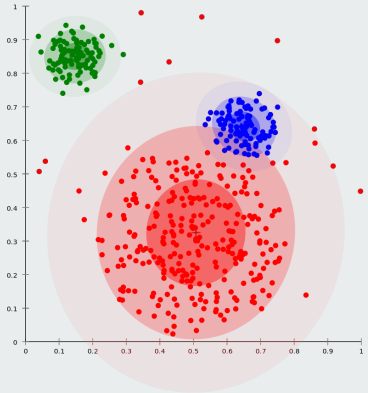
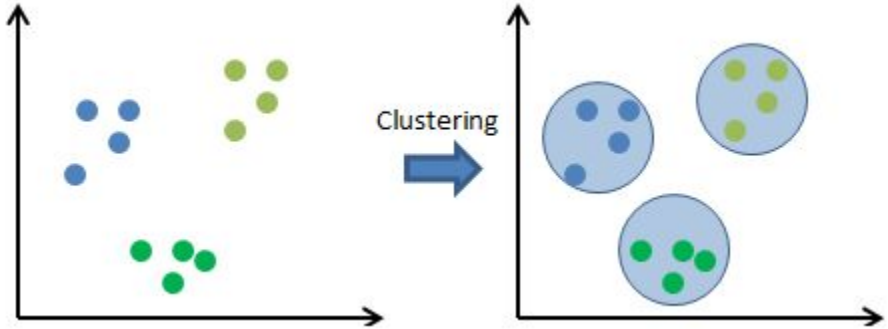
Agrupamiento (Clustering)



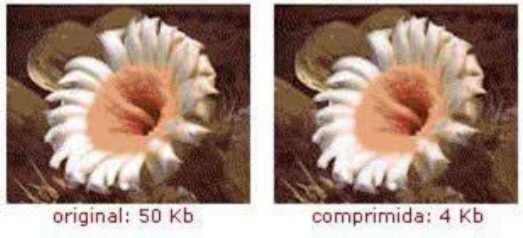
¿Qué es el agrupamiento?

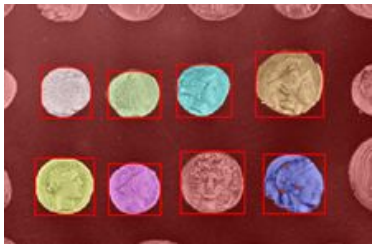
Entre los algoritmos de aprendizaje no supervisado están los algoritmos de agrupamiento. Estos se enfocan en agrupar objetos según sus características intrínsecas o similaridades. El objetivo de los algoritmos de agrupamiento es encontrar las estructura de los datos.

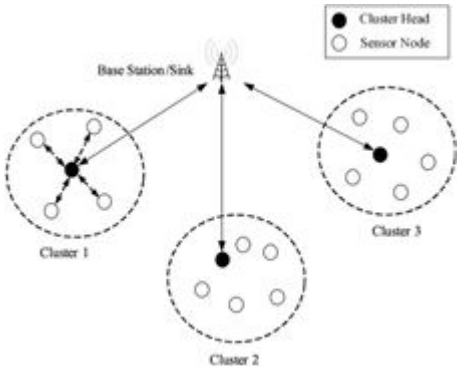


Algunas aplicaciones

- Compresión de información.
- Segmentación de imágenes.
- Sintonización de controladores.
- Organización de resultados en motores de búsqueda.
- Segmentación del hablante.
- Estadística multivariada.

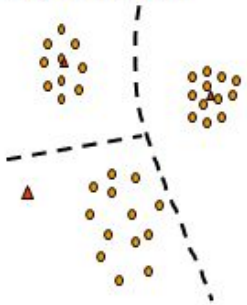




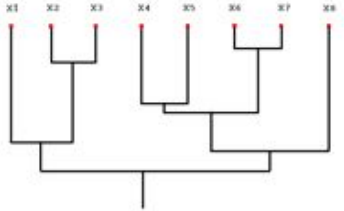


¿Cómo se clasifican según el algoritmo utilizado?

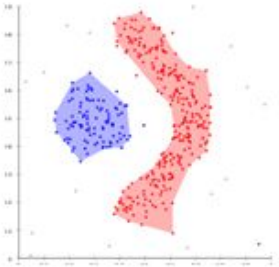
Particional:



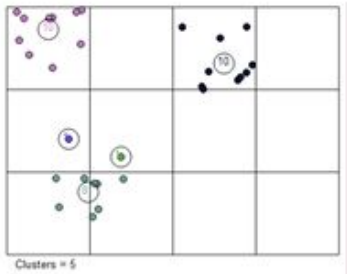
Jerárquico:



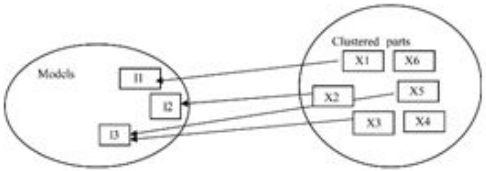
Basados en densidad:



Basados en rejillas:



Basados en modelos:



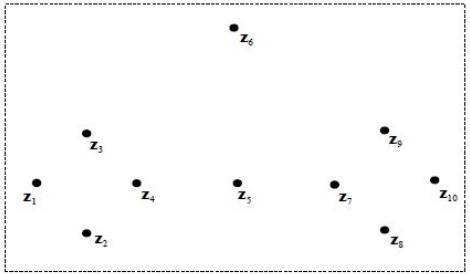
¿Cómo se clasifican según el particionamiento resultante?

Antes de hablar como se clasifican los algoritmos según el particionamiento resultante, se debe hablar de cómo se representa un resultado de un algoritmo de clustering. Para esto se define una matriz U, de ***c*** filas y **N** columnas. Cada columna representa los grados de pertenencia de la muestra **n** a cada uno de los grupos.

$$U_{c,N} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,N} \\ \cdots & \cdots & \cdots \\ u_{i,1} & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ u_{c,1} & \cdots & u_{c,N} \end{bmatrix}$$

Pertenencia de la muestra 1, al cluster i

¿Cómo se clasifican según el particionamiento resultante?



Conjunto de datos en R^2.

Particionamiento concreto (hard):

Estos métodos se basan en la teoría clásica de conjuntos, y requieren que un objeto pertenezca o no a un grupo.

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Particionamiento difuso (fuzzy):

Estos métodos permiten que los objetos pertenezcan a varios clústeres simultáneamente, con diferentes grados de pertenencia. En muchas situaciones, el agrupamiento difuso es más natural que el agrupamiento concreto.

$$U = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 0.8 & 0.5 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.5 & 0.5 & 0.8 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

Particionamiento posibilista (possibilistic):

Estos métodos eliminan la restricción de que la suma de los grados de pertenencia de un dato a los grupos sea igual a 1. Así permite la detección de outliers y la creación de nuevos grupos.

$$U = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.2 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

Algoritmo K-Means (1)

Su objetivo es la partición de un conjunto de m objetos en k grupos/clústeres. Cada observación pertenece al grupo más cercano, según alguna métrica de similaridad.

El algoritmo comienza con K clústeres, representados cada uno por un centro. Estos K centros se inicializan aleatoriamente y son vectores de dimensión $1 \, x \, p$. Donde p es el número de descriptores.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{1,2} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,p} \end{bmatrix}$$

$$C_1 = \begin{bmatrix} c_{1,1} & \cdots & c_{1,p} \end{bmatrix}$$
$$\vdots$$
$$C_k = \begin{bmatrix} c_{k,1} & \cdots & c_{k,} \end{bmatrix}$$

En realidad la matriz X, se puede representar por los K subconjuntos A_i creados, los cuales cumplen:

$$(A_i \mid 1 < i < K) \subset P(X)$$

$$\bigcup_{i=1}^{K} A_i = X$$

$$A_i \cap A_j = \emptyset$$

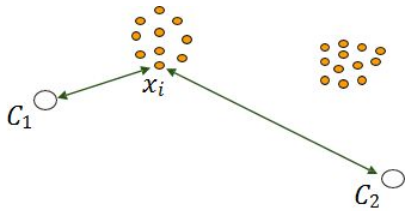
Algoritmo K-Means (2)

Luego de inicializar los centros, a cada muestra se le asigna el grupo más cercano. Para esto se utiliza una medida de distancia, en este caso, utilizaremos la distancia Euclidiana:

$$d_{i,j} = \|x_i - C_j\| = \sqrt{(x_{i,1} - c_{j,1})^2 + \ldots + (x_{i,p} - c_{j,p})^2}$$

El grupo más cercano a una muestra es aquel que tiene la menor distancia de su centro a la muestra.

Esto es gráficamente, para una muestra x_i:

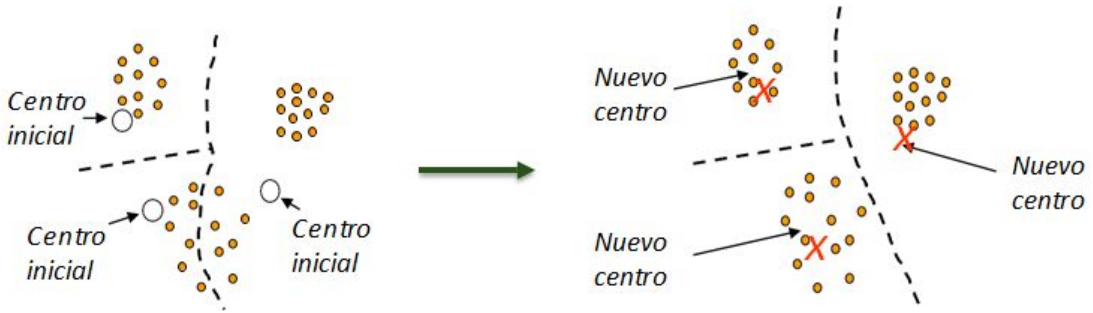


Algoritmo K-Means (3)

Una vez cada muestra asignada a su clúster más cercano, se recalculan los centros de los clústeres.

$$C_j = \frac{1}{n_i} \sum_{x_i \in C_j} x_i$$

El nuevo centro es el promedio de todos las muestras que quedaron en el C_j n_j es el número de elementos en C_j.



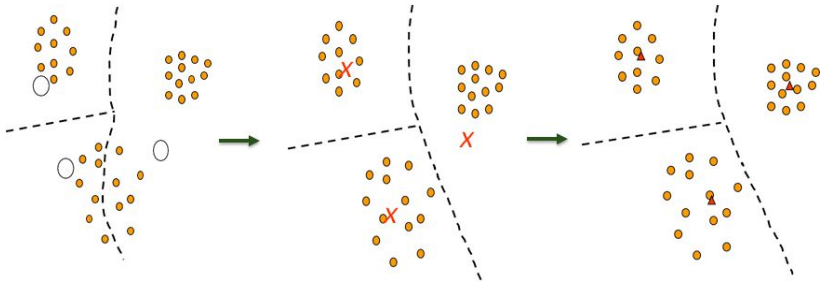
Algoritmo K-Means (4)

El proceso de asignación de las muestras a cada nuevo clúster y el proceso de volver a calcular los centros se repiten iterativamente, hasta que se llegue a un criterio de parada.

El criterio de parada más común es verificar si la suma total de las distancias de las muestras a sus respectivos clústeres no cambia.

$$S = \sum_{j} \sum_{x_i \in C_j} \|x_i - C_j\|$$

Gráficamente el algoritmo se ve así:



Algoritmo Fuzzy C-Means (1)

Similarmente al algoritmo K-Means, su objetivo es la partición de un conjunto de m objetos en K grupos/clústeres. Pero en este caso cada observación tiene cierto grado de pertenencia a cada clúster.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{1,2} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,p} \end{bmatrix}$$

$$C_1 = \begin{bmatrix} c_{1,1} & \cdots & c_{1,p} \end{bmatrix}$$
$$\vdots$$
$$C_k = \begin{bmatrix} c_{k,1} & \cdots & c_{k,} \end{bmatrix}$$

En este algoritmo es de mayor utilidad la matriz de grados de pertenencia U, donde cada columna corresponde a los grados de pertenencia de una muestra a los clústeres obtenidos.

$$U = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,N} \\ \mu_{2,1} & \mu_{1,2} & \cdots & \mu_{2,N} \\ \vdots & \vdots & & \vdots \\ \mu_{K,1} & \mu_{K,2} & \cdots & \mu_{K,N} \end{bmatrix}$$

$$\sum_{k=1}^{K} (\mu_{j,i}) = 1$$

Algoritmo Fuzzy C-Means (2)

Primero, la matriz U debe ser creada aleatoriamente. Una vez creada se utiliza la siguiente fórmula para calcular los centros (Minimización realizada por Dunn,1974 y Bezdek, 1981):

$$C = \frac{\sum_{i=1}^{N} (\mu_{j,i})^{m} x_i}{\sum_{i=1}^{N} (\mu_{j,i})^{m}}$$

Una vez calculados los centros se utiliza la siguiente fórmula para asignar a cada muestra el grado de pertenencia:

$$\mu_{j,i} = \frac{1}{\sum_{k=1}^{K} \left(\frac{\|x_i - C_j\|}{\|x_i - C_k\|} \right)^{\frac{2}{m-1}}}$$

El parámetro m se utiliza para escoger que tan difusa va a ser la partición (grado de sobrelape difuso de los clústeres). m toma valores de 1 a ∞.
Donde $m \to 1$ es una partición no difusa (K-Means), y donde $m \to \infty$ es una partición totalmente difusa.

Algoritmo Fuzzy C-Means (2)

El algoritmo funciona iterativamente como el K-Means:

- Cálculo de los centros basándose en los grados de pertenencia.
- Estimación de los grados de pertenencia a cada clúster.
- Verificar si el criterio de parada se cumple:

$$S = \sum_{j} \sum_{x_i \in C_j} (\mu_{j,i})^m \|x_i - C_j\|$$

Pero… y por qué siempre la distancia Euclidiana? (1)

De hecho la distancia Euclidiana presenta varias limitaciones. Una de ellas es que da más importancia a las variables de mayor orden (Ver “Ejemplo práctico” del pescador en Wikipedia: Distancia Malahanobis).

Un pescador quiere poder medir la similitud entre dos salmones, por ejemplo porque quiere clasificarlos en dos tipos para su venta y poder así vender los grandes más caros. Para cada salmón mide su anchura y su longitud. Con estos datos construye un vector x_i para cada salmón.

La longitud de los salmones pescados es una variable aleatoria que toma valores entre 50 y 100cm, mientras que su anchura está entre 10 y 20cm. Si el pescador usase la distancia Euclidiana, al ser las diferencias de anchura menos cuantiosas que las de longitud, les estará dando menos importancia.

$$d_{i,j} = \|x_i - x_j\| = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2}$$

Vectorialmente, esto es:

$$d_{i,j} = \|x_i - x_j\| = \sqrt{(\vec{x_i} - \vec{x_j})^T (\vec{x_i} - \vec{x_j})}$$

Pero… y por qué siempre la distancia Euclidiana? (2)

El pescador decide entonces incorporar la estadística de los datos a la medida de distancia, ponderando según su varianza: las variables con menos varianza tendrán más importancia que las de mayor varianza.

$$d_{i,j} = \|x_i - x_j\| = \sqrt{\left(\frac{x_{i,1} - x_{j,1}}{\sigma_1}\right)^2 + \left(\frac{x_{i,2} - x_{j,2}}{\sigma_2}\right)^2}$$

Esto es lo mismo vectorialmente a:

$$d_{i,j} = \|x_i - x_j\| = \sqrt{(\vec{x_i} - \vec{x_j})^T S^{-1} (\vec{x_i} - \vec{x_j})}$$

Donde S es una matriz diagonal cuyos elementos en la diagonal $S_{ii} = \sigma_i^2$

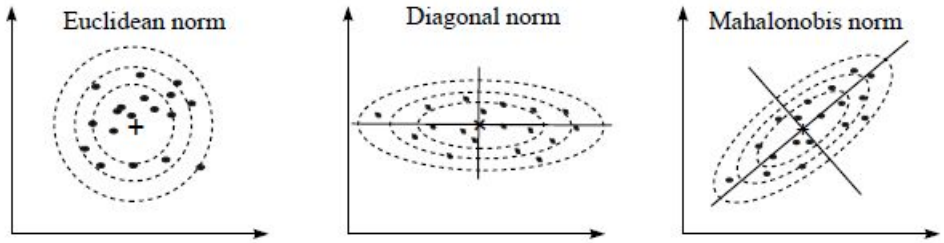
$$S = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \sigma_p^2 \end{bmatrix} \qquad S^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ 0 & 0 & \dots & \frac{1}{\sigma_p^2} \end{bmatrix}$$

Pero… y por qué siempre la distancia Euclidiana? (3)

Pero la expresión anterior tiene un problema, y es que la longitud y anchura de los salmones no son independientes; es decir, la anchura depende en cierta forma de la longitud, pues es más probable que un salmón largo sea también más ancho. Para incorporar la dependencia entre las dos variables. el pescador puede sustituir la **matriz diagonal** S por la **matriz de covarianza** $\sum$:

$$d_{i,j} = \|x_i - x_j\| = \sqrt{(\overrightarrow{x_i} - \overrightarrow{x_j})^T \Sigma^{-1} (\overrightarrow{x_i} - \overrightarrow{x_j})}$$

Diferencia entre las tres distancias:



Algoritmo Fuzzy GK-Means (1)

Una limitación común de los algoritmos de agrupación basados en distancia de diagonal es que tal distancia obliga a la función objetivo a preferir los grupos de cierta forma incluso si no están presentes en los datos.

Para solucionar este problema Gustafson and Kessel propusieron una extensión del Fuzzy C-Means utilizando una distancia adaptativa, con el fin de identificar grupos de diferentes formas geométricas.

Para cada grupo entonces se tiene una distancia asociada con una matriz de covarianza Σ_i :

$$d_{i,j\,\Sigma_i} = \|x_i - x_j\| = \sqrt{(\overrightarrow{x_i} - \overrightarrow{x_j})^T \Sigma_i^{-1} (\overrightarrow{x_i} - \overrightarrow{x_j})}$$

Permitiendo así que cada grupo se adapte a la estructura local de los datos.

Algoritmo Fuzzy GK-Means (2)

Al desarrollar el algoritmo, se tuvo que restringir el determinante d$_{\Sigma_i}$:

$$|\Sigma_i| = p_i \qquad p_i > 0$$

De esta manera se permite qu$_{\Sigma_i}$ varíe con su determinante fijo, que es lo mismo que optimizar la forma del grupo pero manteniendo su volumen constante. Luego del desarrollo matemático hecho por Gustafson and Kessel, encontraron que la matriz de covarianza para el grupo i debe ser:

$$\Sigma_i^{-1} = [\mathrm{p_i}\ \det(F_i)]^{\frac{1}{p}} F_i^{-1} \qquad F_i = \frac{\sum_{i=1}^{N} (\mu_{j,i})^m (\overrightarrow{x_i} - \overrightarrow{C_j})(\overrightarrow{x_i} - \overrightarrow{C_j})^T}{\sum_{i=1}^{N} (\mu_{j,i})^m} \qquad 1 < \mathrm{i} < \mathrm{K}$$

F_i es llamada la matriz de covarianza difusa del grupo i.
Sin ningún conocimiento previo, ρi (Volumen del cluster) simplemente se fija en 1 para cada grupo. El algoritmo GK sólo puede encontrar clusters de volúmenes aproximadamente iguales.

Algoritmo Fuzzy GK-Means (3)

El proceso del GK-Means se puede resumir en:

- Inicializar la matriz U aleatoriamente.
- Calcular los centros de los grupos con la fórmula del Fuzzy C-Means.
- Calcular las matrices difusas de covarianzas por grupo Σ_i
- Calcular la distancia de cada muestra a cada grupo utilizando para cada grupo Σ_i
- Actualizar la matriz U, teniendo en cuenta que la formula de las distancias debe utilizar Σ_i cuando se calcula la distancia a cada grupo.

1

$$U = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,N} \\ \mu_{2,1} & \mu_{1,2} & \cdots & \mu_{2,N} \\ \vdots & \vdots & & \vdots \\ \mu_{K,1} & \mu_{K,2} & \cdots & \mu_{K,N} \end{bmatrix}$$

2

$$C = \frac{\sum_{i=1}^{N} (\mu_{j,i})^m x_i}{\sum_{i=1}^{N} (\mu_{j,i})^m}$$

3

$$\Sigma_i^{-1} = [\mathrm{p_i} \det(F_i)]^{\frac{1}{p}} F_i^{-1}$$

4

$$d_{i,C_j\,\Sigma_i} = \|\overrightarrow{x_i} - \overrightarrow{C_j}\| = \sqrt{(\overrightarrow{x_i} - \overrightarrow{C_j})^T \Sigma_i^{-1} (\overrightarrow{x_i} - \overrightarrow{C_j})}$$

5

$$\mu_{j,i} = \frac{1}{\sum_{k=1}^{K} \left(\frac{d_{i,C\_j}}{d_{i,C\_k}}\right)^{\frac{2}{m-1}}}$$

Comparación Fuzzy C-Means y GK-Means

Clustering Fuzzy c-means

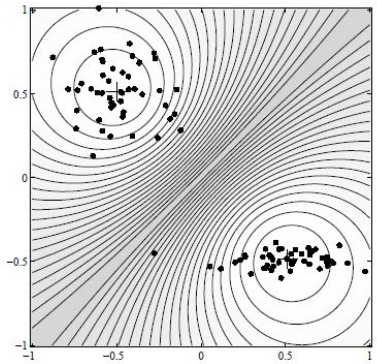


Figure 4.4. The fuzzy c-means algorithm imposes a spherical shape on the clusters, regardless of the actual data distribution. The dots represent the data points, '+' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

Clustering con Matriz de Covarianza Difusa (GK)

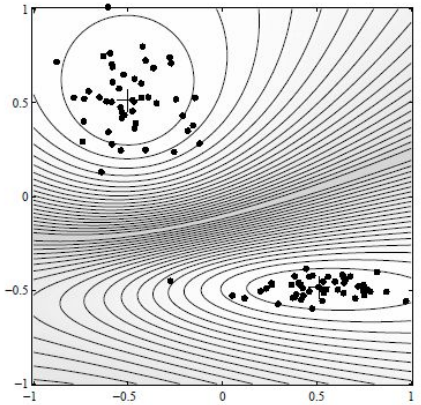
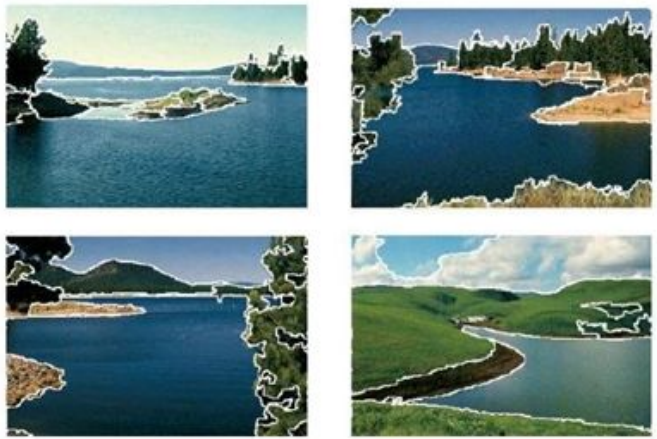


Figure 4.6. The Gustafson–Kessel algorithm can detect clusters of different shape and orientation. The points represent the data, '+' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

Ejemplo: Segmentación de Imágenes

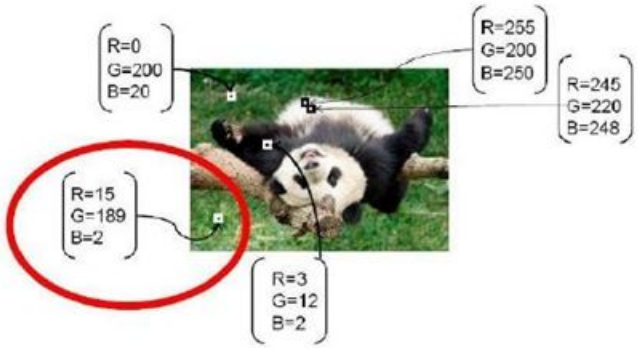
Segmentación: Dividir la imagen en regiones/secuencias con propiedades coherentes.



Un método muy conocido en la segmentación de imágenes es la segmentación basada en color por medio de algoritmos de agrupamiento.

Ejemplo: Segmentación de Imágenes

Lo que se hace es obtener los valores de las componentes RGB de los pixeles.

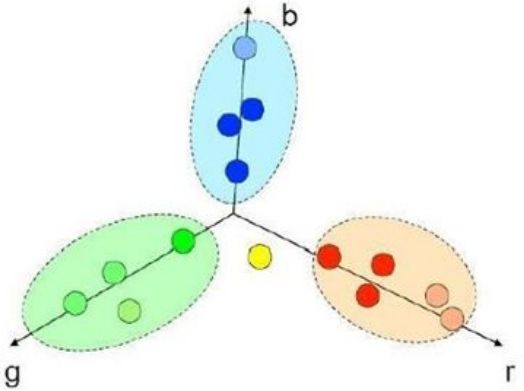


Lo que convierte cada píxel en una muestra x_j

Ejemplo: Segmentación de Imágenes

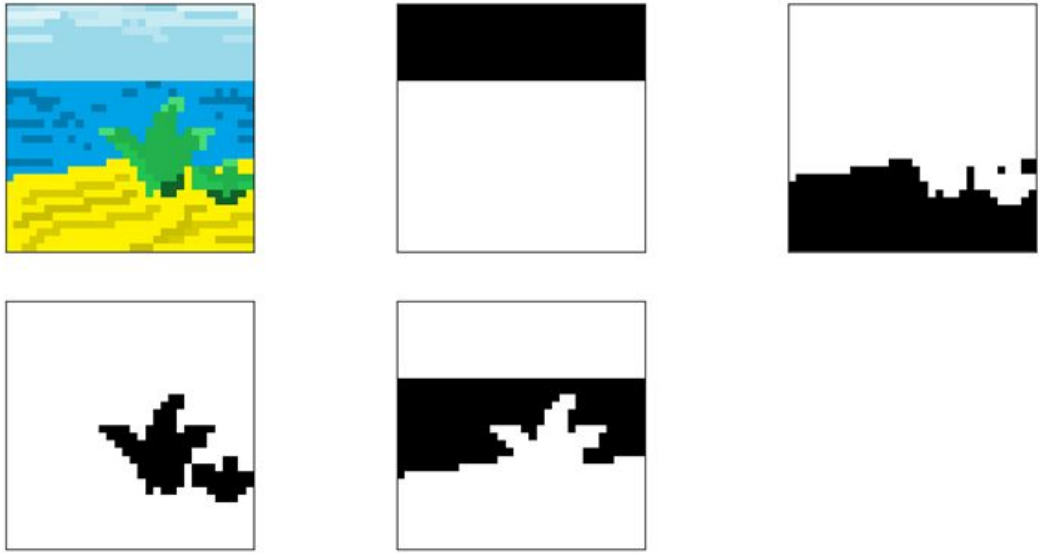
Ahora nuestra imagen es representada en un nuevo espacio, en el cual aplicaremos la técnicas de agrupamiento estudiadas para agrupar la imagen por colores.

Para agrupar objetos basados en color y en ubicación en la imagen se agrega la posición del pixel, nuestra matriz de entrenamiento X quedaría como:



$$X = \begin{bmatrix} p_1 & R_1 & G_1 & B_1 \\ p_2 & R_2 & G_2 & B_2 \\ \vdots & \vdots & \vdots & \vdots \\ p_N & R_N & G_N & B_N \end{bmatrix}$$

Ejemplo: Segmentación de Imágenes



LAMDA

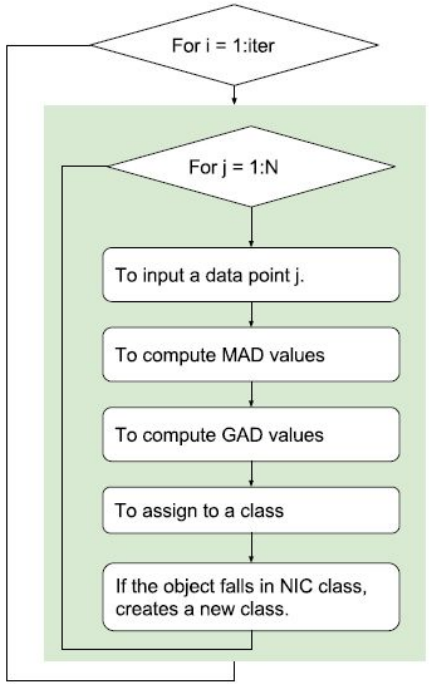


Figure 7: Flow diagram Lamda algorithm.

MADs y GADs (LAMDA)

Binomial:

$$\mathrm{MAD}(x_{i,d}\,|\,\rho_{k,d}) = \left(\rho_{k,d}\right)^{x_{i,d}} \left(1-\rho_{k,d}\right)^{(1-x_{i,d})}$$

Gaussian:

$$\mathrm{MAD}(x_{i,d}\,|\,\rho_{k,d}\,,\,\varphi_{k,d}) = \exp^{-\frac{1}{2}\left(\frac{x_{i,d}-\rho_{k,d}}{\varphi_{k,d}}\right)^2}$$

MADs y GADs (LAMDA)

Min-Max:

$$\mathrm{GAD}(x_i\,|\,k) = (\alpha)\min\left(\mathrm{MAD}(x_{i,1}\,|\,k), ..., \mathrm{MAD}(x_{i,p}\,|\,k)\right) + (1-\alpha)\max\left(\mathrm{MAD}(x_{i,1}\,|\,k), ..., \mathrm{MAD}(x_{i,p}\,|\,k)\right)$$

3 PI:

$$\mathrm{GAD}(x_i\,|\,k) = \frac{\prod_{d=1}^{p}\mathrm{MAD}(x_{i,d}\,|\,k)}{\prod_{d=1}^{p}\mathrm{MAD}(x_{i,d}\,|\,k) + \prod_{d=1}^{p}\left[1-\mathrm{MAD}(x_{i,d}\,|\,k)\right]}$$