Trabajo final: localización de robots usando modelos ocultos de Markov

Este trabajo consta de tres partes bien diferenciadas. En la primera parte se pide la implementación en Python de tres algoritmos para, respectivamente, muestreo, filtrado y decodificación en modelos ocultos de Markov. En la segunda, se aplicarán los algoritmos a un problema simple de localización de robots. Y en la tercera parte, se trata de hacer determinados experimentos para estudiar cómo de buenos son los resultados obtenidos.

1. Implementación de los algoritmos de muestreo, avance y Viterbi

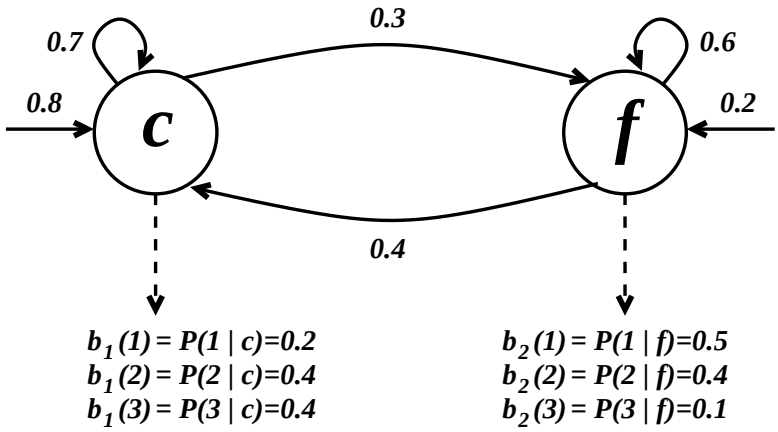
Se pide realizar una implementación en Python de los siguientes algoritmos:

- Algoritmo de muestreo: recibiendo como entrada un modelo oculto de Markov y un número natural n, genera una secuencia de n estados y la correspondiente secuencia de n observaciones, siguiendo las probabilidades del modelo. Ilustramos más adelante este algoritmo, con un ejemplo.
- Algoritmo de avance (*forward*): recibiendo como entrada un modelo oculto de Markov y una secuencia de n observaciones, realiza un filtrado. Es decir, calcula el estado más probable en el instante n, dadas las observaciones. Este algoritmo está explicado en el tema 4 de teoría.
- Algoritmo de Viterbi: recibiendo como entrada un modelo oculto de Markov y una secuencia de n observaciones, encuentra la secuencia de estados más probable, dadas las observaciones. Este algoritmo está explicado en el tema 4 de teoría.

Nótese que previo a la implementación de los algoritmos, es necesario diseñar una representación en Python para modelos ocultos de Markov. Se recomienda definir una clase, de la que los algoritmos anteriores sean métodos. La implementación debe ser general, y no depender de carácterísticas específicas de modelos ocultos de Markov concretos.

En esta primera parte, se pueden probar las implementaciones con los dos ejemplos de modelos ocultos de Markov vistos en el tema 4.

Pasamos a continuación a ilustrar con un ejemplo el primero de los algoritmos que se piden (muestreo), ya que es el único que no se ha visto en la teoría. Para ello, usamos el siguiente modelo oculto de Markov (ejemplo 1 del tema 4):



El problema es generar una secuencia de estados, con las correspondientes observaciones, siguiendo las probabilidades del modelo. Supondremos que disponemos de un generador de números aleatorios entre 0 y 1, con probabilidad uniforme. Vamos a generar una secuencia de 3 estados y la correspondiente secuencia de 3 observaciones.

El primer estado ha de ser generado siguiendo el vector de probabilidades iniciales. En este caso $\pi_1 = P(c) = 0{,}8$ y $\pi_2 = P(f) = 0{,}2$. Supongamos que al generar un número aleatorio, obtenemos 0,65. Esto significa que el primer estado en nuestra secuencia es c.

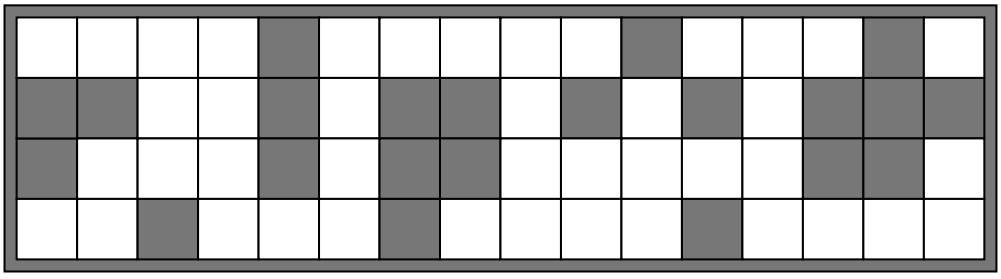
Ahora tenemos que generar la observación correspondiente, y para ello usamos las probabilidades de la matriz de observaciones. Puesto que el estado actual es c, las probabilidades de generar cada observable son: $b_1(1) = P(1|c) = 0{,}2$, $b_1(2) = P(2|c) = 0{,}4$ y $b_1(3) = P(3|c) = 0{,}4$. Si obtenemos aleatoriamente el número 0,53, eso significa que la observación correspondiente es 2, ya que es la primera de las observaciones cuya probabilidad acumulada $(0{,}2 + 0{,}4)$ supera a 0,53.

Generamos ahora el siguiente estado. Para ello usamos las probabilidades de la matriz de transición. Como el estado actual es c, las probabilidades de que cada estado sea generado a continuación son: $a_{11} = P(c|c) = 0{,}7$ y $a_{12} = P(f|c) = 0{,}3$. Si aleatoriamente obtenemos el número 0,82, significa que hemos obtenido f como siguiente estado.

Para la siguientes observaciones y estados, procedemos de la misma manera. Por ejemplo, si el siguiente número aleatorio que obtenemos es 0,29, la observación correspondiente es 1. Si a continuación obtenemos los números aleatorios 0,41 y 0,12, el estado siguiente es f y la observación sería 1. En resumen, hemos generado la secuencia de estados (c, f, f) con la correspondiente secuencia de observaciones $(2, 1, 1)$.

2. Aplicación a localización de robots

En este apartado se pide aplicar las implementaciones anteriores a un problema simple de localización de la posición de un robot que se mueve por una cuadrícula con obstáculos. Esta aplicación está descrita en la sección 15.3.2 del libro *Artificial Intelligence: A Modern Approach (3rd edition)* de S. Russell y P. Norvig. Consideremos la siguiente cuadrícula, por la que un robot se desplaza (las casillas sombreadas están bloquedas y no son transitables):



El robot puede inicar su movimiento en cualquiera de las casillas libres, con igual probabilidad. En cada instante, el robot se mueve de la casilla en la que está a una contigua: al norte, al sur, al este o al oeste, siempre que dicha casilla no esté bloqueda. El movimiento del robot está sujeto a incertidumbre, pero sabemos que se puede mover con igual probabilidad a cada casilla vecina no bloquedada.

Desgraciadamente, el robot no nos comunica en qué casilla se encuentra en cada instante de tiempo. Lo único que el robot puede observar en cada casilla son las direcciones hacia las que existen obstáculos. Por ejemplo una observación NS representa que el robot ha detectado que desde la casilla en la que está, al norte y al sur no puede transitar, pero que sí puede hacerlo a las casillas que están al este y al oeste.

Para acabar de complicar la cosa, los sensores de obstáculos que tiene el robot no son perfectos, y están sujetos a una probabilidad de error. Supondremos que hay una probabilidad ε de que la detección de obstáculo en una dirección sea errónea (y por tanto, hay una probabilidad $1 - \varepsilon$ de que sea correcta).

Supondremos también que los errores en cada una de las cuatro direcciones son independientes entre sí. Esto nos permite calcular la probabilidad de las observaciones dados los estados, como ilustramos a continuación.

Por ejemplo, supongamos que X y E son, respectivamente, las variables aleatorias que indican la casilla en la que está el robot y la observación que realiza el robot. Supongamos también que c es una casilla que hacia el norte y el este tiene obstáculos, y que tiene casillas transitables al sur y al oeste. Si por ejemplo el robot informara que existen obstáculos al sur y al este, la probabilidad de esto sería $P(E = SE|X = c) = \varepsilon^2(1-\varepsilon)^2$ (ya que habría errado en dos direcciones, norte y sur, y acertado en otras dos, este y oeste). Por el contrario, la probabilidad de que el robot informara de obstáculos al norte, sur y este, sería $P(E = NSE|X = c) = \varepsilon(1-\varepsilon)^3$ (ya que habría errado en una dirección y acertado en tres).

Se pide aplicar los algoritmos implementados en la primera parte para:

- Dada una secuencia de observaciones, dar una estimación de la casilla final en la que está el robot.
- Dada una secuencia de observaciones, dar la secuencia de casillas más probable por las que ha pasado el robot.

Nótese que para esta aplicación, es necesario representar la cuadrícula con alguna estructura de datos y, a partir de la cuadrícula, generar el correspondiente modelo oculto de Markov. **La implementación debe ser general y válida para cualquier cuadrícula que se reciba como entrada (no necesariamente la del dibujo)**. Por ello, se pide definir el modelo oculto de Markov correspondente a una cuadrícula dada, como subclase de los modelos ocultos de Markov definidos en la sección anterior. El constructor de esta subclase debe recibir como entrada la cuadrícula y la tasa de error ε.

3. Experimentar con la implementación

Una manera de evaluar cómo de buenas son las estimaciones obtenidas por los algoritmos anteriores es la siguiente. Podemos generar una secuencia de estados y la correspondiente secuencia de observaciones usando el algoritmo de muestreo. La secuencia de observaciones obtenida se puede usar como entrada a los problemas anteriores y comparar las respuestas con la secuencia de estados real que ha generado las observaciones. Se pide ejecutar con varios ejemplos y comprobar cómo de ajustados son los resultados obtenidos.

La valoración cuantitativa de la bondad de las estimaciones dependerá de si se trata del problema del filtrado o del problema de la secuencia más probable. Para el problema de filtrado, podemos medir la distancia Manhattan entre el estado final estimado y el estado final real. Para el problema de la secuencia más probable, podemos dar la proporción de estados coincidentes entre la secuencia real y la estimada, de entre el total de estados de la secuencia.

Se pide dar una serie de estadísticas (al estilo de las que aparecen en las gráficas de la figura 15.8 del libro de texto) que muestren el rendimiento medio de los algoritmos en sus respectivos problemas. Ir aumentando la longitud de las secuencias de observaciones (desde $n = 1$ hasta $n = 40$, por ejemplo) y medir el rendimiento en cada uno de los dos problemas. Para dar un rendimiento más ajustado, es conveniente que para cada longitud dada, se hagan un número de experimentos y se calcule el rendimiento medio. Se pide también las estadísticas con varios valores de ε.

Probar estos mismos experimentos con varias cuadrículas. Para ello, implementar una función que recibiendo como entrada el ancho, el alto y el número de obstáculos de la cuadrícula, genere aleatoriamente una cuadrícula con ese ancho, alto y número de obstáculos.