

# Clasificación difusa de imágenes digitales basada en la coloración de grafos

**Daniel Gómez**

Dept. Estadística e I.O. III  
Universidad Complutense  
dagomez@estad.ucm.es

**Javier Montero**

Dept. Estadística e I.O. I  
Universidad Complutense  
javier\_montero@mat.ucm.es

**Javier Yáñez**

Dept. Estadística e I.O. I  
Universidad Complutense  
jayage@mat.ucm.es

## 1 Introducción

Una de las principales críticas que se hacen a los algoritmos de clasificación de imágenes digitales en el caso no supervisado, es que la mayoría de ellos no tienen en cuenta la información del entorno que rodeaba a un pixel. Por este motivo en este trabajo se presenta una vía para obtener una clasificación difusa que tiene en cuenta esta información.

El objetivo de este trabajo será el de obtener una clasificación difusa no supervisada que tenga en cuenta el entorno y la situación de cada pixel. Para obtener esta clasificación difusa primero se modelizará la imagen digital como un grafo difuso. De la coloración de este grafo difuso se obtendrá una clasificación nítida que nos permitirá determinar las principales regiones homogéneas de la imagen, esto dará lugar a una posterior clasificación difusa de la imagen digital.

Este trabajo está dividido de la siguiente forma: en la sección 2 se introduce el grafo difuso de pixels asociado a la imagen digital. En la sección 3 se presenta el algoritmo estricto de coloración. Este algoritmo tiene algunos problemas computacionales, y por este motivo se presenta en la sección 4 el algoritmo de coloración relajado. Esta coloración dará lugar a una clasificación difusa (sección 5).

## 2 Grafo difuso de pixels

Una imagen puede entenderse como un mapa de puntos (pixels), cada uno de los cuales está caracterizado por una serie de medidas como pueden ser intensidad del blanco, rojo, azul, altitud, etc.

Matemáticamente, se llamará  $P$  al conjunto de pixels

$$P = \{v = (i, j) / i \in \{1, \dots, r\} \ j \in \{1, \dots, s\}\}$$

de una imagen  $I$ . Cada pixel es caracterizado por  $b$  medidas numéricas. La imagen  $I$  puede ser caracterizada por esos valores numéricos, así se tiene que:

$$I = \{\overline{x_{i,j}} = (x_{i,j}^1, \dots, x_{i,j}^b) / (i, j) \in P\}$$

Dada una imagen  $I$ , un problema estándar de clasificación nítida es el de la búsqueda de una *buena* partición del conjunto de pixels en regiones. Estas regiones serán consideradas como candidatos para una nueva clase, en el caso en el que la región sea lo suficientemente homogénea. Así pues en primer lugar se tratará de determinar una familia de pixels  $\{A_1, \dots, A_c\}$  de forma que  $P = \cup_{k=1}^c A_k$  pero  $A_l \cap A_m = \emptyset, \forall l \neq m$ .

Debido a las propiedades de adyacencia entre pixels en una imagen, se modelizará la imagen  $I$  como un grafo planar difuso cuyos nodos son pixels, descritos por medio de sus coordenadas cartesianas  $i \in \{1, \dots, r\}$  y  $j \in \{1, \dots, s\}$ .

El grafo será planar en el sentido de que dos pixels  $v = (i, j)$  y  $v' = (i', j')$  no pueden ser conectados si  $|i - i'| + |j - j'| > 1$ . Consecuentemente, dos pixels podrían ser adyacentes solamente si ellos comparten una coordenada, siendo la otra contigua.

La primera definición de grafo difuso fue propuesta por Kaufmann en [10], desde las relaciones difusas introducida por Zadeh en [14]. Aunque Rosenfeld introdujo en [13] otra definición elaborada, incluyendo nodos difusos y aristas difusas, en este capítulo se tratará con nodos nítidos y aristas difusas (ver [11]).

Sea  $\tilde{G} = (V, \tilde{E})$  un grafo difuso, donde  $V$  es el conjunto de nodos, y  $\tilde{E}$  el conjunto de aristas difusas caracterizados por la matriz  $\mu = (\mu_{v,v'})_{v,v' \in V}$ :

$$\mu_{v,v'} = \mu_{\tilde{E}}(\{v, v'\}) \quad \forall v, v' \in V$$

siendo  $\mu_{\tilde{E}} : V \times V \rightarrow M$  su función de pertenencia.

Cada elemento  $\mu_{v,v'} \in M$  representa el nivel de intensidad de una arista  $\{v, v'\}$  para cada  $v, v' \in V$ . En este sentido, un grafo difuso puede ser también denotado como  $\tilde{G} = (V, \mu)$ .

El conjunto  $M$  es linealmente ordenado de forma que dos valores  $\mu_{v,v'} \prec \mu_{u,u'}$  verifican que el nivel de intensidad de la arista  $\{v, v'\}$  es menor que el nivel de intensidad de la arista  $\{u, u'\}$ . El conjunto  $M$  permite la graduación literal del conjunto de aristas; por ejemplo, si  $M = \{n, l, h\}$  las aristas pueden ser graduados como nulo ( $n$ ), bajo ( $l$ ) o alto ( $h$ ).

Un grafo difuso  $\tilde{G}$  puede ser considerado como una generalización del grafo  $G$ , ya que, tomando  $M = \{0, 1\}$ ,  $\tilde{G}$  es un grafo nítido si la matrix  $\mu$  es definida como

$$\mu_{v,v'} = \begin{cases} 1 & \text{if } \{v, v'\} \in E \\ 0 & \text{en otro caso} \end{cases} \quad \forall v, v' \in V$$

**Definición 2.1** Dada la imagen  $I$ , y una distancia  $d$  entre pixels, el grafo difuso de pixels es definido como el par  $\tilde{G}(I) = (P, \tilde{E})$  siendo  $\tilde{E}$  igual a

$$\left\{ \left\{ \mu_{v,v'} = d(\overline{x_{i,j}}, \overline{x_{i',j'}}) \right\} / \vee \left[ \begin{array}{l} i = i' \quad |j - j'| = 1 \\ j = j' \quad |i - i'| = 1 \end{array} \right] \right\}$$

Teniendo en cuenta la topología de  $\tilde{G}(I)$ , los pixels del grafo difuso pueden ser también caracterizados por el conjunto  $P$  y las dos matrices  $r \times s$ ,  $D^1$  y  $D^2$ , donde

$$D^1_{i,j} = d(\overline{x_{i,j}}, \overline{x_{i+1,j}}) \quad \forall (i, j) \in \{1, \dots, r-1\} \times \{1, \dots, s\}$$

y

$$D^2_{i,j} = d(\overline{x_{i,j}}, \overline{x_{i,j+1}}) \quad \forall (i, j) \in \{1, \dots, r\} \times \{1, \dots, s-1\}$$

Así pues el conjunto de aristas que modeliza la imagen queda definido como sigue:

El proceso de coloración propuesto en este capítulo estará basado en esta representación alternativa, de ahora en adelante se identificará el grafo difuso de pixels  $\tilde{G}(I)$  por  $(r, s, D^1, D^2)$ .

**Ejemplo 2.1** Sea  $(r, s, D^1, D^2)$  grafo de pixels difuso con  $r = 3$ ,  $s = 4$  y

$$D^1 = \begin{pmatrix} 1.5 & 2.0 & 1.9 & 3.6 \\ 1.7 & 1.8 & 0.9 & 1.8 \end{pmatrix} \quad D^2 = \begin{pmatrix} 3.1 & 1.9 & 3.5 \\ 1.6 & 4.0 & 1.8 \\ 2.1 & 0.7 & 0.8 \end{pmatrix}$$

EL grafo de pixels difuso puede verse en la figura 1.

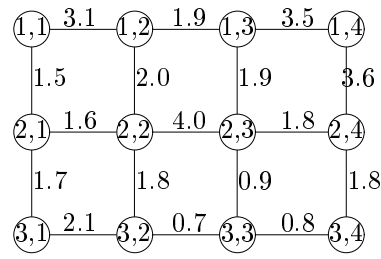


Figura 1: Grafo difuso de pixels del ejemplo 2.1

### 3 Algoritmo estricto de Coloración

Una  $c$ -coloración de un grafo  $G = (V, E)$  (ver [12]) es una función

$$C : V \longrightarrow \{0, \dots, c-1\}$$

que verifica que  $C(v) \neq C(v')$  si  $\{v, v'\} \in E$ . Una  $c$ -coloración induce una clasificación nítida del conjunto de nodos  $V$ , asociando a cada color una clase:

$$V_C(k) = \{v \in V / C(v) = k\} \quad k \in \{0, \dots, c-1\}$$

Una coloración binaria de un grafo  $G = (V, E)$  es un caso particular de una 2-coloración

$$col : V \longrightarrow \{0, 1\}$$

El objetivo que se persigue es obtener una clasificación del conjunto de pixels por medio de una  $c$ -coloración  $C$  del grafo difuso de pixels  $\tilde{G}(I)$ : el pixel  $(i, j) \in P$  estará clasificado como  $k \in \{0, \dots, c-1\}$  si su color es  $C(i, j) = k$ .

El problema de coloración de un grafo difuso puede verse en [10]. Se define la familia de  $G_\alpha$ , a los grafos nítidos definidos mediante  $\alpha$ -cortes del conjunto de aristas

$$E_\alpha = \{\{v, v'\} \in P \times P / \mu_{v,v'} \geq \alpha\}$$

Los valores del parámetro  $\alpha$  serán seleccionados de forma que un sucesivo proceso de coloración binario será aplicado a algunos subgrafos parciales difusos de  $\tilde{G}(I)$ .

La primera coloración binaria analiza el conjunto de pixels  $P$  clasificando cada pixel como 0 ó 1. La segunda coloración binaria es aplicada por separado a cada subgrafo generado por aquellos pixels coloreados como 0 obteniendo así las clases 00 y 01, y para el subgrafo generado por aquellos pixels coloreados como 1, para obtener las clases 10 y 11.

Este proceso jerárquico de coloración binaria es repetido en el proceso de coloración estricto. En este sentido

una  $c$ -coloración  $C$  será definida en  $\widetilde{G(I)}$ : Si, por ejemplo,  $C(i, j) = 5$ , al ser 101 la representación binaria de 5, el pixel  $(i, j)$  será coloreado tres veces como 1, 0 y 1 respectivamente; en este caso se necesitan tres procesos binarios de coloración fijando en cada uno de ellos el primer, segundo y tercer dígito de la representación binaria del color.

### 3.1 Procedimiento de coloración binaria

El procedimiento de coloración binaria clasifica dos pixels adyacentes como 0 y 1 si sólo si la distancia ente ellos es mayor o igual que un valor prefijado  $\alpha$ . Esta es una aproximación alternativa al problema de clasificación clásico ya que en la aproximación clásica dos pixels serán clasificados en la misma clase si son similares sin tener en cuenta si son adyacentes o no.

Formalmente, para definir el primer procedimiento de coloración binario, dado un valor  $\alpha$ , sea  $G_\alpha$  el  $\alpha$ -corte del grafo difuso  $\widetilde{G(I)}$ :

$$G_\alpha = (P, E_\alpha)$$

donde  $E_\alpha$  es el conjunto de todos los pares de pixels adyacentes con distancia  $d$  menor que  $\alpha$ .

Sea  $col : P \rightarrow \{0, 1\}$  una coloración binaria de  $G_\alpha$ . Teniendo en cuenta la topología del grafo difuso de pixels, la primera coloración binaria puede ser obtenida asignando color arbitrario ("0" o "1") a un pixel concreto, y definiendo un orden en el cuál cada pixel será coloreado; si se supone, por ejemplo, que el primer pixel es el  $(1, 1)$  en la parte superior izquierda de la imagen, los pixels serán coloreados desde la izquierda hasta la derecha y de arriba a abajo con un valor de  $\alpha$  fijo.

Dado un pixel coloreado  $(i, j)$ , los pixels adyacentes  $(i+1, j)$  y  $(i, j+1)$  son coloreados; el pixel  $(i+1, j+1)$  puede ser coloreado desde el pixel  $(i+1, j)$  y desde el pixel  $(i, j+1)$ . Una restricción natural es que ambos colores deben ser iguales, en otro caso el pixel será denotado como *inconsistente*.

**Definición 3.1** Dado el conjunto de pixels  $P$  un cuadrado es un subconjunto de cuatro pixels

$$sq(i, j) \equiv \{(i, j); (i+1, j); (i, j+1); (i+1, j+1)\}$$

siendo  $i \in \{1, \dots, r-1\}$  y  $j \in \{1, \dots, s-1\}$ .

Denotaremos entonces por  $PS$  el conjunto de todos los cuadrados.

**Definición 3.2** Dado un grafo difuso de pixels  $(r, s, D^1, D^2)$ , un cuadrado  $sq(i, j) \in PS$  es consistente a nivel  $\alpha$  si, dado un color arbitrario  $col(i, j)$ , el

anterior procedimiento de coloración binaria asigna el mismo color al pixel  $(i+1, j+1)$ , tanto si en coloreado desde el pixel  $(i, j+1)$  o desde el pixel  $(i+1, j)$ . En otro caso se dirá que el cuadrado de pixels es inconsistente.

En el ejemplo 2.1, el pixel  $sq(1, 1)$  es inconsistente al nivel  $\alpha = 3.1$ , pero consistente al nivel  $\alpha = 3.2$ .

**Definición 3.3** Dado un valor de  $\alpha$ , el grafo difuso de pixels  $(r, s, D^1, D^2)$  es consistente al nivel  $\alpha$  si todos los cuadrados  $sq(i, j) \in PS$  son consistentes al nivel  $\alpha$ .

Existen dos valores extremos que garantizan la consistencia de todos los cuadrados:

- $\bar{\alpha} = \max_{(i,j) \in P} \{d_{i,j}^1; d_{i,j}^2\}$

Si se fija un valor  $\alpha > \bar{\alpha}$ , entonces toda la imagen es considerada como una clase única ( $col(i, j) = col(1, 1) \forall (i, j) \in P$ ).

- $\underline{\alpha} = \min_{(i,j) \in P} \{d_{i,j}^1; d_{i,j}^2\}$

En el caso  $\alpha < \underline{\alpha}$ , la imagen parece un tablero de ajedrez, siendo todos los pixels adyacentes clasificados como "0" y "1":

$$col(i, j) = \begin{cases} col(1, 1) & \text{si } (i+j) \text{ es impar} \\ 1 - col(1, 1) & \text{en otro caso} \end{cases}$$

En este sentido, solamente el intervalo  $[\underline{\alpha}, \bar{\alpha}]$  debería ser considerado. Determinar un nivel  $\alpha$  intermedio apropiado no es una cuestión trivial.

Dado un nivel  $\alpha$ , los cuadrados inconsistentes pueden ser detectados con la función binaria  $inconsis_\alpha(i, j, D^1, D^2)$  que devuelve 1 si ese cuadrado es consistente y 0 si no lo es.

Se puede notar que esta función binaria es muy importante, así que se debería escoger un valor apropiado para  $\alpha$ . Esta elección dará lugar a la coloración binaria  $col$ . Sin embargo la función  $inconsis_\alpha(i, j, D^1, D^2)$  no depende del color particular del pixel  $(i, j)$ .

El grafo difuso del ejemplo 2.1 es inconsistente al nivel  $\alpha = 3.1$ , pero es consistente al nivel  $\alpha = 4.1$ .

La coloración que se propone en este artículo está basada en sucesivas coloraciones binarias para diferentes niveles consistentes (que van decreciendo), por este motivo nos interesa empezar con el máximo valor ( $\alpha^*$ ) que hace que el grafo difuso de pixels sea consistente.

**Definición 3.4** Dado el grafo difuso de pixels  $(r, s, D^1, D^2)$ , el nivel de consistencia, denotado como  $\alpha^*$ , es el máximo valor  $\alpha \in [\underline{\alpha}, \bar{\alpha}]$  para el cual el grafo difuso es consistente.

La existencia de este nivel de consistencia está siempre asegurada, al menos mientras la imagen contenga un número finito de pixels. Si alguna inconsistencia es detectada para algún valor de  $\alpha$  dado, puede ser introducido un procedimiento decreciente para encontrar un valor  $\alpha^*$  menor que asegure la consistencia. Este procedimiento será inicializado con  $\alpha^* = \bar{\alpha}$ .

En el ejemplo 2.1, el cuadrado  $sq(1, 2)$  es inconsistente al nivel  $\alpha = \bar{\alpha} = 4.0$ , ya que  $col(1, 2) = col(1, 3) = col(2, 2)$  pero  $col(2, 3) = col(1, 3)$  y  $col(2, 3) = 1 - col(2, 2)$ . Este cuadrado no puede ser consistente para un nivel  $\alpha \in (2.0, 4]$ . Para  $\alpha = 2.0$ , sin embargo, el cuadrado  $sq(2, 1)$  es inconsistente para cualquier valor de  $\alpha \in (1.8, 2.0]$ . De hecho, puede ser chequeado que si  $\alpha = 1.8$ , el grafo difuso de pixels es consistente siendo inconsistente para cualquier valor de  $\alpha \in (1.8, 4.0]$ . En este caso,  $\alpha^* = 1.8$ .

Para determinar  $\alpha^*$  se busca entre aquellos cuadrados inconsistentes  $sq(i, j)$ : Si este cuadrado es inconsistente al nivel  $\alpha$ , se puede computar el máximo valor  $\alpha'$  para el cuál este cuadrado será consistente, por medio de una nueva función *newalpha*, que será evaluada para cada cuadrado.

En el ejemplo 2.1, se puede ver que

$$\begin{aligned} newalpha_{4.0}(1, 2, D^1, D^2) &= 2.0 \\ newalpha_{2.0}(2, 1, D^1, D^2) &= 1.8 \end{aligned}$$

En este ejemplo, todos los pixels han sido coloreados al principio, después, se ha buscado el valor  $\alpha^*$  que asegura la consistencia. Así pues se tiene a los pixels clasificados en las clases "0" o "1". Se debería ahora proceder a conseguir un color más preciso para ambas clases (la clase "0" debe a su vez ser dividida en la clase "00" o "01"). Esto será realizado de forma separada, activando alternativamente solo una de las clases que ya están coloreadas en una etapa anterior.

De la misma forma se procederá en las posteriores etapas, de forma que el anterior proceso de coloración binaria es aplicado a aquellos pixels activados, es decir un subconjunto de pixels  $P' \subset P$ . Este subconjunto de pixels  $P'$  puede ser caracterizado por una matriz *act* tal que  $act(i, j) = 1, \forall (i, j) \in P'$  y  $act(i, j) = 0 \quad \forall (i, j) \notin P'$ .

Se puede computar el intervalo  $[\underline{\alpha}, \bar{\alpha}]$  para los pixels activados (proceso de llamaremos en el desarrollo del trabajo *initalpha*( $r, s, D^1, D^2, act$ )). En el caso en que no existan dos pixels adyacentes activados el proceso debería parar.

De nuevo se puede observar que un cuadrado dado puede ser consistente para un valor  $\alpha$  pero inconsistente para otro valor  $\alpha' < \alpha$ . Aquí un procedimiento decreciente repetido para el conjunto de cuadrados de pixels

*PS* encontrará el nuevo valor  $\alpha^*$  asegurando que todos los cuadrados son hechos consistentes en la nueva coloración.

La siguiente función, *consislevel*, es el núcleo de nuestro algoritmo: esta función será evaluada iterativamente obteniendo así el nivel de consistencia  $\alpha^*$  para la familia de pixels que han sido activados. Los inputs del procedimiento *consislevel* son los pixels del grafo difuso ( $r, s, D^1, D^2$ ) y la matriz *act*. Esta función evalúa el nivel de consistencia  $\alpha^*(act)$ , para un subconjunto de pixels activados  $P' \subset P$ . Hay que observar que inicialmente cuando  $act(i, j) = 1 \quad \forall (i, j) \in P$ , entonces  $\alpha^*(act) = \alpha^*$ , el nivel de consistencia del grafo difuso de pixels.

Siguiendo un orden estándar en los pixels activados, el nivel  $\alpha^*(act)$  asegura un procedimiento de coloración binaria válido *col* que permite determinar el color  $col(i + 1, j)$  con dos opciones:

- $col(i, j) \quad \forall (i, j) \in P$  tal que  $act(i, j) * act(i + 1, j) = 1$  y  $d_{i,j}^1 < \alpha^*(act)$
- $1 - col(i, j) \quad \forall (i, j) \in P$  tal que  $act(i, j) * act(i + 1, j) = 1$  y  $d_{i,j}^1 \geq \alpha^*(act)$

Análogamente, se determina  $col(i, j + 1)$ :

- $col(i, j) \quad \forall (i, j) \in P$  tal que  $act(i, j) * act(i, j + 1) = 1$  y  $d_{i,j}^2 < \alpha^*(act)$
- $1 - col(i, j) \quad \forall (i, j) \in P$  tal que  $act(i, j) * act(i, j + 1) = 1$  y  $d_{i,j}^2 \geq \alpha^*(act)$

Todos estos cálculos se agruparán en el proceso que denotamos por *bincol* $_{\alpha}(r, s, D^1, D^2, act, col)$ . Este proceso calcula la coloración binaria de los pixels activados en el nivel  $\alpha$ . Este procedimiento se inicializa  $col(i, j) = 0 \quad \forall (i, j) \in P$ .

Volviendo de nuevo al ejemplo 2.1, y a la asignación  $act(i, j) = 1 \quad \forall (i, j) \in P$  y  $\alpha = \alpha^* = 1.8$ , el procedimiento *bincol* computa la coloración binaria vista en la figura 2.

Después de una primera etapa, los pixels coloreados son clasificados en dos clases: 0 y 1, dependiendo de cuál es la distancia considerada entre pixels adyacentes.

### 3.2 Algoritmo estricto de coloración

El proceso de clasificación inducido por la coloración binaria puede ser redefinido, aplicado a cada matriz *act*, para clasificar pixels de la clase 0 de la siguiente forma:

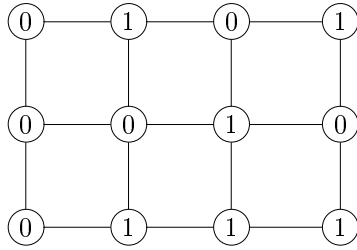


Figura 2: Primera coloración binaria del grafo de pixels difuso del ejemplo 2.1

$$act(i, j) = \begin{cases} 1 & \forall (i, j) \in P / col(i, j) = 0 \\ 0 & \text{en otro caso} \end{cases}$$

El procedimiento *bincol* es entonces aplicado con el valor  $\alpha = consislevel(r, s, D^1, D^2, act)$ . Los pixels activados serán clasificados de nuevo como 0 o 1 dando lugar a las clases "00" y "01".

Análogamente, las clases 10 y 11, son definidas cuando este segundo proceso de coloración es repetido para la clase 1, de forma que cuatro clases son obtenidas: "00", "01", "10" y "11".

Los pixels clasificados pueden ser identificados por la función:

$$C : P \longrightarrow \{0, 1, 2, 3\}$$

donde  $C(i, j)$  es el número entero asociado con el número binario de la clase.

El mismo procedimiento de coloración puede ser sucesivamente aplicado a cada familia de pixels pertenecientes a la misma clase de color, siempre y cuando existan pixels adyacentes con al menos dos distancias distintas entre ellos. (es decir, siempre que se cumpla que  $consislevel(r, s, D^1, D^2, act) > 0$ ). Así, si este proceso de coloración es sucesivamente aplicado  $t$  veces, obtendremos  $2^t$  clases a lo sumo.

En el caso en que se añada un nuevo nivel de clasificación, incrementando el parámetro  $t$ , es porque existen al menos dos pixels adyacentes  $((i, j)$  y  $(i', j')$ ) en la misma clase ( $C(i, j) = C(i', j')$ ) verificando que  $d((i, j), (i', j')) \geq \alpha^*(act)$ , cuando  $act(i, j) = act(i', j') = 1$ ; este número de pares será denotado por *adp*. En otro caso, el proceso de coloración para. Este proceso de coloración es denotado por **crucol** $(r, s, D^1, D^2, C)$ .

Este procedimiento de clasificación puede verse como un método jerárquico divisible [7]. Estos métodos de clasificación empiezan con un cluster que contiene a todos los elementos, y en cada paso se dividen las clases obteniendo mayor número de clases en la siguiente

iteración. Si el proceso lo dejásemos correr, finalmente se tendrían tantas clases como objetos.

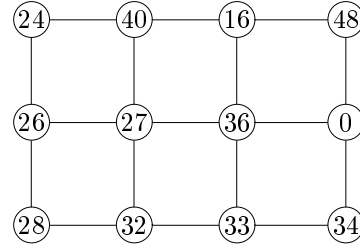


Figura 3: Coloración del grafo difuso de pixels del ejemplo 2.1

## 4 Algoritmo relajado de coloración

Cuando se trabaja con imágenes digitales de tamaño medio o alto, el número de inconsistencias en relación con el tamaño de la imagen suele ser muy pequeño. Sin embargo puede ocurrir, que cuando se busca en el proceso de decrecimiento el valor  $\alpha^*(act)$  este sea el valor 0. Consecuentemente los pixels no son clasificados.

Para paliar este problema, se propone relajar las restricciones de consistencia dando lugar así a un nuevo procedimiento de coloración binaria, permitiendo algunas inconsistencias (por ejemplo un 1 % de cuadrados inconsistentes). La complejidad computacional queda resuelta fijando un pequeño número de iteraciones (4 o 5, por ejemplo).

Denotamos por *incratio* como el porcentaje de inconsistencias del proceso de coloración binario, es decir el número de pixels inconsistentes divididos por  $(r-1)(s-1)$ . Como se puede observar si permitimos valores grandes de *incratio*, el valor de  $\alpha$  que necesitamos no decrecerá mucho y necesitaremos por tanto un número de clases pequeño. Por otro lado, a mayor valor de *incratio* mayor número de pixels serán coloreados erróneamente.

Así pues, se buscará un valor de *compromiso* entre estas dos situaciones. Esta solución de compromiso debe tener en cualquier caso ratios de inconsistencia pequeños (por ejemplo del orden de 0.01, de forma que un porcentaje muy pequeño de pixels serán coloreados de forma equivocada, en el caso de que la imagen sea grande esa clasificación errónea no podrá ser percibida por el ojo humano). Cada pixel inconsistente deberá entonces ser *aislado* de forma que su inconsistencia no induzca más inconsistencias en los pixels adyacentes.

Sea *col* una coloración binaria. Inicialmente,  $col(i, j) = 0 \forall (i, j) \in P$ , y entonces se procederá a asignar nuevos colores a cada pixel. El orden con el

que los pixels serán de nuevo coloreados será el mismo que teníamos en el algoritmo estricto de coloración (de izquierda a derecha y de arriba a abajo).

En el proceso de coloración de la primera fila, desde la izquierda hasta la derecha, no se producirán inconsistencias, así como el proceso de coloración vertical de las columnas desde la primera. Las inconsistencias, si existen, aparecerán cuando exista un cuadrado  $sq(i, j)$  tal que el pixel  $(i + 1, j + 1)$  sea coloreado de forma distinta si se hace desde el pixel  $(i + 1, j)$  que si se hiciese desde el pixel  $(i, j + 1)$ . Denotaremos por  $ninc$  al número total de pixels inconsistentes.

Para que el proceso de coloración funcione correctamente, dado un pixel activado  $(i, j) \in P$ , es muy importante distinguir entre aquéllos que todavía no han sido coloreados,  $act(i, j) = 2$ , y aquéllos que han sido coloreados,  $act(i, j) = 1$ . Inicialmente  $act(i, j) = 2$ , para todos los pixels activados  $(i, j)$ .

Así pues, para cada pixel activado  $(i, j) \in P$ , se puede asociar el valor  $act(i, j) = 2$  si este pixel no ha sido coloreado todavía y  $act(i, j) = 1$  en otro caso. Inicialmente,  $act(i, j) = 2$  para todos los pixels activados. Una vez que un pixel  $(i, j)$  es coloreado,  $act(i, j) = 1$  si es consistente (siempre al actual nivel  $\alpha$ ); en otro caso  $act(i, j) = -1$ . como se ha mencionado anteriormente, estos pixels inconsistentes deben ser aislados para que no *contaminen* a los pixels adyacentes.

Se distinguirán los pixels activados  $act(i, j) \neq 0$  de los no activados  $act(i, j) = 0$ , en el primer caso caso  $act(i, j) = 2$  si no han sido coloreados y  $act(i, j) = 1$  cuando lo sean consistentemente y  $act(i, j) = -1$  en otro caso. Como se ha mencionado anteriormente, estos pixels inconsistentes deben ser aislados para que no *contaminen* a los pixels adyacentes.

Un pixel inconsistente puede ser arbitrariamente coloreado. Iteraciones posteriores suavizaran los efectos de esta coloración arbitraria.

Para detener el crecimiento exponencial de las iteraciones del algoritmo estricto, el número de iteraciones  $t$  debe estar acotado (en este sentido se tiene un control a priori sobre el número de particiones). Se denota por  $it_M$  dicha cota.

En cada iteración, el valor de  $\alpha$  se escoge teniendo en cuenta los diferentes valores de las distancias entre pixels. Por otra parte el parámetro  $it_M$  deber ser elegido teniendo en cuenta muchos factores. Empíricamente, así como para mejorar la visualización de los resultados un valor razonable es  $it_M = 6$

Una vez se ha fijado el valor de  $it_M$ , diferentes valores de  $\alpha$  pueden ser elegidos. Denotamos por  $\alpha_t$ , con  $t \in \{1, \dots, it_M\}$ , la familia de esos valores. El pro-

cedimiento *vecalpha* devuelve eso valores en el vector  $(\alpha_1, \dots, \alpha_{it_M})$ .

## 5 Clasificación difusa

A partir de los algoritmos introducidos anteriormente, se tiene una clasificación nítida. Sean  $C_1, \dots, C_{2^i}$  las clases de dicha clasificación.

**Definición 5.1** Dada una clase  $C_k$  Diremos que región  $R \subset P \times P$  es homogénea respecto a dicha clase si  $\forall (i, j) \in R, (i, j) \in C_k$  y además está rodeada de pixels que no pertenecen a  $C_k$

De esta forma se obtienen las diferentes regiones homogéneas dentro de la imagen digital. Denotaremos por  $NREG$  al número de regiones identificadas por ese proceso de clasificación.

**Ejemplo 5.1** Sea  $(P, I)$  una imagen digital, de tamaño  $3 \times 7$ . Una vez hecha la clasificación utilizando cuatro colores, tenemos cuatro clases: 0, 1, 2, 3. A continuación se ve el grafo asociado a dicha clasificación.

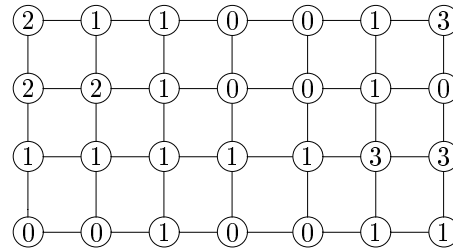


Figura 4: Regiones homogéneas de la red de pixels

Como puede verse la región  $R = \{(1, 1), (2, 1), (2, 2)\}$  es una región homogénea respecto a la clase 2. En esta red de pixels se distinguen diez regiones homogéneas.

Las regiones homogéneas aquí obtenidas son fruto de una coloración de un grafo difuso que representa una imagen digital. Sin embargo cuando la red de pixels es grande, la forma de representar los datos es esencial para el posterior desarrollo de la clasificación.

Para representar las diversas regiones homogéneas se usará una nueva imagen digital donde cada región es representada por un color que resuma las principales características espectrales de la región (ver [8]).

Una vez hecho esto cada región quedará representada por un color que presenta la vaguedad de la región definida. Esta representación (difusa) de las regiones homogéneas puede dar lugar (según sea el interés del

estudio) a una clasificación difusa de la imagen digital (sin más que aplicar algoritmos de clasificación difusa de datos como los algoritmos vistos en [5]), o a la obtención de funciones de pertenencia para las distintas regiones homogéneas (mediante funciones de verosimilitud a cada región) que han sido determinadas en la imagen teniendo en cuenta el entorno de cada pixel.

Aplicando este algoritmo a mas de 20 imágenes digitales se observa que con pocos colores se obtiene una clasificación buena de las diversas regiones homogéneas. Si el número de colores es suficientemente grande se tendrán tantas regiones homogéneas como se tienen en la fotografía original. Una vez que se tiene una buena información acerca del número de clases y sus características se está en disposición de realizar una clasificación difusa (ver [2, 3, 4]).

## 6 Conclusiones Finales

Los principales algoritmos no supervisados que abordan el problema de la clasificación no tienen en cuenta el entorno que rodea a cada pixel, haciendo imposible la clasificación correcta en determinados casos. La clasificación que aquí se propone tiene en cuenta las disimilitudes entre pixels adyacentes para una clasificación nítida inicial. En este trabajo se ha hecho hincapié en la fase de la determinación de regiones homogéneas de la imagen digital, por este motivo esta propuesta, debe ser entendida como una pieza de un algoritmo mucho mas complejo en que se aborden temas que aquí se tratan de pasada como: la determinación de la adecuada función de pertenencia, número de clases difusas que obtienen una buena representación de los datos, procesos de agregación de la información ...etc.

## Referencias

- [1] A. Amo, J. Montero y G. Biging: *Relevancia y redundancia en sistemas de clasificación difusa*. En: Proceedings Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2000 (A. Ollero, S. Sánchez, B. Arrue y I. Baturone, eds., Universidad de Sevilla, Sevilla); pp. 521-525 (2000).
- [2] A. Amo, D. Gómez, J. Montero y G. Biging: *Relevance and redundancy in fuzzy classification systems*. *Mathware and Soft Computing* 8, 203-216 (2001)
- [3] A. Amo, J. Montero, A. Fernández, M. López, J. Tordesillas y G. Biging: *Spectral fuzzy classification: an application*. *IEEE TRANS.SYST.MAN AND CYB. C* 32, 42-48 (2002)
- [4] A.Amo, D. Gómez, J. Montero: *Spectral fuzzy classification: a supervised approach*. *Mathware and Soft Computing* 10:141-154 (2003)
- [5] A. Amo, J. Montero, G. Biging y V. Cutello: *Fuzzy classification systems*. *EUR. JOUR. OPER. RES.* (to appear)
- [6] A. Amo, J. Montero y V. Cutello: *On the principles of fuzzy classification*. En: *Proceedings North American Fuzzy Information Processing Society Conference*; 675–679 (1999).
- [7] B. S. Everitt *Cluster analysis*. 1993.
- [8] D. Gómez. *Algunas aportaciones sobre relaciones de preferencia*. Ph.D. Thesis, Complutense University of Madrid, Spain (in Spanish).
- [9] J. R. Jensen: *Introductory digital image processing*. 1986 Prentice-Hall.
- [10] A. Kaufmann: *Introduction à la théorie des sous-ensembles flous, 1. Eléments théoriques de base*. Masson et Cie, Paris, France, (1976).
- [11] L. Kóczy *Fuzzy graphs in the evaluation and optimization of networks*. *Fuzzy sets and systems*; 46, (1992) 307-319.
- [12] P. M. Pardalos, T. Mavridou, J. y Xue *The Graph Coloring Problem: A Bibliographic Survey*. In: Du, D.-Z., and Pardalos, P.M. (Eds.). *Handbook of Combinatorial Optimization*, vol. 2 (Kluwer Academic Publishers, Boston, 1998), pp. 331-395.
- [13] A. Rosenfeld *Fuzzy graphs*. In: Zadeh, L.A. Fu, K.S., y Shimura, M., eds, *Fuzzy sets and their applications to cognitive and decision processes*, Academic Press New York, (1975) 77-95.
- [14] L.A. Zadeh: *Similarity relations and fuzzy orderings*. *Information Science* 3:177–200 (1971).