
Trabajo 5

Introducción a los modelos ocultos de Markov

Introducción

Los modelos ocultos de Markov (HMM) son una clase de modelos estadísticos útiles para el análisis de una serie de observaciones en tiempo discreto, tales como un flujo de muestras acústicas extraídas de una señal de voz. Los HMM son muy apropiados para el reconocimiento de voz debido a su capacidad inherente de representar eventos acústicos de duración variable y a la existencia de algoritmos eficaces para computar automáticamente los parámetros del modelo a partir de los datos de entrenamiento.

Software

En esta práctica utilizará el programa `hmm_tool`. Este programa funciona sobre los HMM de observación discreta. Posee la capacidad de generar secuencias de observación aleatorias desde un HMM, aplicar el algoritmo de búsqueda de Viterbi sobre una secuencia de observación para un HMM en concreto, y calcular los parámetros de un HMM utilizando el algoritmo de Baum-Welch. También utilizará el programa `hmm_tool` sobre datos artificiales para que se familiarice con las propiedades de los HMM y sus algoritmos asociados. El programa `hmm_tool` opera directamente desde la línea de comandos de UNIX. Los HMM y las secuencias de observación se almacenan en archivos ASCII y se pasan a `hmm_tool` mediante las opciones de la línea de comando. Para enumerar el grupo total de opciones de la línea de comando disponibles con `hmm_tool`, escriba:

```
% hmm_tool -help
```

Cómo empezar

Para empezar esta práctica, escriba el siguiente comando en la ventana de símbolo del sistema de UNIX :

```
% start_lab5.cmd
```

Esto realizará una copia local (en su directorio local) del grupo de archivos necesarios para esta práctica. Estos contienen los modelos ocultos de Markov más las secuencias de observación utilizadas durante la práctica. También se mostrará el uso de `hmm_tool`.

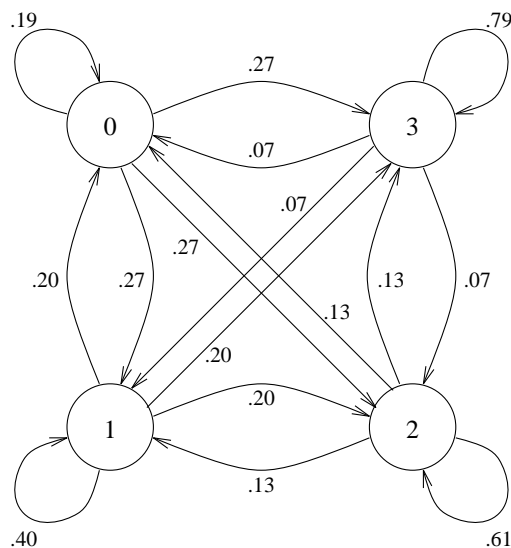
Parte I: Modelos HMM simétricos

Comenzaremos con una clase de modelos HMM llamados “simétricos”. La estructura de los HMM simétricos es suficientemente simple para permitir la estimación directa de los parámetros del modelo desde los datos observados, en lugar de la estimación iterativa mediante el algoritmo de reestimación de Baum-Welch. Un HMM simétrico de estado N posee las siguientes propiedades:

1. El tamaño del alfabeto de salida es igual al número de estados N .
2. Cada estado puede imprimir sólo un símbolo y dos estados no pueden imprimir el mismo símbolo. Más concretamente, la matriz B es un N por la matriz de identidad N . Esto básicamente significa que el modelo ya no está “oculto” .
3. Todas las transiciones que salen de un nodo poseen la misma probabilidad (excepto las autotransiciones); ej., para cualquier estado i $a_{ij} = a_{ik} \forall j, k \neq i$ where $0 \leq i, j, k < N$.
4. Un HMM simétrico de estado N está totalmente determinado por sus probabilidades de transición N autobucles

$$a_{00}, a_{11}, \dots, a_{(N-1)(N-1)}.$$

Aquí tenemos un ejemplo de un HMM simétrico de 4 estados:



T1: Muestre los parámetros de los dos modelos HMM simétricos que utilizaremos y examine sus propiedades con los comandos:

```
% hmm_tool -hmm_in symmetric1.hmm -print_models
% hmm_tool -hmm_in symmetric2.hmm -print_models
```

Use `symmetric1.hmm` (que es de hecho igual al modelo mostrado arriba) para generar una secuencia de observación de 100 símbolos con el siguiente comando:

```
% hmm_tool -hmm_in symmetric1.hmm -generate 1 100 \
-print_obs -obs_out seq1.obs
```

Observe que la secuencia de observación se escribe en el archivo `seq1.obs`. Compute y grabe el número de veces que los 16 bigramas de símbolos de salida han aparecido en la secuencia de observación, utilizando el comando:

```
% hmm_tool -hmm_in symmetric1.hmm -obs_in seq1.obs \
           -count_bigrams
```

Genere otra secuencia de observación de 10.000 símbolos y compute y grabe los cálculos del bigrama de nuevo (no utilice ahora la opción `-print_obs` para evitar imprimir 10.000 observaciones por pantalla)

P1: Derive una fórmula que estime las probabilidades de transición autobucle directamente desde los cálculos del bigrama que usted calculó. Emplee esta fórmula junto con los recuentos observados para estimar las probabilidades de transición autobucle desde ambas secuencias de observación. ¿Producen los cálculos de la secuencia de observación más larga estimaciones más exactas de lo esperado?

T2: El segundo modelo HMM, `symmetric2.hmm`, difiere del primero sólo en la probabilidad de transición₃₃ que es igual a 1.0. Genere una secuencia de observación de 5.000 símbolos empleando `symmetric2.hmm` y calcule los cálculos del bigrama.

P2: ¿Qué sucede cuando trata de estimar las probabilidades de transición autobucle para este HMM y por qué? ¿Ayudaría la observación de secuencias más largas? ¿Qué puede hacer usted para estimar las transiciones autobucle para este modelo?

Parte II: Entrenamiento mediante el algoritmo de Baum-Welch

En esta parte de la práctica examinaremos algunas cuestiones relacionadas con el entrenamiento de modelos HMM, utilizando el algoritmo de avance-retroceso de Baum Welch. Nos ocuparemos de un modelo de Markov de 5 estados de izquierda a derecha, definido por las siguientes matrices de transición y de probabilidad de salida.

$$A = \begin{bmatrix} .8 & .1 & .1 & 0 & 0 \\ 0 & .9 & .1 & 0 & 0 \\ 0 & 0 & .8 & .1 & .1 \\ 0 & 0 & 0 & .9 & .1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} .7 & .3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .8 & .2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .2 & .8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & .7 \end{bmatrix}$$

Observe que A es triangular superior, que hace que el modelo vaya de izquierda a derecha (las transiciones de retroceso están prohibidas). Generaremos algunas observaciones utilizando este modelo, y luego utilizaremos el algoritmo de Baum Welch para entrenar un nuevo modelo HMM y examinar hasta qué punto se ajustan las estimaciones a aquellas del modelo real. Exploraremos el efecto de las distintas cantidades de datos de entrenamiento y el resultado de las distintas matrices iniciales A y B.

T3: Examine el modelo `5-state-lr.hmm` para ver qué corresponde a las matrices A y B anteriores con el comando:

```
% hmm_tool -hmm_in 5-state-lr.hmm -print_models
```

Genere 10 secuencias de observación, cada una de 50 símbolos de longitud, utilizando este modelo:

```
% hmm_tool -hmm_in 5-state-lr.hmm -generate 10 50 \  
-print_obs -obs_out seq2.obs
```

Examine el HMM `5-state-init1.hmm`, que posee los mismos elementos cero en sus matrices A y B (ej., las mismas transiciones y salidas prohibidas) que el modelo real, sino que los elementos no cero están inicializados de modo uniforme. Utilizaremos `5-state-init1.hmm` como modelo inicial en el algoritmo de reestimación de Baum-Welch para entrenar un HMM desde los datos observados.

Para llevar a cabo el entrenamiento de Baum-Welch utilizando la secuencia de observación que usted acaba de generar, escriba el comando:

```
% hmm_tool -hmm_in 5-state-init1.hmm -obs_in seq2.obs \  
-bw 20 -print_models
```

Observe la puntuación de probabilidad logarítmica de los datos de entrenamiento después de cada iteración y note con qué rapidez converge el algoritmo de Baum-Welch.

Después, genere 100 secuencias de observación de longitud 50 y utilice de nuevo `5-state-init1.hmm` como modelo inicial para reestimar los parámetros de HMM desde el nuevo grupo de 100 secuencias de observación. Compare los parámetros de los modelos estimados desde los 10 datos de secuencia y los 100 datos de secuencia, con los parámetros del modelo real que generó las secuencias.

P3: ¿Cuál es la proporción entre el número de puntos de datos en el grupo de datos de entrenamiento y el número de parámetros libres que fueron estimados en las dos pasadas de entrenamiento que realizó antes? ¿Existe una diferencia notable en la calidad de las estimaciones de las matrices A y B en las dos pasadas de entrenamiento?

T4: Estimaremos ahora los parámetros HMM utilizando modelos HMM iniciales más generales. Examine los tres modelos HMM iniciales:

- `5-state-init2.hmm`
- `5-state-init3.hmm`
- `5-state-init4.hmm`

Observe que `5-state-init2.hmm` es un modelo de izquierda a derecha mientras que `5-state-init3.hmm` y `5-state-init4.hmm` no lo son. Utilice el algoritmo de Baum-Welch para estimar los parámetros HMM utilizando los datos de la secuencia de observación 100 sobre los datos generados en **T3**, comenzando con el modelo inicial `5-state-init2.hmm`. Grabe los valores de probabilidad logarítmica de los datos de entrenamiento después de cada iteración. Cuando haya finalizado el entrenamiento, compare los valores del parámetro de los HMM estimados, con aquellos del modelo real.

Repita esta tarea para `5-state-init3.hmm` y `5-state-init4.hmm`.

- P4:** En el mismo grupo de ejes, trace la probabilidad logarítmica como una función del número de iteraciones observadas al entrenar `5-state-init2.hmm`, `5-state-init3.hmm` y `5-state-init4.hmm`. ¿Qué puede explicar sobre la relativa dificultad de entrenar modelos más generales frente a aquellos más restringidos? ¿Qué conclusión obtiene acerca de la importancia de la inicialización de las matrices A y B antes de comenzar el entrenamiento?
- P5:** ¿Qué propiedad del algoritmo de Baum-Welch podría explicar las principales discrepancias que usted puede haber observado entre los parámetros estimados y los verdaderos?

Parte III: Representación de Trellis y búsqueda de Viterbi

Tanto el algoritmo de avance-retroceso utilizado en la reestimación de Baum-Welch como el algoritmo de decodificación de Viterbi, se basan en la representación del trellis (o grafo) de los modelos ocultos de Markov. La primera sección de esta práctica se centra en el siguiente HMM de 2 estados:

$$A = \begin{bmatrix} .4 & .6 \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} .8 & .1 & .1 \\ .2 & .2 & .6 \end{bmatrix}$$

- T5:** Genere y apunte una secuencia de 5 observaciones del modelo mostrado arriba con el comando:

```
% hmm_tool -hmm_in 2-state-lr.hmm -generate 1 5 \
-print_models -print_obs
```

- P6:** Utilice la hoja de la respuesta adjunta para dibujar un trellis para la secuencia que usted observó. Compute todas las probabilidades de las ramas y determine la trayectoria más probable a través del trellis.
- P7:** Estime aproximadamente la longitud de una secuencia de observación que provocaría agotamiento en un ordenador típico si no se utilizan los logaritmos; ej., ¿cuánto tiempo tendrá una secuencia de observación una probabilidad del orden de 10^{-100} ?

Parte IV: Reconocimiento con modelos HMM

En un sistema típico de reconocimiento de palabras basado en modelos HMM, la acústica de cada palabra del vocabulario está representada por un modelo oculto de Markov individual. El reconocimiento se puede realizar mediante el algoritmo de Viterbi, para computar la secuencia de estado oculta más probable de cada uno de los HMM candidatos, para luego seleccionar la palabra correspondiente al HMM con la secuencia de estado más probable.

T6: Cada uno de estos archivos

- testA.obs
- testB.obs
- testC.obs

contiene una secuencia de 50 observaciones generadas a partir de uno de los tres modelos HMM:

- recognition1.hmm
- recognition2.hmm
- recognition3.hmm

Los HMM son modelos de 5 estados de izquierda a derecha que difieren sólo en sus matrices B. Utilice el algoritmo de búsqueda de Viterbi para computar la puntuación de probabilidad logarítmica de la secuencia de estado oculta más probable a través de cada uno de los modelos para cada secuencia de observación (asegúrese de que computa las 9 probabilidades). Por ejemplo, para computar la puntuación entre testA.obs y recognition1.hmm, podemos decir:

```
% hmm_tool -hmm_in recognition1.hmm \  
            -obs_in testA.obs -viterbi
```

P8: Cree una tabla 3×3 mostrando las puntuaciones de Viterbi computadas para cada uno de los modelos candidatos y cada una de las secuencias de observación. ¿Qué modelo presentaba más probabilidad de generar cada secuencia de pruebas basadas en estas puntuaciones? ¿Qué otro método de puntuación puede utilizarse para realizar esta labor de reconocimiento?

Parte V: Modelos HMM de densidad continua

En clase, solucionamos los tres principales problemas de los HMM para los casos en que las observaciones fueron caracterizadas como símbolos discretos seleccionados de un alfabeto finito, y por tanto, podríamos utilizar una densidad de probabilidad discreta dentro de cada estado de un modelo HMM. Una clase más común de HMM son los HMM de densidad continua donde las observaciones son continuas. La representación más general de la función de densidad de probabilidad (pdf) para la que se ha formulado un procedimiento de reestimación, es la mezcla de Gaussianas de la forma:

$$b_j(o_t) = \sum_{k=1}^M c_{jk} N(o_t, \mu_{jk}, U_{jk})$$

donde \mathbf{w}_t es el vector de observación modelado en tiempo t , c_{jk} es el coeficiente de mezcla para la mezcla k th en el estado j y N es la densidad gaussiana con vector de medias μ_{jk} y matriz de covarianza U_{jk} para el componente mezcla k th en el estado j .

En este problema trataremos un sencillo enfoque para la derivación de una fórmula de reestimación de medias para el caso simplificado de una función pdf simple de observación gaussiana, es decir, cuando el número de mezclas $M = 1$:

$$b_j(o_t) = N(o_t, \mu_j, U_j)$$

P9: Para hallar la fórmula de reestimación para μ_j , cuantificamos la variable aleatoria continua utilizando un conjunto de binarias derivadas de la distribución continua. Sean los valores s o_t cuantificados tal que $o_t \approx o_k$ para algún valor k . Dése cuenta que o_k es ahora equivalente a un símbolo de observación del caso discreto. Expresé μ_j como una función de o_k y $b_j(k)$ donde $\bar{b}_j(k)$ es la fórmula de reestimación del símbolo k th en el estado j th para el caso discreto y está determinado por:

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j) \text{ s.t. } o_t = o_k}$$

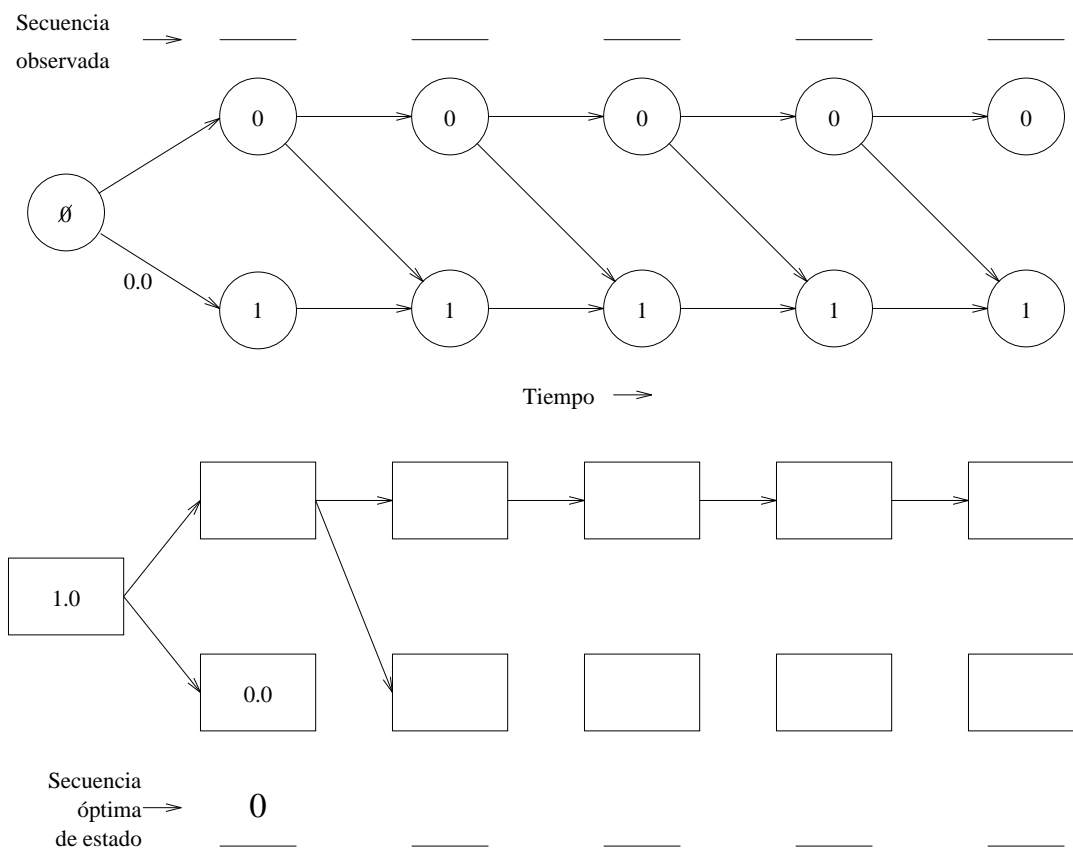
Recuerde que $\gamma_t(j)$ es la probabilidad de estar en el estado j en tiempo t .

Pista: Comience con la definición de la media para una variable aleatoria continua y aproxime la integración con una suma sobre los valores cuantificados.

P10: Muestre que la fórmula de reestimación para μ_j , la media de la gaussiana para el estado j , está determinado por:

$$\mu_j = \frac{\sum_{t=1}^T \gamma_t(j) \cdot o_t}{\sum_{t=1}^T \gamma_t(j)}$$

Hoja de respuestas para el trabajo 5 P6



Indicaciones:

1. Introduzca los 5 símbolos que usted ha observado (de la T5) en la línea superior.
2. Etiquete cada arco del diagrama superior con la probabilidad de unión de que se adopte esa transición y la correspondiente salida generada en el próximo estado.
3. En el segundo diagrama, introduzca en cada recuadro la probabilidad de la trayectoria parcial más probable a través de ese estado, y trace flechas que conecten los estados en las mejores trayectorias parciales.
4. Lea la secuencia de estado oculta más probable del segundo diagrama y escriba el resultado en el espacio facilitado.