Introducción al Procesamiento de Lenguaje Natural

Grupo de PLN - InCo

Métodos de Clasificación Supervisada

Clasificación

- Reconocer entidades agrupándolas en un conjunto discreto de clases

- Básicamente, asignar una categoría de un conjunto discreto de valores a un objeto
  - calificar una película como buena o mala
  - calificar un mail como spam
  - reconocer transacciones fraudulentas
  - reconocer comentarios positivos, negativos o neutros de películas, autos , computadoras, etc.
  -

Clasificación

- reconocer el idioma de un texto
- asignar tópicos a documentos
- dar valores a signos de puntuación
- ...

En nuestro caso, los objetos son textos

Métodos de Clasificación

- Algoritmos específicos *ad hoc*

- Reglas manuales
  - Es usual usar expresiones regulares para reconocer *patterns* semifijos en el texto
    - fechas, dinero, ...

Métodos de Clasificación

Aprendizaje automático

- no supervisado
  - *clustering* , ej., armar grupos en redes sociales, encontrar *bots*
  - LDA (*Latent Dirichlet Allocation)* Encontrar tópicos en documentos

- supervisado
  - ejemplos con el valor de la clase anotado
    - spam
    - análisis de sentimientos ...

Clasificación supervisada

Dada una instancia **x**

un conjunto de clases de salida

$Y = y_1 , y_2 , \ldots, y_n$

y un conjunto de entrenamiento

(d_1 , y_1) … (d_n , y_n)

retornar una clase **y ε Y** para la instancia **x**

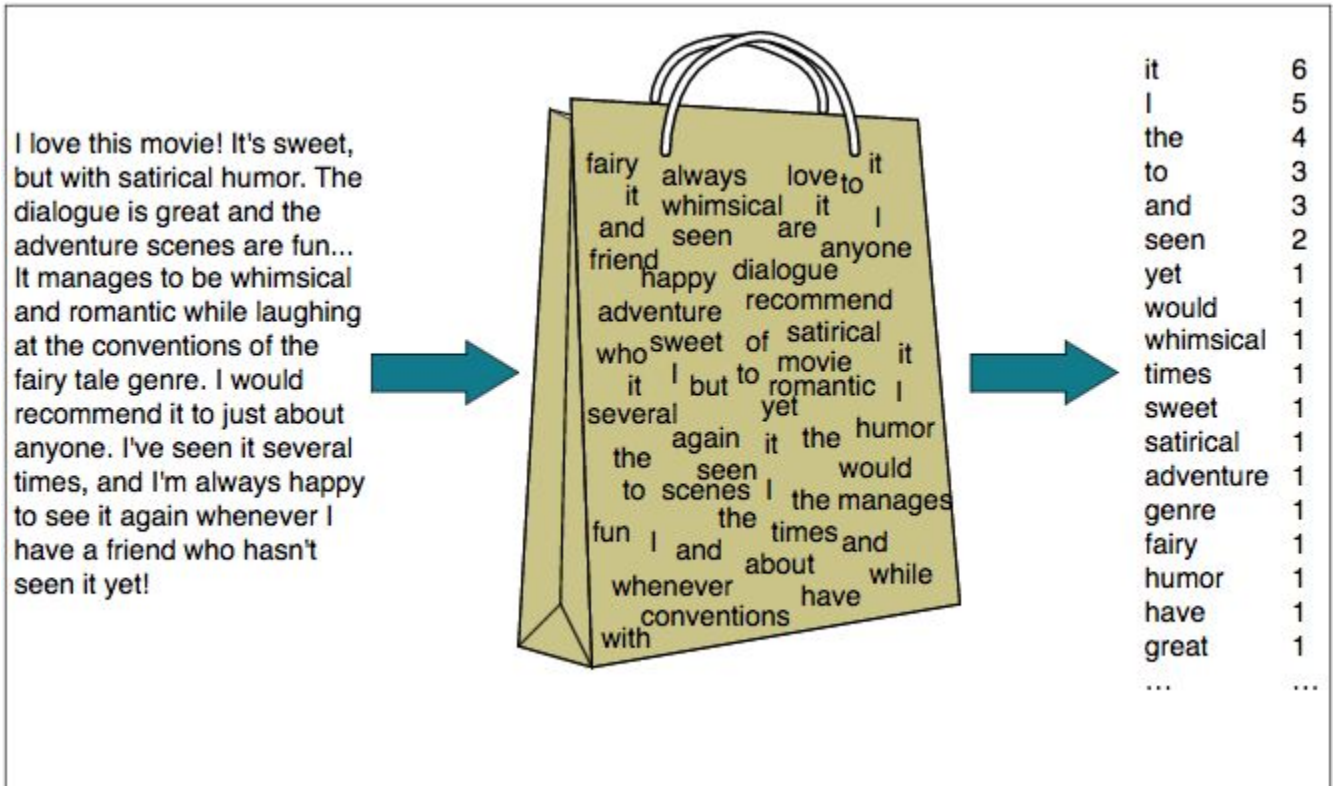
Métodos probabilistas

Para clasificar, los métodos probabilistas intentan obtener una *distribución de probabilidad* sobre las clases dados los atributos de cada cada instancia, es decir:

$$P(C|X_1, X_2, \ldots X_n)$$

Con esto, clasificar consiste en elegir la clase con probabilidad más alta

Clasificación de documentos



Métodos generativos

Intentan modelar la distribución de probabilidad conjunta $P(c, x_1, x_2, \ldots, x_n)$ de los atributos y las etiquetas. ¿Por qué?

$$P(C|X_1, X_2, \ldots X_n) = \frac{P(C, X_1, \ldots, X_n)}{P(X_1, \ldots, X_n)} = \frac{P(C)P(X_1, \ldots, X_n|C)}{P(X_1, \ldots, X_n)}$$

Tanto la probabilidad a priori P(c) como la probabilidad de verosimilitud (*likelihood*) pueden estimarse fácilmente a partir de los datos... siempre que simplifiquemos el problema

Naïve Bayes

Supone independencia entre los atributos

$$\frac{P(C, X_1, \ldots, X_n)}{P(X_1, \ldots, X_n)} = \frac{P(C)P(X_1, \ldots, X_n|C)}{P(X_1, \ldots, X_n)} = \frac{P(C)P(X_1|C)P(X_2|C)\ldots P(X_n|C)}{P(X_1, \ldots, X_n)}$$

Estimo las probabilidades utilizando máxima verosimilitud en el corpus de entrenamiento

¿Cómo construye un clasificador a partir de la distribución?

- La clase que maximiza la probabilidad (observar que $P(x_1, x_2, \ldots, x_n)$ no importa, no depende de la clase).

- Funciona *sorprendentemente* bien.

Clasificación de documentos

positions $\leftarrow$ all word positions in test document

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c)$$

Estimo las probabilidades utilizando máxima verosimilitud en el corpus de entrenamiento

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

$$\hat{P}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

Clasificación de documentos

$$\text{positions} \leftarrow \text{all word positions in test document}$$

$$c_{NB} = \underset{c \in C}{\text{argmax}}\, P(c) \prod_{i \in positions} P(w_i|c)$$

Estimo las probabilidades utilizando máxima verosimilitud en el corpus de entrenamiento

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

$$\hat{P}(w_i|c) = \frac{count(w_i,c)+1}{\sum_{w \in V}(count(w,c)+1)} = \frac{count(w_i,c)+1}{\left(\sum_{w \in V} count(w,c)\right)+|V|}$$

Clasificación de documentos

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“with”}|-) = \frac{0+1}{14+20} \qquad P(\text{“with”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“originality”}|-) = \frac{0+1}{14+20} \qquad P(\text{“originality”}|+) = \frac{0+1}{9+20}$$

Clasificación de documentos

Si S= "predictable with no originality"

$$P(S|-)P(-) = \frac{3}{5} \times \frac{2 \times 1 \times 2 \times 1}{34^4} = 1.8 \times 10^{-6}$$

$$P(S|+)P(+) = \frac{2}{5} \times \frac{1 \times 1 \times 1 \times 1}{29^4} = 5.7 \times 10^{-7}$$

Sentiment Analysis

Igual que antes, pero...

- No contar múltiples ocurrencias de una palabra en un mismo documento (Binary Naive Bayes)

- Manejo de la negación: analizar "didnt like this movie, but" como "didnt NOT_like NOT_this NOT_movie, but"

- Usar lexicones de sentimiento y usar como features la cantidad de palabras en un lexicon positivo y la cantidad de palabras en un lexicon negativo

POS tagging

Ejemplo: calcular POS tag, si tengo la palabra, y los POS tags de las palabras anterior y siguiente en el contexto

$$P(C = Adj|X_{ant} = Det, X_{sig} = Nom, X_s = blanco) = P(C = Adj)P(X_s = blanco|C = Adj)P(X_{ant} = Det|C = Adj)P(X_{sig} = Nom|C = Adj)$$

Métodos discriminativos

- Modelan directamente la dependencia de la clase de los atributos, sin intentar modelar la relación entre los atributos
- En un marco probabilístico, se modela la probabilidad condicional $P(c| x_1,x_2,..., x_n)$
- Diferentes aproximaciones, algunas probabilísticas (MaxEnt), otras no (Averaged Perceptron, Support Vector Machines)

Modelos de Entropía Máxima

También conocidos como Regresión multinomial logística, son modelos log-lineales para clasificación

$$P(C|X_1 \ldots X_n) = \frac{1}{Z} e^{\Sigma\, w_i \cdot f_i}$$

Modelos de Entropía Máxima

Regresión lineal: suponemos que el objetivo es real, y una combinación lineal de los atributos

$$y = \sum w_i \times f_i = w \cdot f$$

Buscamos el w que minimiza la suma de cuadrados de las diferencias entre valores y predicciones, por métodos analíticos o numéricos (por ejemplo, descenso por gradiente)

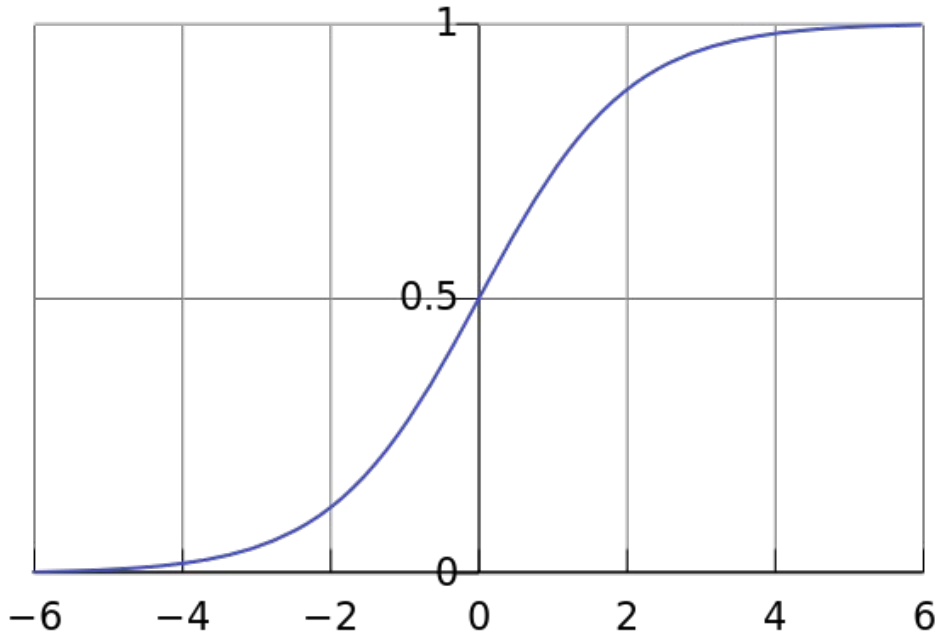
Modelos de Entropía Máxima

$$P(Y = true|x) = \frac{e^{w \cdot f}}{1 + e^{w \cdot f}} = \frac{1}{1 + e^{-w \cdot f}}$$

Modelos de Entropía Máxima

- Función logística

$$\frac{1}{1+e^{-t}}$$



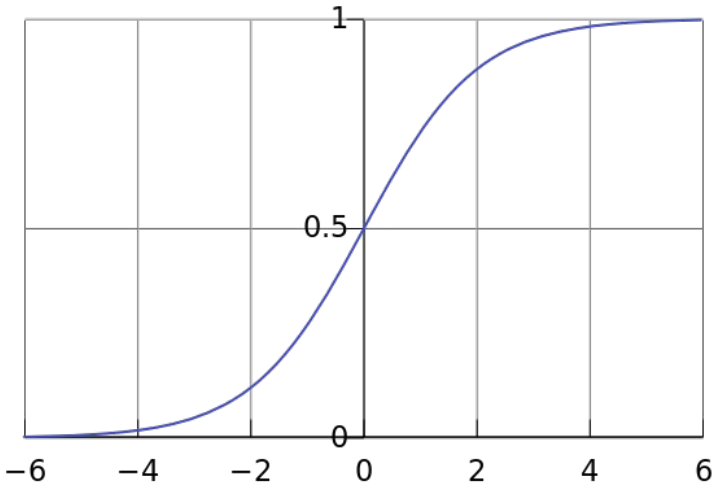
Modelos de Entropía Máxima

- ¿Cómo clasificamos?

Si P(y=true|x) > P(y=false|x)

Entonces exp(w·f) > 1

Entonces w·f > 0!



La función logística es como una versión *smooth* de una función que vale 1 si w·f > 0 y vale 1 si no.

Modelos de Entropía Máxima

- ¿Cómo estimamos los pesos? Conditional Maximum Likelihood Estimation

$$\hat{w} = \underset{w}{\operatorname{argmax}} \prod_i P(y^{(i)}|x^{(i)})$$

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_i y^{(i)} \log \frac{e^{-w \cdot f}}{1 + e^{-w \cdot f}} + (1 - y^{(i)}) \log \frac{1}{1 + e^{-w \cdot f}}$$

Buscamos el máximo de una función convexa... existen diferentes métodos numéricos (descenso por gradiente, L-BFGS, gradiente conjugado...)

Modelos de Entropía Máxima

- ¿Y si hay más de dos clases? MaxEnt (Multinomial Logistic Regression)

$$p(c|x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i(c,x)\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i(c',x)\right)}$$

Observar que tenemos atributos indicadores que dependen de una observación y una clase

Nos quedamos con la clase que maximiza P(c|x)

Maximum Entropy Models

- ¿Por qué Maximum Entropy?
  - Si vas a elegir un modelo de un conjunto de distribuciones de probabilidad, elige el que tenga máxima entropía:
    - p* = argmax H(p)

$$H(x) = -\sum_{x} P(x) \log_2 P(x)$$

  - La solución a esto es una distribución de probabilidad para un modelo logístico multinomial, cuyos pesos maximizan la verosimilitud en los datos de entrenamiento!

Maximum Entropy Models

- Ejemplo: POS tagging

que/CONJ no/ADV sobre/?? comida/N

$f_1(c,x)$ = 1 si word(i)=”sobre” & c = N, 0 en otro caso

$f_2(c,x)$ = 1 si t(i-1)=ADV & c = V, 0 en otro caso

$f_3(c,x)$ = 1 si sufijo(word(i))=”ndo” & c = V, 0 en otro caso

$f_4(c,x)$ = 1 si es_minúscula(word(i)) & c = V, 0 en otro caso

$f_5(c,x)$ = 1 si word(i)=”sobre” & c = V, 0 en otro caso

$f_6(c,x)$ = 1 si t(i-1)=ADV & c = N, 0 en otro caso

Cada feature va a tener un peso, indicando qué tanto afecta la feature correspondiente para ese tag.

Maximum Entropy Models

- Ejemplo: POS tagging

que/CONJ no/ADV sobre/?? comida/N

Supongamos que w(f2,V)=0.8, w(f4,V)=0.01, w(f5,V)=0.1, w(f1,N)=0.8, w(f6,N)=-1.3

Entonces,

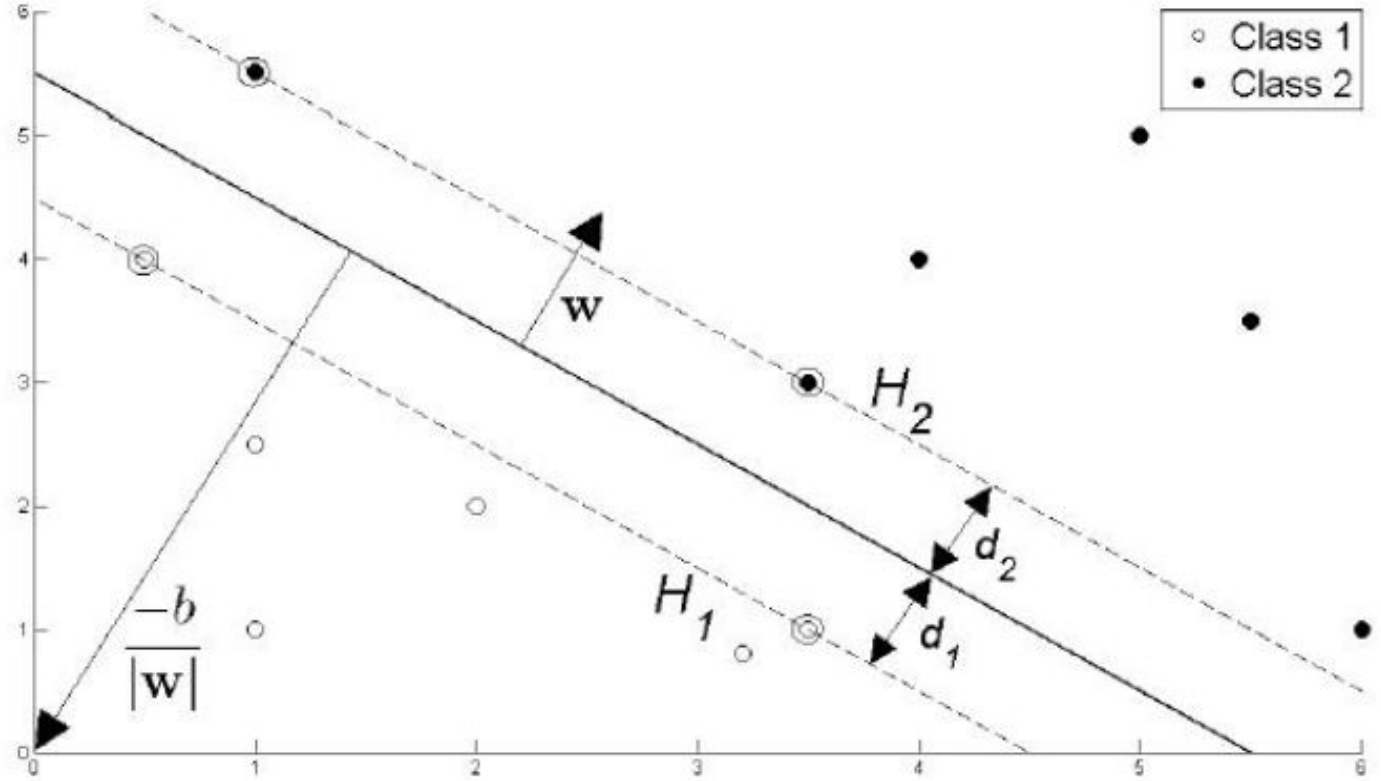
P(N|x)= exp(0.8)exp(-1.3)/Z = 0.20

P(V|x)= exp(0.8)exp(0.01)exp(0.1)/Z= 0.80

Z= exp(0.8)exp(-1.3) + exp(0.8)exp(0.01)exp(0.1)

Para clasificar, buscamos la clase cuya probabilidad estimada por el modelo es máxima

Support Vector Machines



Support Vector Machines

minimize $\frac{1}{2}||\vec{w}||^2$

sujeto a $y_i(\vec{x_i} \cdot \vec{w} - \rho) \geq 1 \ \forall i$

Resolver este problema de minimización con restricciones nos permite obtener w y ρ. Para resolverlo, se utilizan técnicas de programación cuadrática, utilizando el método de los multiplicadores de Lagrange.

Se puede ver que w es una combinación lineal de los ejemplos de entrenamiento... en particular, de los vectores de soporte.

Otros métodos de clasificación

- knn – Vecinos más cercanos
    - Basados en memoria
    - "Lazy"
    - Más costo de clasificación
- Árboles de decisión – No muy usados en PLN
- Random Forests – Combinación de clasificadores
- etc...

Clasificación secuencial

- En lugar de un objeto, tengo una secuencia de objetos, y a cada uno quiero clasificarlo.
    - La función de clasificación tiene secuencias como dominio y como codominio
- Ejemplos en PLN:
    - POS-tagging
    - Reconocimiento de Entidades con Nombre
    - Chunking
    - ... etc!

Clasificación secuencial

- Aproximación computacional al reconocimiento de *spans* de texto:
  - BIO:
    - El primer elemento del span tiene clase B
    - Los restantes en el span tienen clase I
    - El resto es O (Other)

Clasificación secuencial

- Aproximación computacional al reconocimiento de *spans* de texto:
  - BIO (NER– CoNLL 2002):
    - Wolff /B-PER ,/O currently/O a/O journalist/O in/O Argentina/B-LOC ,/O played/O with/O ...

Clasificación secuencial

- Aproximación computacional al reconocimiento de *spans* de texto:
  - BIO (Hedge Cues – CoNLL 2010):

Token	Surface Form	Hedge
1	Cotransfection	O
2	studies	O
3	with	O
4	this	O
5	cDNA	O
6	indicate	B
7	that	I
8	it	O
9	can	B
10	repress	O
11	basal	O
12	promoter	O
13	activity	O
14	.	O

Clasificación secuencial

- Aproximación computacional al reconocimiento de *spans* de texto:
  - FOL(Alcance de Hedge Cues – CoNLL 2010):

Token	Word	Lemma	POS	Hedge	Scope
1	This	This	DT	O	O
2	finding	finding	NN	O	O
3	suggests	suggest	VBZ	O	O
4	that	that	IN	O	O
5	the	the	DT	O	F
6	BZLF1	BZLF1	NN	O	O
7	promoter	promoter	NN	O	O
8	may	may	MD	B	O
9	be	be	VB	O	O
10	regulated	regulate	VBN	O	O
11	by	by	IN	O	O
12	the	the	DT	O	O
13	degree	degree	NN	O	O
14	of	of	IN	O	O
15	squamous	squamous	JJ	O	O
16	differentiation	differentiation	NN	O	L
17	.	.	.	O	O

Clasificación secuencial

- Medidas de Evaluación
  - Las medidas son las mismas, pero...
    - ... ¿qué se considera un TP?
    - En general, se asume *exact match*, aunque hay variantes.

Hidden Markov Models

Generativo: Hidden Markov Models

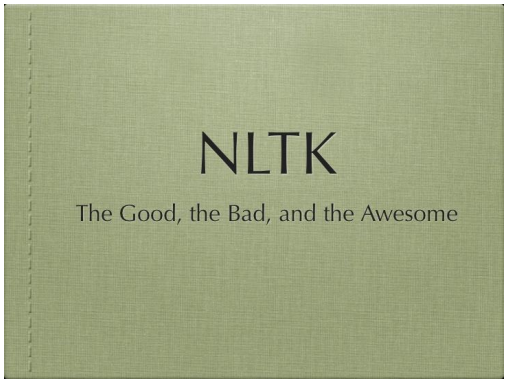
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

Conditional Random Fields

- Es un algoritmo *discriminativo* para clasificación secuencial
- Calcula directamente una estimación de P(y|x):

$$P(y|x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_i f_i(y_{j-1}, y_j, x, j)\right)$$

Software



CRF++: Yet Another CRF toolkit

SVM*light*

Support Vector Machine

TensorFlow

Referencias

- Martin & Jurafsky, Capítulo 7: Classification: Naive Bayes, Logistic Regression, Sentiment
- Manning - Information Retrieval - Capítulo 13 - Text Classification and Naive Bayes
- Abney - Semi Supervised Learning - Capítulo 4 - Classification