Modelo oculto de Márkov

Daniel de Roux

QUANTIL S.A.S.

7 de septiembre de 2015

Motivación

Supongamos que Avianca recibe diariamente miles de llamadas de compradores solicitando comprar pasajes.Avianca dispone de varios trabajadores que se encargan de recibir estas llamadas. Pensado que mantener a tanta gente en este trabajo es un gasto excesivo de recursos, la empresa decide construir un robot que maneje automáticamente estas solicitudes.

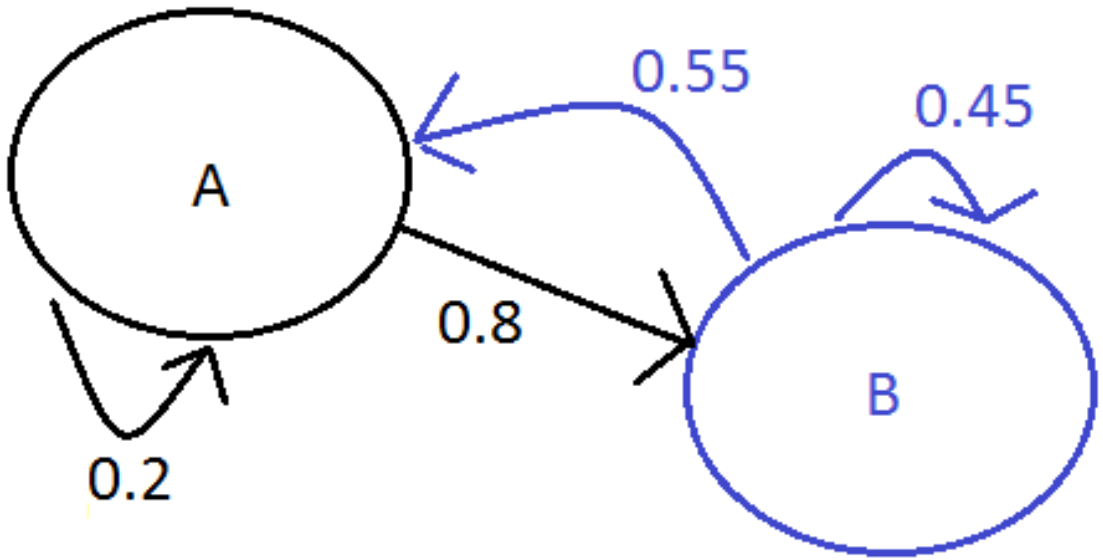
Nuestro robot debería poder clasificar las palabras para poder hacer su trabajo. Por lo tanto, el problema que vamos a atacar es el de POS (Part of speech). Este problema consiste en clasificar un conjunto de palabras usando una serie de etiquetas.Para esto, vamos a usar el modelo oculto de Márkov.

Cadenas de Márkov

Modelo de Márkov

Una cadena de Márkov es un proceso aleatorio que consiste de transiciones de un estado a otro. El conjunto de estados se conoce como el Espacio de estados. La probabilidad de pasar de un estado a otro solo depende del estado en el que se encuentra el proceso, y no de los eventos anteriores. Esta propiedad se llama propiedad de Márkov.

Un ejemplo sencillo



Cadena de Márkov

Más formalmente, una cadena de Márkov es una sucesión de variables aleatorias $X_1,\ X_2, X_3, \ldots$ que satisface la propiedad de Márkov. Es decir:

$$Pr(X_{n+1} = x | X_1 = x_1,\ X_2 = x_2,\ \ldots X_n = x_n) = Pr(X_{n+1} = x | X_n = x_n).$$

Los posibles valores que toman las variables aleatorias X_i son los estados de la cadena. Como vimos en el ejemplo, las cadenas de Márkov se suelen representar como un grafo dirigido.

Matriz de transición

Otra forma de representar una cadena de Márkov es con una matriz de transición. La matriz de transición del ejemplo es:

	A	B
A	0,2	0,8
B	0,55	0,45

La matriz de transición es muy útil pues nos permite hacer cálculos rapidamente. Si por ejemplo La cadena en la iteración n encuentra en el estado 2, y queremos saber la distribución de probabilidad dentro de 3 iteraciones, tenemos:

Matriz de transición

$$x^{(n+3)} = x^{(n+2)}P = (x^{(n+1)}P)P = x^{(n+1)}P^2 = (x^nP)P^2 = x^nP^3.$$

Como en el estado n nos encontramos en el estado B, tenemos que

$$x^{(n+3)} = (0,1) \cdot \begin{pmatrix} 0{,}382 & 0{,}618 \\ 0{,}4249 & 0{,}5751 \end{pmatrix} = (0{,}4249, 0{,}5751)$$

Modelo oculto de Márkov (HMM)

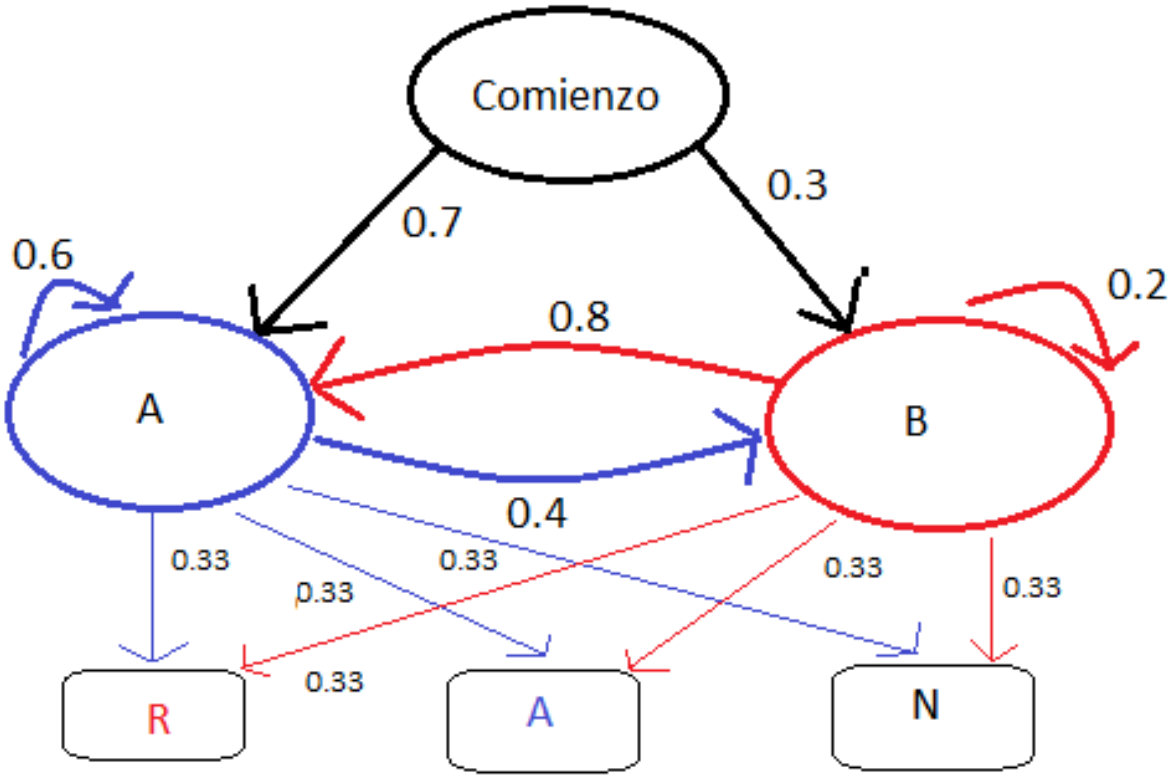
Definición

Un modelo oculto de Márkov (HMM) es un modelo donde se asume que el sistema que se modela es una cadena de Márkov, pero los estados son desconocidos (están escondidos).Veamos un ejemplo para aclarar las ideas.

Ejemplo

Supongamos que en cuarto oscuro, cuyo interior está oculto para un observador. Hay un genio y dos urnas $X_1,\ X_2$. Cada urna contiene 3 bolas, una roja, una azul y una negra. El genio escoge una urna y toma una bola al azar de esta urna. Luego, procede a mostrarnos la bola que salió y luego la vuelve a insertar en la urna de donde la sacó. El observador conoce la sucesión de bolas que fueron sacadas, pero no conoce la sucesión de urnas de las que fueron sacadas. Es importante decir que el genio tiene un proceso para escoger las urnas. Este proceso solo depende de una probabilidad y de la urna anterior.

Ejemplo



Probabilidad de una sucesión observada

Digamos que deseamos conocer, dados los parametros del problema, la probabilidad de una sucesión de eventos. Por ejemplo, cuál es de probabilidad de que saquemos la sucesión de bolas *RRNA*. La probabilidad de una sucesión

$$Y = y(0), y(1), ...y(N-1)$$

de longitud N esta dada por:

$$P(Y) = \sum_X P(Y|X)P(X),$$

donde esta suma se toma sobre todos los estados ocultos posibles:

$$X = x(0), x(1), x(2), ..x(N-1).$$

El problema principal

El problema, naturalmente, es que dada una sucesión de eventos, por ejemplo *NNRA*, no conocemos como fueron usadas las urnas para extraer las bolas. Por lo tanto, desearíamos , mirando la sucesión, determinar cual es la sucesión más probable de estados.
Notemos que esto es bien difícil de calcular! Incluso para nuestro pequeño ejemplo. Para calcular esto, se usa programación dinámica. Este método, a grandes razgos, consiste en separar un problema en problemas más pequeños, solucionarlos y luego combinar las soluciones para formar la solución global.

Part of speech

Part of speech (POS)

Como ya dijimos, el POS consiste en asignar etiquetas (o taggs) a un conjunto de palabras. Por ejemplo, dada la palabra 'casa', queremos asignarle una etiqueta que indique si la palabra es un verbo, un adjetivo, un nombre, etc. Esto no es trivial pues muchas palabras tienen significados distintos según el contexto y por lo tanto tienen etiquetas distintas.

Ejemplo

- El es un ser feliz.
- El quiere ser mejor.

En la primera frase, 'ser' es un nombre. En la segunda es un verbo. Un tagger (un programa que etiqueta las palabras) debe tomar muchas decisiones. Esto se debe a que las palabras ambiguas suelen usarse mucho.

Precisión

Qué porcentaje de las etiquetas que asignamos a las palabras es correcto? Con el método de HMM, la precisión del modelo es del 95 %. Esto parece ser muy bueno, pero notemos que el baseline (el método más bruto que se nos puede ocurrir) la precisión es del 90 %. Esto es, etiquetar cada palabra con la etiqueta más frecuente (usando unos datos de entrenamiento) y las palabras desconocidas etiquetarlas como nombres.

HMM y POS

Podemos pensar en el conjunto de etiquetas como los estados ocultos de nuestra cadena de Márkov. Las palabras que observamos son el conjunto de observaciones. Dada una frase, queremos saber cual es la secuencia de etiquetas que mejor corresponde a la frase.Construyamos un ejemplo.

HMM y POS

Supongamos que observamos una frase de n palabras. Llamamos a esta frase w_n. Queremos ahora encontrar la sucesíon de etiquetas que maximise la probabilidad de encontrar esta sucesión de palabras. Esto es bastante difícil de calcular, asi que asumimos dos los siguientes hechos para facilitar el cálculo:

- la probabilidad de ocurrencia de una palabra es independiente de las palabras que se encuentras antes y después de esta.
- La probabilidad de una etiqueta solo depende de la etiqueta inmediatamente anterior (propiedad de Márkov!).

HMM y POS

Si denotamos una sucesión de n etiquetas por t_i, buscamos t^* que maximice la siguiente expresión:

$$t^* = arg\,\max_{t_i} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}).$$

Cómo econtramos este t^*. Por ejemplo, podemos hacerlo por una de las dos siguientes maneras:

- Por fuerza bruta.
- Usando el algoritmo de Viterbi (programación dinámica.

El algoritmo de Viterbi

Intuitivamente, el algoritmo de Viterbi hace lo siguiente: Supongamos que nos encontramos en un estado en el tiempo t (estamos etiquetando la palabra número t). Lo más probable, es que haya solo una manera para llegar a este estado.Por lo tanto, si varios caminos llegan al estado t, y queremos calcular la probabilidad de transición al tiempo $t+1$, solo debemos mirar los caminos de mayor probabilidad llegando a t.Si hacemos esto para cada instante t, esto reduce el número de cálculos de N^T a $T * N^2$ (hay N etiquetas y T palabras). Construyamos un ejemplo.

El algoritmo de Viterbi

Para implementar el algoritmo de Viterbi, usamos una defición recursiva: Ponemos S como el conjunto de etiquetas. Suponemos que nuestra frase tiene longitud n. Para cada $k \in \{1, .., n\}$definimos :

$$\pi(k, u) = \max_{w \in S} \pi(k-1, w) \cdot P(u|w) \cdot P(x_k|u)$$

donde x_k es la $k-$ésima palabra. Finalmente, para inicializar el algoritmo, encontramos, por fuerza bruta, el mejor camino usando las dos primeras palabras.