Clasificación de textos

ITALICA
Universidad de Sevilla
José A. Troyano

Índice

Introducción

Definición general

Clasificación (o Categorización): Asignar objetos de un universo a dos o más clases o categorías predefinidas.

Muchas de las tareas del PLN pueden ser consideradas, en última instancia, como procesos de clasificación:

Problema	Objeto	Categorías
etiquetado	contexto de una palabra	etiquetas gramaticales
desambiguación	contexto de una palabra	etiquetas semánticas
análisis sintáctico	frase	árboles sintácticos
identificación de autores	documento	autores
identificación de lenguaje	documento	lenguaje
clasificación de textos	documento	temáticas

Introducción

Elementos necesarios para clasificar

Conjunto de entrenamiento: Objetos ya clasificados de los que se pretende extraer regularidades (conocimiento).

Modelo de representación: Sistema que permite codificar los datos de entrenamiento. Por ejemplo:

$$(x,c)$$

donde x es un vector de medidas y c es la etiqueta asignada.

Modelo de clasificación: Un tipo parametrizado de clasificadores. Por ejemplo:

$g(x)= w.x + w_0$ (si $g(x)>0$ se elige c_1, si no se elige c_2)

Entrenamiento: Proceso que estima los parámetros del modelo de clasificación (en el caso anterior w y w_0).

Introducción

Evaluación del clasificador (I)

Conjunto de prueba: Objetos ya clasificados pero no utilizados en el entrenamiento.

Para clasificadores binarios, se define una tabla de contingencias que recoge todos los casos posibles. Supongamos dos categorías A y B:

- A_c: Objetos clasificados en A correctamente
- B_c: Objetos clasificados en B correctamente
- A_i: Objetos clasificados en A incorrectamente
- B_i: Objetos clasificados en B incorrectamente

La medida obvia para la corrección es:

$$correccion = (A_c+B_c) / (A_c+B_c+A_i+B_i)$$

Introducción

Evaluación del clasificador (II)

Si una de las categorías es más interesante, por ejemplo en el caso de la recuperación de documentos en internet, los clasificamos en relevantes (A) e irrelevantes (B):

$$certeza = A_c / (A_c+A_i)$$
$$cobertura = A_c / (A_c+B_i)$$

En el caso de más de dos categorías, se construye una tabla de contingencias (frecuencias) para cada una de ellas (ci frente a ¬ci), a partir de ellas se pueden hacer dos cosas:

Macro-ponderar: Calcular la medida (p.e. certeza) para cada tabla y extraer la media. Da el mismo peso a cada categoría.

Micro-ponderar: Construir una tabla unificada, sumando las frecuencias, y luego calcular la medida. Da el mismo peso a cada objeto.

Introducción

La colección Reuters

Es la base de datos más popular para la evaluación de los clasificadores de textos.

Se trata de una colección de artículos de la agencia de noticias Reuters del año 1987.

- 9603 artículos de entrenamiento
- 3299 artículos de prueba
- 100 categorías (fusiones y adquisiciones, tipos de interés, ganancias, ...)

Introducción

Un artículo de Reuters (categoría ganancias)

```
<REUTERS NEWID="11">
<DATE>26-FEB-1987 15:18:59.34</DATE>
<TOPICS><D>earn</D></TOPICS>
<TEXT>
<TITLE>COBANCO INC &lt;CBCO> YEAR NET</TITLE>
<DATELINE>    SANTA CRUZ, Calif., Feb 26 - </DATELINE>
<BODY> Shr 34 cts vs 1.19 dlrs
     Net 807,000 vs 2,858,000
     Assets 510.2 mln vs 479.7 mln
     Deposits 472.3 mln vs 440.3 mln
     Loans 299.2 mln vs 327.2 mln
     Note: 4th qtr not avaliable. Year includes 1985
extraordinary gain from tax carry forward of 132,000 dlrs,
or five cts per shr.
  Reuter
</BODY></TEXT>
</REUTERS>
```

Introducción

El modelo de representación (I)

Por ejemplo para construir un clasificador para una categoría determinada, podemos representar cada documento D_j, como:

> "un vector x_j de medidas s_{ij} para las 20 palabras w_i que tengan el estadístico χ^2 (mide la dependencia) más alto para dicha categoría"

$$s_{ij} = \text{redondeo}\left(10x\frac{1+\log(tf_{ij})}{1+\log(l_j)}\right)$$

tf_{ij}= número de ocurrencias de la palabra w_i en el documento D_j

l_j= longitud del documento D_j

Introducción

El modelo de representación (II)

Para la categoría "ganancias" las 20 palabras más significativas son:

vs	mln	cts	;	&	000	loss	'	"	3
profit	dlrs	1	pct	is	s	that	net	lt	at

El vector del documento anterior sería:

(5,5,3,3,3,4,0,0,0,4,0,3,2,0,0,0,0,3,2,0)

La elección del modelo es ya en sí misma un problema, y va a influir de manera determinante en el resto del proceso.

Podemos optar entre una obtención manual y automática del modelo de representación. La obtención automática conlleva la elección de símbolos extraños como ; " ó &.

Introducción

Entropía: una medida de la incertidumbre

Dada una función de distribución de probabilidades p(x) asociada a una variable aleatoria X:

$$p(x) = P(X=x), \qquad x \in X$$

Se define la entropía de la siguiente forma:

$$H(p) = H(X) = \sum_{x \in X} p(x) \log_2(1/p(x))$$

La entropía mide la incertidumbre:

Quiniela de dos partidos

R. Madrid-Sp. Gijón	70%	20%	10%
Sevilla-Betis	50%	30%	20%

p(X)={0.35,0.21,0.14,0.10,0.06,0.04,0.05,0.03,0.02}

H=2.642

2^H=6.241

Dado de nueve caras

P(x)=1/9

H=3.169

2^H=9

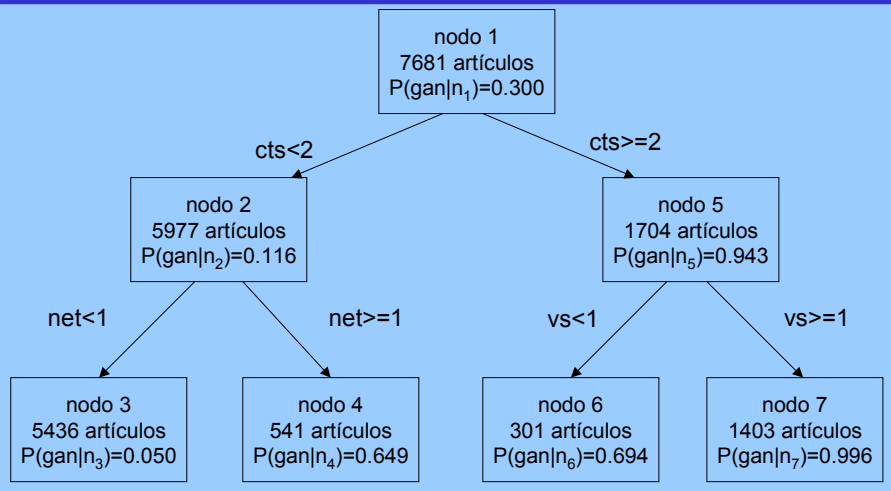
También puede ser interpretada como cantidad de información.

Índice

- Introducción
- **Árboles de decisión**
- Modelado de máxima entropía
- Perceptrones
- Vecinos más cercanos

Árboles de decisión

Un árbol de decisión para la categoría “ganancias”



Sólo hay que enfrentar el vector de un documento al árbol, para calcular la probabilidad de pertenecer a la categoría.

Árboles de decisión

La construcción del árbol (entrenamiento)

Criterio de división: Nos dice cuándo dividir un determinado nodo y la característica que provoca la división.

Criterio de parada: Nos dice cuando un nodo no puede ser dividido más. Por ejemplo cuando todos sus elementos tienen la misma representación o cuando pertenecen a la misma categoría.

Criterio de poda: Nos dice cuándo eliminar parte del árbol construido. Es útil cuando el árbol eliminado no recoge una “regularidad” importante.

Árboles de decisión

Criterio de división

Dividiremos un nodo de manera que la Ganancia de información sea máxima:

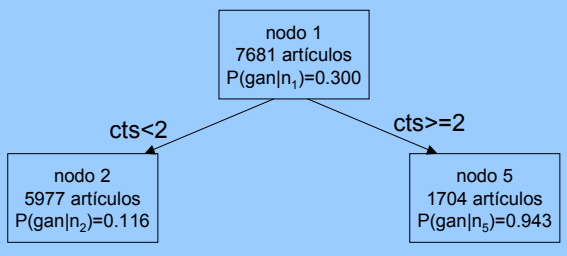
$$G(a,y) = H(t) - (p_L H(t_L) + p_R H(t_R))$$

Donde

- a es el atributo que sirve para dividir
- y es el valor de corte de dicho atributo
- t es la distribución de probabilidad del nodo padre
- p_L y p_R son las proporciones de elementos que pasan a los nodos izquierdo y derecho.
- t_L y t_R son las distribuciones de probabilidades de los nodos izquierdo y derecho

Árboles de decisión

Criterio de división: ejemplo



Entropía del nodo 1: 0.611

Entropía del nodo 2: 0.359

Entropía del nodo 5: 0.219

Suma ponderada de la entropía de los hijos: (5977/7681)*0.359 + (1704/7681)*0.219=0.328

Ganancia de entropía: 0.611-0.328 = 0.283

Cuanto mayor sea mejor ya que la subdivisión habrá mermado la incertidumbre en mayor medida.

Árboles de decisión

Criterio de poda

Una vez que se ha construido el árbol completo se procede a eliminar los nodos irrelevantes o incluso dañinos:

Repetir hasta que el árbol t quede vacío

n = Nodo menos relevante de t %con alguna medida

Eliminar n de t

Almacenar t en st % colección de todos los árboles

Fin repetir

Elegir el elemento de st más adecuado % mediante validación

Árboles de decisión

Conclusiones

Inconvenientes

- Son más complejos que otros clasificadores.
- Dividen el conjunto de entrenamiento en muchos subconjuntos pequeños lo que favorece la aparición de distorsiones

Ventajas

- Son fácilmente interpretables por una persona

Índice

- Introducción
- Árboles de decisión
- **Modelado de máxima entropía**
- Perceptrones
- Vecinos más cercanos

Modelado de máxima entropía

Motivación

• Puede integrar características complejas y heterogéneas (aplicable a conceptos lingüísticos)

• Cada característica se considera una restricción sobre el modelo.

• Una vez definidas las características, se calcula el modelo de máxima entropía.

Modelo incierto pero más fiel a los datos	+ ENTROPÍA -	Modelo con más información pero menos fiel a los datos

CONCLUSIÓN: Nos quedamos con la información “justa” que nos dan los datos. No generalizamos ninguna conclusión que pueda incluir información no contrastable “empíricamente”.

Modelado de máxima entropía

Elección de las características

• En nuestro caso este aspecto se ha obviado para simplificar la presentación de las ideas:

  Escogemos las mismas 20 características del modelo de representación anterior.

• Sin embargo, es quizá la cuestión más importante de este tipo de técnicas. (similar al etiquetado transformacional)

• En un modelado real, la elección de las características se hace al mismo tiempo que el entrenamiento.

• El propio proceso de entrenamiento nos dice si merece la pena, o no, mantener una determinada característica dentro del modelo.

Modelado de máxima entropía

Obtención del modelo

PASO 1: Las características f_i, son funciones binarias calculadas de la siguiente forma para un documento x_j:

$$f_i(x_j,c) = \begin{cases} 1 \text{ si } s_{ij} > 0 \text{ y } c=1 \\ 0 \text{ en otro caso} \end{cases}$$

PASO 2: Se calcula la *esperanza empírica* $E\,f_i$ (media) de cada una de las características en base al corpus de entrenamiento.

PASO 3: Para cada característica se define una restricción R_i que consiste en exigir que la esperanza de dicha característica en el modelo final sea igual a $E\,f_i$

PASO 4:De todas las distribuciones de probabilidad que satisfacen las restricciones R_i, se escoge el de mayor entropía.

Modelado de máxima entropía

Entrenamiento y criterio de clasificación

La función de distribución que define el modelo, y para la que pretendemos maximizar la entropía es:

$$p(x,c) = 1/Z \prod_{i=1..K} \alpha_i^{fi(x,c)}$$

- Z es una constante de normalización
- K es el número de características
- α_i es el peso para la característica f_i, precisamente el algoritmo de entrenamiento calcula estos pesos para que la entropía de p sea máxima.

Los pesos se calculan con el algoritmo *Generalized Iterative Scaling.*

Se decide que la categoría de *x* es *c* si $p(c|x) > p(\neg c|x)$

Índice

Perceptrones

Conceptos básicos

Descenso del gradiente:

- Se intenta minimizar la función de error.
- Es un algoritmo iterativo.
- En cada paso se corrigen los parámetros del modelo en la dirección del gradiente.

El modelo de clasificación (binario,*sí/no*):

sí si $w.x-\theta > 0$

no en otro caso

x es el vector de entrada (modelo de representación)

w es un vector de pesos

θes el umbral de decisión

Otros modelos: No binarios, redes neuronales

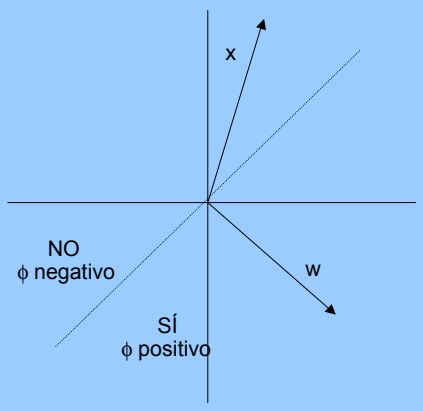
Perceptrones

Algoritmo de clasificación

```
Comentario: Clasificación binaria (sí/no)
Func Decisión(x,w,θ)
  Si w.x- θ>0 entonces
    Decisión = sí
  si no
    Decisión = no
  Fin si
Fin func
```

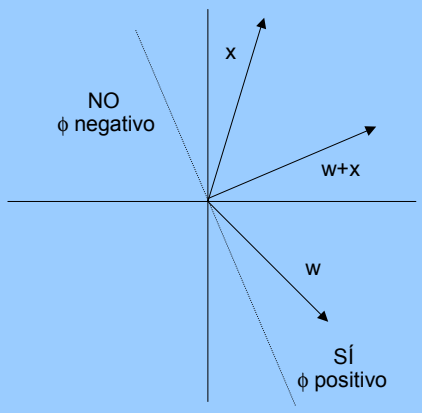
Perceptrones

Corrección del vector de pesos: suma



Hay un error: sabemos que x debería encuadrarse en la categoría SÍ.

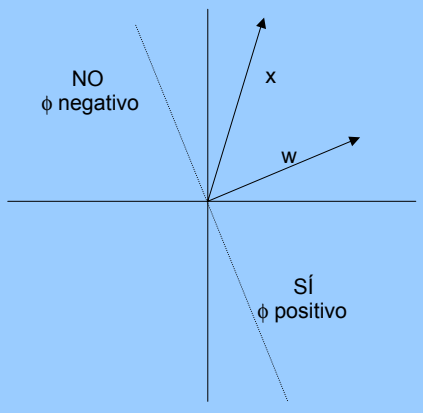
x debería quedar en el lado positivo. Por tanto para corregir hay que sumar.



El error se corrige sustituyendo w por w+ x.

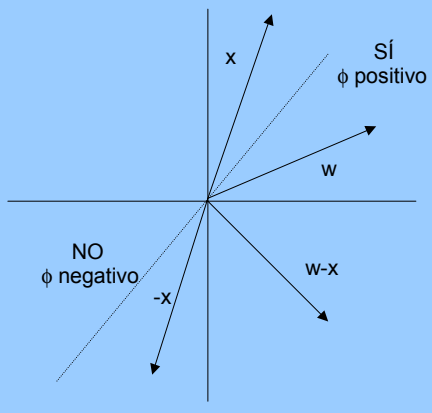
Perceptrones

Corrección del vector de pesos: resta



Hay un error: sabemos que x debería encuadrarse en la categoría NO.

x debería quedar en el lado negativo. Por tanto para corregir hay que restar.



El error se corrige sustituyendo w por w-x.

Perceptrones

Corrección y aprendizaje

Integramos θ en el producto escalar a través de la función ϕ:

$$\phi(w') = w'.x' = (w_1,...w_k,\theta).(x_1,...,x_k,-1)$$

El objetivo del algoritmo de aprendizaje es encontrar w y θ.
Cada vez que detectemos un error (sobre el corpus clasificado) lo corregiremos así:

Si la clase de x es **sí** y $\phi<0$ (el modelo dice **no**)

$w'=w'+x'$

Si la clase de x es **no** y $\phi>0$ (el modelo dice **sí**)

$w'=w'-x'$

En realidad estamos corrigiendo según el gradiente de ϕ:

$$\nabla\phi(w')=x'$$

que es el vector que provoca el mayor cambio en ϕ

Perceptrones

Algoritmo de aprendizaje

Comentario: Inicialización
w=(0,...,0)
θ = 0
Comentario: Aprendizaje
Mientras no haya convergencia **hacer**
  Para todos los elementos x_j del corpus **hacer**
    d= Decisión(x_j,w, θ)
    Si (clase(x_j)=d) **entonces** continuar
      | (clase(x_j)=sí y d=no) **entonces** w'=w'+x'
      | (clase(x_j)=no y d=sí) **entonces** w'=w'-x'
    Fin si
  Fin para
Fin mientras

Se considerará que hay convergencia cuando un porcentaje de los documentos estén clasificados correctamente.

El 100% sólo se podrá alcanzar si el problema es *linealmente separable* (por un hiperplano).

Índice

- Introducción
- Árboles de decisión
- Modelado de máxima entropía
- Perceptrones
- **Vecinos más cercanos**

Vecinos más cercanos

Conceptos básicos

• Para clasificar un nuevo objeto, buscamos el objeto (ya clasificado) más cercano (según alguna medida de distancia) y le asignamos la misma categoría.

• Una generalización obvia es utilizar k vecinos, en este caso la clasificación es más robusta.

• Uno de los puntos más críticos de esta técnica es la elección de la medida.

• En PLN hay varias medidas basadas en vectores que pueden servir. Por ejemplo la medida del coseno:

$$\frac{X \cap Y}{\text{raiz}(|X|.|Y|)}$$ %concordancia (p.e. no-cero)

Vecinos más cercanos

Algoritmo

Comentario: Clasifica *y* en dos clases *A* y *B* en base al corpus de entrenamiento *X* aplicando la técnica de 1-vecino

DistanciaMinima(y) = max $_{x \in X}$ sim(x,y) %similaridad máxima

SIM_A={x $\in$ SIM| clase(x)=A $\wedge$ sim(x,y) = DistanciaMinima(y)}

SIM_B={x $\in$ SIM| clase(x)=B $\wedge$ sim(x,y) = DistanciaMinima(y)}

P(A|y)= $|SIM_A| / (|SIM_A| + |SIM_B|)$

P(B|y)= $|SIM_B| / (|SIM_A| + |SIM_B|)$

Si P(A|y)>P(B|y) **entonces**

  clase(y) = A

si no

  clase(y)= B

Fin si