ANÁLISIS DE DATOS.

Jesús García

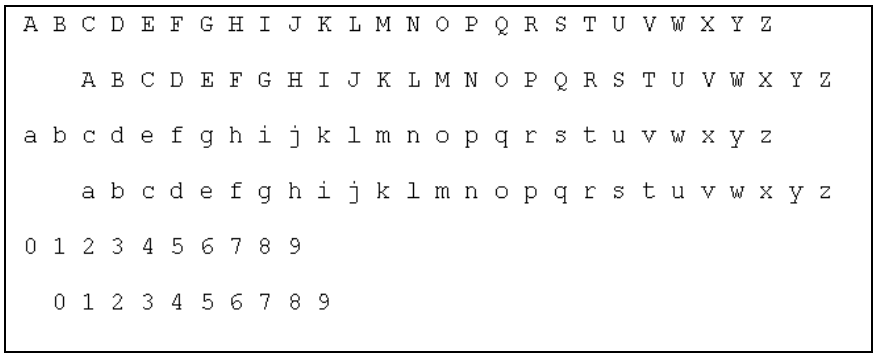
1. Descripción

Se pretende desarrollar un sistema de reconocimento automático de caracteres (*OCR, Optical Character Recognition*) para extraer información textual a partir de imágenes digitalizadas. De todos los componentes de este sistema, se centrará este trabajo en el diseño de un sistema clasificador que opere sobre atributos numéricos extraídos de la subimagen correspondiente a cada carácter.

Se dispone de un programa de extracción de atributos, *"analisisCaracteres.c"*, que toma una imagen de entrada en formato PBM (*Portable Bit Map*) y lleva a cabo los pasos siguientes:

- segmentación de cada caracter dentro de la imagen
- cálculo de atributos
- almacenamiento en fichero de salida

Por simplicidad en la resolución de los problemas típicos en la primera fase (fuera del interés de este estudio), se supone que la imagen únicamente tiene dos niveles (fondo blanco y las regiones que representan los caracteres están en negro), y que los caracteres están suficientemente separados, horizontal y verticalmente. Bajo estas hipótesis podemos tener la seguridad de que los caracteres son aislados correctamente por el programa, y los atributos calculados a partir de sus sub-imágenes son adecuados. A continuación se muestra un ejemplo de imagen de entrada, "*L_arial.pbm":*



Aparecen las letras mayúsculas, minúsculas y dígitos numéricos. Cada carácter aparece dos veces con objeto de tener muestras en las que varíe la posición horizontal y vertical con respecto a la disposición de los píxeles de la imagen. Se proporciona un conjunto de plantillas correspondientes a diversas fuentes distintas.

El programa tiene varios modos de funcionamiento, que se activan con los argumentos de la invocación (hasta 5):

ejecutable <ficheroimagen.pbm> <fichero_salida> [-t] [-b] [-v]

así, los dos primeros argumentos (obligatorios) son los nombres de los ficheros de entrada y salida, y los tres últimos son opcionales, con el significado siguiente:

ANÁLISIS DE DATOS.

-t (test): si está presente, indica que la imagen no contiene un texto conocido, por lo que no se conocen los caracteres que originan cada grupo de atributos. En caso contrario, se supone que la imagen sigue una plantilla tal y como la de la figura anterior, lo que permite almacenar cada caracter con sus atributos, necesario para realizar un entrenamiento. Por tanto, con esta misma imagen, la salida sin la opción -t sería:

```
A, 12, 36, 46, 49, 2755, ...
B, 15, 58, 39, 52, 2448, ...
C, 13, 37, 40, 53, 2534, ...
...
9, 17, 54, 45, 52, 3003, ...
```

mientras que con la opción -t tendríamos:

```
?, 12, 36, 46, 49, 2755, ...
?, 15, 58, 39, 52, 2448, ...
?, 13, 37, 40, 53, 2534, ...
...
?, 17, 54, 45, 52, 3003, ...
```

-b (borrar): si está presente, indica que el fichero de salida se creará de nuevo, borrando si existiese un previamente fichero con el mismo nombre. En caso contrario, el resultado se añadirá al final del fichero. Cuando se quiera generar un fichero de aprendizaje a partir de muchas plantillas, basta ejecutar el programa sobre cada una y con el mismo nombre de fichero de salida, desactivando esta opción.

-v (verboso): esta opción permite depurar el funcionamiento del programa. Si está presente, el fichero de salida no contendrá todos los atributos en una única fila para cada caracter, sino que aparece la subimagen aislada, y a continuación cada uno de los atributos precedido por su descripción. La codificación de la subimagen es: {***espacio_blanco*** para fondo, ***1*** para píxel del carácter, ***3***, para píxel de fondo en un agujero del caracter}. A continuación se muiestra un ejemplo para el primer carácter de la plantilla, al utilizar esta opción:

```
Caracter: A
tiene la forma siguiente:
========== PIXELES DE CARACTER 'A' ==========
====          111           ====
====          111           ====
====         11311          ====
====         11311          ====
====         11311          ====
====        1133311         ====
====        1133311         ====
====        1133311         ====
====       113333311        ====
====       111111111        ====
====      11111111111       ====
====      11        11      ====
====      11        11      ====
====     11          11     ====
====     11          11     ====
=================================================
aspecto: 12, densidad: 36, mediax: 46, mediay: 49
mediax2: 2755, mediay2: 3107, mediaxy: 2268
circularidad: 73, rectangularidad: 36, densidadAgujeros: 9
valores de densidad (en rejilla 9x9): 17 67 17 45 25 45 48 20 48
valores de signatura (cada 30°): 28 58 17 17 58 28 22 27 51 51 27 22
```

ANÁLISIS DE DATOS.

Por defecto, si no se indican los tres últimos argumentos, el modo de funcionamiento será no verboso, no borrado (extendiendo el fichero de salida en caso de existir previamente), y no test (suponiendo que la imagen sigue la plantilla de entrenamiento para añadir el campo con el valor del carácter).

Los atributos se extraen a partir de la imagen binaria, definida como

$$I_{ij} = \left\{ \begin{array}{l} 1, \quad \text{si el pixel (i, j) pertenece al caracter (negro)} \\ 0, \quad \text{en caso contrario (blanco)} \end{array} \right\}; \quad i \in [1:N], j \in [1:M]$$

y están todos redondeados a números enteros:

- **aspecto**: 10*N/M, donde N mide la altura del carácter, y M la anchura, ambas expresadas en píxeles de imagen.
- **densidad:** es el porcentaje del área total ocupada por la imagen del carácter:

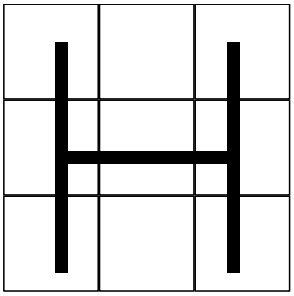
$$d = 100 \frac{NumPix}{NM}; \quad NumPix = \sum_{j=1}^{N} I_{ij}$$

- **momentos:** son el resultado de promediar las coordenadas horizontal y vertical, todos los términos hasta segundo orden, sobre la imagen del carácter I_{ij}. Además, están normalizados con respecto a las dimensiones para que no dependan de los tamaños específicos de la imagen.

$M_x = \frac{1}{N\,NumPix} \sum_{i=1}^{M} \sum_{j=1}^{N} jI_{ij}$	$M_y = \frac{1}{M\,NumPix} \sum_{i=1}^{M} \sum_{j=1}^{N} iI_{ij}$
$M_{x^2} = \frac{1}{N^2\,NumPix} \sum_{i=1}^{M} \sum_{j=1}^{N} j^2 I_{ij}$	$M_{y^2} = \frac{1}{M^2\,NumPix} \sum_{i=1}^{M} \sum_{j=1}^{N} i^2 I_{ij}$
$M_{xy} = \frac{1}{M\,N\,NumPix} \sum_{i=1}^{M} \sum_{j=1}^{N} ijI_{ij}$	

los términos M_x, M_y están relacionados con la posición relativa del centroide del carácter (cuanto más simétrico, los valores serán más próximos a 50%, 50%), y los términos M_{x2}, $M_{y2,}$ M_{xy} están relacionados con la dispersión y la correlación de ambas coordenadas.

- **rejilla**: son 9 valores que representan el porcentaje de área ocupada por el carácter a lo largo de una rejilla 3x3, tal y como se indica en el ejemplo de la figura:



10	0	10
22	16	22
10	0	10

ANÁLISIS DE DATOS.

- **circularidad**:

$$C = \frac{\text{perimetro}^2}{4\pi\,\text{Area}}$$

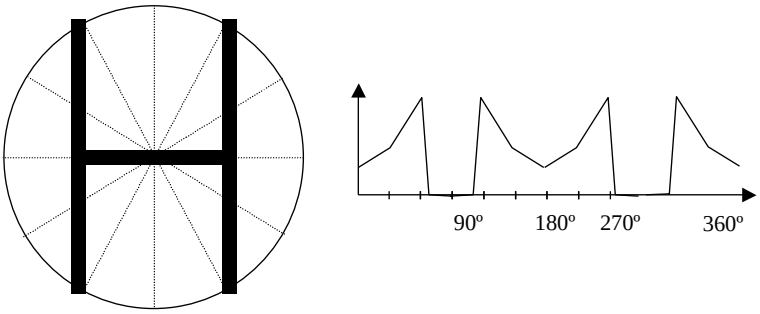
- **rectangularidad**:

$$R = \frac{\text{area caja}}{\text{area caracter}}$$

- **densidad de agujeros**: es el porcentaje de píxeles que forman parte de agujeros

$$d_A = 100\frac{\text{NumPixAg}}{NM};\quad \text{NumPixAg} = \sum_{j=1}^{N}(I_{ij} \mid (i,j) \text{ es agujero})$$

- **signatura**: es un conjunto de 12 valores del "radio" del carácter, medido desde el centroide, obtenidos cada 30º, tal y como se indica en el ejemplo de la figura:



estos valores del radio también están normalizados, con respecto al lado mayor de la caja que contiene el carácter, para evitar dependencias con el tamaño.

2. Planteamiento

El objetivo básico del trabajo es llevar a cabo un análisis detallado de la posibilidad de clasificar caracteres a partir de los atributos extraídos de las imágenes, generalizando sus propiedades comunes independientes de los tipos de fuentes. Para ello se hará uso de una herramienta que integra técnicas de análisis de datos, WEKA.

2.1. Parte obligatoria

Con el programa de análisis deberán generarse diferentes muestras de atributos extraídos con diferentes tipos de fuentes de caracteres (se proporcionan 29 plantillas). Estos ficheros de datos serán la entrada a la herramienta de análisis utilizada después para determinar las prestaciones de diferentes técnicas de clasificación posibles. Se pide:

1. Analizar la capacidad de clasificación de los diferentes caracteres a partir de los atributos generados. Deben considerarse al menos: árboles de decisión, sistemas de reglas, sistemas bayesianos y sistemas de clasificación mediante regresión, detallándose los modelos generados para cada técnica y parámetros a fijar más

importantes, de forma que pudieran implementarse en caso de considerarse adecuados.

2. Se identificarán aquellos caracteres más difícilmente distinguibles, llegando a un conjunto mínimo de caracteres con una discriminación aceptable (superior al 90%). Por ejemplo, puede analizarse el efecto de restringir el problema a letras mayúsculas, números, etc. Pueden así mismo separarse aquellas fuentes de mayor dificultad, para llegar al conjunto de fuentes con "mayor regularidad", que es capaz de tratar correctamente el sistema.

3. Se analizará la posibilidad de distinguir letras mayúsculas de minúsculas, y letras de caracteres, añadiendo los nuevos campos de clase correspondientes.

4. Deberán detallarse los diferentes tipos de error cometidos, determinando si las diferencias de prestaciones con las mejores técnicas son significativas o no (herramienta "experimenter" de WEKA). Sería interesante analizar el efecto de variar los costes de los diferentes tipos de error, así como la dependencia con el tamaño de la muestra de entrenamiento.

5. Se considerará el efecto de filtrar los atributos de entrada para distintas técnicas, determinándose los más significativos. En primer lugar se compararán las prestaciones alcanzadas por cada grupo por separado (momentos, densidad y agujeros, rejilla, signatura), para a continuación aplicar filtros de selección automática de atributos. Se indicará la ganancia obtenida con este filtrado en la precisión final.

6. Se analizará la salida de algunas técnicas de agrupamiento, determinando si existen grupos de caracteres con características comunes, y si estos grupos tienen alguna relación de interés con las clases reales.

7. Finalmente, deberá obtenerse la solución a un mensaje desconocido, aplicando el mejor clasificador obtenido a los atributos de una secuencia de caracteres no conocidos. Esta secuencia se proporcionará más adelante. A modo de ejemplo, una posible imagen podría ser la siguiente:

el conjunto de atributos para esta secuencia, "*prueba.arff*" se adjunta como ejemplo a clasificar

2.1. Parte optativa

Podrá realizarse cualquier tipo de extensión que se considere apropiada, siempre que sea adecuadamente justificada y documentada en la memoria de la práctica, razonando la motivación por la que se ha decido abordar. A continuación se indican algunas sugerencias.

ANÁLISIS DE DATOS.

8. El conjunto de fuentes puede extenderse con las que se considere oportuno, incluso aportando caracteres manuscritos (por ejemplo, generados con una herramienta de dibujo). Un ejercicio de interés sería analizar la degradación de las prestaciones al cambiar la resolución de las fuentes disponibles.

9. Además de filtrar algunos atributos de entrada para los diferentes clasificadores considerados, es interesante analizar el efecto de incluir atributos resultantes de resultar ciertas operaciones con los de entrada (ejemplo: se pueden obtener los momentos "centrados", definidos como M_{x2}-M_x^2, M_{y2}-M_y^2, M_{xy}-M_xM_y, y ver cual es su efecto en la tasa de error). Del mismo modo, puede analizarse el efecto de extender algunos de los vectores, como el tamaño de la rejilla, o el número de radios para la signatura, modificando algunos parámetros del código del programa.

10. Puede analizarse el impacto de algunas distorsiones en la imagen de entrada sobre la capacidad de discriminación. Por ejemplo, añadir ruido en los píxeles, generado con una probabilidad de intercambiar un píxel aleatoriamente. Así mismo, es interesante analizar si es mejor incluir este efecto en el conjunto de entrenamiento o no.