



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica



Tema 2. Representación de objetos

Percepción (PER)

Curso 2021/2022

Departamento de Sistemas Informáticos y Computación

Índice

- 1 Introducción ▷ 3
- 2 Representación de imágenes ▷ 7
- 3 Representación de voz ▷ 29
- 4 Representación de texto ▷ 41

Índice

- 1 *Introducción* ▷ 3
- 2 Representación de imágenes ▷ 7
- 3 Representación de voz ▷ 29
- 4 Representación de texto ▷ 41

Extracción de características

- Captura y representa aquella información *discriminativa* del objeto a clasificar:
 - La similitud entre las representaciones de dos objetos debe estar en relación directa con la similitud con la que se suelen *percibir* dichos objetos
 - Objetos de la misma clase deben tener representaciones similares, mientras que objetos de clases distintas deben tener representaciones diferentes
- Representaciones invariantes a transformaciones/distorsiones usuales
 - Escalado, rotación, translación, oclusión
 - Variaciones de un mismo objeto deben tener representaciones similares
- Representación vectorial o simbólica del objeto
- Representación definida dentro de un *espacio* de representación E

Conceptos básicos

Clases: $\mathbb{C} = \{1, \dots, C\}$ si no se dice lo contrario

- Cada objeto o se manifiesta en un *Espacio Primario* o Universo U
- Suponemos que cada objeto $o \in U$ pertenece a una única *clase* $\varsigma(o)$
- \mathbb{C} denota el conjunto de posibles *identificadores* o *etiquetas de clase*

Espacio de representación: Generalmente $E = \mathbb{R}^D$ ó $E = \Sigma^*$

- Sea $\mathbf{x} = f(o)$ el resultado de la extracción de características de $o \in U$
- E incluye las representaciones del objeto o : $\{\mathbf{x} : \mathbf{x} = f(o), o \in U\} \subset E$
- f no es inyectiva: dos objetos pueden tener la misma representación

Conceptos básicos

Clasificador: $G : E \rightarrow \mathbb{C}$

- G se aprende con *muestras etiquetadas* $(\mathbf{x}_1, c_1), \dots, (\mathbf{x}_N, c_N) \in E \times \mathbb{C}$
- Recordad que $G = \{g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_C(\mathbf{x})\}$ y $G(\mathbf{x}) \equiv \operatorname{argmax}_c g_c(\mathbf{x})$
- El objetivo es maximizar el acierto del clasificador

$$\sum_{o \in U} \delta(G(f(o)), \varsigma(o)) \quad \delta(i, j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Índice

- 1 Introducción ▷ 3
- 2 *Representación de imágenes* ▷ 7
- 3 Representación de voz ▷ 29
- 4 Representación de texto ▷ 41

Representación de imágenes

- Imagen: soporte de uno de los medios más importantes del sistema perceptivo
- Aplicaciones:
 - Reconocimiento óptico de caracteres (OCR)
 - Reconocimiento de huellas dactilares
 - Reconocimiento facial
 - Reconocimiento de elementos en imágenes
 - Robótica
 - ...
- Reto: gran cantidad de información a manejar en algunas ocasiones
- El procesado de imágenes tiene interés *per se*, con independencia de su uso para el reconocimiento

Adquisición

- **Imagen:** Función $f(x, y)$ es el *brillo* en cada punto de *coordenadas* x, y
- **Imagen Digital:** $f(x, y)$ discretizada en su *dominio* (coordenadas) y *rango* (brillo)

Muestreo:

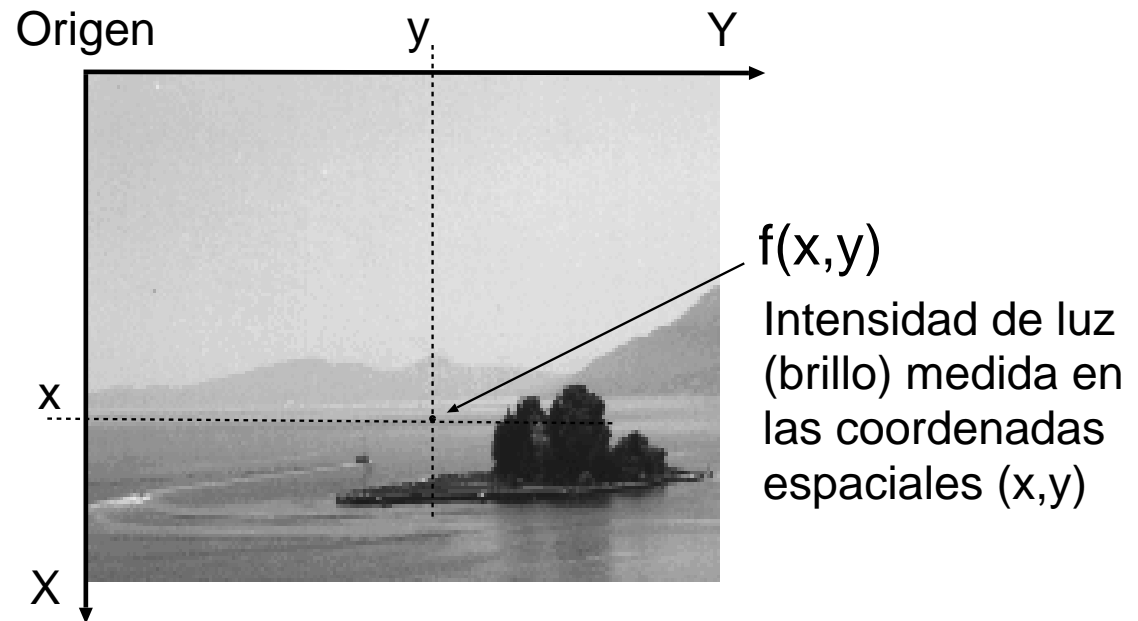
Discretización del *dominio*

Resolución: píxeles/pulgada

Cuantificación:

Discretización del *rango*

Niveles o colores: bits/píxel



Adquisición: escáner

Scanner plano



Resolución óptica: 600 ppp

Modos de exploración:

1 bit (blanco y negro)

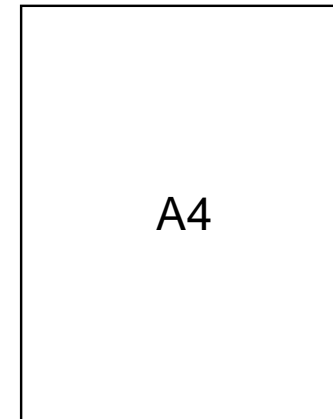
4 bits (16 niveles de gris)

8 bits (256 niveles de gris)

24 bits (16,7 millones colores)

8,27 p (21 cm)

11,69 p (29,7 cm)



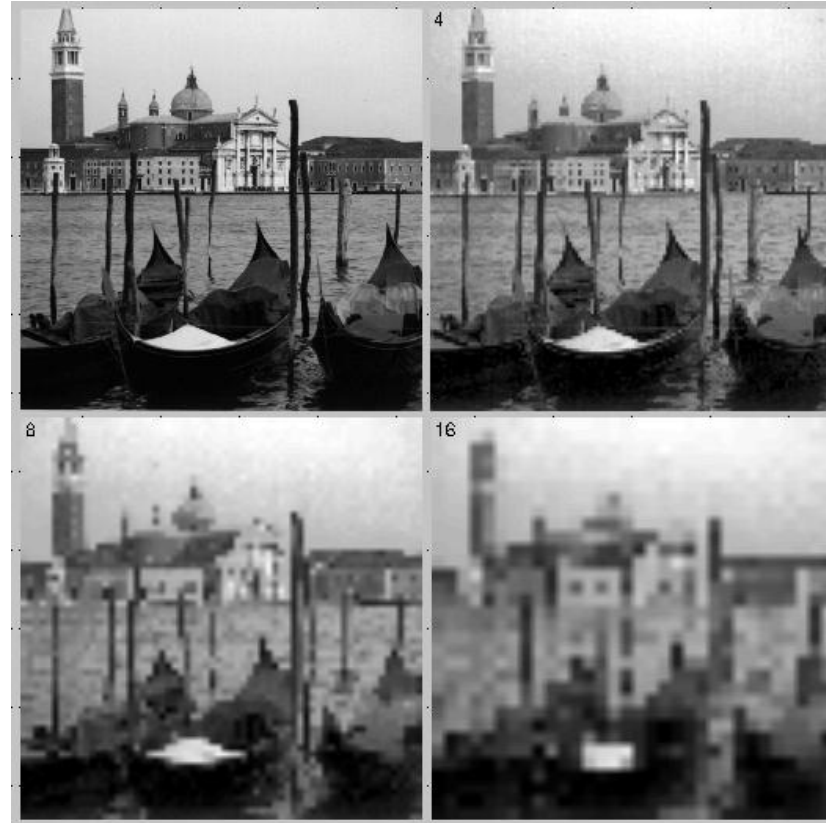
A 100 ppp y 8 bits:

*827 pixels/linea x 1169 lineas x 1 byte/pixel
= 1 Mbyte*

Límites de resolución

Imagen original

Muestreada 128 ppp



Muestreada 64 ppp

Muestreada 32 ppp

La frecuencia espacial $P = \frac{T_r}{T_d}$ donde T_r : tamaño referencia y T_d : tamaño detalle

Aliasing y teorema del muestreo

- Imagen Original: 1x1 pulgadas
- Detalles más pequeños a capturar: ≈ 0.02 pulgadas
- Frecuencia espacial de los detalles: $P \approx \frac{1}{0.02} = 50ppp$

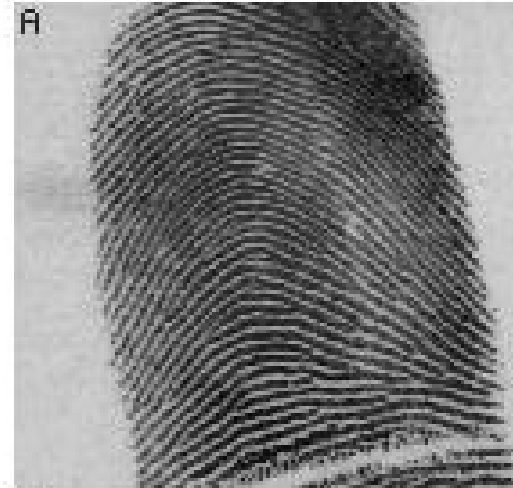
- **Frecuencia de Nyquist**

(Teorema del Muestreo):

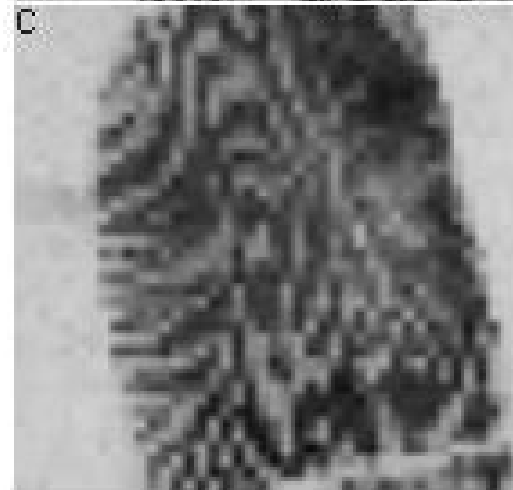
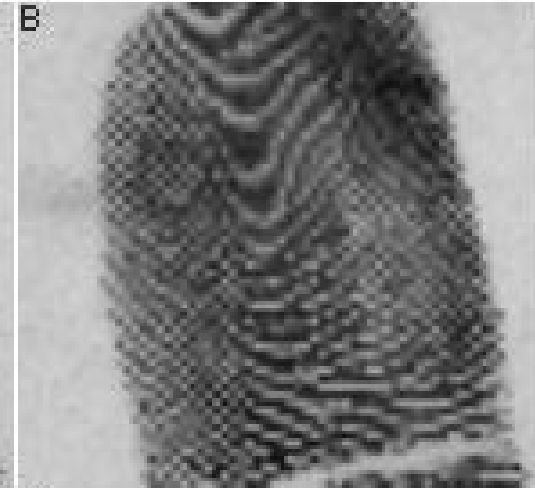
Si P es la frecuencia espacial de los detalles más pequeños a capturar en una imagen y F es la resolución de muestreo, la imagen original sólo podrá ser fielmente reproducida si:

$$F > 2P$$

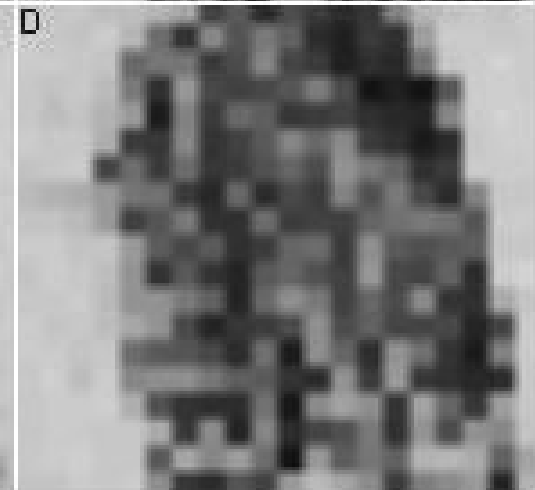
140ppp



70ppp



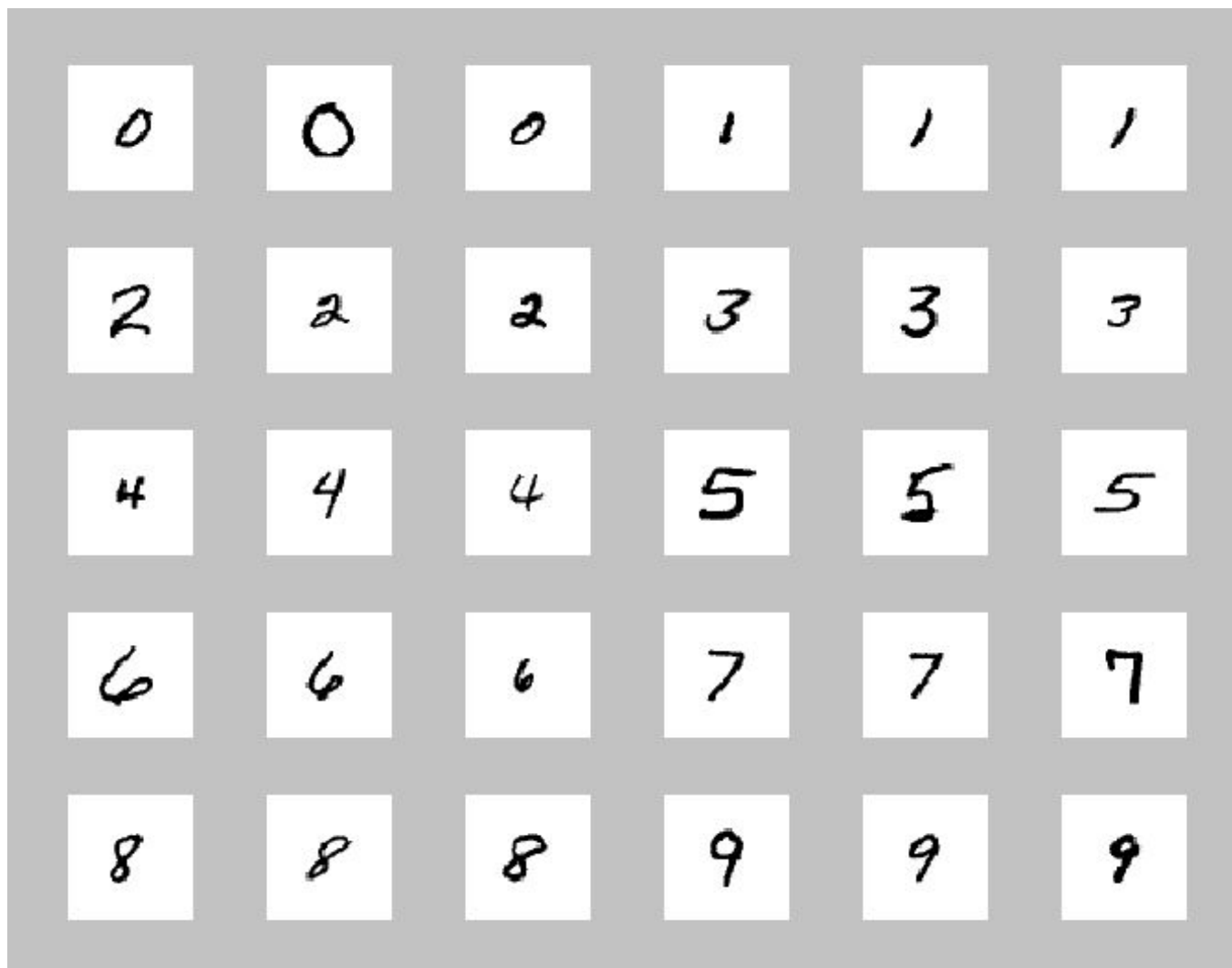
44ppp



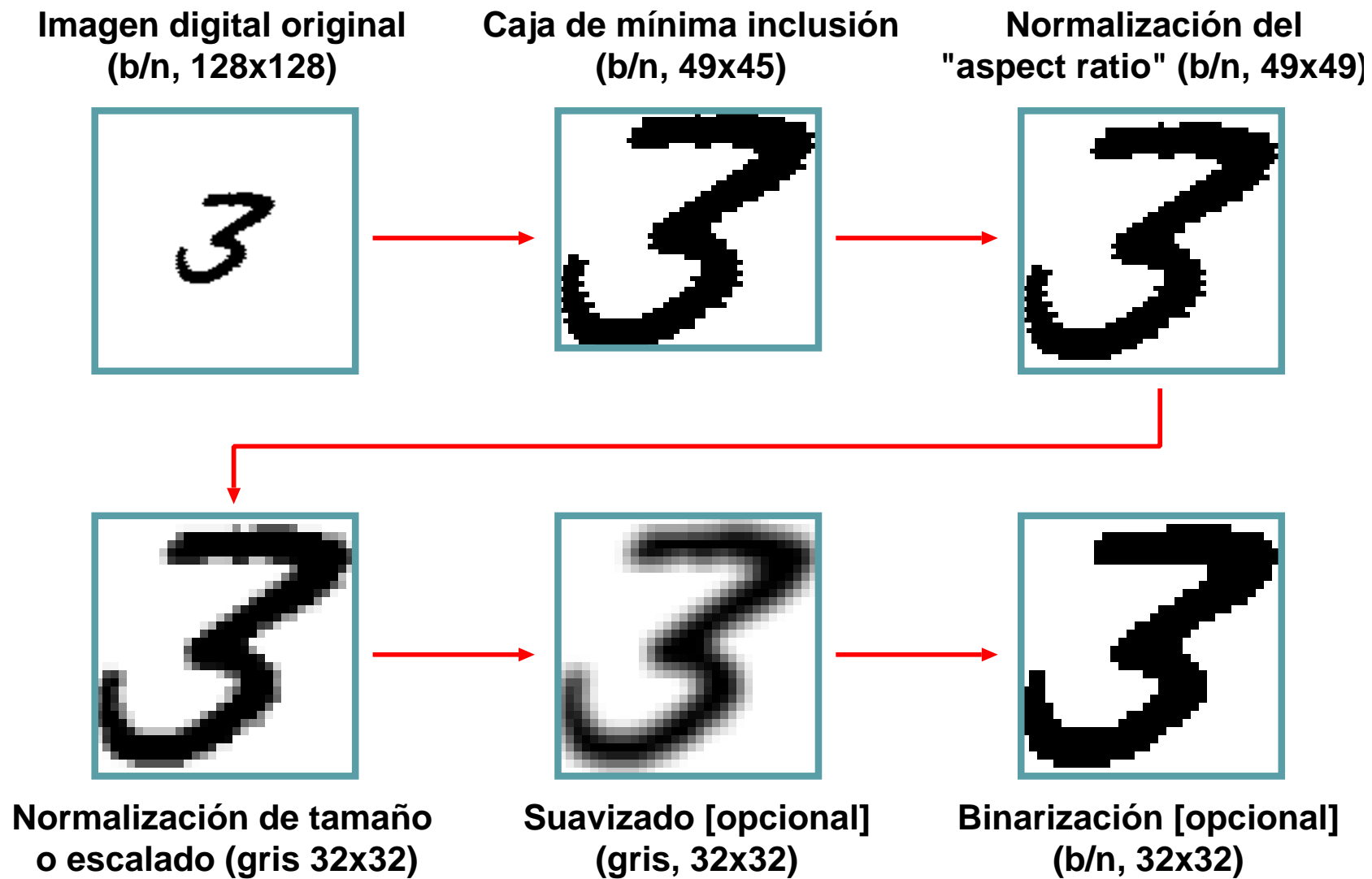
22ppp

Reconocimiento de caracteres manuscritos (OCR)

Carácteres manuscritos, aislados y “Off-line”



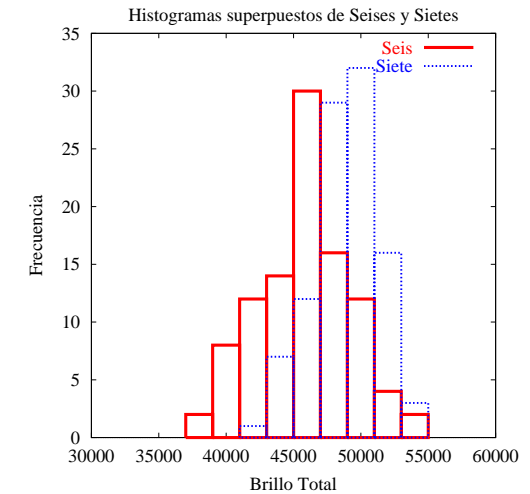
Preproceso de imágenes en OCR



Extracción de características en OCR

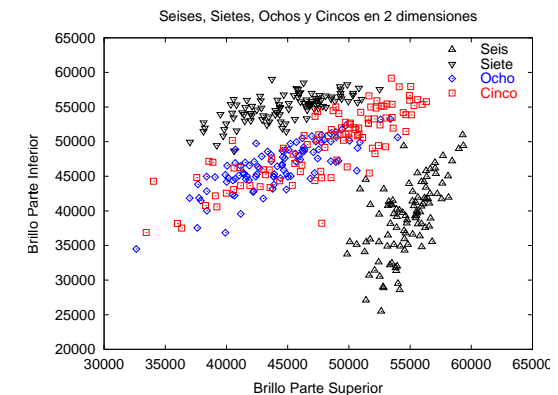
Aproximación básica: una característica por imagen

- P. ej., brillo total
- Ventaja: fácil representación (histograma)
- Inconveniente: baja capacidad discriminativa



Un paso más: dos características por imagen

- P. ej., brillo parte superior e inferior
- Ventaja: fácil representación (bidimensional)
- Inconveniente: mejora capacidad discriminativa pero aún es baja



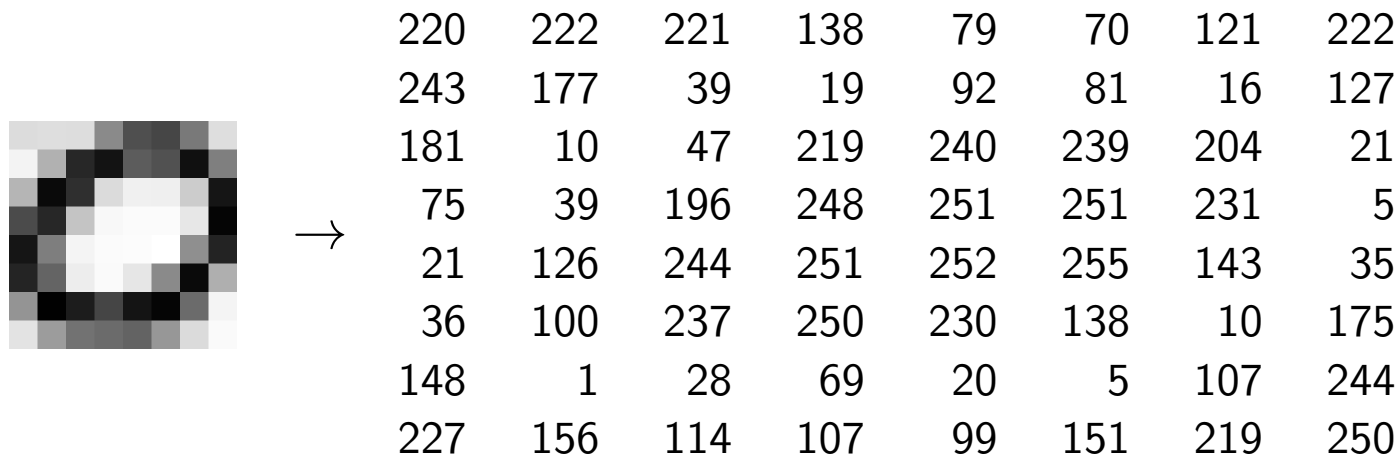
Aumentar la dimensionalidad mejora la capacidad discriminativa

Alternativa final: tomar los valores de los píxeles como características

Extracción de características en OCR

Una técnica de extracción de características en OCR es la representación **directa**:

- Preproceso y normalización a un tamaño vertical y horizontal fijo $I \times J$
- Generación de imagen en escala grises (formato PGM)
- El nivel de gris de cada píxel es una componente de la representación vectorial

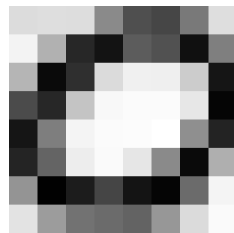


Vector de características de 8×8 componentes: (220, 222, 221, 138, . . . , 219, 250)

Extracción de características en OCR

Otra posibilidad es la representación **por histograma**:

- Tantas componentes como niveles de gris (usualmente, 256)
- La componente i es el número de píxeles con nivel de gris i en la imagen



220	222	221	138	79	70	121	222
243	177	39	19	92	81	16	127
181	10	47	219	240	239	204	21
75	39	196	248	251	251	231	5
21	126	244	251	252	255	143	35
36	100	237	250	230	138	10	175
148	1	28	69	20	5	107	244
227	156	114	107	99	151	219	250

Nivel	0	1	2	3	4	5	6	...	254	255
Número de píxeles	0	1	0	0	0	2	0	...	0	1

Vector de características de 256 componentes: (0, 1, 0, 0, 0, 2, 0, . . . , 0, 1)

Extracción de características en OCR

Cada representación ocupa un tamaño distinto en memoria

Dados n píxeles y l niveles de gris

- Representación directa: $n \cdot \left\lceil \frac{\log_2 l}{8} \right\rceil$ bytes

$$(220, 222, 221, 138, \dots, 219, 250) \rightarrow 64 \cdot 1 = 64 \text{ bytes}$$

- Representación por histograma: $l \cdot \left\lceil \frac{\log_2(n+1)}{8} \right\rceil$ bytes

$$(0, 1, 0, 0, 0, 2, 0, \dots, 0, 1) \rightarrow 256 \cdot 1 = 256 \text{ bytes}$$

Extracción de características: métodos locales

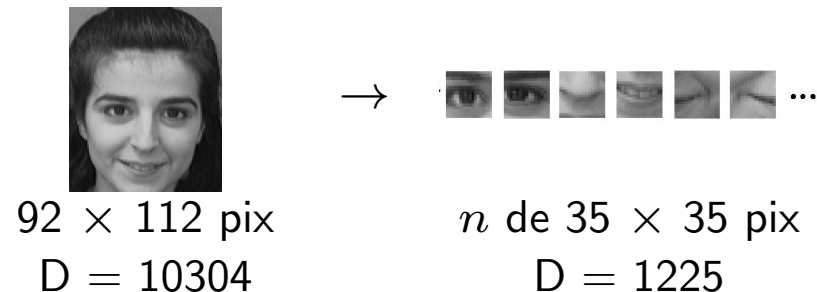
- Globalmente dos imágenes del mismo objeto pueden diferir



- Localmente existen partes (*patches*) semejantes
- Demo en PoliformaT

Extracción de características: métodos locales

- Cada imagen es representada por varias partes de la misma
- Se escogen ventanas de la imagen que sean informativas (discriminativas); por ejemplo, ventanas con alta varianza en niveles de grises



- Un objeto se representa por varios vectores de *características locales* (CL)
- Se obtienen representaciones de una dimensionalidad menor
- No se almacena la posición de la característica local dentro de la imagen
- Las representaciones locales son invariantes a traslación y oclusiones



Puntos de interés

- Puntos de interés: píxeles de los cuales extraer CL
- Criterio para definir puntos de interés:
 - Deben de poder ser formalmente (matemáticamente) definidos
 - La información *local* alrededor debe de ser discriminativa
 - En algunas aplicaciones, robustas a ciertas transformaciones: iluminación, perspectiva, distorsión, . . .
- Basados en información y varianza
 - Detectores de contorno
 - Detectores de esquinas
- Basados en exploración espacial de la imagen
 - Extracción a partir de una rejilla
 - Extracción aleatoria

Puntos de interés: detectores de contorno

- Evitan obtener CL de zonas homogéneas de la imagen
- Se detectan los contornos y se umbralizan
- Puntos de interés con alta respuesta al detector de contorno (alta varianza)



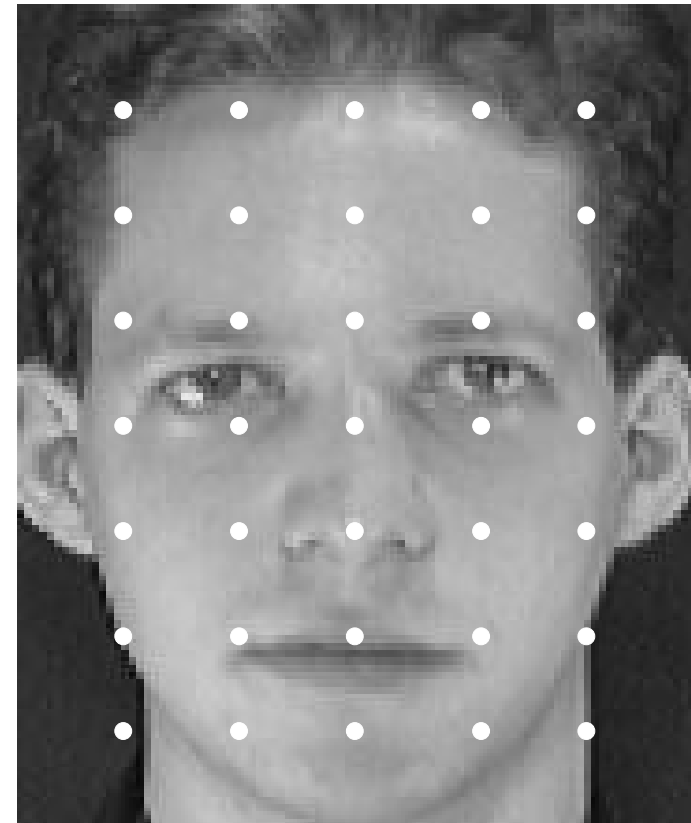
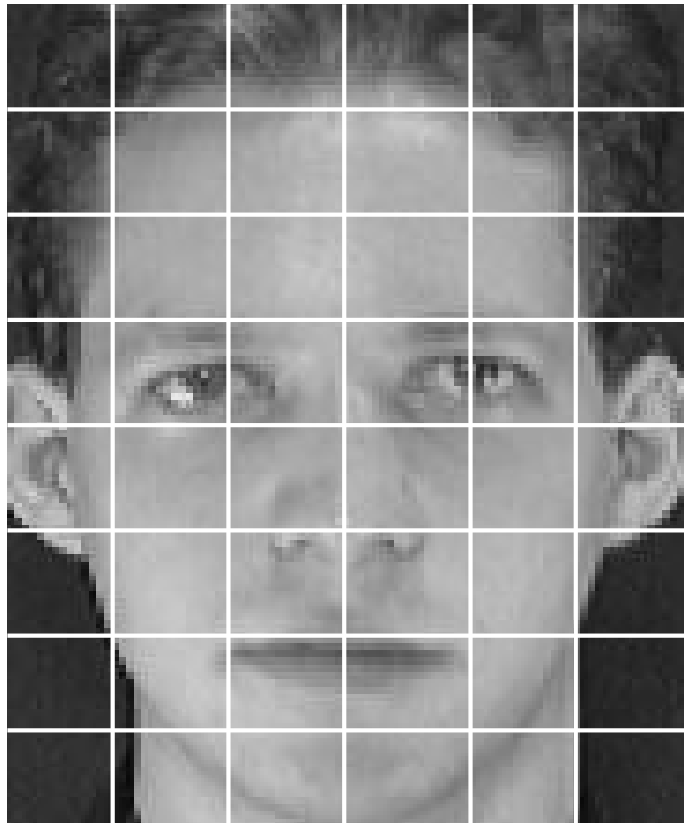
Puntos de interés: detectores de esquinas

- Más restrictivos que el detector de contorno
- Puntos de interés con alta respuesta a un detector de esquinas
- Se usan habitualmente en tareas de detección de objetos



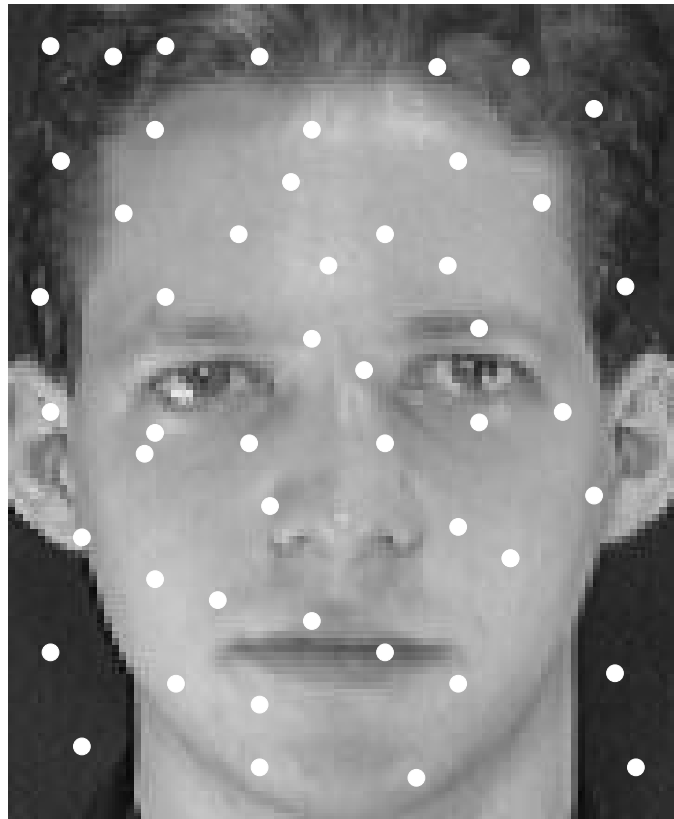
Puntos de interés: extracción por rejilla

- Puntos de interés: puntos de unión de una rejilla
- Puntos espaciados en un número de píxeles fijo, tanto horizontal como vertical (puede ser distinto en cada eje)



Puntos de interés: extracción aleatoria

- Puntos de interés: subconjunto aleatorio de los píxeles de la imagen
- Suele ser un buen complemento a la extracción por rejilla



Extracción de características: métodos locales

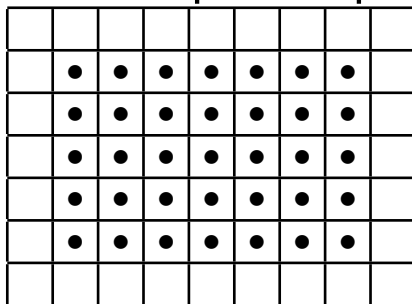
La representación por CL ocupa una memoria dependiente de:

- Número de puntos de interés/ventanas tomado (n)
- Tamaño (en bytes) de la representación de cada ventana (s)

n : dependiente de extracción y tamaño de ventana
 s : como en extracción global \rightarrow Tamaño final:
 $n \cdot s$ bytes

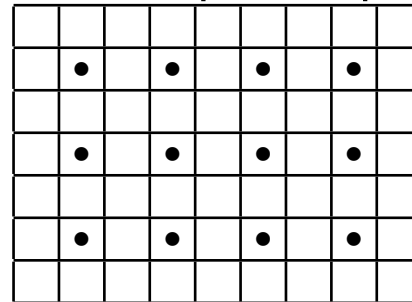
Ejemplo: extracciones por rejilla (ventana, desplazamiento horizontal y vertical)

3×3 , 1 píx., 1 píx.



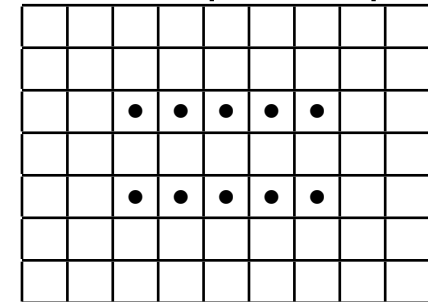
$$n = (9 - 3 + 1) \cdot (7 - 3 + 1) = 35$$

3×3 , 2 píx., 2 píx.



$$n = \left\lceil \frac{(9-3+1)}{2} \right\rceil \cdot \left\lceil \frac{(7-3+1)}{2} \right\rceil = 12$$

5×5 , 1 píx., 2 píx.

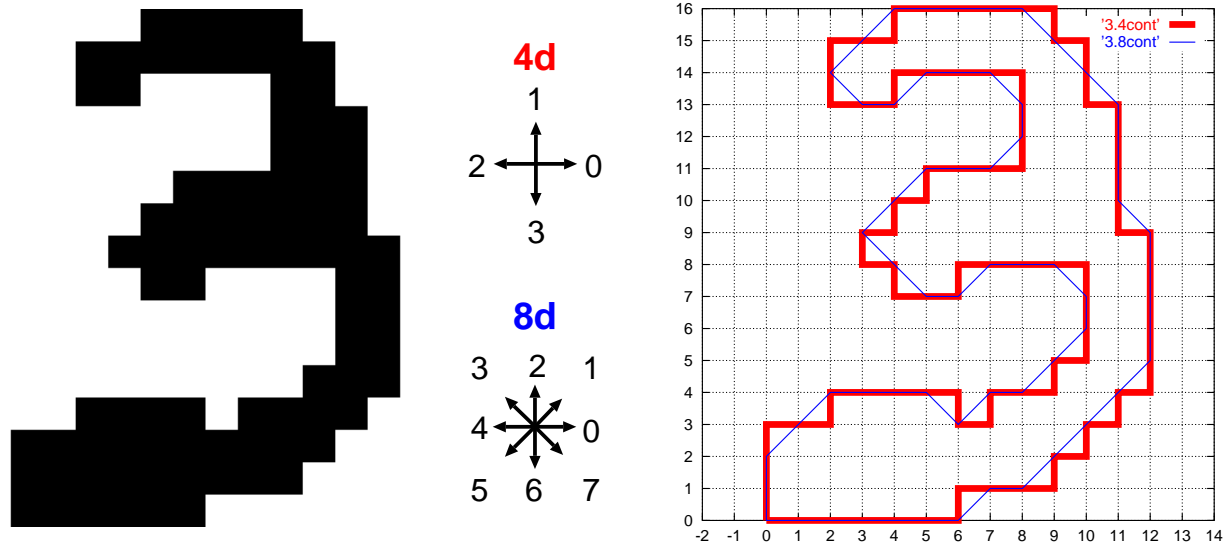


$$n = (9 - 5 + 1) \cdot \left\lceil \frac{(7-5+1)}{2} \right\rceil = 10$$

Representación estructural: códigos de contorno

Aproximación estructural/sintáctica al RF: objetos compuestos por la concatenación de *primitivas* (subobjetos elementales), representación por *modelos sintácticos* (gramáticas)

Extracción de primitivas basada en *códigos de contorno de 4 y 8 direcciones*

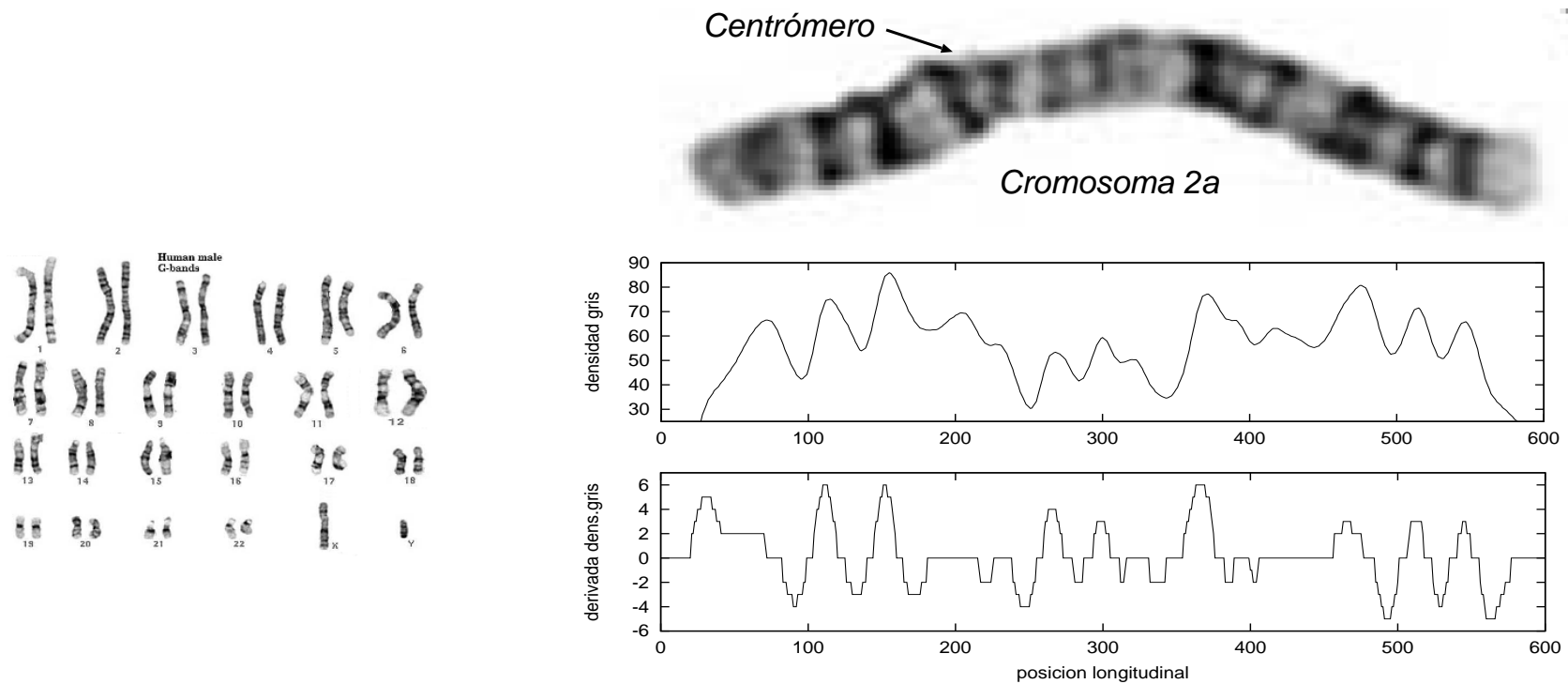


4d: 00000303303333033333232322232222221110010000301001011122223221210101000111222232211001

8d: 0000777666766665555454444442211000710112344543311001234454311

Representación estructural: otras representaciones

Representación estructural de un cromosoma



"====CDFDCBBBBBBBA==bcd==DGFB=bccb== ==cffc=CCC==cdb==BCB==dfdcb===="

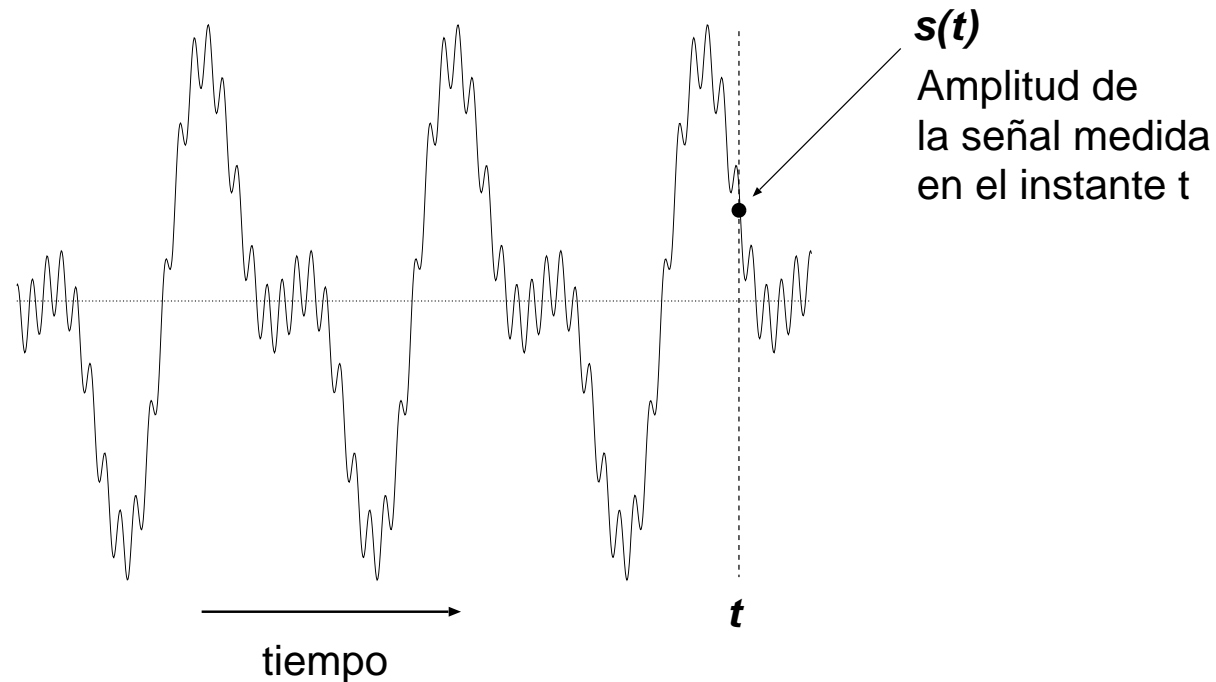
Cadena de Primitivas

Índice

- 1 Introducción ▷ 3
- 2 Representación de imágenes ▷ 7
- 3 *Representación de voz* ▷ 29
- 4 Representación de texto ▷ 41

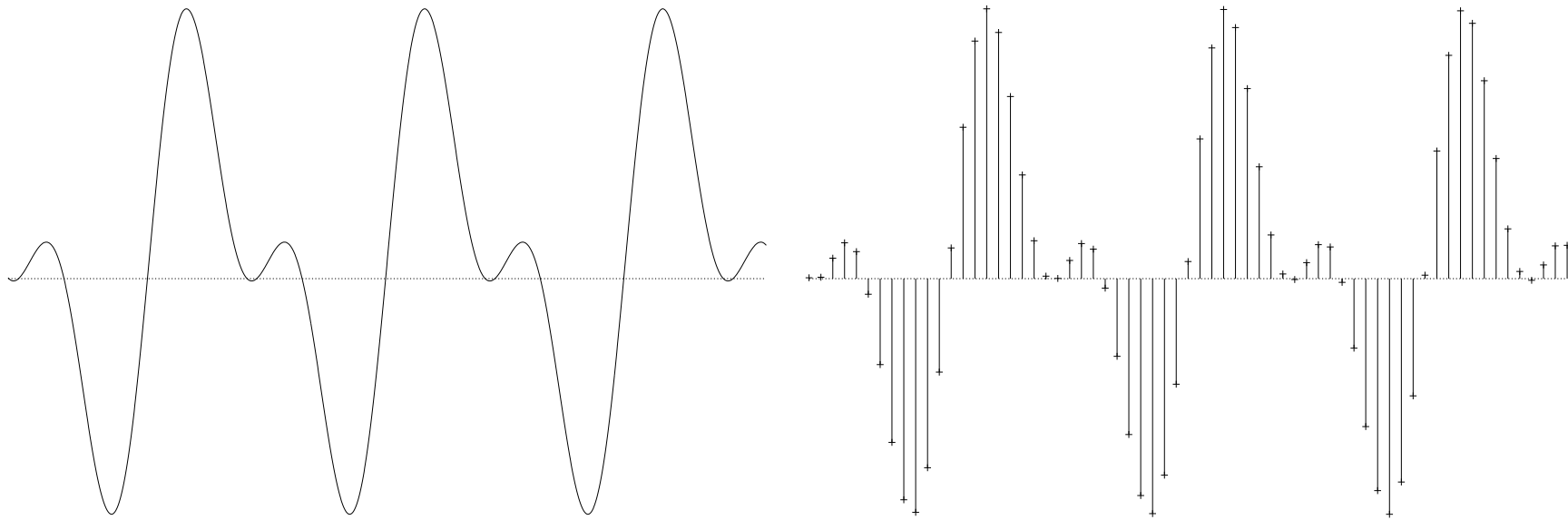
Adquisición y preprocesado

- Señal acústica: función temporal de variaciones de *amplitud* de la presión del aire $s(t)$
- Digitalización: discretización de $s(t)$ a nivel:
 - Temporal (dominio): *muestreo*
 - Amplitud (rango): *cuantificación*



Adquisición y preprocesado

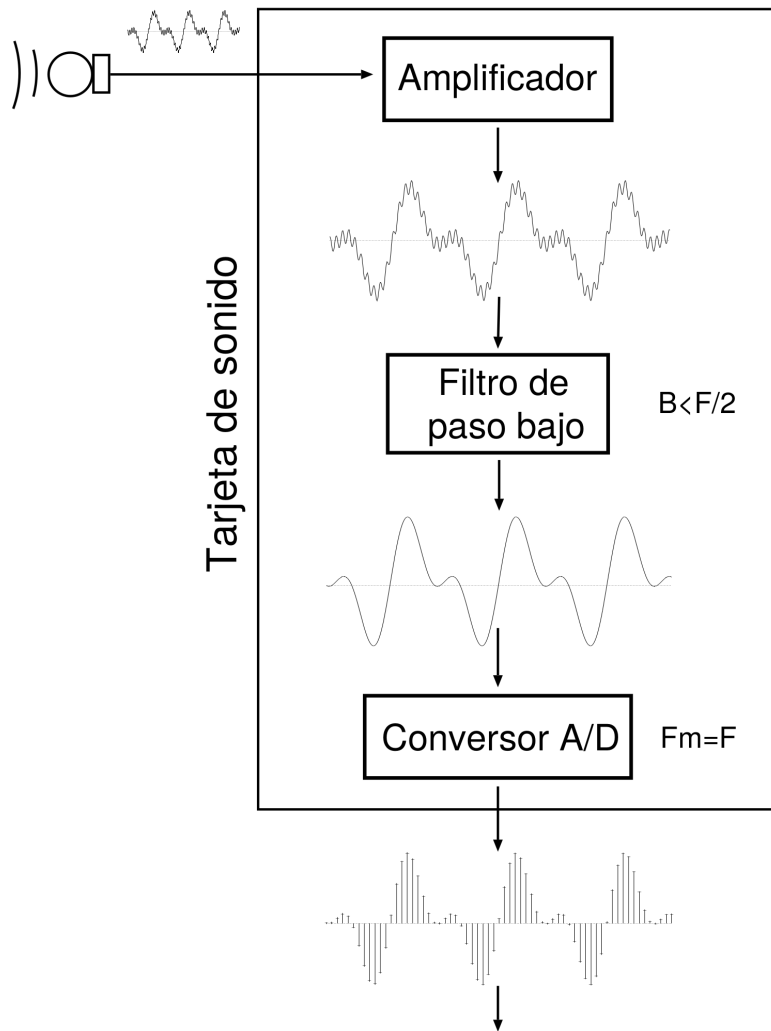
Ejemplo de **muestreo**:



Teorema del muestreo: para reconstruir una señal de ancho de banda B , la frecuencia de muestreo F_m debe cumplir $F_m > 2 \cdot B$

Cuantificación: discretizar valores en el rango de la amplitud a una cierta representación digital (número de bits)

Adquisición y preproceso



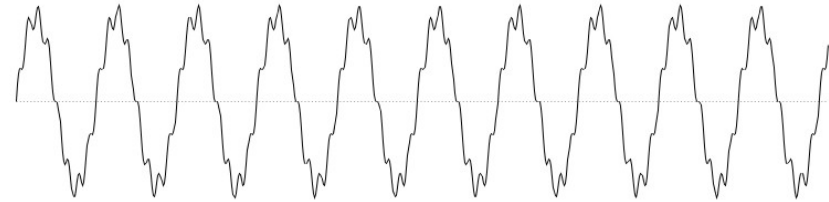
	Voz telefónica B=3.6kHz	Voz Calidad B=8kHz	Audio CD (HI-FI) B=20kHz
Muestreo (kHz)	8	16	44.1
Cuantificación (bits)	8	16	2x16
Flujo de datos (Mbytes/hora)	27.5	109.9	605.6
Segundos en 1 Mbyte	131.1	32.8	5.9

Adquisición y preproceso

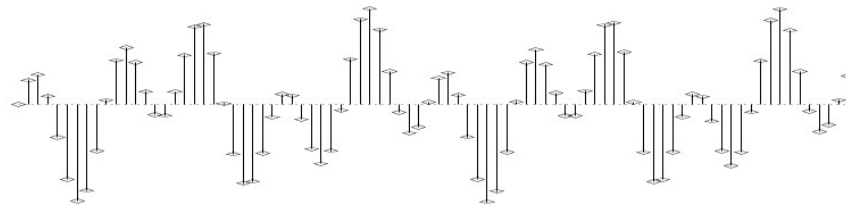
Violación del teorema de muestreo



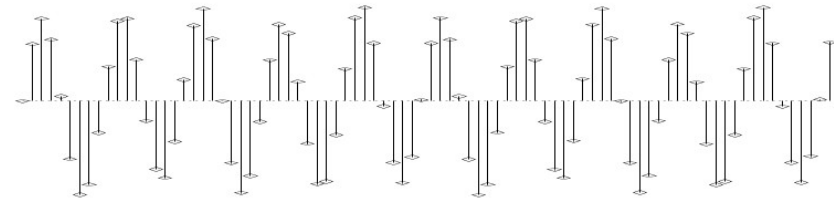
Señal original (sin filtrar): $F_0=600\text{hz}$; $F_1=4800\text{hz}$



Señal original (filtrada): $F_0=600\text{hz}$; $F_1=4800\text{hz}$



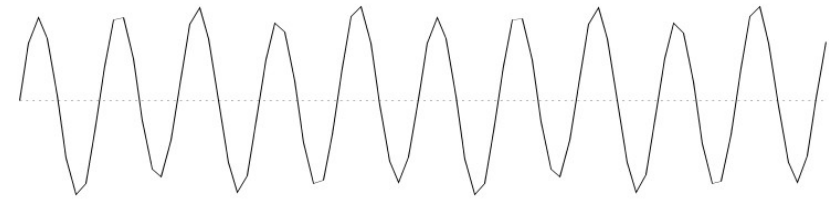
Señal muestreada: $F_m = 5000 \text{ hz} < 2 \cdot 4800 \text{ hz}$



Señal muestreada: $F_m = 5000 \text{ hz} > 2 \cdot 600 \text{ hz}$



Señal muestreada reconstruida

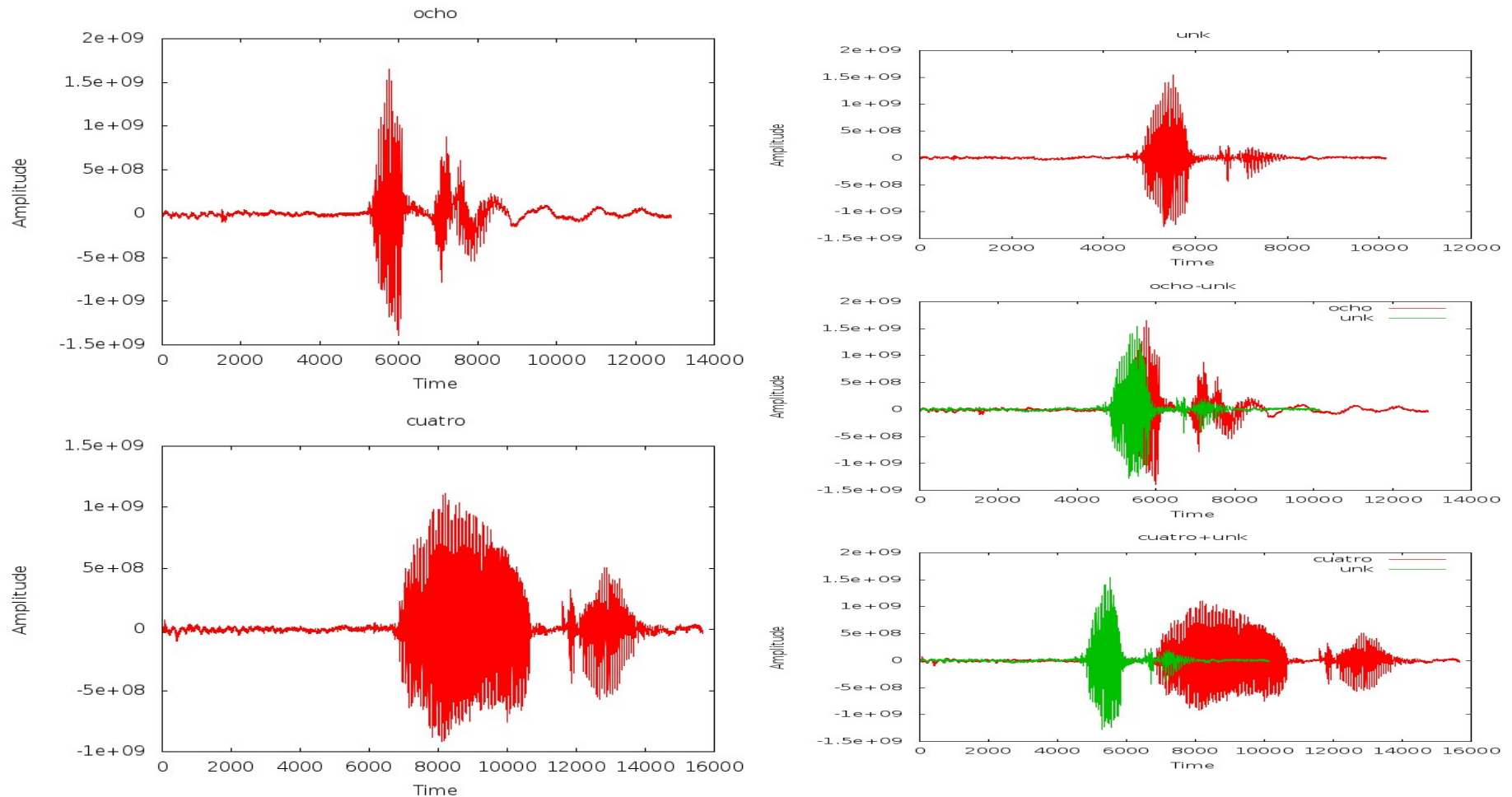


Señal muestreada reconstruida

Audios en PoliformaT

Adquisición y preproceso

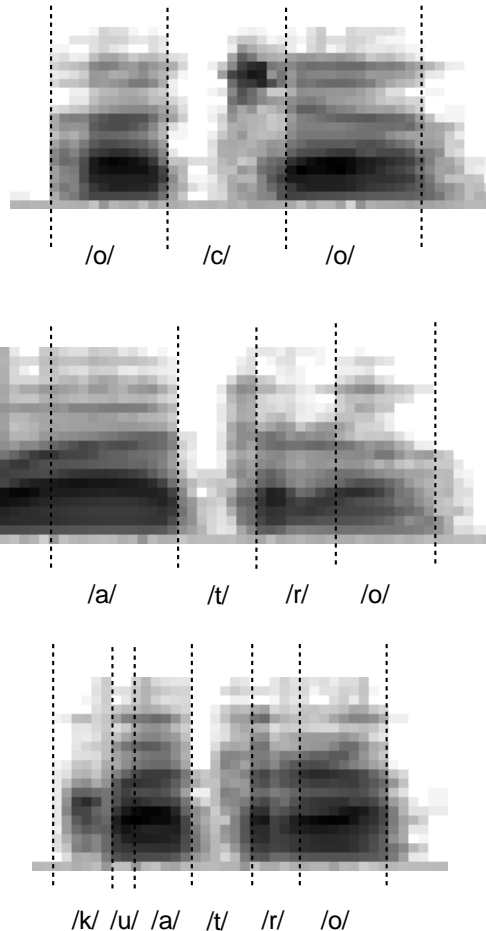
La representación temporal no es lo bastante discriminativa



Alternativa: **representación frecuencial** (*espectrograma*)

Adquisición y preproceso

Representación
en el dominio
de la frecuencia

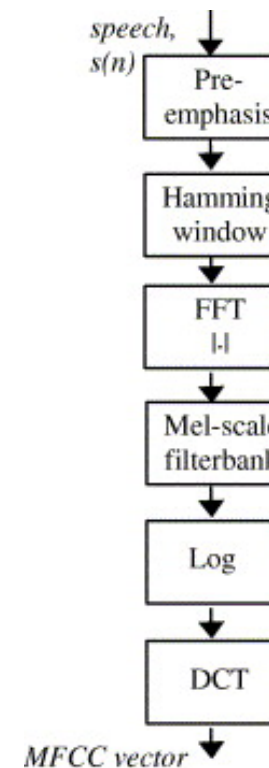


- Mayor capacidad discriminativa
- Distintas regiones identifican distintos sonidos
- Problema: longitud de las regiones no uniforme (variabilidad temporal no lineal)
 - Vocálicos: elásticos
 - Consonánticos: duración más regular

Extracción de características

Proceso habitual para obtención de coeficientes ceptrales (**estándar ETSI**):

- Preénfasis
 - Paso alto, equilibrado frecuencial
- Ventana de Hamming
 - Obtención de subseñales (marcos, *frames*)
- Transformada rápida de Fourier (FFT)
 - Paso a dominio frecuencial
- Banco de filtros en la escala de Mel
 - Filtro basado en percepción humana
- Logaritmo
 - Sensibilidad no lineal
- Transformada discreta del coseno (DCT)
 - Tracto vocal, decorrelado

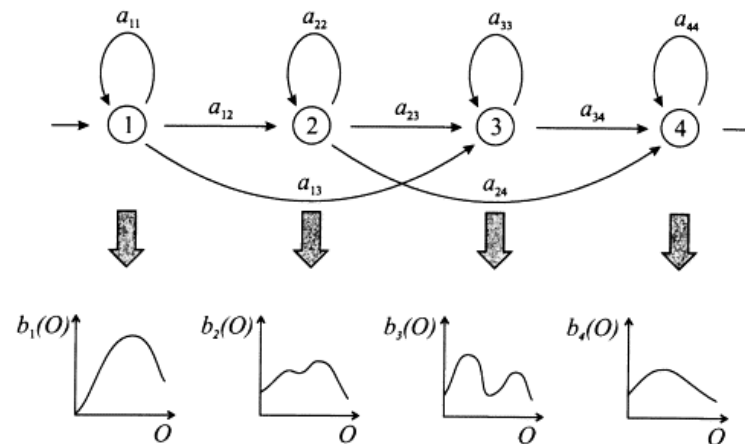


<http://ars.els-cdn.com/content/image/>

[1-s2.0-S0167639305002359-gr2.jpg](#)

Representación continua: coeficientes cepstrales

- La representación continua permite el uso de múltiples modelos estadísticos
- Representación directa del vector de características (MFCC)
- Modelo tradicional: modelo oculto de Markov (HMM) (1970s-2010s)
 - MFCC se usan para estimar los parámetros del HMM
 - Distribución Gaussiana para probabilidad de emisión en los estados
 - Posible reemplazar Gaussiana por una red neuronal profunda (DNN)



<http://ars.els-cdn.com/content/image/1-s2.0-S0262885699000554-gr1.gif>

Representación discreta: cuantificación vectorial

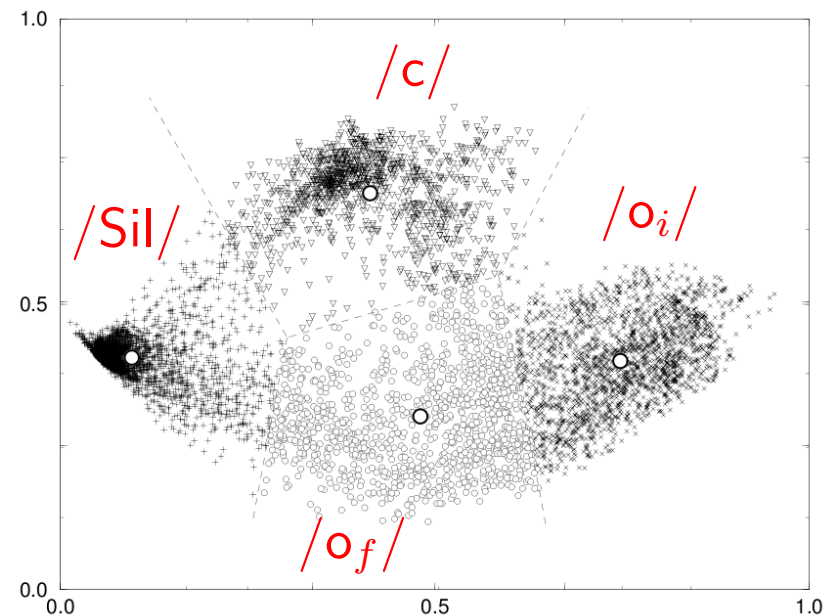
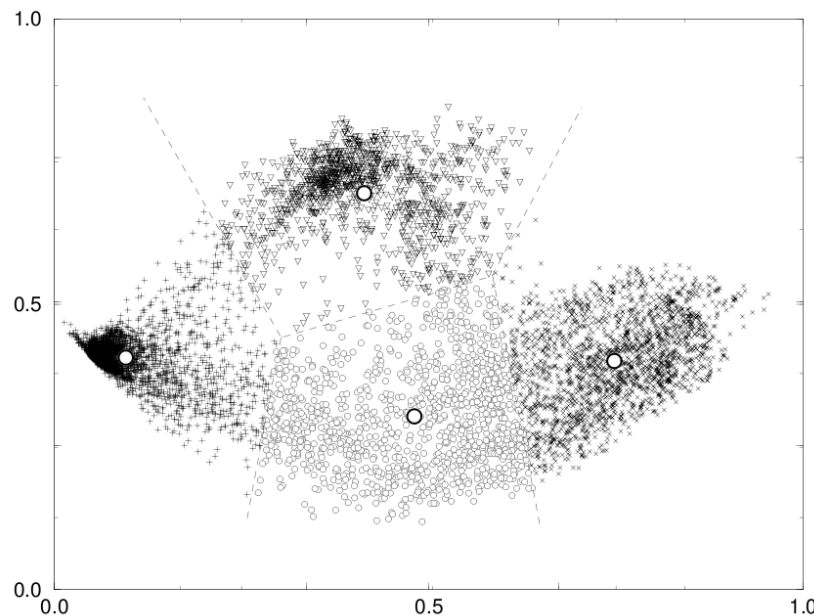
- Para tareas sencillas (palabras aisladas) puede aplicarse esta alternativa
- Idea general: asociar un símbolo a cada tipo de vector de características
- De secuencia de vectores de características a cadena de símbolos
- Proceso en dos pasos:
 1. Obtención de tipos de vectores (*codebook*)
 2. Cuantificación vectorial (asignación de vectores a símbolos)

Representación discreta: cuantificación vectorial

Proceso de creación del *codebook*

- Se toman vectores de características de entrenamiento
- Se elige un número k de tipos de vectores (k etiquetas)
- Se realiza un particionado en k *clusters* (p.ej., con k -medias)
- Se obtiene la media de cada *cluster* como prototipo o *codeword*
- Se asocia una etiqueta a cada *codeword*

Ejemplo: vectores de características de “ocho” proyectados en dos dimensiones



Representación discreta: cuantificación vectorial

Proceso de cuantificación vectorial

- Se toma el *codebook* obtenido del paso previo
- Se toman los vectores de características a decodificar
- Se asigna a cada vector la etiqueta del *codeword* más cercano (en distancia euclídea)

Espectrograma
(21 canales, 100hz)



52x21

Cadena de
etiquetas acústicas

<bbffbfffeehhhhhhhhhhhhhhhhhhhhggddddddddddddddff>

52 símbolos
($|\Sigma| = 16$)

Índice

- 1 Introducción ▷ 3
- 2 Representación de imágenes ▷ 7
- 3 Representación de voz ▷ 29
- 4 *Representación de texto* ▷ 41

Extracción de texto

- Selección de la unidad de documento
 - Varios ficheros pueden constituir un único documento
 - Un fichero grande puede requerir dividirse en varios documentos
- Problema de determinar la granularidad de la unidad de documento
- Texto almacenado en ficheros como secuencia de bytes
- Conversión a secuencia de caracteres con cierto formato
 - Ejemplo: extracción desde un fichero Word, PDF, comprimido, XML, etc.
- Necesidad de determinar el formato de fichero y su extractor asociado
- La secuencia de bytes puede contener información útil

Extracción de texto

Fases de extracción de características:

- Tokenización
 - Separación palabras, signos puntuación, etc.
- Normalización
 - A minúsculas, sin tildes, etc.
- Eliminación de *stop words*
 - Palabras con alta frecuencia
- *Stemming*
 - Eliminación de terminaciones (algoritmo de Porter)
- Lematización
 - Análisis morfológico, eliminación de desinencias

Más información: [Introduction to Information Retrieval](#) Cap. 2

Extracción de texto: ejemplo

- Extraído de la tarea *20 Newsgroups*¹
- Consiste en clasificar los mensaje enviados en 20 grupos de noticias
- Ejemplo de mensaje enviado al grupo de noticias *alt.atheism*

From: b711zbr@utarlg.uta.edu (JUNYAN WANG)
Subject: Bible contradictions

I would like a list of Bible contadictions from those of you who dispite being free from Christianity are well versed in the Bible.

- La codificación es ASCII y la unidad de documento es el mensaje
- Elimina mensajes duplicados y algunos campos de la cabecera del mensaje

¹Disponible en <http://qwone.com/~jason/20Newsgroups>

Extracción de texto: ejemplo

Original

From: b711zbr@utarlg.uta.edu (JUNYAN WANG)

Subject: Bible contradictions

I would like a list of Bible contadictions from those of you who dispite being free from Christianity are well versed in the Bible.

Tokenización

From : b711zbr @ utarlg.uta.edu (JUNYAN WANG)

Subject : Bible contradictions

I would like a list of Bible contadictions from those of you who dispite being free from Christianity are well versed in the Bible .

Normalización

from : b711zbr @ utarlg.uta.edu (junyan wang)

subject : bible contradictions

i would like a list of bible contadictions from those of you who dispite being free from christianity are well versed in the bible .

Eliminación

stop words

++++ : b711zbr @ utarlg.uta.edu (junyan wang)

subject : bible contradictions

+ +++++ like + list ++ bible contadictions +++++ +++++ ++ +++ +++ dispite being free +++++ christianity +++ well versed ++ +++ bible .

Stemming

++++ : b711zbr @ utarlg.uta.edu (junyan wang)

subject : bibl- contradict----

+ +++++ like + list ++ bibl- contadict---- +++++ +++++ ++ +++ +++ dispit-be--- free +++++ christian--- +++ well vers-- ++ +++ bibl- .

Lematización

++++ : b711zbr @ utarlg.uta.edu (junyan wang)

subject : bibl- contradict----

+ +++++ like + list ++ bibl- contadict---- +++++ +++++ ++ +++ +++ dispit-be--- free +++++ christ@@@--- +++ well vers-- ++ +++ bibl- .

Representación *Bag-Of-Words* (BOW)

- Determinar el vocabulario V (tokens diferentes) de la colección de D documentos
- Cada documento d es representado mediante un vector \mathbf{x}_d cuya dimensionalidad es igual al tamaño de vocabulario $|V|$
- Cada componente t del vector está asociada a un token del vocabulario
- Representaciones:
 - Binaria: los valores del vector indican la presencia (1) o no (0) de un determinado token en el documento que está representando

$$x_{dt} \in \{0, 1\}$$

- Entera (***term-frequency***): los valores del vector indican el número de ocurrencias de dicho token en el documento

$$x_{dt} \in \mathbb{N}$$

Representación *BOW*

3 mensajes enviados a alt.atheism

⇒

0	0	0	windows
4	11	3	god
0	0	0	dod
0	3	0	government
2	0	1	writes
14	7	15	people
0	0	0	team
0	0	0	bike
0	0	0	game
0	3	0	car
0	1	0	article
0	0	0	hockey
0	0	0	rutgers
0	0	0	encryption
0	0	0	israel
4	1	3	jesus
0	0	0	clipper
1	2	11	christians
8	8	0	bible
7	4	3	christian

3 mensajes enviados a comp.windows.x

⇒

17	17	9	windows
0	0	0	god
0	0	0	dod
0	0	0	government
0	1	0	writes
4	3	5	people
1	0	0	team
0	0	0	bike
0	1	0	game
0	0	0	car
3	0	8	article
0	0	0	hockey
0	0	0	rutgers
0	0	0	encryption
0	0	0	israel
0	0	0	jesus
0	0	0	clipper
0	0	0	christians
2	0	0	bible
0	1	0	christian

3 mensajes enviados a rec.sport.hockey

⇒

0	0	0	windows
0	0	0	god
0	0	0	dod
1	0	0	government
0	0	2	writes
8	0	7	people
9	10	0	team
0	0	0	bike
3	13	10	game
0	0	0	car
0	0	0	article
8	2	5	hockey
0	0	0	rutgers
0	0	0	encryption
0	0	0	israel
0	0	0	jesus
0	0	0	clipper
0	0	0	christians
0	0	0	bible
0	0	0	christian

Representación *BOW*

Tendremos un conjunto de documentos $\{T_1, T_2, \dots, T_D\}$

El tamaño de la representación de un documento T_d en memoria depende de:

- El tamaño de vocabulario: $|V|$
- La longitud del documento d : l_d (máximo de ocurrencias de una palabra)

Para un documento T_d :

$$|V| \cdot \left\lceil \frac{\log_2(l_d + 1)}{8} \right\rceil \text{ bytes}$$

Para la colección $\{T_1, T_2, \dots, T_D\}$, con $l_m = \max_{d=1, \dots, D} l_d$

$$D \cdot |V| \cdot \left\lceil \frac{\log_2(l_m + 1)}{8} \right\rceil \text{ bytes}$$

Representaciones en el área de extracción de información

- Las representaciones *BOW* sólo tienen en cuenta un documento
- La capacidad discriminativa puede depender de la colección de documentos
- Posible solución: definir el peso de un token como el producto de una función local (documento) por una global (colección)

$$w_{dt} = L(d, t) \cdot G(t)$$

- Representaciones *BOW*:
 - Funciones locales:
 - Conteo: $L(d, t) = x_{dt}$
 - Logaritmo: $L(d, t) = \log(x_{dt} + 1)$
 - Función global unitaria: $G(t) = 1$

Representaciones en el área de extracción de información

Representaciones globales:

- Funciones locales: las mismas que para *BOW*
- Funciones globales más utilizadas:

$$\text{Normal} \quad G(t) = \left(\sum_d x_{dt}^2 \right)^{-\frac{1}{2}}$$

$$\text{GfIdf} \quad G(t) = \frac{\sum_d x_{dt}}{\sum_{d:x_{dt}>0} 1}$$

$$\text{Idf} \quad G(t) = \log \frac{D}{\sum_{d:x_{dt}>0} 1}$$

- *Idf* es el más usado al atenuar tokens con presencia en muchos documentos

Representación: secuencia de tokens (n-gramas)

- La representación basada en *BOW* pierde información de contexto

- Ejemplo:

Mary is quicker than John
John is quicker than Mary

- Número de ocurrencias de secuencias de tokens de longitud n (n -gramas)
- Captura relación entre tokens consecutivos
- Puede presentar problemas de dimensionalidad ($|V|^n$) y dispersión
- Ha demostrado obtener mejor rendimiento que la aproximación *BOW*

Representación: secuencia de tokens (n-gramas)

3 mensajes enviados a *alt.atheism*

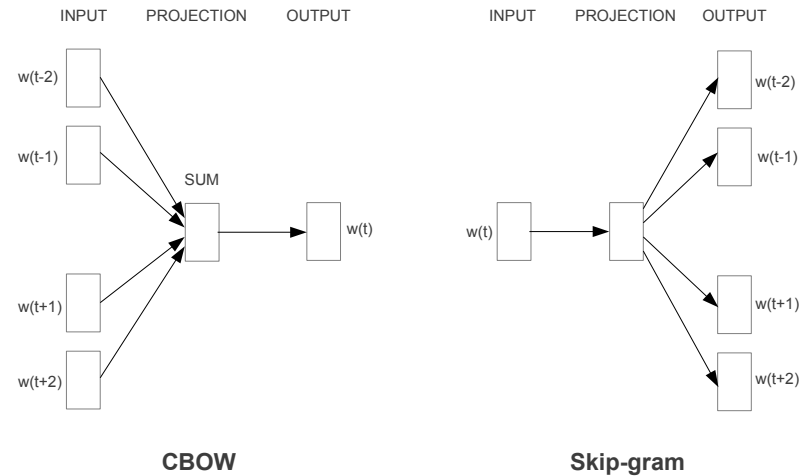
0	0	0	x windows
2	3	0	in god
0	0	0	dod security
0	1	0	government of
1	0	0	writes about
7	2	4	other people
0	0	0	team sponsored
0	0	0	your bike
0	0	0	best game
0	1	0	a car uses
0	0	0	article contains
0	0	0	hockey playoffs
0	0	0	rutgers university
0	0	0	inexpensive encryption
0	0	0	in israel
2	0	1	jesus appeared
0	0	0	clipper project
1	1	3	respectable christians
4	2	0	christian bible
3	1	1	christian morality

3 mensajes enviados a *alt.atheism*

0	0	0	for x windows
1	2	0	believe in god
0	0	0	meets dod security
0	1	0	government of the
1	0	0	writes about radical
3	1	2	other people have
0	0	0	team sponsored by
0	0	0	ride your bike
0	0	0	the best game
0	1	0	a car uses
0	0	0	this article contains
0	0	0	hockey playoffs have
0	0	0	rutgers university newark
0	0	0	inexpensive encryption devices
0	0	0	voting in israel
1	0	0	suddenly jesus appeared
0	0	0	delta clipper project
0	0	1	many respectable christians
2	2	0	the christian bible
2	0	0	that christian morality

Representación vectorial de palabras

- Representación de una palabra mediante un vector de características D -dimensional
- Capturan relación semántica entre palabras. P. ej.: king - man + woman \approx queen
- La representación vectorial más conocida es *word2vec*²: continuous *BOW* y *skip-gram*



- Se utilizan millones de frases para su entrenamiento
- Uso en tareas de PLN: modelado de lenguaje, reconocimiento del habla, traducción automática, etc.

²<https://arxiv.org/abs/1301.3781>