

# Statistical structured prediction

Question set (Part 1-A)

Adrián Vázquez Barrera

Polytechnic University of Valencia

**November 2021**

## Question 1

Para un lenguaje generado por una gramática probabilística, deben cumplirse tres condiciones para ser definido como lenguaje probabilístico:

- La probabilidad  $\phi(x)$  debe ser cero si  $x$  no está definida en el lenguaje.
- La probabilidad de los elementos definidos debe estar en el intervalo  $(0, 1]$ .
- La suma de todas las probabilidades debe ser 1.

Hay gramáticas probabilísticas para las que, sus probabilidades de sus árboles de derivación no suman uno, lo que las hace inconsistentes. Si un lenguaje se produce utilizando una gramática inconsistente, la tercera condición no se cumple.

Hay lenguajes probabilísticos  $(L, \phi)$  que no pueden ser generados por una gramática probabilística  $G_\theta = (G, P)$ .

Siguiendo el ejemplo visto en clase:

$$\varphi(x) = \frac{1}{en!}$$

La producción del lenguaje es exponencial. Una gramática está definida por un conjunto de reglas, lo que significa que el grado de las producciones es polinómico, aunque el número de combinaciones pueda ser infinito. Por lo tanto, no está garantizado que a cada  $\phi(x)$  se le pueda asignar correctamente un  $P_\theta(x)$ .

## Question 2

Three-augmented Naive Bayes classifier:

$$P(x_1^T) = P(y) \cdot P(x_1|y) \cdot P(x_2|x_1, y) \cdot P(x_3|x_1, y) \cdot P(x_4|x_3, y)$$

Trigram Hidden Markov Model:

$$P(x_1^T, y_1^T) = \prod_{t=1}^T q(y_t|y_{t-2}, y_{t-1}) \cdot e(x_t|y_t) = \\ (y_1) \cdot e(x_1|y_1) \cdot q(y_2|y_1) \cdot e(x_2|y_2) \cdot (y_3|y_1, y_2) \cdot e(x_3|y_3) \cdot \dots (y_T|y_{T-2}, y_{T-1}) \cdot e(x_T|y_T)$$

## Question 3

**Definición:**

$$\beta_t(s) := \sum_{y_1^t; y_t=s} \prod_{i=t+1}^T \Psi_i(y_{i-2}, y_i, x_i)$$

**Inicialización:**

$$\beta_T(s) = 1$$

**Recursión:**

$$\beta_t(s) = \sum_{s' \in y} \Psi_{t+1}(y_{t-1} = s, y_{t+1} = s', x_{t+1}) \cdot \beta_{t+1}(s')$$

**Resultado final:**

$$\beta(s) = \sum_{s \in y} \Psi_1(y_{-1} = \#, y_1 = s, x_1) \cdot \beta_1(s)$$

## Question 4

La fórmula recursiva original del algoritmo inside se define como:

$$e(A, i, i + l) = \sum_{B, C \in N} p(A \rightarrow BC) \sum_{k=1, \dots, l-1} e(A, i, i + k) \cdot e(A, i + k, i + l)$$

Sabiendo que se trata de una "right-branching grammar", podemos establecer como constante el punto de corte  $k$  igual a 1.

**Definición:**

$$e(A, i, T) := P_{\theta}(A \xRightarrow{*} x_{i+1} \dots x_T)$$

**Inicialización:**

$$e(A, i, i + 1) = p(B \rightarrow b) \cdot \delta(b, x_{i+1})$$

**Recursión:**

$$e(A, i, T) = \sum_{B, C \in N} p(A \rightarrow BC) \cdot p(B \rightarrow b) \cdot \delta(b, x_{i+1}) \cdot e(A, i + 1, T)$$

**Resultado final:**

$$P_{\theta}(x) = e(S, 0, T)$$

Al hacer sólo una llamada recursiva a la vez, el árbol de derivación generado se asemeja a un árbol binario desbalanceado. Por lo que podemos afirmar de que el coste computacional del algoritmo inside para las right-branching grammars es lineal.

## Question 5

**Definición:**

$$\alpha_t(s) := \sum_{y_1^t; y_t=s} \prod_{i=1}^T \Psi(y_{i-2}, y_{i-1}, y_i, x_i)$$

**Inicialización:**

$$\alpha_1(s) = \Psi(y_{-1}, y_0, y_1 = s, x_1)$$

**Recursión:**

$$\alpha_t(s) = \sum_{s' \in y} \alpha_{t-1}(s') \sum_{s'' \in y} \Psi(y_{t-2} = s'', y_{t-1} = s', y_t = s, x_t)$$

**Resultado final:**

$$\alpha_t(s) = \sum_s \alpha_t(s)$$

Al tener en cuenta ahora todos los dos posibles estados anteriores, se ha modificado la recursión en base al nuevo requerimiento. En este caso  $s'$  representa el conjunto de estados inmediatamente anterior al estado actual y  $s''$  al conjunto de estados dos veces anterior.

## Practical assignments

Para todos los modelos, el que proporciona la menor perplejidad para cualquier corpus es G3. El modelo G1 es peor que el G3, sin embargo, el modelo G2 es, con diferencia, el peor de todos ellos, con una perplejidad muy superior a la del G3, especialmente para los triángulos equiláteros, donde observamos que la perplejidad es incluso dos órdenes de magnitud superior a la del G3.

Statistical evaluation (Perplexity)			
	EQ	IS	SC
G1	99095.20	26581.81	33618.35
G2	635136.42	52217.60	43543.30
G3	986.21	1254.90	1218.04

### Classification

Se puede observar como efectivamente el modelo G3 ofrece el menor error, sin embargo, los tres modelos fallan bastante en la clasificación global. Cabe destacar que el modelo G2 es capaz de clasificar los triángulos equiláteros con el menor error, lo mismo ocurre con el modelo G3 y los triángulos equiláteros. Los triángulos isósceles son el talón de Aquiles común a todos los modelos.

G1					
	EQ	IS	SC	Error	Error (%)
EQ	597	285	118	403	40.3%
IS	88	471	441	529	52.9%
SC	71	406	523	477	47.7%
Error	46.97%				

G2					
	EQ	IS	SC	Error	Error (%)
EQ	281	211	508	719	71.9%
IS	71	215	714	785	78.5%
SC	81	190	729	271	27.1%
Error	59.17%				

G3					
	EQ	IS	SC	Error	Error (%)
EQ	789	102	109	211	21.1%
IS	178	512	310	488	48.8%
SC	7106	421	473	527	52.7%
Error	40.87%				