UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

Departamento de sistemas informáticos y computación

Predicción Estructurada Estadística

Cuestionario 1-B

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Autor
Francisco Javier Gil-Terrón Rodríguez

2021 - 2022

Tabla de contenidos

1. Cuestiones teóricas

1.1 Cuestión 1

En esta primera cuestión se pide analizar el comportamiento del algoritmo Inside-Outside cuando las probabilidades iniciales asociadas a las reglas son equiprobables. Para ellos, se partirá del ejemplo de las diapositivas.

Bajo esta premisa, se modificarán las probabilidades del ejemplo tal que, la probabilidad se repartirá equitativamente entre todas las opciones de cada regla, por ejemplo, la probabilidad de que un nombre sea cualquiera de las opciones (vieja, ayuda, pelea, mujer y demanda) será de 0.2 ya que existen cinco opciones, o la probabilidad de que un prefijo se componga de un verbo o de un verbo y un nombre será de 0.5 para ambos casos ya que sólo existen estas dos opciones. Es decir, puesto que las probabilidades iniciales de las reglas son equiprobables todas las formas sintácticas que sean iguales (por ejemplo, Art Nom) tendrán la misma probabilidad independientemente de los elementos que la compongan.

Con esto en mente, y dadas las muestras del ejemplo D = {la vieja demanda ayuda, la mujer oculta pelea, la vieja ayuda}, calculamos las nuevas probabilidades de las posibles ramas de las muestras:

P_Θ (la vieja demanda ayuda) = 0.0016 + 0.00416 = 0.00583

P_Θ (la mujer oculta pelea) = 0.0016 + 0.00416 = 0.00583

P_Θ (la vieja ayuda) = 0.0083

Y calculamos la nueva estimación de la regla (Suj → Art Nom Adj) en estas muestras con el algoritmo Inside-Outside:

$$\bar{p}(Suj \rightarrow ArtNomAdj) = \frac{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj \rightarrow ArtNomAdj, t_x) P_\Theta(x, t_x)}{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj, t_x) P_\Theta(x, t_x)} =$$

$$= \frac{\frac{1}{0.00583} 0.00416}{\frac{1}{0.00583}(0.0016 + 0.00416) + \frac{1}{0.00583}(0.0016 + 0.00416) + \frac{1}{0.0083}(0.0083)} =$$

$$= \frac{0.7142}{3} = 0.238$$

1.2 Cuestión 2

El siguiente caso consiste en calcular la estimación de la regla (Suj →Art Nom) con el ejemplo anteriormente usado, mediante el algoritmo Inside-Outside para las muestras D = {(la vieja) (demanda ayuda), la mujer oculta pelea, la vieja ayuda}.

En este caso, ya que partimos del ejemplo, únicamente habrá que calcular la probabilidad de las cadenas de la primera muestra, ya que las otras dos ya las da el ejemplo.

P_Θ ((la vieja) (demanda ayuda)) = 0.0009

P_Θ (la mujer oculta pelea) = 0.0009 + 0.01176 = 0.01266

P_Θ (la vieja ayuda) = 0.007

Con esta información ya podemos estimar la regla (Suj →Art Nom) con Inside-Outside:

$$\bar{p}(Suj \rightarrow ArtNom) = \frac{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj \rightarrow ArtNom, t_x) P_\Theta(x, t_x)}{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj, t_x) P_\Theta(x, t_x)} =$$

$$= \frac{\frac{1}{0.0009} 0.0009 + \frac{1}{0.01266} 0.0009 + \frac{1}{0.007} 0.007}{\frac{1}{0.0009}(0.0009) + \frac{1}{0.01266}(0.0009 + 0.01176) + \frac{1}{0.007}(0.007)} =$$

$$= \frac{2.071}{3} = 0.69$$

1.3 Cuestión 3

Para repetir el caso anterior empleando el algoritmo de Viterbi, bastará tener en cuenta que este algoritmo contabiliza el número de veces que se encuentra la regla (en este caso (Suj → Art Nom)) en el árbol de derivación más probable.

$$\bar{p}(Suj \rightarrow ArtNom) = \frac{\sum_{x \epsilon \mathcal{D}} N(Suj \rightarrow ArtNom, \widehat{t_x})}{\sum_{x \epsilon \mathcal{D}} N(Suj, \widehat{t_x})}$$

Ya que la primera y tercera muestra tienen un único árbol de derivación posible, y en ambos aparece la regla, se contabilizarán. Mientras que, en el caso de la segunda muestra, se compone de dos árboles de derivación y en el mejor árbol (o el más probable), es decir, el de probabilidad 0.011176, no se encuentra la regla, por lo que no se contabilizará. En conclusión, la estimación de la regla (Suj → Art Nom) con el algoritmo de Viterbi para las muestras propuestas, será 2/3 = 0.666.

Como se puede ver, el algoritmo Inside-Outside ofrecía un mejor resultado, ya que su estimación daba 0.69 para las mismas muestras frente a 0.66 del algoritmo de Viterbi, aunque computacionalmente es menos caro este último Algoritmo.

1.4 Cuestión 4

A continuación, se pide la estimación de la regla (Suj →Art Nom) con el mismo ejemplo mediante el algoritmo Inside-Outside para las muestras D = {la vieja demanda ayuda, la mujer oculta pelea, la vieja mujer oculta demanda ayuda}.

En primer lugar, será necesario añadir una regla a la gramática, ya que con la del ejemplo no se puede generar ningún árbol de derivación para la última muestra, para lo que se añadirá la regla (Suj → Art Adj Nom Adj) y se redistribuirá la probabilidad entre las reglas de la siguiente manera:

0.4 (Suj → Art Nom)

0.2 (Suj → Art Adj Nom)

0.2 (Suj → Art Nom Adj)

0.2 (Suj → Art Adj Nom Adj)

Con esta modificación, las probabilidades para los árboles de derivación de las muestras serán:

P_θ (la vieja demanda ayuda) = 0.00072 + 0.00168 = 0.0024

P_θ (la mujer oculta pelea) = 0.00072 + 0.01176 = 0.01248

P_θ (la vieja mujer oculta demanda ayuda) = 0.0002268

Esta última muestra producirá un único árbol de derivación, tal como se puede ver en la siguiente ilustración:

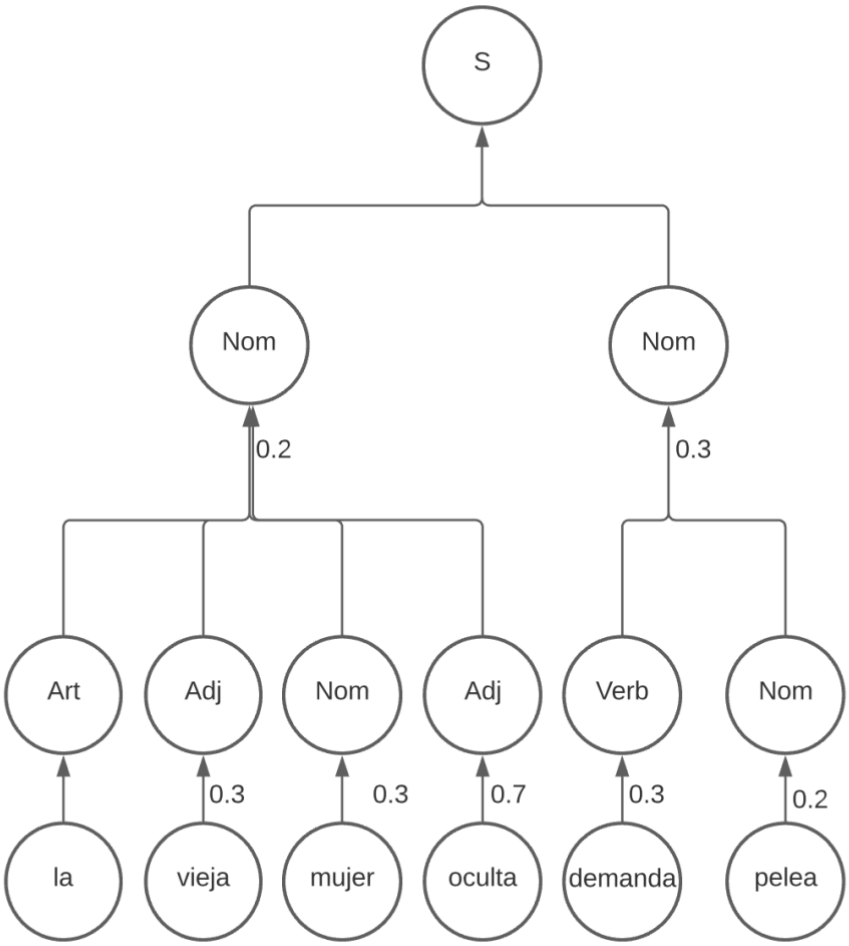

Figura 1 – Árbol de derivación

Con esta información ya podemos estimar la regla (Suj →Art Nom) con Inside-Outside:

$$\bar{p}(Suj \rightarrow ArtNom) = \frac{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj \rightarrow ArtNom, t_x) P_\Theta(x, t_x)}{\sum_{x \epsilon \mathcal{D}} \frac{1}{P_\Theta(x)} \sum_{t_x} N(Suj, t_x) P_\Theta(x, t_x)} =$$

$$= \frac{\frac{1}{0.0024} 0.00072 + \frac{1}{0.01248} 0.00072}{\frac{1}{0.0024}(0.00072 + 0.00168) + \frac{1}{0.01248}(0.00072 + 0.01176) + \frac{1}{0.0002268}(0.0002268)}$$

$$= \frac{0.3577}{3} = 0.1192$$

1.5 Cuestión 5

Repetir el caso anterior empleando el algoritmo de k-mejores con k=2 daría excatamente el mismo resultado que empleando el algoritmo Inside-Outside, ya que el algoritmo k-mejores tomaría los dos mejores árboles de derivación de cada muestra, pero como se puede ver en el apartado anterior, ninguna de las tres muestras propuestas genera más de dos árboles de derivación.

2. Cuestiones prácticas

En este último apartado práctico de la tarea se propone, empleando una vez más el toolkit SCFG para la clasificación de triángulos, evaluar el comportamiento de los algoritmos Inside-Outside y Viterbi con muestras parentizadas o con braquets, y sin ellos.

2.1 Cuestión 7

La primera cuestión consiste en estudiar la variación en el número de triángulos rectángulos en función de la cantidad de símbolos no terminales. Siguiendo el ejemplo propuesto en el boletín y generando 1000 triángulos se han obtenidos los siguientes resultados:

No terminales	**Triángulos rectángulos**
5	29
10	63
15	61
20	84

Tabla 1 – Triángulos rectángulos en función del nº de no terminales

Tal como se puede ver en la tabla, existe una relación proporcional entre el número de no terminales y de triángulos rectángulos generados, lo que se puede deber a que, cuanto mayor es el número de no terminales, más flexibilidad tendrá el modelo para aprender.

2.2 Cuestión 8

A continuación, se estudiarán los resultados de clasificación de los triángulos según el algoritmo empleado y si la muestra tenía brackets o no. Para ello, se analizarán los resultados a través de las matrices de confusión:

	Equi	Isos	Righ	ERR	ERR%
Equi	396	452	152	604	60.4
Isos	387	319	294	681	68.1
Righ	440	363	197	803	80.3
Total ERR				2088	69.6

Tabla 2 – Matriz de confusión Inside-Outside con brackets

	Equi	Isos	Righ	ERR	ERR%
Equi	705	212	83	295	29.5
Isos	460	311	229	689	68.9
Righ	393	267	340	660	66.0
Total ERR				1644	54.8

Tabla 3 – Matriz de confusión Inside-Outside sin brackets

	Equi	Isos	Righ	ERR	ERR%
Equi	707	293	0	293	29.3
Isos	518	277	205	723	72.3
Righ	276	280	444	556	55.6
Total ERR				1572	52.4

Tabla 4 – Matriz de confusión Viterbi con brackets

	Equi	Isos	Righ	ERR	ERR%
Equi	311	379	310	689	68.9
Isos	435	263	302	737	73.7
Righ	253	359	388	612	61.2
Total ERR				2038	67.93

Tabla 5 – Matriz de confusión Viterbi sin brackets

Como se puede ver en las tasas de error de las matrices de confusión de los modelos, la clasificación de triángulos isósceles y rectángulos es la que mayor número de fallos tiene (variando ligeramente en función del modelo) mientras que en la clasificación de equiláteros hay una gran variabilidad de un modelo a otro.

Más en detalle, a partir de la matriz de cada modelo, podemos analizar dónde se producen los errores en cada caso, por ejemplo, a partir de la matriz de confusión del modelo generado con el algoritmo Inside-Outside con muestras con brackets, se puede saber que este clasifica erróneamente una gran cantidad de triángulos isósceles confundiéndolos con equiláteros y viceversa, mientras que los triángulos rectángulos son confundidos tanto con equiláteros como con isósceles, siendo el tipo de triángulos que peor clasifica.

Estos resultados son mejores para este algoritmo en el caso de muestras sin brackets, donde mejora drásticamente el reconocimiento de triángulos equiláteros y moderadamente el de rectángulos, aunque estos últimos se siguen confundiendo con los otros dos. Por su parte, los isósceles se siguen confundiendo con equiláteros.

Por otro lado, en el caso del modelo generado con algoritmo de Viterbi, podemos observar, el caso con brackets, que se han obtenido unos resultados similares a los del modelo inmediatamente anterior, clasificando ligeramente peor los isósceles y mejor los rectángulos.

Por último, en la versión del algoritmo de Viterbi sin brackets, se puede ver que es similar al primer caso, aunque parece que la tendencia es clasificar al azar, ya que independientemente del tipo de triángulo, lo confunde con los otros dos tipos de triángulos.

2.3 Cuestión 9

En este último apartado se estudiará, de manera similar a cómo se hizo en el ejercicio anterior, la variación en las prestaciones de los algoritmos de Viterbi e Inside-Outside, pero en este caso, en función del tamaño del conjunto de entrenamiento. Hasta ahora se trabajaba con un conjunto de 1000 muestras, por lo que en esta sección se probará tanto a reducir como aumentar este valor para observar el impacto que tiene en los algoritmos. De esta manera se experimentará con tamaños de 500 y 1500 para los conjuntos de entrenamiento.

Los resultados obtenidos se pueden ver en las siguientes tablas:

	Equi	Isos	Righ	ERR	ERR%
Equi	248	436	316	752	75.2
Isos	328	450	222	550	55.0
Righ	439	313	248	752	75.2
Total ERR				2054	68.47

Tabla 6 – Matriz de confusión Inside-Outside con 500 muestras de entrenamiento

	Equi	Isos	Righ	ERR	ERR%
Equi	251	596	153	749	74.9
Isos	390	397	213	603	60.3
Righ	439	326	235	765	76.5
Total ERR				2117	70.57

Tabla 7 – Matriz de confusión Inside-Outside con 1500 muestras de entrenamiento

	Equi	Isos	Righ	ERR	ERR%
Equi	368	470	162	632	63.2
Isos	396	428	176	572	57.2
Righ	281	398	321	679	67.9
Total ERR				1883	62.77

Tabla 8 – Matriz de confusión Viterbi con 500 muestras de entrenamiento

	Equi	Isos	Righ	ERR	ERR%
Equi	630	293	77	370	37.0
Isos	487	269	244	731	73.1
Righ	318	267	415	585	58.5
Total ERR				1686	56.2

Tabla 9 – Matriz de confusión Viterbi con 1500 muestras de entrenamiento

Los resultados particulares de cada una de las matrices son similares a los que se vieron en el apartado anterior, por lo que no se hará hincapié en analizarlos, sino que nos centraremos en estudiar la variación que hay con respecto al número de muestras de entrenamiento.

Si comparamos con los resultados obtenidos en el apartado anterior para el algoritmo de Viterbi, (en la versión con brackets, ya que las nuevas muestras generadas tienen brackets), como es de esperar, reducir la cantidad de datos de entrenamiento empeora los resultados, mientras que si aumenta el conjunto de entrenamiento no se mejoran las prestaciones, sino que se obtiene un valor similar, aunque ligeramente peor.

Por otro lado, en cuanto al algoritmo Inside-Outside, pese al haber reducido la cantidad de datos de entrenamiento los resultados a penas se han visto modificados, lo que ocurre de igual manera al aumentar el tamaño del conjunto de entrenamiento, es decir, los resultados obtenidos son similares independientemente del número de muestras de entrenamiento.