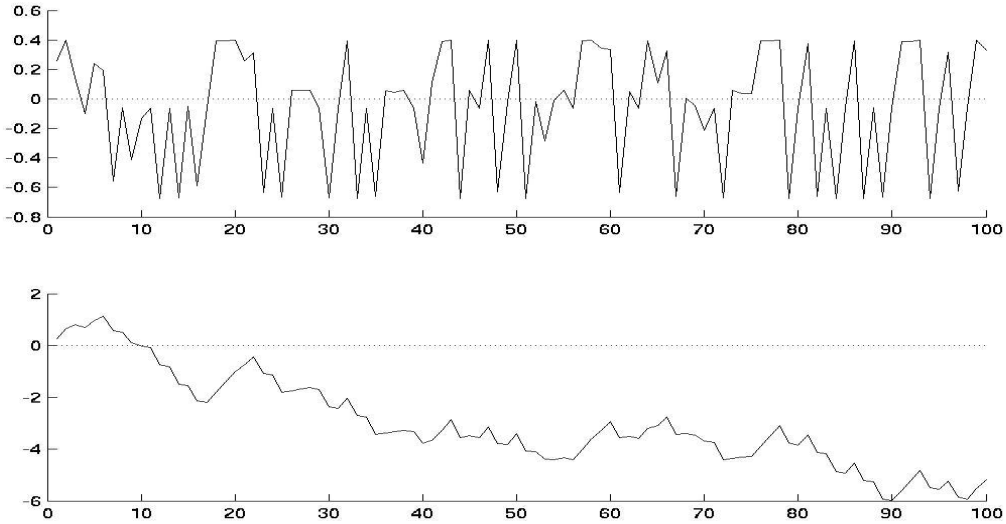


figura 4.13 Arriba, diferencia entre el logaritmo de la suma de las alfas del modelo original y el del modelo con matriz de transición homogénea. Debajo, su suma acumulada (representación en el tiempo de diferencia entre ambas verosimilitudes). Como puede observarse, el modelo original para el boquerón (matriz de transición no homogénea), resulta más verosímil.

5. Cálculo del camino óptimo

5.1 Criterios de optimalidad

Este problema se nos plantea cuando tenemos que decidir, dada un modelo, cual es la secuencia de estados óptima respecto a una secuencia de observación. Por qué estados es más probable que pase el sistema para adaptarse a la observacion.

Lo primero que tenemos que decidir es a qué llamamos secuencia de estados óptima. Supongamos que nos ceñimos al caso de que el criterio de optimalidad sea la máxima verosimilitud. Aunque nos ciñamos a usar este criterio, aún nos quedan varias alternativas. La secuencia óptima podría ser

i. Aquella que esté compuesta por aquellos estados que para cada instante de tiempo maximizen $p(q_t=i/\boldsymbol{o},\lambda)$, i.e una secuencia de *estados óptimos*. En este caso se escoge en cada paso el eslabón con maximiza verosimilitud y luego se añade a la cadena.

ii. Aquella que de entre todas las posibles secuencias de estados, maximize $p(\boldsymbol{o}/q,\lambda)$, i.e una *secuencia óptima* de estados. Es el camino individual más verosímil, la secuencia de estados que más presumiblemente el sistema habría seguido de haber generado la observación.

5.2 Secuencia de estados óptimos

En el primer caso estamos considerando en cada instante el estado más probable dado que se ha observado una secuencia. Formalmente buscamos

$$\underset{i}{argmax}\ p(q_t=i/o,\lambda)$$

A la probabilidad de estar en i en el instante t dada una observación se le asigna una variable que será de utilidad cuando nos enfrentemos a la resolución del problema de la reestimación. Se define

$$\gamma_t(i) = p(q_t=i/o,\lambda) \tag{5.1}$$

Intuitivamente, el estado más probable en un instante t es aquel en el que será más probable que acaben los caminos que se adecuan a las observaciones pasadas y del que es más probable que salgan caminos que se adecuen a la observaciones futuras. Si recordamos la definición de alfa y beta, $\alpha_t(j)$ compendiaba, hasta el instante t, la verosimilitud de todos los caminos que llegaban al estado j. Por su parte, $\beta_t(j)$ condensaba la verosimilitud de todos los caminos que, en el instante t, nacían en j. En términos matemáticos

$$p(o,q_t=i/\lambda) = p(o_1 \ldots o_t, q_t=i/\lambda)\, p(o_{t+1} \ldots o_T / q_t=i,\lambda) = \alpha_t(i)\beta_t(i)$$

De modo que hemos hallado la probabilidad conjunta de observar la secuencia *o* y estar en el estado i en el instante t. Gamma se deriva a partir de ella usando Bayes, o notando que es necesario un coeficiente de normalización que haga que $\sum_{\forall j} \gamma_t(j) = 1$:

$$\gamma_t(i) = \frac{p(o,q_t=i/\lambda)}{p(o/\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{p(o/\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{\forall j} \alpha_t(j)\beta_t(j)} \tag{5.2}$$

Para encontrar nuestro camino $Q^* = \left\{q_1^*, q_2^*, \ldots q_T^*\right\}$ de eslabones óptimos sólo es necesario calcular alfa y beta, construir gamma y escoger aquellos estados q_t^* que, en cada instante cumplan

$$q_t^* = \underset{j}{argmax}\ \gamma_t(j) \tag{5.3}$$

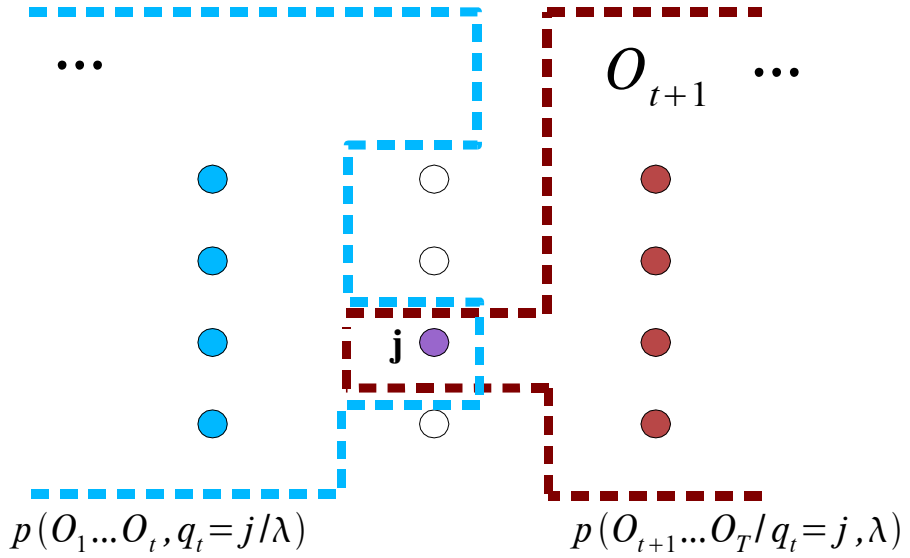


figura 5.1 Representación gráfica de cómo contribuyen la variable hacia delante (izquierda) y hacia atrás (derecha) al cálculo de la probabilidad instantánea de ocupación de estado, dada una secuencia de observaciones

Finalmente un momento para la reflexión. ¿Qué podemos decir del camino así obtenido? Lo único que podemos decir es lo que emana de la definición: no es sino la concatenación de aquellos estados en es el camino cuyos eslabones son, individualmente y para cada instante, los que tienen más papeletas de haber dado lugar a la observación. Maximiza el número esperado de estados correctos.

A esos estados individuales se puede llegar desde multitud de caminos, pero no necesariamente desde cualquier camino. Si algún término de la matriz de transición fuera cero (es decir, la transición por ejemplo de i a j está prohibida) nada impide que aun así el estado i sea el más probable en un instante, y el j en el posterior. El camino de eslabones óptimos puede que ni tan siquiera sea transitable, puede que no sea un camino válido que el modelo pueda recorrer para generar la observación.

5.3 Secuencia óptima de estados

Una solución al problema anterior es cambiar el criterio de optimalidad para que abarque pares o tripletes de estados en lugar de estados individuales. La modificación más ampliamente aceptada y que aquí se expone es la de buscar la única secuencia completa de estados óptima. Para ello existe una técnica basada en métodos de programación dinámica llamada algoritmo de Viterbi. Para poder encontrar la secuencia de estados $q = \{q_{1,} q_{2,} \ldots q_T\}$ que con más probabilidad hubiera generado la observación $o = \{o_1, o_{2,} \ldots o_T\}$ definimos la cantidad

$$\delta_t(i) = \max_{q_1, q_2, \ldots q_{t-1}} p(q_1 q_2 \ldots q_t = i, o_1 o_2 \ldots o_t / \lambda)$$

La expresión define $\delta_t(i)$, que representa la probabilidad del camino más probable de entre todos los caminos parciales que, teniendo en cuenta las observaciones hasta ese instante, desembocan en el estado i en el instante t. En un instante t, cada estado tendrá su propio mejor camino, de entre los que en él desembocan, y $\delta_t(i)$ es la *probabilidad* de ese mejor camino parcial. Delta se puede calcular recursivamente notando que por definición

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] \cdot b_j(o_{t+1})$$

$\delta_T(i)$ nos proporcionará al final la probabilidad del mejor camino completo de todos los que, tomando en cuenta las observaciones, mueren en el estado i. Nos da la probabilidad del camino, pero no el camino en sí. Para no perder el rastro según calculamos delta, hay que ir guardando las elecciones que hemos hecho a cada paso. Así al final, podremos tirar del hilo y recuperar el camino completo. Para mantener la traza de los estados visitados se usa la matriz $\psi_t(i)$, que almacena en cada instante de tiempo cuál es el estado que *precede* al estado i.

1) Inicialización

$$\delta_1(i) = \pi_i b_i(o_1) \qquad \forall i$$
$$\psi_1(i) = 0$$

2) Recursión

$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] \cdot b_j(o_t) \qquad \forall t, \forall i, \forall j$$
$$\psi_t(j) = \operatorname*{argmax}_i [\delta_{t-1}(i) a_{ij}]$$

3) Terminación

$$p^* = \max_i [\delta_T(i)]$$
$$q_T^* = \operatorname*{argmax}_i [\delta_T(i)]$$

4) Recuperación del camino

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

Vemos que es muy similar al cálculo de la variable hacia delante, salvo por la de recuperación del camino. Usando esta técnica podemos hacernos una idea de la sucesión de estados por la que ha pasado el sistema para generar la salida. Esto, claro está, si es que los estados de nuestro modelo se corresponden con algo parecido a un estado en el sistema real.

5.4 Simulación

Esta simulación persigue dejar patente le hecho de que la secuencia de estados óptimos puede ser distinta a la secuencia óptima de estados. Para ello, usaremos un modelo ampliado del comportamiento de una especie ya expuesta el el capítulo anterior, la sardina. En este nuevo modelo se ha añadido un tercer estado, que podríamos etiquetar como “durmiendo”. Este nuevo estado sólo es accesible desde el estado

uno (“comiendo”) - en ningún caso desde el estado dos o “huyendo”- y se caracteriza por una baja velocidad de desplazamiento y un tamaño del banco más o menos entre los tamaños que caracterizan a los otros estados.

Modelo a ampliado.

$$\boldsymbol{A}^a = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.8 & 0 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} \qquad \pi^a = \begin{bmatrix} 0.3390 & 0.5932 & 0.0678 \end{bmatrix}$$

$$b_1(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_1, \boldsymbol{U}_1); \qquad \mu_1^a = \begin{bmatrix} 0.1 & 7 & 1.3 \end{bmatrix} \qquad \boldsymbol{U}_1^a = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

$$b_2(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_2, \boldsymbol{U}_2); \qquad \mu_2^a = \begin{bmatrix} 2.2 & 13 & 5.2 \end{bmatrix} \qquad \boldsymbol{U}_2^a = \begin{bmatrix} 1.9 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1.6 \end{bmatrix}$$

$$b_3(\boldsymbol{o}_t) = N(\boldsymbol{o}_t, \mu_3, \boldsymbol{U}_3); \qquad \mu_3^a = \begin{bmatrix} 0.04 & 9 & 3 \end{bmatrix} \qquad \boldsymbol{U}_3^a = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}$$

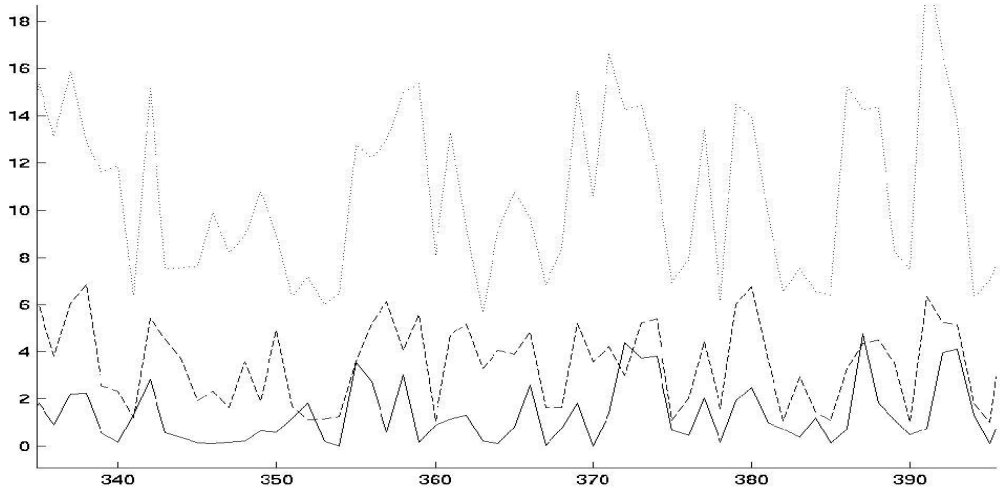


figura 5.2 Detalle de la secuencia de observaciones para la simulación 5a. La linea punteada se corresponde con la longitud del eje mayor del banco, la rayada con la excentricidad, y la continua con la celeridad del desplazamiento. Los puntos alrededor de t=364 serán de especial interés para la simulación.

Confrontamos este modelo con la observación mostrada (en parte) en la figura 5.2. Calculamos en primer lugar la probabilidad de estar en el instante t en el estado i, dado el modelo y las observaciones, que no es sino la definición de $\gamma_t(i)$. El cálculo no introduce ningún concepto nuevo que no hayamos visto hasta ahora, ya que se basa enteramente en las variables hacia delante y hacia atrás. La única consideración a tener en cuenta es el problema del escalado. Siempre que trabajamos con dichas variables lo hacemos con su versión escalada, por el problema ineludible de la falta de rango dinámico. Lo que se hace es sustituir en la expresión de gamma las variables por su versión escalada. Puede demostrarse que si utilizamos los mismos coeficientes de escalado c_t para $\hat{\alpha}_t$ y $\hat{\beta}_t$, estos coeficientes se contrarrestan en el cálculo de la $\gamma_t(i)$ así definida.

Para ello recordamos del capítulo anterior las expresiones 4.3a y 4.3b, y formulamos su equivalente para la variable hacia detrás, partiremos de la base de que

$$c_t = \frac{1}{\sum_{\forall j} \hat{\alpha}_t(j)} \qquad \begin{aligned} \bar{\alpha}_t &= c_t \hat{\alpha}_t \\ \bar{\beta}_t &= c_t \hat{\beta}_t \end{aligned}$$

$$\hat{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \bar{\alpha}_t(i) a_{ij}\right] b_j(o_{t+1}) = c_t \left[\sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij}\right] b_j(o_{t+1})$$

$$\hat{\beta}_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \bar{\beta}_{t+1}(j) = c_{t+1} \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \hat{\beta}_{t+1}(j)$$

$$\alpha_t(i) = \hat{\alpha}_t(i) \prod_{k=1}^{t-1} \frac{1}{c_k} \qquad \beta_t(i) = \hat{\beta}_t(i) \prod_{k=T}^{t+1} \frac{1}{c_k}$$

El uso de los mismos coeficientes de escalado para ambas variables no conduce al desbordamiento de $\hat{\beta}_t$, dado que en principio ambas son del mismo orden. sustituyendo estas últimas expresiones en el cálculo

de gamma tenemos que

$$C = \prod_{k=1}^{T} c_k$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{\forall j}\alpha_t(j)\beta_t(j)} = \frac{(\frac{c_t}{C})\hat{\alpha}_t(i)\hat{\beta}_t(i)}{\sum_{\forall j}(\frac{c_t}{C})\hat{\alpha}_t(j)\hat{\beta}_t(j)} = \frac{\hat{\alpha}_t(i)\hat{\beta}_t(i)}{\sum_{\forall j}\hat{\alpha}_t(j)\hat{\beta}_t(j)}$$

De modo que gamma puede calcularse a partir de las variables escaladas sin mayor problema, y así lo hacemos.

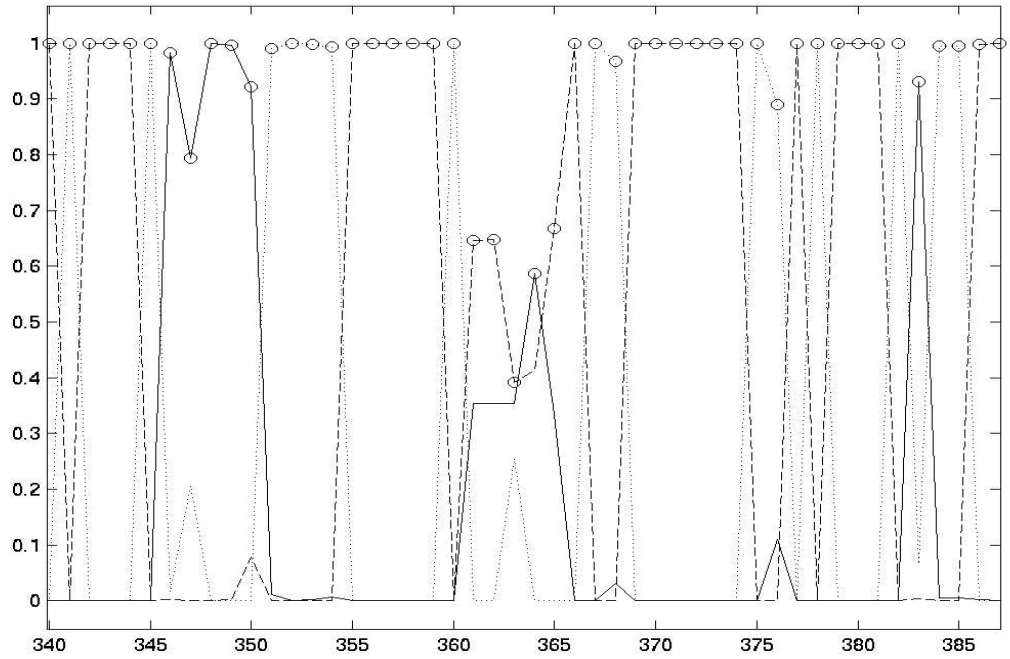


figura 5.3 Probabilidad de que el modelo se encuentre en uno u otro estado dado el modelo y las observaciones ($\gamma_t(i)$), en el intervalo temporal de interés para la simulación. La línea punteada se corresponde con el estado uno (comiendo), la rayada con el estado dos (huyendo) y la continua con el estado tres (durmiendo). Nótese como en el instante t= 364 lo más probable es que el banco esté durmiendo.

Es de reseñar que por su propia definición, $\gamma_t(i)$ suma uno sobre los estados en cualquier instante. El la gráfica se muestra el valor de gamma para cada uno de los estados, coronando en cada instante con un círculo el valor máximo de entre ellos. Si miramos con atención el modelo y las observaciones, podemos reconocer el funcionamiento de gamma. Por ejemplo en torno a t = 357 podemos observar que la velocidad del banco es grande y el tamaño también. Esas observaciones casan bastante bien con la variable aleatoria asociada al estado dos (huyendo) y $\gamma_t(2)$ toma para ellas valores cercanos a la unidad.

Si atendiéramos a la primera definición del capítulo y tomásemos como mejor secuencia de estados aquellos que cumplan (5.3), tendríamos que aceptar que entre los instantes t=363 y t=364, el sistema pasa del estado dos al estado tres, transición que *está prohibida*. Como ya anticipamos en la teoría, vemos ahora que en la práctica el camino así formado no puede ser recorrido por el modelo (su verosimilitud se hace cero en cuanto haga una transición con probabilidad nula).

Aunque el camino formado por los estados individualmente más probables puede ser intransitable, la mayor parte de las veces sus eslabones coinciden con los del camino único más probable, que obtenemos usando el método de Viterbi.

La verosimilitud total de las observaciones respecto al modelo de halla como siempre, y en este caso toma el valor

$$\ln(p(\boldsymbol{o}/\lambda)) \simeq -2654.7$$

El camino único más probable ofrece un valor de verosimilitud logarítmica de

$$\ln(p(o/q^*,\lambda)) \simeq -2667.1$$

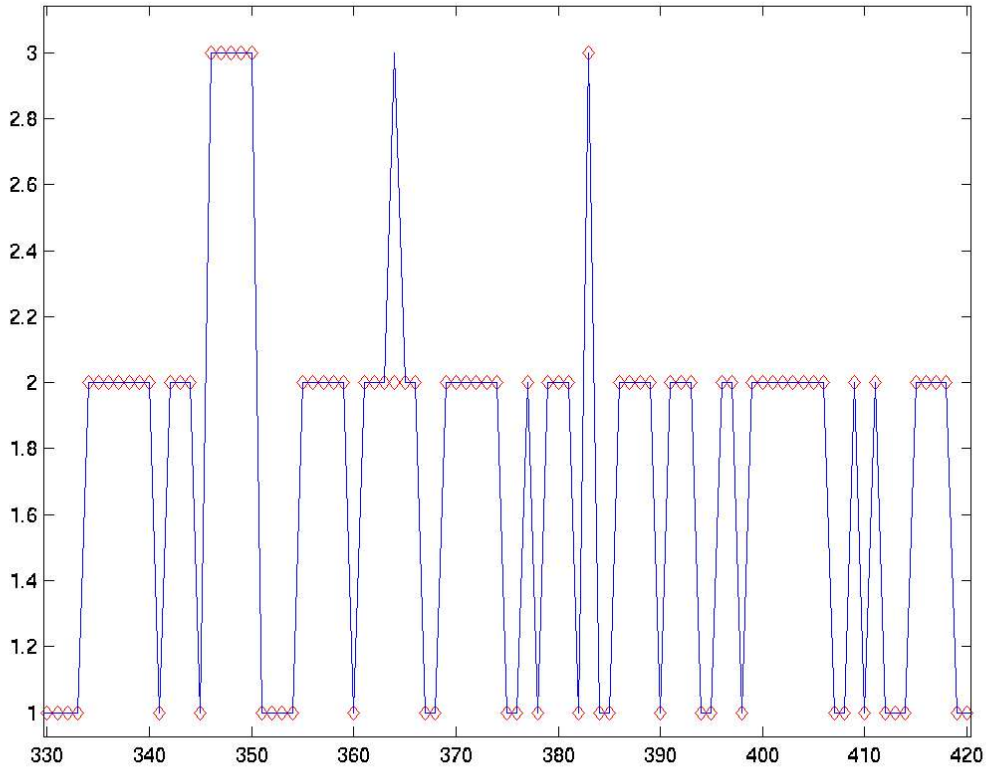


figura 5.4 Gráfico que representa el camino de los estados individualmente más probables y el camino único más probable. En línea continua, tenemos aquellos estados para los que gamma es máxima en cada instante. Los rombos representan los estados que componen el camino único más probable (viterbi).

es decir, unos 12.5 belios por debajo del total, lo que hace que aporte una parte entre $2.7 \cdot 10^5$ del total de la verosimilitud. Esto puede parecer poca contribución, pero si tenemos en cuenta que la observación tiene quinientas muestras, y hay tres estados, podemos deducir que el número total de caminos que contribuyen verosimilitud es del orden de 3^{500} (aproximadamente 10^{239}). Desde luego su aportación supera (con creces) a la media esperada.

6. Reestimación de parámetros

6.1 Introducción

La resolución de este último problema, el más dificil de los tres, ya está suavizada gracias a los conceptos que se introdujeron en la resolución de los otros dos. Variables hacia delante y hacia detrás serán de gran utilidad en la reestimación de los parámetros del modelo.

Lo que perseguimos con la reestimación de los parámetros es conseguir que el modelo refleje lo mejor posible una serie de observaciones, en el sentido como siempre de que la verosimilitud de las mismas respecto al primero sea lo mayor posible. Esta cuestión está claramente relacionada con el entrenamiento del modelo, previo a poder usarlo para reconocimiento.

No se conoce ningún método analítico de encontrar ese conjunto de parámetros que maximizen la verosimilitud de la observación, pero podemos alcanzar máximos locales usando métodos iterativos. El método más usado para maximizar localmente $p(o/\lambda)$ es el de Baum-Welch . Es equivalente a la aplicación del algoritmo EM sobre el HMM y no es la única alternativa (podemos usar otras técnicas como descenso de gradiente, etc...), pero es la que desarrollaremos en este capítulo.

6.2 Baum-Welch

6.2.1 Generalidades

Este método fue desarrollado inicialmente por Baum y sus colaboradores. Una forma intuitiva de ver el funcionamiento del proceso es la siguiente.

Partimos del modelo con unos parámetros iniciales, y una observación respecto a la cual tenemos que maximizar la verosimilitud. Entretejidos en el cálculo de la verosimilitud se encuentran el conjunto de todos los posibles caminos que el modelo podría haber recorrido. Estos caminos abren un abanico inmenso: ramificamos todos los posibles pasados de una única observación, todas las formas que tenga el modelo de generarla aparecen ponderadas cada una por su grado de realismo. Y es sobre de este grupo de caminos fantasmales y sus probabilidades, sobre este grupo ampliado de miles de ejecuciones del modelo, sobre el cual podemos extraer por simple conteo nuevos valores para cualquier parámetro del modelo. Como los caminos que mejor describen la observación pesan más, influirán más en los valores reestimados, y es de esperar que el modelo así reconstituido arroje mejores valores de verosimilitud que su predecesor.

6.2.2 Exposición

6.2.2.3 Sobre la reestimación de A y π

Empezaremos definiendo $\xi_t(i,j)$ como la probabilidad de, estando en el estado i en el instante t, pasar al estado j en t+1, dada una observación y el modelo. Formalmente

$$\xi_t(i) = p(q_t=i, q_{t+1}=j/o, \lambda) \tag{6.1}$$

Para calcular $\xi_t(i,j)$ podemos basarnos en valores ya conocidos. $\alpha_t(i)$ nos aporta la probabilidad de acabar en el estado i en el instante t; $\beta_{t+1}(j)$ se ocupa la probabilidad al partir del estado j en el instante t + 1. Y la probabilidad de pasar del estado i al j dado o_t viene dada por la matriz de transición y por la densidad asociada al estado de destino. Formalmente

$$\xi_t(i,j) = p(q_t=i, q_{t+1}=j/o, \lambda) = \frac{p(o, q_t=i, q_{t+1}=j/\lambda)}{p(o/\lambda)} =$$

$$= \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{p(o/\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{\forall i}\sum_{\forall j}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

El denominador $p(o/\lambda)$ puede obtenerse (como de hecho se hace) como $\sum_{\forall i}\alpha_T(i)$, pero se ha expresado de esta forma para dejar patente el hecho de que su función no es otra que la que le otorga el teorema de Bayes, normalizar a uno la suma de probabilidades.

En el capítulo anterior, cuando buscábamos el camino formado por los estados óptimos, definimos $\gamma_t(i)$ como la probabilidad de encontrarnos en el estado i en el instante t, dada la secuencia de observaciones. Esta cantidad se relaciona fácilmente con $\xi_t(i,j)$. Si sumamos sobre todos los posibles destinos de la transición entre estados que es el eje de $\xi_t(i,j)$, lo que nos queda es precisamente la probabilidad de estar en el origen de dicha transición.

$$\gamma_t(i) = \sum_{\forall j}\xi_t(i,j) = \frac{\alpha_t(i)\beta_t(i)}{p(o/\lambda)} \qquad (6.2)$$

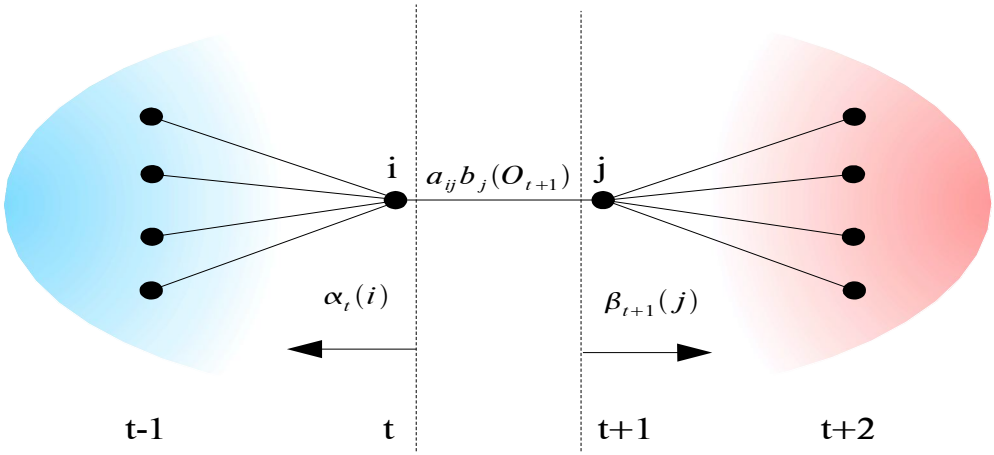


figura 6.2 Representación gráfica de la estructura del cálculo de la probabilidad conjunta de que el modelo esté en el estado i en el instante t, y en el j en t+1, sujeto a la observaciones.

$\gamma_t(i)$ nos aporta mucha información acerca de los estados del modelo. La suma respecto al tiempo puede interpretarse como la esperanza del número de veces que se visita el estado i. O lo que es lo mismo, el el número esperado de veces que una transición se inicia desde el estado i (si excluimo el instante final, $t = T$, desde el cual no hay transición posible). Del mismo modo, la suma de $\xi_t(i,j)$ respecto al tiempo representa el numero esperado de transiciones del estado i al j.

$\sum_{t=1}^{T-1} \gamma_t(i)$= número esperado de transiciones desde el estado i

$\sum_{t=1}^{T-1} \xi_t(i,j)$= número esperado de transiciones desde el estado i al estado j

Usando estas dos interpretaciones no es difícil encontrar un método razonable de reestimación de los parámetros del modelo. Si nos remitimos a la definición de probabilidad en términos de frecuencias de ocurrencia, los coeficientes de la matriz de transición A se reducirían a contar el número de veces que el modelo pasa del estado i al estado j, normalizado por el número total de transiciones desde el estado i (para cumplir la restricción estocástica). Las componentes del vector de probabilidades iniciales de ocupación de estado π vendrían dadas por el número de veces que el modelo estuviera en un estado dado en el instante inicial ($t = 1$) .

$$\bar{\pi}_i = \textit{número esperado de veces en que } q_1 = i = \gamma_1(i) \qquad (6.3)$$

$$\bar{a}_{ij} = \frac{\textit{número esperado de transiciones de } i \textit{ a } j}{\textit{número esperado de transiciones desde } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (6.4)$$

6.2.2.2 Sobre la reestimación de las variables aleatorias asociadas a los estados

La expresión para la reestimación de la densidad b_j(x) en el caso contínuo o de las probabilidades de símbolo $b_j(v_k)$ en el discreto difieren en la forma, pero no en el fondo. $\gamma_t(i)$ nos permite evaluar la probabilidad de que, dada una observación, nos encontremos en un estado. Los parámetros de la variables aleatorias asociada a un estado digamos i se estimarán como se haría normalmente en base a las observaciones, a excepción de que en este caso las muestras vendrán ponderadas según la probabilidad que tengan de pertenecer a este estado i. probabilidad de estar en un estado o en otro

i) variable discreta

Supongamos para el caso discreto que en un estado tenemos una distribución multinomial con probabilidades $p_j(k)$ de generar los símbolos v_k . Lo que haríamos normalmente si conociéramos un conjunto de muestras de entrenamiento de salidas que sabemos pertenecen a esa variable, es contar cuantas veces aparece cada símbolo, y asignar a $p_j(k)$ su frecuencia relativa.

$$\bar{p}_j(k) = \frac{\textit{número de veces que aparece } v_k}{\textit{número total de observaciones pertenecientes al estado } j} = \frac{\sum_{t=1}^{T} \delta_k}{\sum_{t=1}^{T} 1}$$

$$\delta_k = \begin{Bmatrix} 1 & \text{si } O_t = v_k \\ 0 & \text{e.o.c.} \end{Bmatrix}$$

Aquí ocurre igual, con la salvedad de que las muestras no pertenecen con total seguridad a la variable, sino a través de una probabilidad. formalmente:

$$\bar{p}_j(k) = \frac{\textit{nro esperado de veces en el estado } j \textit{ observando } v_k}{\textit{número esperado de veces en el estado } j} = \frac{\sum_{t=1}^{T} \gamma_t(j)\,\delta_k}{\sum_{t=1}^{T} \gamma_t(j)}$$

En este caso los parámetros que rigen el comportamiento de la variable discreta coinciden con la probabilidad de observación de los símbolos. De haber usado otro tipo de distribución (digamos Poissoniana en lugar de multinomial) obviamente esto no sucedería.

ii) variable continua

Si ahora miramos el caso de la variable contínua y tomamos como ejemplo una distribución Gaussiana multidimensional, la densidad de probabilidad vendría dada por $b_j(\boldsymbol{x}) = N(\mu_j, \boldsymbol{U}_j)$, donde los parámetros a optimizar serán la media μ_j y la matriz de covarianzas $\boldsymbol{U}_j$. Los parámetros óptimos en el sentido de máxima verosimilitud para de un conjunto de entrenamiento $\boldsymbol{o} = \{\boldsymbol{o}_1 \boldsymbol{o}_2 \ldots \boldsymbol{o}_T\}$ se obtienen usando

$$\bar{\mu}_j = \frac{\sum_{\forall t} \boldsymbol{o}_t}{\sum_{\forall t} 1} \qquad \bar{\boldsymbol{U}}_j = \frac{\sum_{\forall t} (\boldsymbol{o}_t - \bar{\mu}_j)(\boldsymbol{o}_t - \bar{\mu}_j)^t}{\sum_{\forall t} 1}$$

Siguiendo el mismo razonamiento que en el caso precedente podemos reescribir las fórmulas introduciendo la probabilidad de pertenencia al estado, tendríamos para los valores reestimados

$$\bar{\mu}_j = \frac{\sum_{\forall t} \gamma_t(j)\,\boldsymbol{O}_t}{\sum_{\forall t} \gamma_t(j)} \qquad \bar{\boldsymbol{U}}_j = \frac{\sum_{\forall t} \gamma_t(j)(\boldsymbol{O}_t - \bar{\mu}_j)(\boldsymbol{O}_t - \bar{\mu}_j)^t}{\sum_{\forall t} \gamma_t(j)}$$

iii) mezcla de variables

Tanto en el caso contínuo como en el caso discreto podemos toparnos con que las variables aleatorias asociadas a los estados son en sí mismas una mezcla de variables. Podríamos estar hablando por ejemplo de una mezcla de gaussianas, cuya densidad vendría dada por

$$b_j(x) = \sum_{\forall k} c_{jk} N(\mu_{jk}, \boldsymbol{U}_{jk}) \qquad \text{s.a} \quad \sum_{\forall k} c_{jk} = 1$$

Esta nueva situación nos plantea nuevas incertidumbres. Para poder estimar la media y la varianza de cada una de las gaussianaas que componen la mezcla, necesitaríamos saber cual de ellas ha generado la observación. Es lo mismo que nos pasaba con los estados, no podíamos relacionarlos directamente con las observaciones (como ocurre con las cadenas de Markov). Si en los puntos precedentes solucionamos

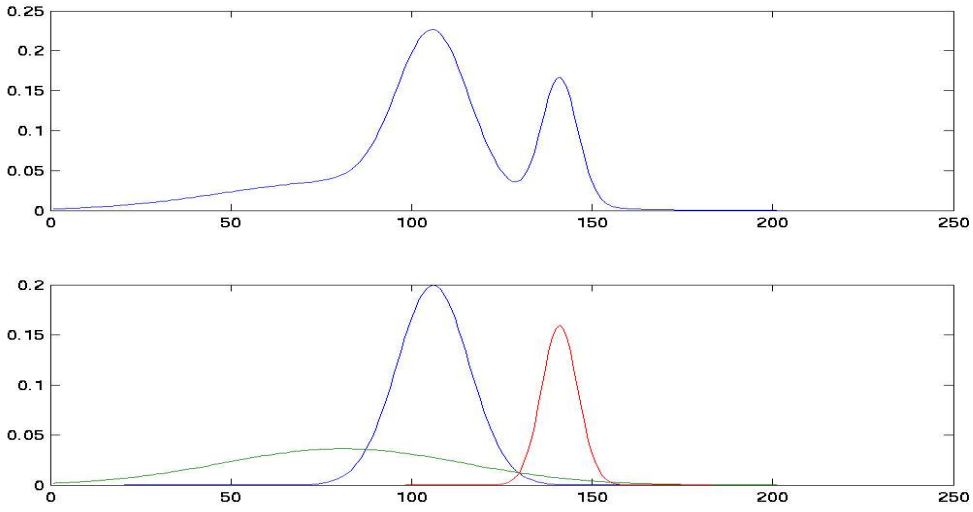


figura 6.3 Función densidad de probabilidad multimodal compuesta por una mezcla de gaussianas. Arriba, densidad completa. Debajo, componentes de la mezcla.

el problema echando mano de $\gamma_t(i)$ (la probabilidad de ocupación del estado i en el instate t dada una secuencia de observaciones) ahora haremos exactamente lo mismo. Si definimos una nueva cantidad $\gamma_t(i,k)$ como la probabilidad de que estemos en el estado i y $\boldsymbol{o}_t$ haya sido generada por la k-ésima componente de la mezcla, dada una secuencia de observaciones $\boldsymbol{o}=\{\boldsymbol{o}_1\boldsymbol{o}_2...\boldsymbol{o}_T\}$. Formalmente

$$\begin{aligned}\gamma_t(i,k) &= \gamma_t(i)\left[\frac{c_{ik}N(\boldsymbol{o}_t,\mu_{ik},\boldsymbol{U}_{ik})}{\sum_{\forall m} c_{im}N(\boldsymbol{o}_t,\mu_{im},\boldsymbol{U}_{im})}\right] = \\ &= \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{\forall j}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{ik}N(\boldsymbol{o}_t,\mu_{ik},\boldsymbol{U}_{ik})}{\sum_{\forall m} c_{im}N(\boldsymbol{o}_t,\mu_{im},\boldsymbol{U}_{im})}\right] \qquad (6.5)\end{aligned}$$

Ya que conocemos con qué probabilidad se corresponde a cada gaussiana cada muestra, reestimamos igual que antes, sólo que ponderando por esta probabilidad.

$$\bar{\mu}_{ik} = \frac{\sum_{\forall t}\gamma_t(i,k)\boldsymbol{o}_t}{\sum_{\forall t}\gamma_t(i,k)} \qquad \bar{\boldsymbol{U}}_{ik} = \frac{\sum_{\forall t}\gamma_t(i,k)(\boldsymbol{o}_t-\bar{\mu}_{ik})(\boldsymbol{o}_t-\bar{\mu}_{ik})^t}{\sum_{\forall t}\gamma_t(i,k)}$$

y los coeficientes de la mezcla se reestimarán como lo que son, una multinomial

$$\bar{c}_{ik} = \frac{\textit{nro esperado de veces en } i \textit{ y genera } k-\textit{ésima}}{\textit{número esperado de veces en el estado } i} = \frac{\sum_{\forall t}\gamma_t(i,k)}{\sum_{\forall t}\sum_{\forall k}\gamma_t(i,k)} \qquad (6.6)$$

Siempre que la elección entre los miembros de una mezcla esté regida por una multinomial, podemos ver que la situación es equivalen-

te a tener un modelo ampliado. El modelo ampliado tendría más estados que el original, y cada estado llevaría asociada una única variable aleatoria simple.

Con esto se quiere dejar patente que las fórmulas y razones expuestas para la reestimación de los parámetros de variables aleatorias compuestas tienen su fundamento en las demostraciones venideras, que se llevarán a cabo usando, por comodidad, variables simples.

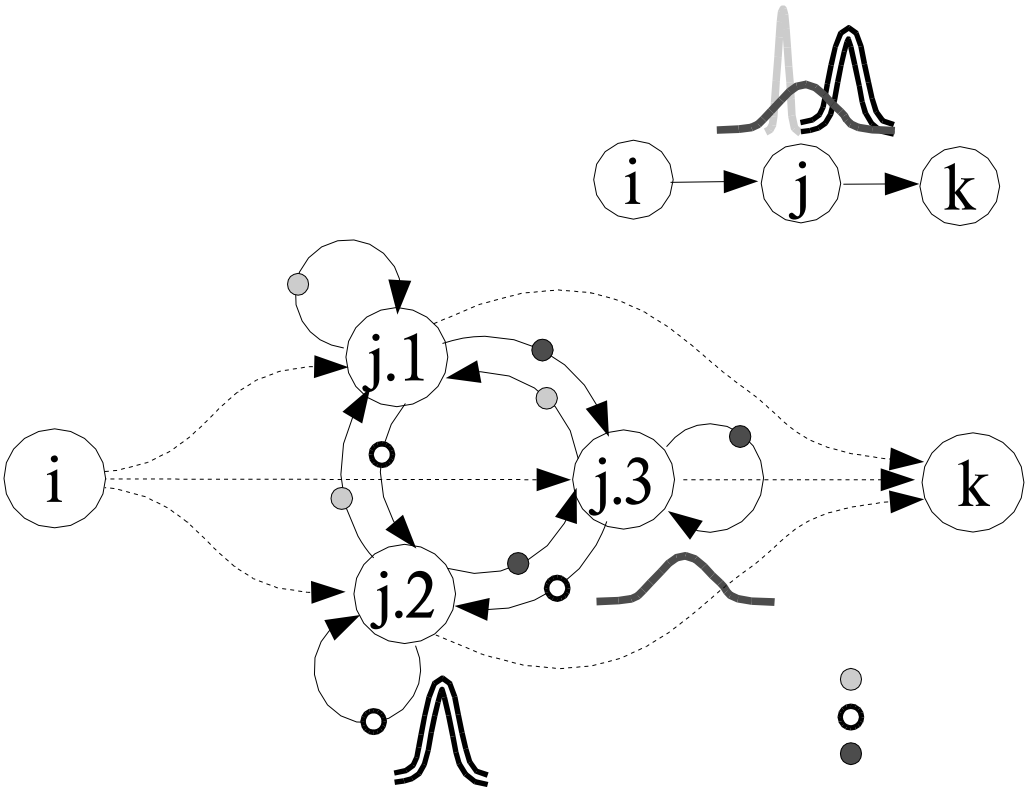


figura 6.4 Representación gráfica de la equivalencia ente un modelo donde los estados tienen asociada una mezcla de variables aleatorias, y un modelo ampliado en el que cada estado tiene una variable aleatoria simple.

6.2.3 Resumen del algoritmo

i. Inicialización.

El intento de escoger un modelo adecuado con unos parámetros iniciales favorables ha de llevarse a cabo en base a conocimientos previos, suposiciones o, en el peor de los casos, al azar.

ii. Recursión.

a) usando $\lambda(\theta)$ calculamos

$$\alpha_t(i), \beta_t(i) \qquad \forall t, \forall i$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{p(O/\lambda)} \qquad 1 \le t \le T-1, \forall i, \forall j$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(O/\lambda)} \qquad \forall t, \forall i$$

$$\gamma_t(i,j) = \gamma_t(i)\left[\frac{c_{ik} N(\boldsymbol{O}_t, \mu_{ik}, \boldsymbol{U}_{ik})}{\sum_{\forall m} c_{im} N(\boldsymbol{O}_t, \mu_{im}, \boldsymbol{U}_{im})}\right] \qquad \forall t, \forall i, \forall j$$

b) Calculamos $\bar{\theta}$, los parámetros reestimados usando

$$\bar{\pi}_i = \gamma_1(i) \qquad \bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \bar{c}_{ik} = \frac{\sum_{\forall t} \gamma_t(i,k)}{\sum_{\forall t} \sum_{\forall k} \gamma_t(i,k)}$$

$$\bar{\mu}_{ik} = \frac{\sum_{\forall t} \gamma_t(i,k) \boldsymbol{o}_t}{\sum_{\forall t} \gamma_t(i,k)} \qquad \bar{\boldsymbol{U}}_{ik} = \frac{\sum_{\forall t} \gamma_t(i,k)(\boldsymbol{o}_t - \bar{\mu}_{ik})(\boldsymbol{o}_t - \bar{\mu}_{ik})^t}{\sum_{\forall t} \gamma_t(i,k)}$$

c) Hacemos $\theta = \bar{\theta}$

iii. Finalización.

Cuando la mejora crezca por debajo de un umbral, empeore respecto a un conjunto de control o cualquier otro criterio.

6.3 Demostración

6.3.1 Motivación

Con lo expuesto hasta ahora estamos en condiciones de reestimar cualquiera de los parámetros del sistema, pero las explicaciones dadas han sido más bien cualitativas. Si utilizamos esas fórmulas para reestimar los parámetros recursivamente, ¿llegaremos a algún tipo de máximo? ¿estamos siguiera mejorando la verosimilitud del modelo? Las expresiones expuestas, en apariencia deducidas por lógica, derivan de la optimización indirecta de la función verosimilitud.

En el trabajo original de Baum la optimización de la verosimilitud se consigue utilizando una función auxiliar $Q(\lambda(\bar{\theta}),\lambda(\theta))$, llamada función de Baum. La función de Baum se define como

$$Q(\bar{\theta},\theta) = \sum_{\forall q} p(o,q/\bar{\theta})\, p(o,q/\theta) \tag{6.7}$$

Donde se ha reemplazado las referencia explícita al modelo ($\lambda(\theta)$) por referencias a los parámetros del mismo (θ) en aras de la claridad. $\bar{\theta}$ representa los nuevos valores que tomarán los parámetros.

En la mayoría de la literatura esta función se utiliza sin más, haciendo referencia a que entronca con la aplicación del algoritmo EM a los HMM, pero en este caso vamos a clarificar su origen antes de pasar a optimizarla.

6.3.2 Algoritmo EM aplicado a un HMM

En esta demostración se utiliza la nomenclatura y los conceptos expuestos en el segundo capítulo. Su lectura es necesaria para entender el origen de la función mayorizante de la verosimilitud, pero será irrelevante en los siguientes apartados.

En este caso de estudio las variables ocultas z del algoritmo EM se corresponderán con el camino recorrido por el modelo para generar la observación. Éste camino viene como siempre representado por la secuencia de estados $q=\{q_1 q_2 ... q_t\}$. Las salidas $y=\{y_1 .... y_T\}$ se harán corresponder (para seguir la notación para los modelos de markov) con las observaciones $o=\{o_1 .... o_T\}$. La función mayorizante de la verosimilitud toma la forma

$$\bar{g}_i(\theta) \;=\; E_{z/y,\theta^i}\{\ln p(y,z/\theta)\} \;\equiv\; \sum_{\forall q} \ln p(o,q/\bar{\theta})\cdot p(q/o,\theta)$$

Esta expresión se puede modificar si aplicamos Bayes al último término, y posteriormente nos fijamos en que de cara a la optimización, $p(q/\theta)$ es una constante

$$\sum_{\forall q} \ln p(o,q/\bar{\theta})\cdot p(q/o,\theta) \;=\; \sum_{\forall q} \ln p(o,q/\bar{\theta})\cdot \frac{p(o,q/\theta)}{p(o/\theta)}$$
$$\equiv \sum_{\forall q} \ln p(o,q/\bar{\theta})\cdot p(o,q/\theta) \;=\; Q(\bar{\theta},\theta)$$

que no es sino la función de Baum (6.7). A partir de aquí las demostraciones por ambos caminos corren parejas.

6.3.3 Optimización de la función de Baum

Ahora optimizaremos la función, para hallar aquellos parámetros que optimizan la verosimilitud. Partiendo de la expresión de la función de Baum

$$Q(\bar{\theta},\theta) \;=\; \sum_{\forall q} p(o,q/\bar{\theta})\, p(o,q/\theta)$$

Si evaluamos el primer término $p(o,q/\bar{\theta})$, en términos como siempre de probabilidades de transición y verosimilitudes respecto al estado, obtenemos la expresión para la probabilidad conjunta

$$p(o,q/\bar{\theta}) = \bar{\pi}_{q_1}\bar{b}_{q_1}(o_1)\prod_{t=1}^{T-1}\bar{a}_{q_t q_{t+1}}\bar{b}_{q_{t+1}}(o_{t+1})$$

$$\ln p(o,q/\theta) = \ln\bar{\pi}_{q_1}+\sum_{t=1}^{T-1}\ln\bar{a}_{q_t q_{t+1}}+\sum_{t=1}^{T}\ln\bar{b}_{q_t}(o_t)$$

Gracias a las propiedades del logaritmo podemos reescribir $Q(\bar{\theta},\theta)$, como expresión de los parámetros separados, $\vec{\pi}$, $\boldsymbol{A}$ y $b_k(x)$.

$$Q(\bar{\theta},\theta) = Q_\pi(\bar{\pi})+\sum_{i=1}^{N}Q_{a_i}(\bar{a}_i)+\sum_{i=1}^{N}Q_{b_i}(\bar{b}_i) \tag{6.8}$$

$$Q_\pi(\bar{\pi}) = \sum_{i=1}^{N}p(o,q_1=i/\theta)\ln\bar{\pi}_i \tag{6.8a}$$

$$Q_{a_i}(\bar{a}_i) = \sum_{j=1}^{N}\sum_{t=1}^{T-1}p(o,q_t=i,q_{t+1}=j/\theta)\ln\bar{a}_{ij} \tag{6.8b}$$

$$Q_{b_i}(\bar{b}_i) = \sum_{t=1}^{T}p(o,q_t=i/\theta)\ln\bar{b}_i(o_t) \tag{6.8c}$$

Como la función de Baum se ha descubierto separable, la optimizaremos optimizando cada una de sus partes. Las se tienen en cuenta las restricciones estocásticas para las probabilidades de transición a_{ij} y las probabilidades iniciales π_i , las funciones auxiliares (6.8a) y (6.8b) exhiben la forma común dada por

$$\sum_{j=1}^{N}w_j\ln y_j \quad \text{sujeto a} \quad \sum_{j=1}^{N}y_j=1$$

Si derivamos esta expresión incluyendo la restricción estocástica mencionada tenemos que

$$L(y_j) = \sum_{j=1}^{N} w_j \ln y_j - \lambda(\sum_{j=1}^{N} y_j - 1)$$

$$\frac{\partial L}{\partial y_j} = \frac{w_j}{y_j} - \lambda = 0 \quad \rightarrow y_j = \frac{w_j}{\lambda}$$

$$\sum_{j=1}^{N} y_j = \frac{1}{\lambda}\sum_{j=1}^{N} w_j = 1 \quad \rightarrow \lambda = \sum_{j=1}^{N} w_j$$

$$y_j = \frac{w_j}{\sum_{j=1}^{N} w_j}$$

Lo que particularizando para las expresiones (6.8a) y (6.8b) nos da los óptimos para el vector de probabilidades iniciales y para la matriz de transición, respectivamente.

$$\bar{\pi}_i = \frac{p(o, q_1 = i/\theta)}{p(o/\theta)}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} p(o, q_t = i, q_{t+1} = j/\theta)}{\sum_{t=1}^{T-1} p(o, q_t = i/\theta)}$$

Que son las mismas expresiones (6.3) y (6.4) que presentamos en el apartado previo a la demostración. La optimización de las densidades de probabilidad asociadas a un estado la resolveremos, como ejemplo, para el caso de gaussianas multidimensionales.

Si la distribución de probabilidad de cada estado viene dada por una v.a gaussiana multidimensionales de D dimensiones tal que para un estado k tengamos

$$b_k(\boldsymbol{o}_t) = N(\boldsymbol{o}, \mu_k, \boldsymbol{U}_k) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{U}_k|}} \exp(-\frac{1}{2}(\boldsymbol{o}_t - \mu_k)^T \boldsymbol{U}_k^{-1}(\boldsymbol{o}_t - \mu_k))$$

$$\ln b_k(\boldsymbol{o}_t) = -\frac{1}{2}(\boldsymbol{o}_t - \mu_k)^T \boldsymbol{U}_k^{-1}(\boldsymbol{o}_t - \mu_k) + c$$

donde c es una constante que desaparecerá al derivar respecto al vector de medias o respecto a los elementos de la matriz de covarianzas, en

cada caso. Derivando e igualando a cero encontramos la estimación óptima de la media.

$$\begin{bmatrix} \partial/\partial\mu_{k1} \\ \partial/\partial\mu_{k2} \\ \vdots \\ \partial/\partial\mu_{kD} \end{bmatrix} Q_{b_k} = \sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta)\boldsymbol{U}_k^{-1}\cdot(\boldsymbol{o}_t - \mu_k) = \boldsymbol{U}_k^{-1}\cdot\sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta)(\boldsymbol{o}_t - \mu_k) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mu_k \sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta) = \sum_{\forall t} \left[p(\boldsymbol{o}, q_t = k/\theta)\boldsymbol{o}_t \right] \Rightarrow \quad \bar{\mu}_k = \frac{\sum_{\forall t} \left[p(\boldsymbol{o}, q_t = k/\theta)\boldsymbol{o}_t \right]}{\sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta)}$$

Operando ahora para los términos de la matriz de covarianzas, tenemos que

$$\frac{\partial \ln(N(\boldsymbol{o}_t, \mu_k, \boldsymbol{U}_k))}{\partial \boldsymbol{U}_k^{-1}} = \frac{\partial\left[\; 0.5\cdot\ln\left|\boldsymbol{U}_k^{-1}\right| - 0.5(\boldsymbol{o}_t - \mu_k)^T\boldsymbol{U}_k^{-1}(\boldsymbol{o}_t - \mu_k) + c \; \right]}{\partial \boldsymbol{U}_k^{-1}}$$

$$= \frac{0.5}{\left|\boldsymbol{U}_k^{-1}\right|}\left(\frac{\partial\left|\boldsymbol{U}_k^{-1}\right|}{\partial \boldsymbol{U}_k^{-1}}\right) - \frac{1}{2}(\boldsymbol{o}_t - \mu_k)(\boldsymbol{o}_t - \mu_k)^T = \frac{1}{2}\frac{adj(\boldsymbol{U}_k^{-1})^T}{\left|\boldsymbol{U}_k^{-1}\right|} - \frac{1}{2}(\boldsymbol{o}_t - \mu_k)(\boldsymbol{o}_t - \mu_k)^T$$

$$= \frac{1}{2}\left[\boldsymbol{U}_k - (\boldsymbol{o}_t - \mu_k)(\boldsymbol{o}_t - \mu_k)^T\right]$$

$$\frac{\partial Q_{b_k}(b_k)}{\partial \boldsymbol{U}_k^{-1}} = \sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta)\cdot\frac{1}{2}\left[\boldsymbol{U}_k - (\boldsymbol{o}_t - \mu_k)(\boldsymbol{o}_t - \mu_k)^T\right] = 0$$

$$\bar{\boldsymbol{U}}_k = \frac{\sum_{\forall t}\left[p(\boldsymbol{o}, q_t = k/\theta)(\boldsymbol{o}_t - \mu_k)(\boldsymbol{o}_t - \mu_k)^T \right]}{\sum_{\forall t} p(\boldsymbol{o}, q_t = k/\theta)}$$

Quedando por tanto demostrada la validez de las expresiones para la reestimación de los parámetros.

6.4 Simulaciones.

Las simulaciones de este capítulo culminan el aspecto de implementación de el presente proyecto. Son fruto de la reestimación mediante Baum-Welch de modelos ocultos de markov de densidades continuas completamente operativos para el reconocimiento del habla aislada aislada.

El modelo en sí opera en un espacio de trece dimensiones, proporcionado por el bloque de preprocesado de voz descrito en el capítulo 7 (preénfasis y mel-cepstrum), por lo que en esta ocasión, a diferencia de lo que ocurria en el segundo capítulo con las mezclas de gaussianas a las que se les aplicaba el algoritmo EM, no podremos recurrir a una visualización directa de las densidades asociadas a los estados. De hecho aunque actuásemos en dos dimensiones tampoco encontraríamos una manera sencilla de representar conjuntamente las densidades asociadas a los estados y las densidades que rigen el desarrollo secuencial en el tiempo del modelo. Por lo tanto, para visualizar la mejora durante el entrenamiento lo haremos precisamente mediante el criterio que determina la optimización, la verosimilitud logarítmica.

Aunque hasta ahora no hemos tratado el problema del cuerpo de entrenamiento, éste siempre ha estado presente. Cuando modelábamos situaciones como la del comportamiento de los peces obteníamos modelos en cierto modo cíclicos. Su matriz de transición era casi completa, y podríamos haberlo entrenado con una sola secuencia de observaciones, una tirada larguísima de muestras obtenidas por días y días de funcionamiento contínuo del radar sobre una población piloto. Cuando afrontamos el problema de modelar palabras emerge naturalmente el uso de modelos con transiciones más restringidas, de comportamiento más se-

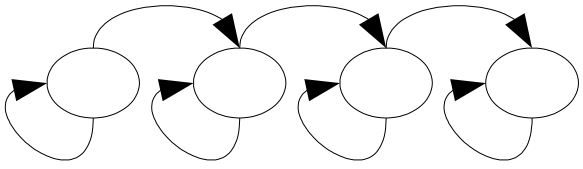


figura 6.5 Modelo secuencial o de izquierda a derecha.

cuencial y finitos en el tiempo. Son los llamados modelos de izquierda a derecha.

Las secuencias de observaciones son ahora limitadas en el tiempo (tanto como lo es la fonación de una palabra), y por lo tanto es necesario utilizar un conjunto suficientemente nutrido de ellas para obtenener una reestimación consistente de los parámetros. De la mano de éste surgen otros problemas ligados a la implementación real como el tamaño insuficiente del conjunto de entrenamiento, etc, eventualidades para las que existen ciertos "trucos" como la utilización conjunta de versiones del modelo con un número menor de estados, la asociación de una misma variable aleatoria a varios estados, o el uso de un suelo mínimo en la reestimación de las covarianzas [Rabiner], que no vamos a detallar, aunque en el programa se aplican algunos de ellos.

Expuesto lo anterior , en esta simulación vamos a utilizar un conjunto de entrenamiento compuesto por 46 ejecuciones de la palabra "nueve". El modelo inicial se ha construido simplemente segmentando en cinco partes una cualquiera de estas alocuciones, y generando a partir de ellas cinco estados. Los primeros cuatro llevan asociadas mezclas de gaussianas y el quinto una gaussiana simple. Dado el tamaño del modelo y la poca utilidad del gesto no voy a precisar en ningún punto los valores concretos de los parámetros del mismo, y será a través de las gráficas por donde contrastemos la correcta aplicación del método.

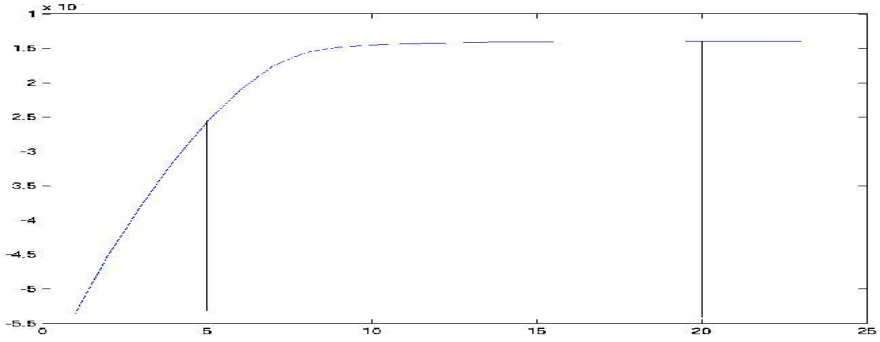


Ilustración 6.6 Verosimilitud logarítmica total del conjunto de entrenamiento respecto al modelo, contra el número de interaciones. En torno a k=20, las mejoras son mínimas.

La gráfica más importante es la de la verosimilitud logarímica. Ésta se muestra en la figura 6.5 para el total de las cuarenta y seis alocuciones de la palabra. Las lineas verticales marcan los puntos donde hemos monitorizado el estado del modelo para hacer un seguimiento preciso de su evolución. En concreto nos hemos centrado en contrastar el modelo respecto a una de las secuencias de observaciones, en concreto la primera (figura 6.6).

La suma de la variable hacia delante para todos los estados en un instante dado refleja la verosimilitud entre modelo y observaciones hasta ese instante. Si usamos su versión escalada, que en cierto modo elimina el efecto acumulativo inherente al cálculo de alfa, podemos ponderar en qué cantidad contribuye cada instante, o dicho de otro

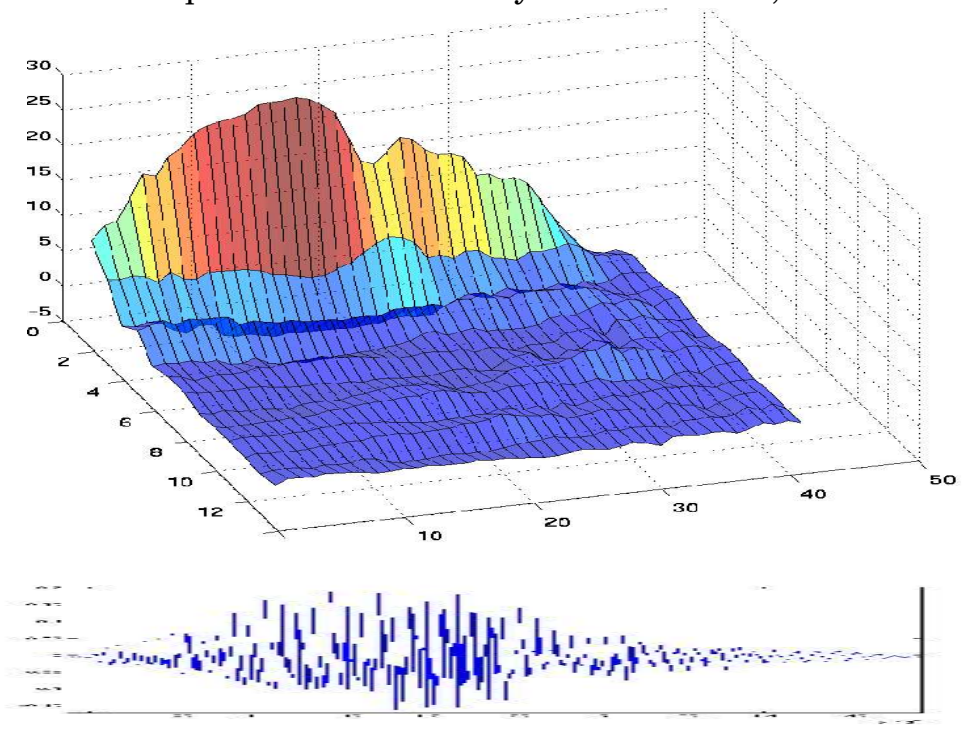


figura 6.7 Primera de las 46 secuencias de observaciones de la palabra "nueve" (11025Hz, mono). Sobre ella, salida del banco de preprocesado constituido por un preénfasis de la señal, la transformada mel-cepstrum (enventanado Hamming de 15ms con solape del 30%, filtrado con 13 bancos entre 20Hz y fs/2)

modo, que intervalos temporales de las observaciones casan mejor con el modelo. Si representamos esta variable hacia delante para todos hitos marcados deberíamos poder observar cómo en promedio cada vez va encajando mejor. Podemos observar estos dos comportamientos en la figura 6.7.

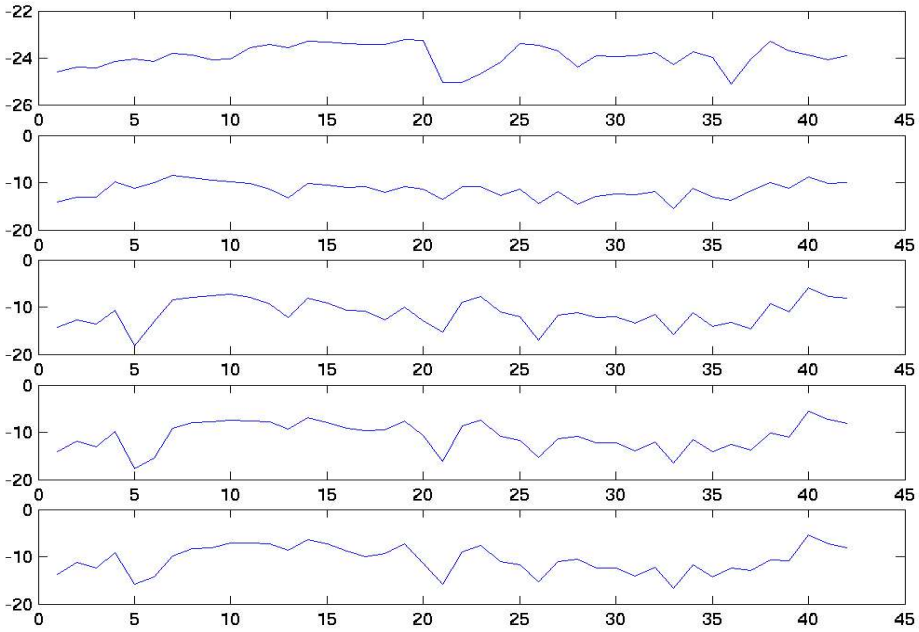


figura 6.8 Logaritmo de la suma de la variable hacia delante escalada para todos los estados en cada instante. De arriba abajo se muestra el valor que toma para la primera alocución del conjunto de entrenamiento y el modelo en la iteración número 0, 5, 10, 15 y 20 respectivamente.

Vemos como según aumenta el número de iteraciones aumenta la verosimilitud en cada instante, y además se van fraguando valles y lomas. Esto es una consecuencia de que las variables asociadas a los estados van convergiendo a un valor estable, óptimo en el sentido de máxima verosimilitud respecto al *total* de las secuencias, por lo que en tanto estas difieren entre sí habrá zonas más beneficiadas que otras, que no demarcan necesariamente un cambio de estado. El crecimiento

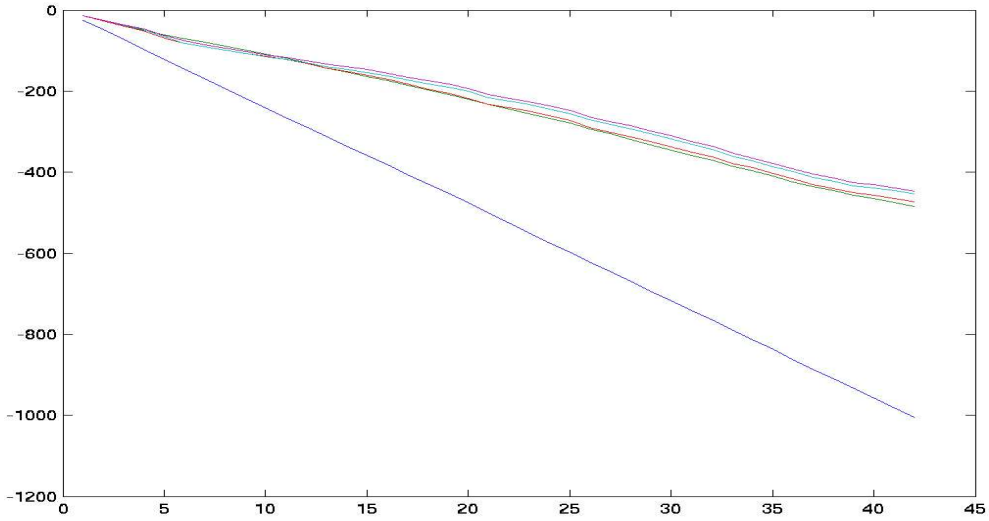


figura 6.9 Suma acumulativa de la verosimilitud logarítmica para una sola ejecución de la palabra y el modelo en distintos puntos de la reestimación.

de la verosimilitud para esta fonación es más visible en la figura 6.8, que refleja la suma acumulada de la variable hacia delante para concluir precisamente al final de la palabra en t=42 en el valor de la verosimilitud logarítmica.

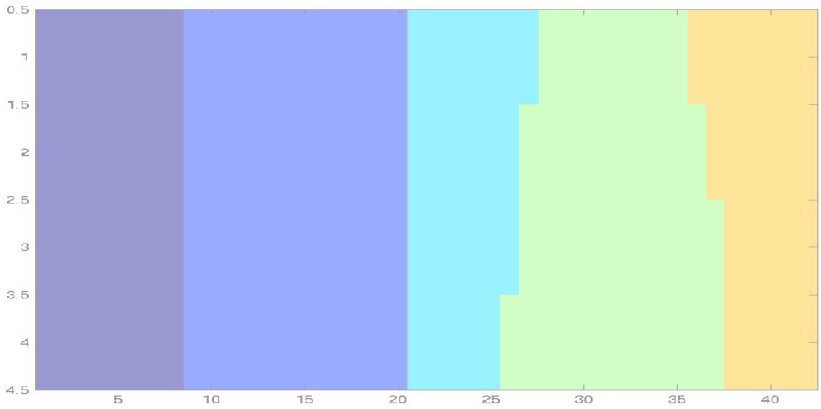


Figura 6.10 Representación del camino de Viterbi en la primera palabra del conjunto de entrenamiento. Cada línea representa uno de los cinco hitos sobre el índice del número de iteraciones, y cada color, un estado.

A título figurativo, un parámetro que nos puede permitir hacernos una idea distinta de como varían las probabilidades de transición y las variables de estado es la secuencia óptima de estados, o el camino de Viterbi. En la figura 6.9 representamos este camino como siempre para la primera alocución de la palabra y para el modelo en distintos puntos de la reestimación. Podemos observar cómo el estado cuatro se va haciendo con más control, en detrimento de sus vecinos (se va viendo favorecido).

Lo que hemos visto para una de las secuencias de observaciones se puede extrapolar a todas. Todas mejoran en mayor o menor grado, de modo que la verosimilitud total aumenta (si consideramos estadísticamente independientes las ejecuciones, ésta no es sino el producto de todas ellas), pero no el modelo no alcanza la misma verosimilitud respecto a todas. Este comportamiento queda patente en la figura 6.10, que muestra cómo varían con el número de iteraciones la media y la de la verosimilitud logarítmica entre las distintas secuencias que componen el conjunto de entrenamiento.

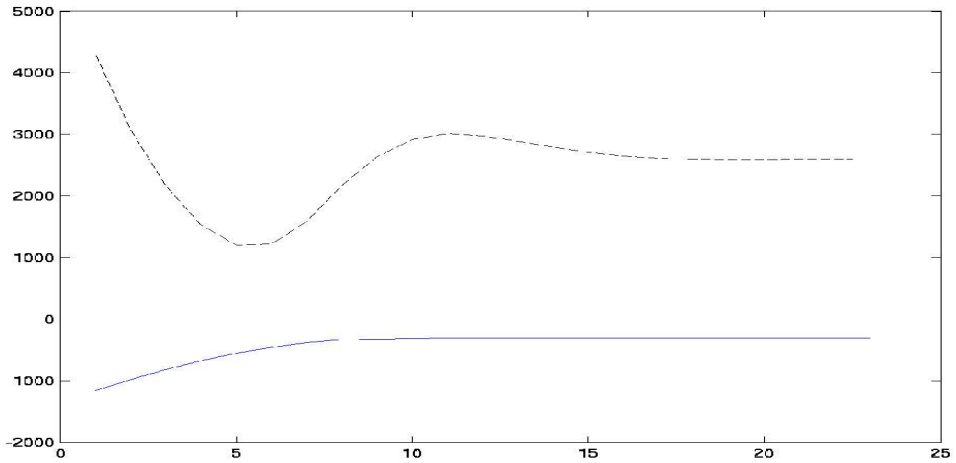


figura 6.11 Varianza (linea discontínua) y media(linea continua) de la verosimilitud logarítmica entre el conjunto de secuencias de entrenamiento.

Para finalizar he intentado hacer algo que refleje mal que bien cómo el modelo se adapta a las observaciones. Para ello en la figura 6.11 se muestra secuencia formada por las medias de las variable asociadas a los estados que conforman el camino óptimo. Obviamente es una representación burda, ya que no refleja en manera alguna las distribuciones de los estados, y la ponderación de las medias es del estilo todo o nada, pero da una idea visual de cómo casan modelo y observaciones. El valor de la transparencia viene dado por la verosimilitud en cada instante, de modo que las regiones más verosímiles son más opacas.

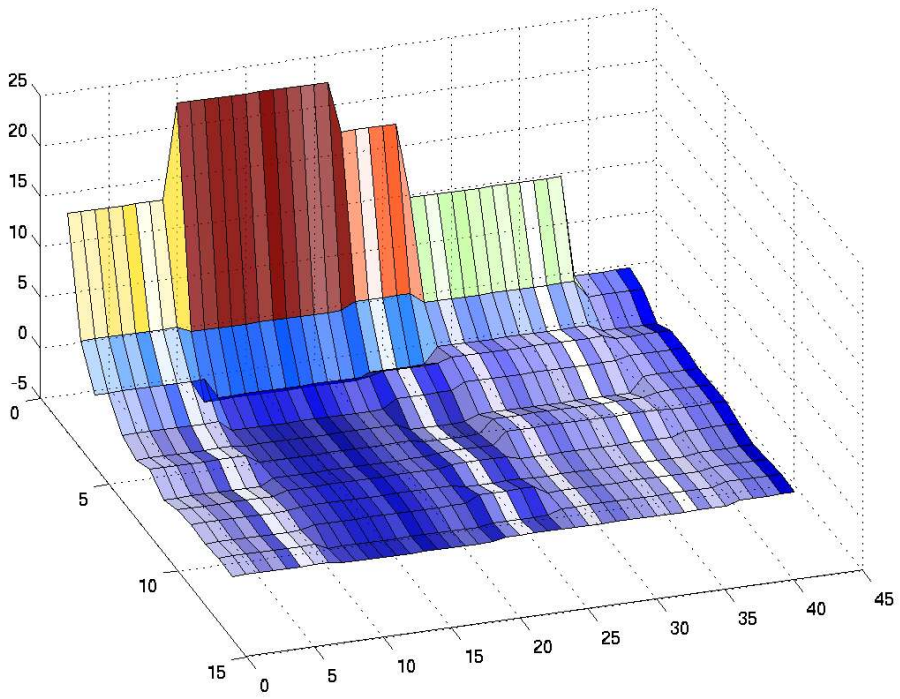


figura 6.12 Representación conjunta del camino óptimo, las medias de los estados y la verosimilitud instantánea para el modelo final respecto a la primera secuencia de observaciones de la palabra nueve. Obsérvese el parecido razonable con la figura 6.6

7. Aplicación al reconocimiento de habla

7.1 Introducción

En este capítulo echaremos un vistazo a los conceptos necesarios para integrar las cadenas ocultas de Markov en el marco de un sistema de reconocimiento de habla. Un sistema de reconocimiento de habla consta de las siguientes partes

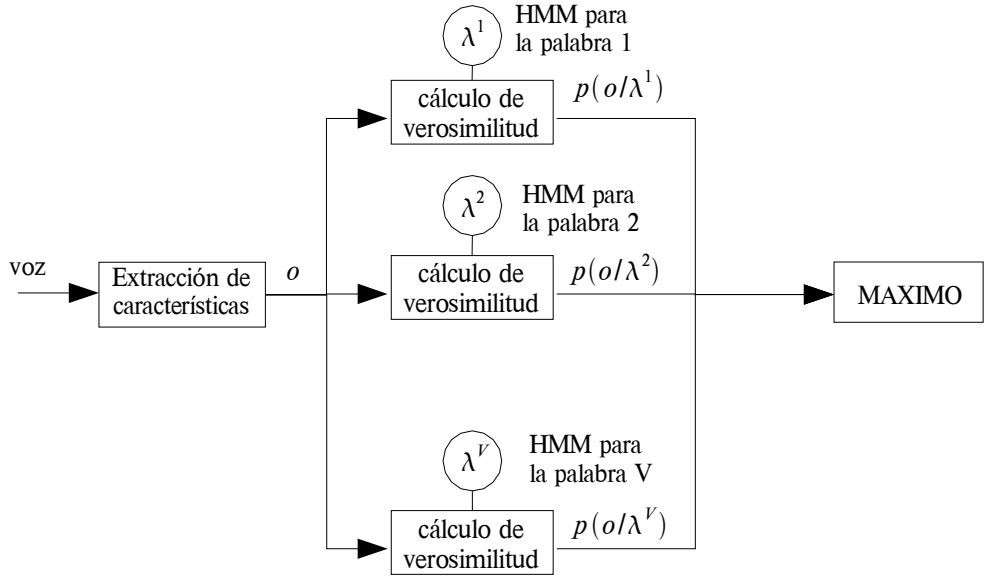


figura 7.1 Diagrama de bloques de las funcionalidades implicadas en el reconocimiento de habla.

La única diferencia que podemos observar respecto a un modelo cualquiera de clasificación es la primera parte de acondicionamiento de señal, denominada *front-end*. Es en esta primera parte donde se realiza la extracción de características de la señal de entrada, entendiendo por extracción de características la transformación de la señal original en

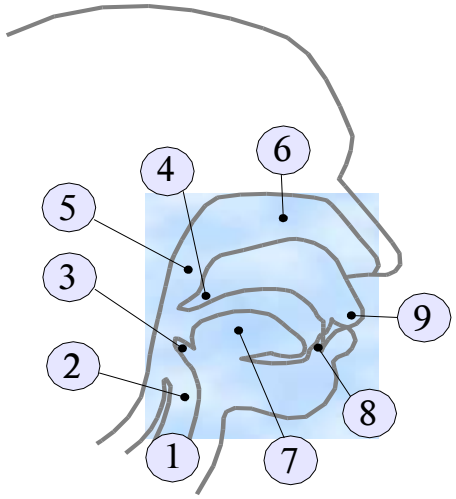
otra N-dimensional más apropiada para la tarea de reconocimiento de habla.

La existencia de un extractor de características adecuado es más que necesaria, dado que la señal de audio no es apropiada para el reconocimiento de patrones. Incluso las pequeñas variaciones de entonación o tiempo que se producen cuando una persona trata de pronunciar de la misma manera una misma palabra provocan en la señal temporal enormes distorsiones. Además, si consideramos por un lado la tasa de bits de una señal de voz (más de 100kbps si la codificamos sin pérdidas), y por otro que la información que transporta puede codificarse como texto usando un canal de apenas 60bps, nos hacemos una idea de la cantidad de redundancia que lleva la señal de voz. La señal de voz puede por tanto variar mucho sin entorpecer la comunicación, es la envoltura de algo, la codificación de algo. Si nos ceñimos a las variaciones temporales de la señal nos estaremos agarrando a algo que es y debe ser voluble y mutable, en tanto que preserve ciertas características que transportan la información.

7.2 Modelo del tracto vocal

El reconocimiento automático de habla desempeña de algún modo el papel de una persona escuchando a una persona que habla. Así se trataron de modelar el acto de habla y el acto de escucha.

La realización de un modelo para la emisión de voz comienza con el estudio del tracto vocal. Tanto los sistemas de compresión de voz como los sistemas de reconocimento de habla aceptan por lo general un modelo estándar de producción de voz, que representa el tracto vocal como una fuente de señal (a ratos periódica, y a ratos aleatoria) mas un filtro que varía lentamente con el tiempo.



1. Cuerdas vocales
2. Laringe
3. Epiglotis
4. Paladar blando
5. Faringe
6. Cavidad nasal
7. Lengua
8. Dientes
9. Labios

Ilustración 13 Este dibujo muestra los órganos más importantes implicados en la fonación. Sombreadol, el tracto vocal, responsable de la generación u modulación de la voz.

La fuente de señal modela dos fenómenos distintos. Cuando la fuente genera una señal periódica, está modelando el tono fundamental generado por el funcionamiento de las cuerdas vocales. Las cuerdas vocales no siempre están vibrando cuando hablamos. De hecho, sólo intervienen en la ejecución de las vocales y de algunos fonemas cuya pronunciación es indisociable de las mismas (en español la m, n o ñ). El otro modo de operación, como fuente aleatoria de señal, aparece cuando intentamos pronunciar consonantes como s, f, etc.

Estas consonantes tienen como base un sonido sibilante o bien procedente de la fricción, que se puede modelar como una fuente de espectro más o menos plano. La disposición de los órganos fonadores como lengua, labios o el paladar blando quedan representados en la respuesta del filtro. Ésta respuesta se puede considerar aproximadamente constante durante la ejecución de un fonema, y desde luego varía más lentamente que cualquiera de las señales generadas por la fuente.

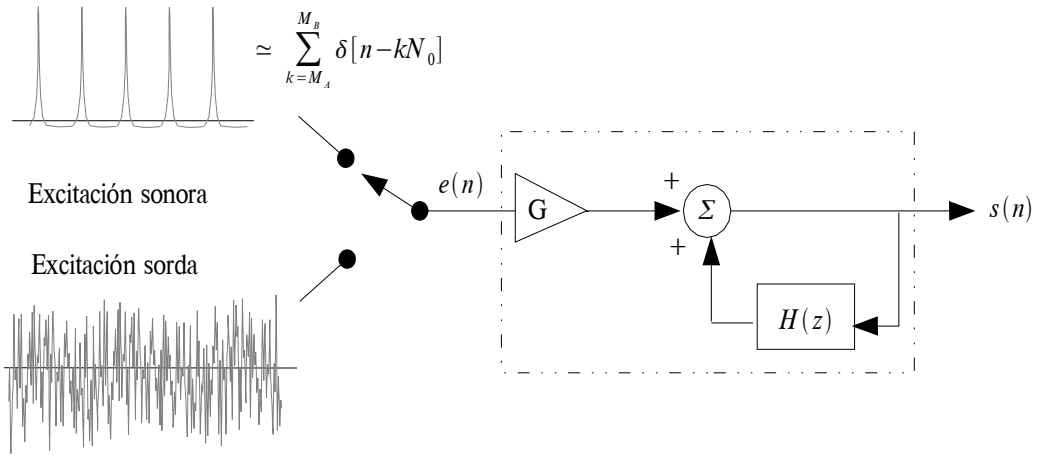


figura 7.14 Diagrama de bloques del modelo de tracto vocal usado en reconocimento de habla y en codificadión de voz.

7.3 Extracción de características

En el campo de los codificadores de voz o *vocoders* de baja tasa, la excitación del tracto vocal es importante y se representa normalmente por tres características: si es sorda o sonora, amplitud y una estimación de la frecuencia fundamental (*tono*) de la fuente, si es sonora. Sin embargo, en reconocimiento de habla se da por supuesto que la forma del espectro contiene información suficiente para el reconocimiento. Los métodos más famosos para obtener características clasificables del ámbito espectral son la codificación lineal predictiva (LPC), los bancos de filtros y los basados en el *cepstrum*. Estos últimos son los que se han venido usando preeminentemente en reconocimiento de habla desde 1980 y es uno de ellos el que se ha utilizado en la aplicación.

7.3.1 Cepstrum

El cepstrum se define como la transformada inversa del logaritmo del espectro de la voz. Formalmente

$$C(\cdot) = Z^{-1}(\ln(Z(\cdot)))$$

La expresión presentada también se conoce como cepstrum com-

plejo (el logaritmo que aparece es complejo), en contraposición con el cepstrum real (que ignora la información de fase y es el que se usa en voz). Como normalmente la voz se representa como una excitación $e(n)$ convolucionada la respuesta impulsiva del tracto vocal $h_T(n)$, podemos observar que la transformada cepstrum "deconvoluciona" ambas señales:

$$\begin{aligned} s(n) &= e(n) * h_T(n) \\ |S(\omega)| &= |E(\omega)||H_T(\omega)| \\ \ln|S(\omega)| &= \ln|E(\omega)| + \ln|H_T(\omega)| \\ c(n) &= \hat{e}(n) + \hat{h_T}(n) \end{aligned}$$

Donde el circunflejo distingue la señal original de la transformada inversa de la amplitud de su espectro. Se puede demostrar que las primeras muestras (primeros milisegundos) de la señal c(n) se corresponden con $\hat{h_T}(n)$, que decae exponencialmente con n, mientras que los valores de c(n) para n alto se corresponden aproximadamente con $\hat{e}(n)$. Esta es la base de la efectividad del análisis cepstrum. Una demostración rápida de estos hechos se obtiene modelando en primer lugar la excitación como un tren de pulsos con cierto decaimiento $\alpha \in (0,1)$

$$e[n] = \sum_{k=0}^{M-1} \alpha^k \delta[n-kN]$$

$$\downarrow Z$$

$$E(z) = 1+\alpha z^{-N}+...+(\alpha z^{-N})^{M-1} = \frac{1-(\alpha z^{-N})^M}{1-\alpha z^{-N}}$$

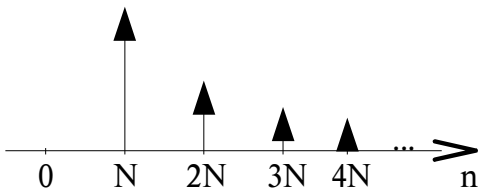
$$\ln E(z) = \ln(1-(\alpha z^{-N})^M) - \ln(1-\alpha z^{-N}) = -\sum_{k=1}^{\infty} \frac{\alpha^{kM}}{k} z^{-kMN} + \sum_{k=1}^{\infty} \frac{\alpha^k}{k} z^{-kN}$$

$$\downarrow Z^{-1}$$

$$\hat{e}[n] = -\sum_{k=1}^{\infty} \frac{\alpha^{kM}}{k} \delta[n-kMN] + -\sum_{k=1}^{\infty} \frac{\alpha^k}{k} \delta[n-kN]$$

Si M (el número de muestras transformadas) es considerable, es el segundo término de $\hat{e}[n]$ el que domina. En el límite tenemos

$$\lim_{\substack{\alpha\to 1\\ M\to\infty}} \hat{e}[n] = \sum_{k=1}^{\infty} \frac{1}{k}\delta[n-kN]$$



De modo que la señal de excitación así modelada en principio sólo estaría presente para valores de $n \geq N$. Por otra parte veamos cómo quedaría la transformada del filtro que modela la forma del tracto vocal.

$h[n]$

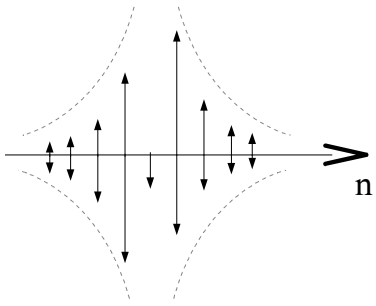
$\downarrow Z$

$$H(z) = |A| \frac{\prod_{k=1}^{M_a}(1 - a_k z^{-1}) \prod_{k=1}^{M_n}(1 - b_k z)}{\prod_{k=1}^{M_c}(1 - c_k z^{-1}) \prod_{k=1}^{M_d}(1 - d_k z)} \qquad s.a \ \ |a_k|,|b_k|,|c_k|,|d_k| \in [0,1)$$

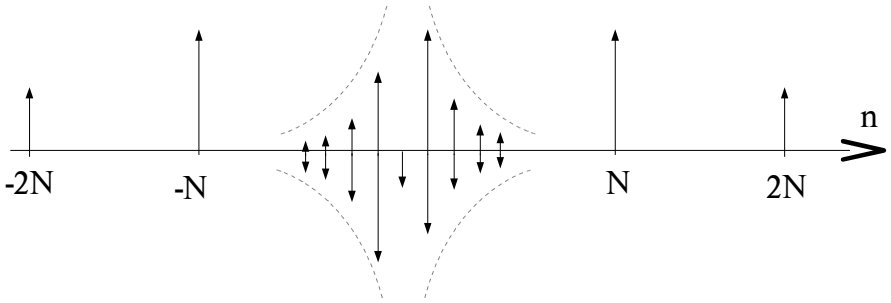
$$\ln H(z) = \ln|A| + \sum_{k=1}^{M_a} \ln(1 - a_k z^{-1}) + \sum_{k=1}^{M_b} \ln(1 - b_k z) - \sum_{k=1}^{M_c} \ln(1 - c_k z^{-1}) - \sum_{k=1}^{M_d} \ln(1 - d_k z)$$

$\downarrow Z^{-1}$

$$\hat{h}[n] = \begin{cases} \sum_{k=1}^{N_b} \frac{b_k^{-n}}{n} - \sum_{k=1}^{N_d} \frac{d_k^{-n}}{n} & n<0 \\ \ln|A| & n=0 \\ \sum_{k=1}^{N_c} \frac{c_k^{-n}}{n} - \sum_{k=1}^{N_a} \frac{a_k^{-n}}{n} & n>0 \end{cases}$$



Será dominante aquel polo o cero más cercano a l círculo unidad, pero en cualquier caso todos decaen como mínimo como 1/n, por lo que la respuesta sólo es significativa en un estrecho margen alrededor del origen, alrededor de unos 3ms. Si muestreamos a 8KHz se traduce en 24 muestras. Por lo tanto la respuesta conjunta será



Dibujo 2 Transformada cepstrum conjunta de la excitación y el filtro.

La $\hat{h}_T(n)$ tiene una relación mucho más clara con la forma del tracto vocal que $S(\omega)$, que está llena de armónicos de la excitación, y sus componentes son por tanto mucho más adecuadas para ser utilizadas como características en la labor de reconocimiento, por lo que enventanamos alrededor del origen y nos quedamos con ella.

En general, en reconocimiento de habla no se considera de gran importancia la fase de las señales, dada la naturaleza del funcionamiento del oído humano y la dificultad de forzar la continuidad de la fase en presencia de ruido. Por ello se utiliza el cepstrum real, que se obtiene como la antitransformada del logaritmo de la amplitud del espectro.

$$C_{\Re}(\cdot) = Z^{-1}(\ln|Z(\cdot)|) = \Re\{C(\cdot)\}$$

Puede demostrarse usando las propiedades de la transformada cepstrum que el resultado de esta operación se corresponde con la parte real del cepstrum complejo.

7.3.2 Coeficientes cepstrum en escala Mel

Al proceso del cálculo del cepstrum se añade una etapa intermedia de filtrado. Esta está formada por una serie de bancos de filtros distribuidos según la escala Mel.

La escala Mel contempla la percepción que hacemos de las frecuencias. Si una frecuencia es el doble de otra en la escala mel, entonces suena “el doble de aguda”. La no linealidad del tono respecto a la frecuencia no es la única peculiaridad de la percepción humana del sonido. Existen multitud de fenómenos que aparecen por la interacción de frecuencia y amplitud, y uno nos será especialmente relevante para justificar el uso de los MFCC (Mel Frecuency Cepstral Coefficients). Se conoce como enmascaramiento el fenómenos que hace que, para un tono concreto, aumente el umbral audible si hay otro tono cercano o una banda de ruido. Esto quiere decir que para percibir un tono cercano a otro, o cercano a una banda de ruido, debemos aumentar su amplitud. este enmascaramiento persiste incluso durante algunos milisegundos después de desaparecida la interferencia.

Fletcher [5] descubrió que el ruido sólo afectaba a la percepción del tono si estaba distribuido sobre cierta banda bastante estrecha, centrada en la frecuencia del tono. La sensibilidad diferencial de tono (mínima diferencia de tono discernible) está muy relacionada con la existencia de estas bandas [6], así como un amplio número de fenómenos perceptuales. Este ancho de banda crítico no es constante en toda la banda audible, sino que aumenta con la frecuencia, de modo que para frecuencias altas la percepción de tonos es más “gruesa” que las para frecuencias bajas, con más capacidad de discriminación.

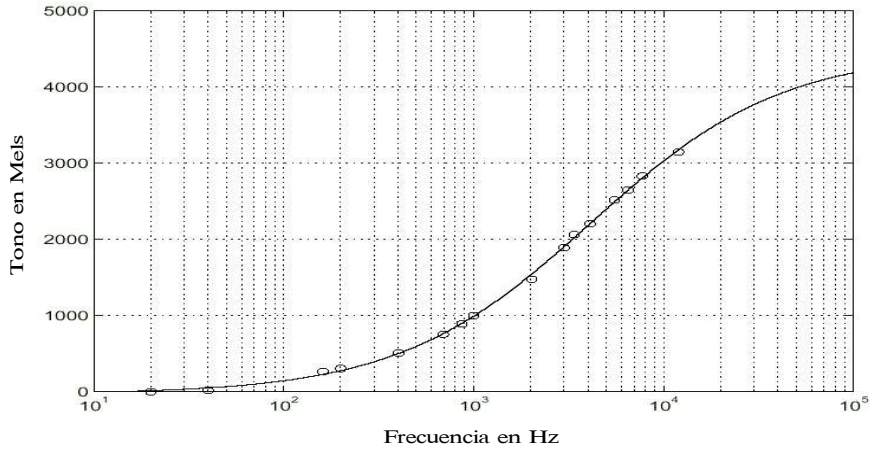
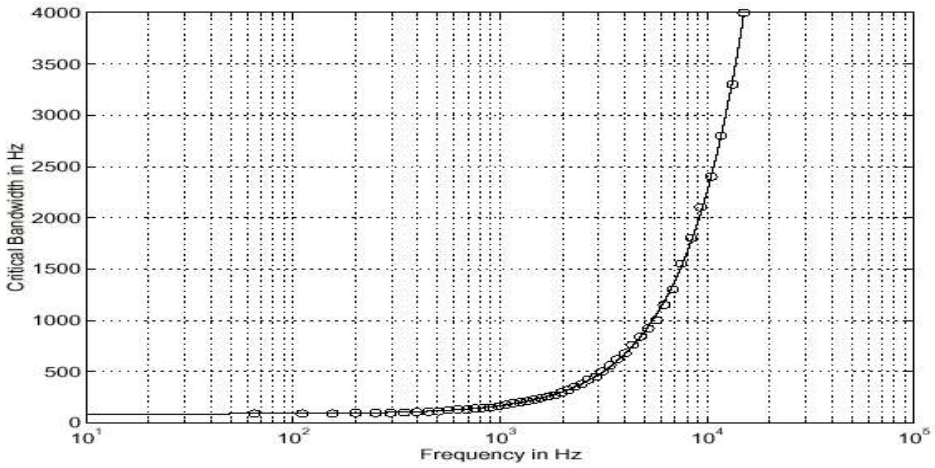


figura 7. 15 Sobre estas lineas, el tono en mels frente a la frecuencia en Hz. Abajo, cómo varía el ancho de banda crítico respecto a la frecuencia.



Basándonos en estos dos fenómenos (variación no lineal de tono y cambio del ancho de banda crítico) podemos diseñar un banco de filtros cuyas salidas se repartan de manera homogénea (perceptualmente) por todo el rango de frecuencias de voz (de 100 a 3000 Hz aprox.), y

además estén homogéneamente distorsionados (perceptualmente). Las fórmulas para obtener el tono en mels y las frecuencias de corte de los filtros [0] son.

$$v(f) = \frac{4491.7}{1 + \exp(7.1702 - 1.9824 \log_{10}(f))} - 30.360$$

$$f_{critLOW} = 1.3056 f^{0.95987} - 64.193$$
$$f_{critHIGH} = 0.70616 f^{1.0497} + 81.288$$

El banco de filtros resultante extenderá sus frecuencias centrales según la escala de Mel, y el ancho de los filtros será proporcional al ancho mínimo discernible. para los filtros del banco suelen usarse un enventanado triangular, como el mostrado en la figura.

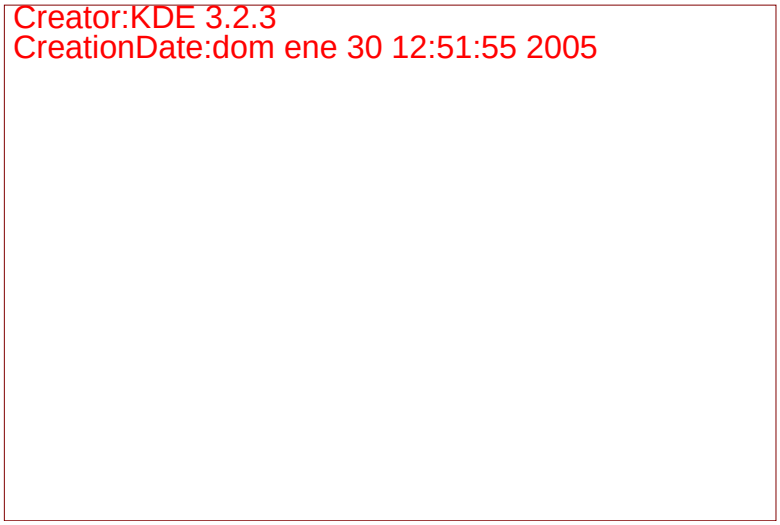


Ilustración 16 Banco de de filtros con escala mel

Los coeficientes Mel-cepstrum se obtienen aplicando este banco

de filtros a la amplitud logarítmica del espectro, justo antes de antitransformar.

7.4 Preénfasis

Una última consideración a tener en cuenta es que el espectro de voz decae unos 6bB/octava en los fonemas sonoros. Para ecualizar el espectro y conseguyir que todas las frecuencias tengan igual preponderancia podemos aplicar un filtro ene le dominio del tiempo tan simple como

$$s(n) = s(n) - \alpha s(n-1)$$

previo a la transformación a frecuencia, y donde $\alpha \simeq 0.9$. Este preénfasis tiene también una contrapartida biológica, ya que el oido externo puede aproximarse por un tubo con una frecuencia de resonancia natural alrededor de los 3KHz, de modo que potencia la energía alrededor de esa franja.

7.5 Uso de la IDCT

En la última transformación que se lleva a cabo para calcular el cepstrum (la transformada inversa del logaritmo de la amplitud espectral) a menudo se utiliza la IDCT (Inverse Discrete Cosine Transform) en lugar de la transformada inversa de fourier. La DCT tiene una serie de carácterísticas que la hacen muy buena para la compresión de señal. Por la propia naturaleza de su proceso de cálculo, elimina los artefactos de frontera, y concentra en los primeros coeficientes la mayoría de la energía de la señal. Estos primeros coeficientes cargarán también con un porcentaje elevado (entorno a un 87%) de la variablilidad de la señal, por lo que se ha dicho de ella que es una aproximación eficiente a la transformada óptima de Karhunen-Loeve (que ortogonaliza las componentes de la señal)[22]. El hecho es que la capacidad de compactación y el bajo coste computacional han hecho que esté presente en los

formatos más extendidos de compresión de imagen (jpeg). Aplicándola en el caso que nos ocupa, podemos reducir el número de características tras el filtrado a los primeros 9 a 15 valores del cepstrum.

7.6 Un algoritmo

A partir de los conceptos tratados, se ha resuelto utilizar en la práctica el siguiente algoritmo. Sea $s[n], 1 \leq n \leq N$ la señal de entrada, aplicamos los siguientes pasos.

i) Preénfasis

$$s[n] = s[n] - \alpha s[n-1] \qquad \alpha = 0.9$$

ii) Descomposición en bloques

La señal se rompe en M bloques de longitud W, que tienen cierto solape de v muestras entre ellos. El resultado se almacena en una matriz $Y_{M \times W}$. W suele corresponderse con las muestras en un intervalo ente 10 y 20 ms. El solape suele variar ente el 20 y el 50% de la longitud del bloque.

$$y_{i,j} = s_{vi+j} \qquad \begin{matrix} j = 0 \ldots W-1 \\ i = 0 \ldots M-1 \end{matrix}$$

iii) Enventanado

Cada bloque se multiplica por una ventana que suaviza las discontinuidades en el dominio temporal para evitar artefactos en frecuencia. Utilizaremos una ventana Hamming.

$$y_{ij} = y_{ij} w_j \qquad \forall i, \ \forall j$$

$$w_j = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi}{W-1}\right) \qquad \forall j$$

iv) Cálculo del espectro de potencia

Se representa como una matriz S_{M times U} donde U viene dado por la longitud más apropiada para la transformada (dependerá

de las limitaciones del algoritmo que usemos para la FFT, si está basado en primos –menos limitante-, o en potencias de dos). Se puede o bien ajustar W desde el principio para que coincida con un valor válido de U, o bien rellenar con ceros hasta el valor válido más cercano.

$$S_i = \left|(FFT(y_i))\right|^2 \qquad \forall i$$

v) Filtrado mel y logaritmo

Consistirá en aplicar K filtros con forma triangular y , en la escala mel, las frecuencias centrales estarán equiespaciadas y todos los filtros tendrán el mismo ancho de banda. suele tomarse como frecuencia mínima 20Hz y como máxima la mitad de la frecuencia de muestreo, o un valor en torno a 4KHz. El número de filtros K no debe ser muy grande (sería como no aplicarlos) ni muy pequeño (nos daría una representación demasiado pobre). Suelen usarse valores entre 13 y 24.

$$\left.\begin{array}{rcl} \gamma & = & \dfrac{W-1}{f_{max}-f_{min}} \\ I_l & = & \gamma(f_l-f_{ini}) \\ I_c & = & \gamma(f_c-f_{ini}) \\ I_h & = & \gamma(f_h-f_{ini}) \end{array}\right\} \rightarrow \quad f_{ij} = \begin{cases} \dfrac{1}{f_c-f_l}\left(\dfrac{j}{\gamma}+f_{min}-f_l\right) & \lfloor I_l \rfloor \le j \le \lfloor I_c \rfloor \\ \dfrac{1}{f_c-f_h}\left(\dfrac{j}{\gamma}+f_{min}-f_c\right) & \lfloor I_c \rfloor \le j \le \lfloor I_h \rfloor \\ 0 & e.o.c \end{cases}$$

Como cada filtro ocupa más rango de frecuencia, y porta por tanto más energía, se suele normalizar dividiendo por el área bajo el filtro. Si además aplicamos el logaritmo, esto resulta en una matriz $\boldsymbol{P}_{M\times K}$ donde

$$p_{ij} = \log_{10}\left(\frac{1}{A_j}\sum_{k=0}^{U-1} S_{ik} f_{jk}\right), \qquad \begin{array}{l} 0 \le j \le K-1 \\ 0 \le i \le M-1 \end{array}$$

$$A_j = \sum_{k=0}^{U-1} f_{jk}$$

vi) IDCT

Resulta en una matriz $Q_{M\times L}$ tal que

$$q_{ij} = \frac{1}{K}\sum_{k=0}^{K-1} p_{ik}\cos((k-0.5)\frac{\pi j}{L}) \qquad \begin{matrix} 0\le j\le L-1 \\ 0\le i\le M-1 \end{matrix}$$

Normalmente se escoge L menor que K, con unos valores entre 9 y 15.

8. Conclusiones y lineas futuras

8.1 Conclusiones

Hemos visto cómo se trabaja con modelos ocultos de Markov, y cómo podemos utilizarlos en el reconocimiento de habla. Se ha dejado patente su extrema flexibilidad para modelar sucesos que tienen una duración variable y características imprecisas, y cómo a pesar de ello podemos esperar obtener reestimaciones localmente óptimas respecto a un conjunto de observaciones aplicando el algoritmo EM. Y con todo esto no vamos a acentuar ninguna de las bondades del modelo más de lo que lo hace el hecho de que se lleve utilizando veinte años en reconocimiento de habla. La única conclusión posible es que es posible implementarlo con un bajo coste computacional, y obtener resultados en el mundo real sin más que asimilar la teoría.

8.2 Líneas futuras

De continuarse la línea del proyecto, o utilizarse el código desarrollado para otros fines, existen unos cuantos tópicos que pueden resultar de interés en el terreno del reconocimiento de habla

i) La evaluación de las densidades de probabilidad es el paso más costoso y que con más frecuencia se repite cuando evaluamos la verosimilitud de una secuencia de observaciones respecto a un modelo. Este factor es un cuello de botella, y es si es importante en reconocimiento de habla aislada, es crítico en reconocimiento contínuo [12]. Para evitar evaluar las densidades que resultarán menos relevantes, se plantea la creación de un árbol binario que en cada nodo dividirá en dos hiperplanos el espacio de características [13]. Se busca que las hojas de este árbol aproximen lo mejor posible lo que se conoce como una región de Vo-

ronoi (aquella region del espacio para cuyos puntos el vecino más próximo es un vector patrón dado), muy costosa de calcular de forma precisa.

ii) La inclusión de una implementación del criterio MAP[14] para el entrenamiento de los HMM permitiría un entrenamiento más rápido de los modelos para adaptarlos a las peculiaridades un usuario en concreto, con un número menor de palabras en el conjunto de entrenamiento.

Apendice A. Aplicación práctica

A.1 Objetivos

En el momento de empezar el proyecto tenía en mente hacer una implementación lo más portable posible, que funcionase y que lo hiciese rápido. Por eso me decanté por construir un conjunto de clases en c++ que trabajasen en base a librerías matemáticas lo más normalizadas posibles, y que estuviesen extendidas. Una vez finalizado el núcleo estadístico me fui dando cuenta de que al parecer la implementación que se requería para el proyecto podría haberse limitado a una serie de scripts en matlab así que, algo desengañado, acabé de implementar el frontend para extracción de características en este entorno.

A.2 Aplicación desarrollada

A.2.1 Nucleo estadístico

El núcleo estadístico como ya he dicho está construido en c++, y se basa en las librerías del proyecto it++, que se distribuyen bajo licencia GPL, en concreto sobre la versión 3.7.3. Este conjunto de librerías de basan su núcleo de algebra matricial en el proyecto ATLAS (Automatic Tuning Lineal Algebra Subroutines), que ofrece para cualquier plataforma implementaciones optimizadas del conjunto de subrutinas BLAS y LAPACK.

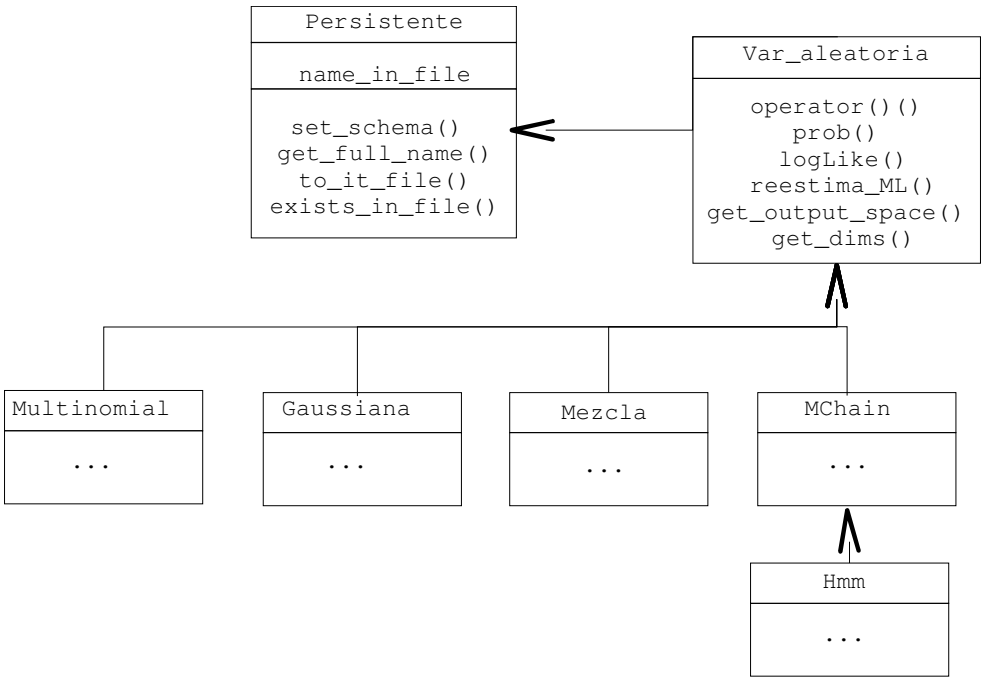


figura A.1 Detalle del modelo de clases.

El modelo de clases se ha desarrollado con la vista puesta en la flexibilidad de los modelos. Definiendo clases que heredeando de *var_aleatoria* podemos ampliar sin mucho esfuerzo el abanico de familias de densidades disponibles. Por ahora sólo se han implementado las correspondientes a gaussianas, multinomiales y mezclas genéricas. Es de reseñar que la clase que representa un modelo de markov desciende también de *var_aleatoria*, de modo que la variable asociada a un estado podría ser otro HMM. Esta capacidad de anidamiento puede ser muy util para ciertas aplicaciones.

De cara a su utilización sin necesidad de implementar código se han preparado dos programas, compara.exe y entrena.exe. El primero compara una secuencia o secuencias de observaciones contra uno o varios modelos, devolviéndonos la verosimilitud para cada uno. El segundo entrena un modelo inicial a partir de una o varias secuencias de observaciones.

A.2.2 Comunicación con Matlab

La comunicación con matlab se hace a través de archivos con extensión .it. Usando it++ tenemos acceso a funciones que permiten escribir y leer matrices y vectores. Para cargar el archivo en el espacio de trabajo de matlab disponemos de los programas *load_it.m* y *save_it.m*.

A.2.3 FrontEnd

El frontend desarrollado en matlab implementa el algoritmo expuesto al final del capítulo siete, incluyendo además un detector de actividad vocal basado en los coeficientes cepstrum. Éste segmenta finalmente las palabras y las almacena en un formato de archivo que es fácilmente accesible desde c++ usando las librerías que ya hemos mencionado. Además, genera un modelo inicial para el entrenamiento a partir de las observaciones.

Apéndice B

B.1 Notación

Consideraciones generales: si algún símbolo aparece en negrita y minúscula, se trarta de un vector o de una secuencia de vectores. Si el tipo de letra es negrita y mayúscula hablamos de matrices.

símbolo	descripción
$\boldsymbol{A}$	Matriz estocástica.
a_{ij}	Probabilidad de transición del estado i al estado j.
Π	Traspuesta de la matriz estocástica.
π	Probabilidad inicial de ocupación de estados.
o	Secuencia de observaciones $o=\{o_1 o_2 ... o_T\}$
o_t	Observación en el instante t.
$b_j(o_t)$	Probabilidad de que la variable asociada al estado j haya generado la observacion o_t
B	Simboliza el conjunto de parámetros que rigen las variables asociadas a los estados de un modelo oculto de markov
θ	Conjunto total de parámetros. Para un HMM, $\theta=(A,\pi,B)$
q	Secuencia de estados $q=\{q_1 q_2 ... q_T\}$
q_t	estado del modelo en el instante t. Suele aparecer como $q_t=i \quad \forall i\in[1,N]$
N	Número de estados del modelo
T	Número de observaciones.

símbolo	descripción
λ	Modelo oculto de Markov.
$\lambda(\theta)$	Modelo oculto de markov, mostrando explicitamente la dependencia respecto a los parámetros.
$\lambda(A,\pi,B)$	Idem.
$\alpha_t(i)$	Variable hacia delante. Se define como $p(o_1 o_2 \cdots o_t q_t = i/\lambda)$
$\beta_t(i)$	Variable hacia atrás. Se define como $p(o_{t+1} \cdots o_T, q_t = i/\lambda)$
$\hat{\alpha}$	Variable escalada
α^*	Valor óptimo.
$\xi_t(i,j)$	Probabilidad de pasar del $q_t = i$ a $q_{t+1} = j$ dada una secuencia de observaciones y un modelo.
$\gamma_t(i)$	Probabilidad de que $q_t = i$ dada una secuencia de observaciones y un modelo.
$Q(\bar{\theta},\theta)$	Función auxiliar de Baum

B.2 Acrónimos

MLE	*Maximum Likelihood Estimation*, estimación de máxima verosimilitud.
MAP	*Maximum A Posteriori.*
MMIE	*Minimum Mutual Information Estimation*, estimación de mínima información mutua.
CMLE	*Conditional Maximum Likelihood Estimation, estimación de máxima verosimilitud condicional.*
EM	*Expectation Maximization*, Promediado Maximización.
HMM	*Hidden Markov Model*, modelos ocultos de Markov.
MFCC	*Mel Frecuency Cepstal Coefficients*, coeficientes del cepstrum en escala Mel
DCT	*Discrete Cosine Transform.*

Bibliografía

(1) Statistical Pattern Classification

(2) *Pattern recognition*

(3) Lawrence R. Rabiner *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*.

(4) Jeff Bilmes *What HMM's Can Do*. bilmes@ee.washington.edu

(5) Sergio A. Cruces *Apuntes de Cátedra de Tratamiento Digital de Voz* Servicio de Publicaciones de la Escuela Superior de Ingenieros de Sevilla, 2004

(6) Douglas O'Shaughnessy *Interacting with computers by voice: automatic speech recognition and synthesis* . Proceedings of the ieee, vol. 91, no 9, september 2003

(7) H.P. Combrinck and E.C. Botha *On The Mel-scaled Cepstrum* Department of Electrical and Electronic Engineering University of Pretoria Pretoria South Africa

(8) J. F. Blinn *What's the deal with the DCT?* IEEE Computer Graphics and Applications, vol. 13, pp. 78 83, July 1993.

(9) Li Tan and Montri Karnjanadecha *Modified Mel-frequency cepstrum coefficient* Department of Computer Engineering Faculty of Engineering Prince of Songkhla University

(10) J.A Haigh & J.S. Mason *A voice activity detector based on cepstral analysis.*

(11) J.A Haigh & J.S. Mason *Robust voice activity detection using cepstral features.*

(12) J Fritsch, I. Rogina *Speeding Up the score computation of Hmm speech recognizers with the Bucket Voronoi intersection algorithm*.

(13) J Fritsch, I. Rogina *The bucket box intersection algorith for fast aproximative evaluation of siagonal mixture gaussians*.

(14) Jean-Luc Gauvain and Chin-Hui Lee *Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains*.