

José Javier Calvo Moratilla
2021/2022

Predicción Estructurada Estadística

Entrega nº2

Ejercicios teóricos:

Question1: Analyze the behavior of the Inside-Outside estimation algorithm when the initial probabilities associated to the rules are equiprobable. Justify your comments with the example in page 50 from Part I.4 Model Parameter Estimation (P-I.4 from now on)

1 S → Suj PRed	Después de actualizar las probabilidades de las reglas como equiprobables se calculan las probabilidades para cada frase:
1/3 Suj → Art Nom	
1/3 Suj → Art Adj Nom	
1/3 Suj → Art Nom Adj	$P(\text{"La vieja demanda ayuda"})$ $= (1 * 0.2 * 0.25 * 0.2 * \frac{1}{3} * 0.5) + (1 * 0.5 * 0.2 * 0.25 * \frac{1}{3} * 0.5)$ $= 5.883 \times 10^{-3}$
0.5 Pred → Verb Nom	
0.5 Pred → Verb	
1 Art → "la"	$P(\text{"La mujer oculta pelea"})$ $= (1 * 0.2 * 0.25 * 0.2 * \frac{1}{3} * 0.5) + (1 * 0.2 * 0.5 * 0.25 * \frac{1}{3} * 0.5)$ $= 5.883 \times 10^{-3}$
0.2 Nom → "vieja"	
0.2 Nom → "ayuda"	
0.2 Nom → "mujer"	$P(\text{"La viaja ayuda"})$ $= (1 * 0.2 * 0.25 * \frac{1}{3} * 0.5) = 8.333 \times 10^{-3}$
0.2 Nom → "pelea"	
0.2 Nom → "demanda"	
0.25 Verb → "demanda"	Se concluye que con la probabilidad equiprobable, las frases de menor longitud son las que obtienen una mayor probabilidad.
0.25 Verb → "ayuda"	
0.25 Verb → "oculta"	
0.25 Verb → "pelea"	
0.5 Adj → "vieja"	
0.5 Adj → "oculta"	

Question2: Compute the estimation of the rule ($Suj \rightarrow Art\ Nom$) with the example in page 50 from P-I.4 with the Inside-Outside algorithm with the following training sample that includes bracketed samples: $D = \{(la\ vieja)(demanda\ ayuda), la\ mujer\ oculta\ pelea, la\ vieja\ ayuda\}$

1 S \rightarrow Suj PRed	$D = \{(La\ vieja)\ (demanda\ ayuda),\ la\ mujer\ oculta\ pelea,\ la\ vieja\ ayuda\} =$
0.5 Suj \rightarrow Art Nom	
0.3 Suj \rightarrow Art Adj Nom	
0.2 Suj \rightarrow Art Nom Adj	$P_{\theta}((La\ vieja)\ (demanda\ ayuda)) =$
0.3 Pred \rightarrow Verb Nom	$= (1 * 0.1 * 0.3 * 0.2 * 0.5 * 0.3)$
0.7 Pred \rightarrow Verb	$= 9.0000e - 04 = 0.00090$
1.0 Art \rightarrow "la"	$P_{\theta}(la\ mujer\ oculta\ pelea) =$
0.1 Nom \rightarrow "vieja"	$= (1 * 0.3 * 0.1 * 0.2 * 0.5 * 0.3) + (1 * 0.3 * 0.7 * 0.4 * 0.2 * 0.7)$
0.2 Nom \rightarrow "ayuda"	$= 0.012660$
0.3 Nom \rightarrow "mujer"	$P_{\theta}(la\ vieja\ ayuda) = (1 * 0.1 * 0.2 * 0.5 * 0.7)$
0.2 Nom \rightarrow "pelea"	$= 7.0000e - 03 = 0.00700$
0.2 Nom \rightarrow "demanda"	$\rho'(Suj \rightarrow Art\ Nom)$
0.3 Verb \rightarrow "demanda"	$= \frac{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj \rightarrow Art\ Nom, t_x) * P_{\theta}(x, t_x)}{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj, t_x) * P_{\theta}(x, t_x)}$
0.2 Verb \rightarrow "ayuda"	$= \frac{\frac{1}{0.00090} * 0.00090 + \frac{1}{0.012660} * 0.00090 + \frac{1}{0.00700} * 0.00700}{\frac{1}{0.00090} * 0.00090 + \frac{1}{0.012660} * 0.012660 + \frac{1}{0.00700} * 0.00700}$
0.1 Verb \rightarrow "oculta"	
0.4 Verb \rightarrow "pelea"	
0.3 Adj \rightarrow "vieja"	
0.7 Adj \rightarrow "oculta"	$= 0,69036$

Question3: Repeat the previous exercise but using the Viterbi-Score estimation algorithm.

1 S → Suj PRed	$D = \{(La\ vieja)\ (demanda\ ayuda),\ la\ mujer\ oculta\ pelea,\ la\ vieja\ ayuda\} =$
0.5 Suj → Art Nom	
0.3 Suj → Art Adj Nom	
0.2 Suj → Art Nom Adj	<p>Se obtiene el camino más probable por viterbi en el árbol de derivación correspondiente, para cada frase, y se comprueba si aparece la regla Suj → Art Nom, si es así lo contabilizamos, sinó no. En último lugar el resultado obtenido se divide por el número total de frases, obteniendo la estimación final.</p>
0.3 Pred → Verb Nom	
0.7 Pred → Verb	
1.0 Art → "la"	$P_{\theta}((La\ vieja)\ (demanda\ ayuda))$ Solo hay un camino posible con probabilidad 0.00090 El camino contiene (Suj → Art Nom)
0.1 Nom → "vieja"	
0.2 Nom → "ayuda"	
0.3 Nom → "mujer"	$P_{\theta}(la\ mujer\ oculta\ pelea)$ Hay 2 caminos posibles $9.0000e - 04$ o 0.011760 El camino más probable no contiene (Suj → Art Nom)
0.2 Nom → "pelea"	
0.2 Nom → "demanda"	
0.3 Verb → "demanda"	$P_{\theta}(la\ vieja\ ayuda)$ Solo hay un camino posible con probabilidad 0.00700 El camino contiene (Suj → Art Nom)
0.2 Verb → "ayuda"	
0.1 Verb → "oculta"	
0.4 Verb → "pelea"	$\rho'(Suj \rightarrow Art\ Nom) = \frac{2}{3} = 0.6667$
0.3 Adj → "vieja"	
0.7 Adj → "oculta"	

Question4: Compute the estimation of the rule (Suj → Art Nom) with the example in page 49 from P-I.4 with the Inside-Outside algorithm with the following training sample: $D = \{la\ vieja\ demanda\ ayuda, la\ mujer\ oculta\ pelea, la\ vieja\ mujer\ oculta\ demanda\ ayuda\}$. You can use the hyp toolkit1 and you have to provide a plot for the last sentence similar to the ones in slide 50 of P-I.4. Compute just one iteration.

1 S → Suj PRed	$D = \{la\ vieja\ demanda\ ayuda, la\ mujer\ oculta\ pelea, \}$
0.3 Suj → Art Nom	$la\ vieja\ mujer\ oculta\ demanda\ ayuda\} =$
0.2 Suj → Art Adj Nom Adj	Después de analizar la tercera frase se comprueba que no existe regla para las categorías gramaticales que forman el sujeto:
0.3 Suj → Art Adj Nom	
0.2 Suj → Art Nom Adj	<div style="text-align: center;"> <p>Sujeto Predicado</p> <p><u>“La vieja mujer oculta demanda ayuda”</u></p> <p>Art Adj Nom Adj Verb Nom</p> </div>
0.3 Pred → Verb Nom	
0.7 Pred → Verb	
1.0 Art → “la”	Se decide crear una nueva regla (Suj → Art Adj Nomb Ajd)
0.1 Nom → “vieja”	Y se coge parte la probabilidad de la regla (Suj → Art Nom) donde se define una probabilidad de 0.2 para la nueva regla y un 0.3 para la original.
0.2 Nom → “ayuda”	
0.3 Nom → “mujer”	Se calculan las probabilidades de cada frase en su respectivo árbol obteniendo los siguientes resultados:
0.2 Nom → “pelea”	
0.2 Nom → “demanda”	$P_{\theta}(La\ vieja\ demanda\ ayuda) =$ $= (1 * 0.1 * 0.3 * 0.2 * 0.3 * 0.3) +$ $(1 * 0.3 * 0.2 * 0.2 * 0.3 * 0.7) = 3.0600 \times 10^{-3}$
0.3 Verb → “demanda”	
0.2 Verb → “ayuda”	
0.1 Verb → “oculta”	$P_{\theta}(La\ mujer\ oculta\ pelea) =$ $= (1 * 0.3 * 0.1 * 0.2 * 0.3 * 0.3) +$ $(1 * 0.3 * 0.7 * 0.4 * 0.2 * 0.7) = 0.012300$
0.4 Verb → “pelea”	
0.3 Adj → “vieja”	$P_{\theta}(La\ vieja\ mujer\ oculta\ demanda\ ayuda) =$ $= (1 * 0.3 * 0.3 * 0.7 * 0.3 * 0.2 * 0.2 * 0.3) = 2.2680 \times 10^{-4}$
0.7 Adj → “oculta”	$\rho'(Suj \rightarrow Art\ Nom)$ $= \frac{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj \rightarrow Art\ Nom, t_x) * P_{\theta}(x, t_x)}{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj, t_x) * P_{\theta}(x, t_x)}$ $= \frac{\frac{1}{3.0600 \times 10^{-3}} * 5.4000 \times 10^{-4} + \frac{1}{0.012300} * 5.4000 \times 10^{-4}}{\frac{1}{3.0600 \times 10^{-3}} * 3.0600 \times 10^{-3} + \frac{1}{0.012300} * 0.012300 + \frac{1}{2.2680 \times 10^{-4}} * 2.2680 \times 10^{-4}}$ $= 2.582 \times 10^{-3}$

Question5: Repeat the previous exercise but using the k-best estimation algorithm with $k = 2$. You can use the hyp toolkit. Compute just one iteration.

1 S → Suj PRed	$D = \{la\ vieja\ demanda\ ayuda, la\ mujer\ oculta\ pelea, \}$ $la\ vieja\ mujer\ oculta\ demanda\ ayuda\} =$
0.3 Suj → Art Nom	
0.2 Suj → Art Adj Nom Adj	Después de analizar la tercera frase se comprueba que no existe regla para las categorías gramaticales que forman el sujeto:
0.3 Suj → Art Adj Nom	
0.2 Suj → Art Nom Adj	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> Sujeto <u>“La vieja mujer oculta demanda ayuda”</u> Art Adj Nom Adj </div> <div style="text-align: center;"> Predicado Verb Nom </div> </div>
0.3 Pred → Verb Nom	
0.7 Pred → Verb	Se decide crear una nueva regla (Suj → Art Adj Nomb Ajd) Y se coge parte la probabilidad de la regla (Suj → Art Nom) donde se define una probabilidad de 0.2 para la nueva regla y un 0.3 para la original.
1.0 Art → “la”	
0.1 Nom → “vieja”	
0.2 Nom → “ayuda”	Se calculan las probabilidades de cada frase en su respectivo árbol obteniendo los siguientes resultados:
0.3 Nom → “mujer”	
0.2 Nom → “pelea”	$P_{\theta}(La\ vieja\ demanda\ ayuda) =$ $= (1 * 0.1 * 0.3 * 0.2 * 0.3 * 0.3) +$ $(1 * 0.3 * 0.2 * 0.2 * 0.3 * 0.7) = 3.0600 \times 10^{-3}$
0.2 Nom → “demanda”	
0.3 Verb → “demanda”	$P_{\theta}(La\ mujer\ oculta\ pelea) =$ $= (1 * 0.3 * 0.1 * 0.2 * 0.3 * 0.3) +$ $(1 * 0.3 * 0.7 * 0.4 * 0.2 * 0.7) = 0.012300$
0.2 Verb → “ayuda”	
0.1 Verb → “oculta”	
0.4 Verb → “pelea”	
0.3 Adj → “vieja”	$P_{\theta}(La\ vieja\ mujer\ oculta\ demanda\ ayuda) =$ $= (1 * 0.3 * 0.3 * 0.7 * 0.3 * 0.2 * 0.2 * 0.3) = 2.2680 \times 10^{-4}$
0.7 Adj → “oculta”	$\rho'(Suj \rightarrow Art\ Nom)$ $= \frac{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj \rightarrow Art\ Nom, t_x) * P_{\theta}(x, t_x)}{\sum_{x \in D} \frac{1}{P_{\theta}(x)} * \sum_{t_x} N(Suj, t_x) * P_{\theta}(x, t_x)}$ $= \frac{\frac{1}{3.0600 \times 10^{-3}} * 5.4000 \times 10^{-4} + \frac{1}{0.012300} * 5.4000 \times 10^{-4}}{\frac{1}{3.0600 \times 10^{-3}} * 3.0600 \times 10^{-3} + \frac{1}{0.012300} * 0.012300 + \frac{1}{2.2680 \times 10^{-4}} * 2.2680 \times 10^{-4}}$ $= 2.582 \times 10^{-3}$

Ejercicios prácticos:

Question7: Complete the following table

# non-terminal symbols	# rectangle triangles
5	29
10	63
15	61
20	84

Question8: Study the classification results depending on the algorithm used for training (Inside-Outside or Viterbi) and the type of samples (bracketed or not). Analyze the results.

Inside-Outside con brackets

	equi	isos	right	Err	Err%
equi	323	0	677	677	67,7
isos	208	84	788	916	91,8
right	106	50	844	156	15,6
Error: 1749/3000 = 58,30%					

Inside-Outside sin brackets

	equi	isos	right	Err	Err%
equi	783	217	0	217	21.7
isos	483	366	151	634	63.4
right	48	187	765	235	23.5
Error: 1086/3000 = 36.20%					

Viterbi con brackets

	equi	isos	right	Err	Err%
equi	77	843	80	923	92.23
isos	70	850	80	150	15.0
right	12	676	312	688	68.8
Error: 1761/3000 = 58.70%					

Viterbi sin brackets

	equi	isos	right	Err	Err%
equi	67	933	0	933	93.3
isos	171	612	217	388	38.8
right	55	372	573	427	42.7
Error: 1748/3000 = 58.27%					

Después de analizar los datos se observa que los modelos entrenados con el algoritmo Inside-Outside presentan un error menor a los modelos entrenados con Viterbi. Cuando los modelos que utilizan datos de entrenamiento sin corchetes demuestran un error menor. Los modelos entrenados con Inside-Outside demuestran un buen rendimiento para detectar triángulos equiláteros y right y en el caso de los modelos entrenados con viterbi destaca el rendimiento superior al 90% en la clasificación de triángulos equiláteros.

Question9: Study the classification results depending on the algorithm used for learning the PCFG and the size of the training data. You can generate more training data as follows: `scfg-toolkit/genFig -F 0 -c 1000 -l 2 -L 10 > DATA/Tr-rightt`
You have to use a different seed for each training set in order to avoid a repeated sequence of training samples. Analyze the results.

Se observa que al utilizar muchos datos ralentiza el experimento, por ello se decide primero ejecutar el algoritmo viterbi con una pequeña cantidad de datos para poder ser comparada con el algoritmo de inside-outside y una gran cantidad de datos para ver una evolución más avanzada, dado que viterbi es el algoritmo que más rápido se ejecuta, obteniendo los siguientes resultados:

Viterbi, 50

	equi	isos	right	Err	Err%
equi	73	129	798	927	92,7
isos	111	147	742	853	85,3
right	85	70	845	155	15,5
Error: 1935 / 3000 = 64,50					

Viterbi, 100

	equi	isos	right	Err	Err%
equi	67	933	0	933	93,3
isos	195	580	225	420	42,0
right	36	476	488	488	51,2
Error: 1865/3000 = 62,17%					

Viterbi, 200

	equi	isos	right	Err	Err%
equi	144	380	476	856	85,6
isos	233	448	319	552	55,2
right	129	200	671	329	32,9
Error: 1737 / 3000 = 57,90%					

Viterbi, 3000

	equi	isos	right	Err	Err%
equi	67	230	703	933	93,3
isos	172	353	476	647	64,7
right	88	210	702	298	28,8
Error: 1878/3000 = 62,60%					

Viterbi, 6000

	equi	isos	right	Err	Err%
equi	0	0	1000	1000	100,0
isos	73	71	856	929	92,9
right	0	0	1000	0	0,0
Error: 1929/3000 = 64,30%					

Viterbi, 9000

	equi	isos	right	Err	Err%
equi	0	0	1000	1000	100,0
isos	0	0	1000	1000	100,0
right	0	6	994	6	0,6
Error: 2006/3000 = 66,87%					

Se observa que a un número concreto de datos de entrenamiento se consigue el error mínimo pero luego sigue una tendencia ascendente del error cuantos más datos se utilizan para el entrenamiento, produciéndose un sobreentrenamiento, en el cual el modelo se especializa en detectar triángulos “Right” en detrimento del resto de triángulos.

Para terminar se prueba el algoritmo Inside-Outside un número menor de datos para poder realizar los cálculos rápidamente, dada la lentitud observada en el algoritmo Inside-Outside, obteniendo los siguientes resultados:

Inside-Outside, 50

	equi	isos	right	Err	Err%
equi	387	613	0	613	61,3
isos	302	583	115	417	41,7
right	82	144	774	226	22,6
Error: 1256/3000 = 41,87%					

Inside-Outside, 100

	equi	isos	right	Err	Err%
equi	721	279	0	279	27,9
isos	473	413	114	587	58,7
right	50	217	733	267	26,7
Error: 1133/3000= 37,77					

Inside-Outside, 200

	equi	isos	right	Err	Err%
equi	927	73	0	73	7,3
isos	665	204	131	796	79,3
right	69	153	778	222	22,2
Error: 1091/3000 = 36,37%					

Con el algoritmo Inside-Outside, con los mismos datos que viterbi se observa un comportamiento parecido al experimento de viterbi pero con un error menor de clasificación en general, detectando con un error mínimo los triángulos equiláteros y right, pero empeorando con los triángulos isósceles.

	equi	isos	right	Err	Err%
equi	783	217	0	217	21.7
isos	483	366	151	634	63.4
right	48	187	765	235	23.5
Error: 1086/3000 = 36.20%					

Si se observa el resultado del algoritmo Inside-Outside del ejercicio anterior, sin brackets, con 1000 datos de entrenamiento la tendencia del entrenamiento demuestra un decremento de su error, pero no se cuentan con datos concluyentes en la memoria para estudiar la evolución del algoritmo debido al tiempo que tarda en realizar los cálculos para una cantidad de datos mayor.