Cadenas de Markov en el procesamiento del lenguaje natural

Mario Becerra Contreras
Estudiante de Matemáticas Aplicadas del ITAM

Introducción

El **procesamiento del lenguaje natural** es un problema de gran interés actual debido a la variedad de aplicaciones que tiene, como en traducción automática de texto, reconocimiento del lenguaje, extracción de información de textos voluminosos, resumir textos, sistemas automáticos de diálogo, corrección de textos, entre otras.

El procesamiento del lenguaje natural (de ahora en adelante abreviado NLP por sus siglas en inglés) presenta una enorme variedad de dificultades, principalmente por el hecho de que existen ambigüedades en el lenguaje y palabras que tienen varios significados y, en el caso de reconocimiento de voz, se tiene el problema de palabras o frases casi homófonas. En este artículo me enfocaré en particular en la parte técnica del NLP modelando el problema mediante **cadenas de Markov**.

Desarrollo

Se tiene un vocabulario finito V de palabras, en la práctica usualmente grande, pues debe "representar" todo un idioma. Como ejemplo, consideremos el vocabulario de juguete $V = \{el, perro, ladra\}$. Sea también V' el conjunto (infinito) de todas las combinaciones posibles de palabras de V. Definamos una oración como una sucesión de palabras $x_1 x_2 \dots x_n$ donde $n \geq 1$ y $x_i \in V$ para todo $i = 1, \dots, n-1$; consideraremos a x_n como un símbolo especial STOP; esto para términos prácticos porque las oraciones pueden tener cualquier longitud, entonces con el símbolo STOP se sabe dónde acaba la oración. Por lo tanto, con esta definición, podemos decir que el conjunto V' es el conjunto de oraciones en el lenguaje. Estas oraciones pueden tener sentido o no. Con vocabulario del ejemplo anterior, oraciones que podrían pertenecer al conjunto V' son *"el STOP"*, *"el perro STOP"*, *"el perro ladra STOP"*, *"el el STOP"*, *"el el perro STOP"*, *"el perro perro STOP"*, etc.

Teniendo esto en mente, el objetivo es encontrar una **distribución de probabilidad** para las oraciones en V', es decir, una función $\mathbb{P}$ tal que $\mathbb{P}(X = x) \geq 0$ para todo $x \in V'$ y $\sum_{x \in V'} \mathbb{P}(X = x) = 1$, donde X es una variable aleatoria discreta que toma valores en V'. Esto se logra a partir de un *set* de entrenamiento, el cual es un conjunto de muchas oraciones, usualmente miles, millones o más, provenientes de periódicos, de la red, de *tweets*, etc. A partir de estas oraciones, se estiman los parámetros de la distribución $\mathbb{P}$.

Un primer método para estimar los parámetros y tal vez el más sencillo en el que alguien podría pensar, pero que sirve para entender estos conceptos, es el que se expone a continuación.

Sea N el número de oraciones en el *set* de entrenamiento, y para cada oración $x \in V'$ sea $c(x)$ el número de veces que aparece la oración x en los datos. Entonces $\mathbb{P}(X = x) = \frac{c(x)}{N}$.

Este primer método claramente tiene muchas deficiencias. Si una oración en particular no se encuentra en el *set* de entrenamiento, le asignará probabilidad 0, por lo que este método no sirve para evaluar oraciones nuevas.

Otra forma que ha resultado ser muy eficiente es asumir la **propiedad de Markov**.

La sucesión de variables aleatorias $\{X_n\}_{n\in\mathbb{N}}$ cumple la propiedad de Markov si

$$\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}).$$

Esta propiedad se puede extender a que X_n dependa no solo de X_{n-1}, sino de los k pasos anteriores, es decir,

$$\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_{n-k} = x_{n-k}).$$

A ésta se le llama *propiedad de Markov de orden k*.

Nuestro objetivo es encontrar la probabilidad conjunta

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n),$$

donde las X_i son variables aleatorias que representan palabras.
Se puede demostrar fácilmente que, si se cumple la propiedad de Markov de orden 1, entonces:

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_1 = x_1) \prod_{i=2}^{n} \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}),$$

y, de la misma forma, si se cumple con la propiedad de Markov de orden 2 se tiene que

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_1 = x_1)\mathbb{P}(X_2 = x_2 | X_1 = x_1) \prod_{i=3}^{n} \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}).$$

Así, tenemos que se puede encontrar la probabilidad de una oración a partir de probabilidades condicionales. A este tipo de modelo se le llama modelo de n-grama. En general, un **n-grama** es una subsucesión de n elementos de una sucesión dada. O sea que en este caso es la probabilidad de una palabra dadas las n palabras anteriores. El caso en que $n = 1$ se llama unigrama, $n = 2$ bigrama, y $n = 3$ trigrama.

Ahora que tenemos un modelo de lenguaje, ¿cómo estimamos las probabilidades? La respuesta es relativamente sencilla. Sabemos que

$$\mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2}) = \frac{\mathbb{P}(X_i = x_i, X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2})}{\mathbb{P}(X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2})}.$$

Se puede demostrar que los **estimadores de máxima verosimilitud** de

$$\mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2}),\ \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}) \text{ y } \mathbb{P}(X_i = x_i)$$

son respectivamente:

$$q_{MV}(x_i | x_{i-1}, x_{i-2}) = \frac{c(x_i, x_{i-1}, x_{i-2})}{c(x_{i-1}, x_{i-2})}$$

$$q_{MV}(x_i | x_{i-1}) = \frac{c(x_i, x_{i-1})}{c(x_{i-1})}$$

$$q_{MV}(x_i) = \frac{c(x_i)}{c()}$$

donde $c(u, v, w)$ es el número de veces que aparece la sucesión de palabras u,v,w en el *set* de entrenamiento, y $c()$ es el número de palabras totales en el *set* de entrenamiento.

Ahora tenemos tres tipos de estimadores para el modelo, pero, ¿cuál deberíamos usar?

Existe una forma ingeniosa de agrupar la información de los tres estimadores. Esta es tomando $q(x_i|x_{i-1}, x_{i-2})$, el estimador de $\mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2})$, como:

$$q(x_i | x_{i-1}, x_{i-2}) = \lambda_1 \times q_{MV}(x_i | x_{i-1}, x_{i-2}) + \lambda_2 \times q_{MV}(x_i | x_{i-1}) + \lambda_3 \times q_{MV}(x_i).$$

Entonces,

$$q(x_i | x_{i-1}, x_{i-2}) = \lambda_1 \frac{c(x_i, x_{i-1}, x_{i-2})}{c(x_{i-1}, x_{i-2})} + \lambda_2 \frac{c(x_i, x_{i-1})}{c(x_{i-1})} + \lambda_3 \frac{c(x_i)}{c()}$$

donde $\lambda_i \geq 0$ para cada $i = 1, 2, 3$ y $\lambda_1 + \lambda_2 + \lambda_3 = 1$. A esta técnica se le conoce como **suavizado por interpolación lineal**. Es fácil demostrar que la función q es de probabilidad, pero en este artículo no se demostrará. Estos parámetros de suavizado se pueden interpretar como un indicador de confianza de cada estimador de trigrama, bigrama y unigrama.

Para escoger las λ óptimas se hace lo siguiente: digamos que tenemos algunos datos adicionales al *set* de entrenamiento, a los cuales llamaremos *datos de desarrollo*. Sea $c'(u, v, w)$ el número de veces que aparece el trigrama (u, v, w) en los datos de desarrollo. Se puede demostrar que la función de log-verosimilitud de los datos de desarrollo es, como función de las λ's, la siguiente:

$$L(\lambda_1, \lambda_2, \lambda_3) = \sum_{u,v,w} c'(u, v, w) \times \ln q(w|u, v).$$

Queremos las λ que maximicen la función L, por lo que los valores que se toman son

$$\arg \max_{\lambda_1,\lambda_2,\lambda_3} \{L(\lambda_1, \lambda_2, \lambda_3)\}$$

sujeto a $\lambda_i \geq 0$ para cada $i = 1, 2, 3$ y $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Existen métodos más sofisticados para encontrar λ que funcionan de acuerdo al contexto y al número de veces en que apareció cada palabra, y también existen otros métodos para suavizar los estimadores, pero por ahora nos quedaremos con este método más sencillo.

Conclusiones

Pudimos llegar a un modelo relativamente sencillo sin mayores complicaciones teóricas a partir del supuesto de que las variables aleatorias $\{X_n\}_{n\in\mathbb{N}}$ que pueden tomar valores en el vocabulario V, cumplen la propiedad de Markov, y así, podemos estimar la probabilidad de que ocurra cierta oración en un lenguaje. Para este artículo no se realizaron experimentos numéricos para probar la efectividad del modelo, pero este ya ha sido probado con anterioridad y ha dado buenos resultados.

Bibliografía

[1] Collins, Michael. *Language Modeling.* Columbia University, 2013. `http://www.cs.columbia.edu/ mcollins/lm-spring2013.pdf`

[2] Jurafsky, Daniel y Martin, James H. *Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing.* 2007.

[3] *Natural Language Processing.* Columbia University, Coursera, 2014. Impartido por Michael Collins.

[4] Wikipedia, la enciclopedia libre. *N-grama.* `http://es.wikipedia.org/wiki/N-grama` (consultada el 21 de julio de 2014).