Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

# Temas Selectos de Análisis de Datos (Tarea 2)

José Antonio Garcia Ramirez

20 de septiembre de 2018

## 1. Ejercicio 1

Este ejercicio es sobre Hidden Markov Models (HMM).
Vimos en clase que el método más usado para el proceso de POS-tagging es HMM, donde asignamos etiquetas gramaticales POS (variables *latentes u ocultas*) a una secuencia de palabras (variables *observables*).
Dado un Corpus de entrenamiento y una secuencia de palabras de prueba, HMM calcula la *secuencia de etiquetas POS* más probable mediante la expresión

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n|w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}),$$

o, en palabras,

$$\hat{t}_1^n \approx \arg\max_{t_1^n} \prod P(\text{emisión})P(\text{transición}).$$

Estas probabilidades están totalmente definidas por el Corpus, y el proceso de entrenamiento de la red se realiza mediante operaciones de conteo de las palabras y las etiquetas contenidas.
Vamos a utilizar un *mini* Corpus tomado de la NYU[1], el cual puedes encontrar en el archivo `POSData.zip`, y contiene el corpus de entrenamiento (`training.pos`) y otro corpus de prueba (`development.pos`).

[1] `http://cs.nyu.edu/courses/spring12/CSCI-GA.2590-001/`

1. *Calcula las probabilidades de emisión y transmisión (verosimilitud y apriori) a partir del corpus de entrenamiento.*

   Después de las dificultades con el corpus (pues el package *tokenizer* borraba los tags con un solo elemento), opte por leerlo como un texto y después dividir cada línea por el carácter tabulador el código que realiza los calculos estan en el script anexo con el nombre 'tarea2.R' (en sus líneas 1 a 93 )

2. *Considera el texto de prueba:*

   ```
   Your contribution to Goodwill will mean more than you may know.
   ```

   *Obten los tokens del texto de prueba. Verifica que, para el corpus de entrenamiento usado, las posibles etiquetas POS para cada token es:*

   ```
   $Your
   [1] "PRP$"
   $contribution
   [1] "NN"
   $to
   [1] "IN" "TO"
   $Goodwill
   [1] "NNP"
   $will
   [1] "NN" "MD"
   $mean
   [1] "JJ"  "VB"  "VBP"
   $more
   [1] "RB"  "JJR" "RBR"
   $than
   [1] "IN"
   $you
   [1] "PRP"
   $may
   [1] "MD"
   $know
   [1] "NN"  "VB"  "VBP"
   $.
   [1] "COMMA" "."
   ```

   Después de leer la documentación de coreNLP configure mi archivo, que debe de llamarse 'en.properties' con el siguiente contenido:

```
annotators=tokenize,ssplit,pos
tokenize.language=english
```

Haciendo uso del package 'coreNLP' tokenize el texto de prueba

Mi salida con las etiquetas POS son las siguientes:

```
$`palabra.tag`
[1] "Your" "PRP$"

$palabra.tag
[1] "contribution" "NN"

$palabra.tag
[1] "to" "IN" "TO"

$palabra.tag
[1] "Goodwill" "NNP"

$palabra.tag
[1] "will" "MD"   "NN"

$palabra.tag
[1] "mean" "JJ"   "VB"   "VBP"

$palabra.tag
[1] "more" "JJR"  "RB"   "RBR"

$palabra.tag
[1] "than" "IN"

$palabra.tag
[1] "you" "PRP"

$palabra.tag
[1] "may" "MD"

$palabra.tag
[1] "know" "NN"   "VB"   "VBP"

$palabra.tag
[1] "."     "."     "COMMA"
```

Donde es de notar que todos los tags son iguales.

3. *Verifica que el número de posibles secuencias (*paths*) para nuestro sencillo texto de prueba es de 216. Esto te puede dar una idea de la complejidad computacional de éste tipo de problemas. Puedes verificarlo computacionalmente.*

   En teoría son más porque del punto de inicio a cualquier tag existe un path lo que incrementa el número de posibles tags (multiplicado por la cardinalidad de tags posibles, en nuestro caso 52). Si consideramos la primera transición del inicio a la primer palabra 'Your' tenemos entonces 12 palabras y 11 transiciones entre tags. Para la primer transición solo hay un path posible (pues solo existe un tag al cual llegar), para la segunda transición hay dos paths posibles (pues existen dos posibles tags para la tercer palabra 'to'), para la tercera solo hay una, para la cuarta hay dos posibilidades, para la quinta y sexta hay 3 posibilidades , para la séptima, octava y novena transiciones solo hay una posibilidad, para la tercera transición existen tres posibles estados y para la última existen dos posibles estados así que el número de posibles paths es: $1 * 2 * 1 * 2 * 3 * 3 * 1 * 1 * 1 * 3 * 2 = 216$

4. Usa el algoritmo Viterbi para estimar la secuencia de etiquetas POS del texto de prueba. Compara tu resultado con el obtenido al usar el Anotador de `coreNLP` de Stanford.

| | palabras | TagViterbi |
|---|---|---|
| 1 | your | PRP |
| 2 | contribution | NN |
| 3 | to | TO |
| 4 | goodwill | NN |
| 5 | will | MD |
| 6 | mean | VB |
| 7 | more | JJR |
| 8 | than | IN |
| 9 | you | PRP |
| 10 | may | MD |
| 11 | know | VB |
| 12 | . | . |

   De la salida anterior podemos observar que las etiquetas del algoritmo 'viterbi()' son un subconjunto de las de coreNLP

5. Generalmente, no todas las palabras están incluidas en el Corpus de entrenamiento. Intenta, por ejemplo, asignar POS-tags al texto:

   `Coming to Goodwill was the first step toward my becoming totally`.

¿ Qué podemos hacer en este caso? Implementa tu idea y verifica su desempeño con textos del corpus de prueba.
Podemos observar que palabra 'totally' no está en el corpus, pruebo con dos opciones. La primera consiste en reemplazar 'totally' con estructura similar como lo es 'total' (lo que es parecido a realizar un procedimiento de stemming en la palabra) que solo tiene una frecuencia de 6 el tag es el siguiente:

| | palabras | Tag |
|---|---|---|
| 1 | coming | ” |
| 2 | to | ” |
| 3 | goodwill | ” |
| 4 | was | ” |
| 5 | the | ” |
| 6 | first | ” |
| 7 | step | ” |
| 8 | toward | ” |
| 9 | my | ” |
| 10 | becoming | ” |
| 11 | total | ” |
| 12 | . | ” |

Podemos ver que las etiquetas no se determinaron, por lo que optamos por aumentar una etiqueta o tag extra que denotaremos como 'XX' para incluirla en la matriz de transición de estados y otra etiqueta 'XX' para posibles palabras desconocidas, como no tenemos información a priori acerca de la frecuencia de palabras desconocidas usaremos una distribución uniforme sobre las matrices de transición y de observación y finalmente reemplazamos la palabra que no aparece en el corpus por el símbolo 'XX'. En el siguiente cuadro mostramos las etiquetas que se dieron con esta nueva idea, lo importante en este caso es que si bien las hay (los tags) estos estan erroneos para las palabras que ya conocemos esto se debe a la distribución que a priori que se utilizó.

| | palabras | Tag |
|---|---|---|
| 1 | coming | VBG |
| 2 | to | TO |
| 3 | goodwill | XX |
| 4 | was | VBD |
| 5 | the | DT |
| 6 | first | JJ |
| 7 | step | NN |
| 8 | toward | XX |
| 9 | my | PRP$ |
| 10 | becoming | XX |
| 11 | XX | XX |
| 12 | . | . |

Repetimos el experimento con probabilidades iguales pero más pequeñas del orden de $10^{-100}$ para esta palabra desconocida por el corpus. Oteniendo como resultado lo siguiente

| | palabras | Tag |
|---|---|---|
| 1 | coming | VBG |
| 2 | to | TO |
| 3 | goodwill | NN |
| 4 | was | VBD |
| 5 | the | DT |
| 6 | first | JJ |
| 7 | step | NN |
| 8 | toward | IN |
| 9 | my | XX |
| 10 | becoming | VBG |
| 11 | XX | NN |
| 12 | . | . |