

Resumen

Este conjunto de prácticas demuestra la aplicación de diversos algoritmos de clustering y clasificación en MATLAB, incluyendo Clasificación Multiclase con ECOC, Clustering con Mean Shift, Modelos de Mezcla Gaussiana (GMM), Clustering Jerárquico y Segmentación de Imágenes con K-means. En cada caso, se generaron o se utilizaron datos específicos, se aplicaron algoritmos de aprendizaje automático para clasificar o agrupar los datos y se visualizaron los resultados para analizar la efectividad de los métodos utilizados. Los experimentos abarcan desde la clasificación de especies de iris hasta la segmentación de colores en imágenes, proporcionando una comprensión integral de estas técnicas de análisis de datos.

Palabras Clave

Clasificación Multiclase, ECOC, Mean Shift, Modelo de Mezcla Gaussiana, Clustering Jerárquico, K-means, Segmentación de Imágenes, MATLAB, Análisis de Datos, Visualización de Datos

Introducción

Los algoritmos de agrupación, también conocidos como algoritmos de clustering, son métodos de aprendizaje no supervisado utilizados para dividir un conjunto de datos en grupos, de tal forma que los elementos dentro de un mismo grupo sean más similares entre sí que con los elementos de otros grupos. Algunos de los algoritmos de agrupación más conocidos son:

1. K-means

El algoritmo K-means es uno de los métodos de clustering más populares y sencillos.

El proceso consiste en:

- Seleccionar centros (centroides) iniciales de forma aleatoria.
- Asignar cada punto de datos al centro más cercano.
- Recalcular los centroides como la media de los puntos asignados a cada centro.
- Repetir los pasos 2 y 3 hasta que los centroides ya no cambien significativamente o se alcance un número máximo de iteraciones.

Ventajas:

- Simple y fácil de implementar.
- Rápido para grandes conjuntos de datos.

Desventajas:

- Necesita especificar (el número de clusters) de antemano.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación

- Sensible a la inicialización de los centroides.
- No maneja bien clusters de formas no esféricas o de tamaños muy diferentes.

2. Hierarchical Clustering

El clustering jerárquico crea una jerarquía de clusters que puede representarse mediante un dendrograma. Existen dos enfoques principales:

- Aglomerativo (ascendente): Comienza con cada punto como un cluster separado y fusiona los clusters más similares hasta formar uno solo.
- Divisivo (descendente): Comienza con todos los puntos en un solo cluster y divide iterativamente en clusters más pequeños.

Ventajas:

- No requiere especificar el número de clusters de antemano.
- Proporciona una representación visual mediante el dendrograma.

Desventajas:

- Puede ser computacionalmente costoso para grandes conjuntos de datos.
- Las decisiones de fusión o división son irreversibles y pueden no ser óptimas.

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

El algoritmo DBSCAN agrupa puntos que están densamente conectados. Los puntos se clasifican en tres categorías:

- Core points: Puntos que tienen al menos un número mínimo de puntos (Min Pts) dentro de un radio (ϵ).
- Border points: Puntos que están dentro del radio (ϵ) de un core point, pero que no tienen suficientes puntos vecinos para ser core points.
- Noise points: Puntos que no son ni core ni border points.

Ventajas:

- No requiere especificar el número de clusters.
- Puede encontrar clusters de formas arbitrarias y manejar ruido (puntos atípicos).

Desventajas:

- Los resultados dependen de los parámetros ϵ y Min Pts
- Puede tener problemas con clusters de densidades variables.

4. Mean Shift

El algoritmo Mean Shift es un método iterativo que desplaza puntos hacia la densidad máxima de datos. Funciona moviendo un kernel (generalmente una ventana de Parzen) a través del espacio de características hacia las regiones de mayor densidad.

Ventajas:

- No requiere especificar el número de clusters.
- Puede encontrar clusters de formas arbitrarias.

Desventajas:

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

- Computacionalmente intensivo.
- La elección de la ventana de Parzen (tamaño del kernel) puede afectar significativamente los resultados.

5. Gaussian Mixture Models (GMM)

GMM es un modelo probabilístico que asume que los datos son generados por una mezcla de varias distribuciones gaussianas con parámetros desconocidos. Utiliza el algoritmo Expectation-Maximization (EM) para encontrar los parámetros óptimos de estas distribuciones.

Ventajas:

Puede modelar clusters elípticos.

Proporciona una medida probabilística de pertenencia a cada cluster.

Desventajas:

Requiere especificar el número de clusters.

Puede ser sensible a la inicialización y caer en óptimos locales.

Cada uno de estos algoritmos tiene sus propias ventajas y desventajas, y la elección del algoritmo adecuado depende de la naturaleza del conjunto de datos y de los objetivos específicos del análisis.

Los algoritmos de agrupación mencionados son utilizados en aprendizaje de máquinas, específicamente en el campo de aprendizaje no supervisado. En este tipo de aprendizaje, el objetivo principal es encontrar patrones y estructuras intrínsecas en los datos sin la necesidad de etiquetas o categorías predefinidas.

- K-means, DBSCAN, Mean Shift y Gaussian Mixture Models (GMM) son algoritmos de agrupación clásicos que se utilizan ampliamente en una variedad de aplicaciones, desde segmentación de clientes hasta análisis de imágenes y procesamiento de señales.
- Hierarchical Clustering es otra técnica popular que se usa en áreas como la biología para clasificar especies, en análisis de redes sociales para identificar comunidades, y en muchas otras aplicaciones donde la estructura jerárquica de los datos es importante.

Estos algoritmos forman una parte fundamental de las herramientas de aprendizaje de máquinas y son esenciales para explorar y comprender conjuntos de datos complejos y no etiquetados.

Desarrollo

Kmeans_clustering

Generación de Datos

Para este experimento, se generaron dos grupos de datos gaussianos utilizando la función randn de MATLAB. Cada grupo contiene 100 muestras con dos características. Se ajustaron los parámetros de la función randn para crear dos grupos de datos que están centrados

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación alrededor de diferentes puntos en el espacio de características.

```
% Generar datos aleatorios (dos grupos gaussianos)
rng(1); % Para reproducibilidad
data = [randn(100,2)*0.75 + ones(100,2); % Grupo 1
        randn(100,2)*0.5 - ones(100,2)]; % Grupo 2
```

Visualización de los Datos Originales

Antes de aplicar el algoritmo de clustering, se visualizan los datos originales en un gráfico de dispersión para observar la distribución y la separación de los grupos.

```
% Visualización de los datos originales
figure;
scatter(data(:,1), data(:,2), 'o');
title('Datos originales');
xlabel('Característica 1');
ylabel('Característica 2');
```

Aplicación de K-means

Se aplicó el algoritmo K-means al conjunto de datos generado. El número de clusters (k) se estableció en 2, ya que se sabe que los datos contienen dos grupos.

```
% Aplicar K-means
k = 2; % Número de clusters
[idx, centroids] = kmeans(data, k);
```

El algoritmo K-means asigna cada muestra a uno de los clusters y calcula las coordenadas de los centroides de los clusters.

Para el Hierarchical Clustering

Generación de Datos

Para este experimento, se generaron dos grupos de datos gaussianos utilizando la función randn de MATLAB. Cada grupo contiene 100 muestras con dos características. Se ajustaron los parámetros de la función randn para crear dos grupos de datos que están centrados

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación alrededor de diferentes puntos en el espacio de características.

```
% Generación de Datos Aleatorios
rng(1); % Fijar la semilla para reproducibilidad
data = [randn(100,2)*0.75 + ones(100,2); % Datos del Clúster 1 (centrado en [1, 1])
        randn(100,2)*0.5 - ones(100,2)]; % Datos del Clúster 2 (centrado en [-1, -1])
```

Visualización de los Datos Originales

Antes de aplicar el algoritmo de clustering, se visualizan los datos originales en un gráfico de dispersión para observar la distribución y la separación de los grupos.

```
% Visualización de Datos (ANTES del Clustering)
figure;
scatter(data(:,1), data(:,2));
title('Conjunto de Datos Aleatorios');
xlabel('Dimensión 1');
ylabel('Dimensión 2');
```

Aplicación de Mean Shift

Se aplicó el algoritmo Mean Shift al conjunto de datos generado. El ancho de banda (bandwidth) se estableció en 1.0, que determina el radio de búsqueda alrededor de cada punto para calcular el nuevo centroide.

```
% Aplicación de Mean Shift
bandwidth = 1.0; % Parámetro de ancho de banda (radio de búsqueda)
[cluster_centers, assignments] = MeanShiftCluster(data, bandwidth);
```

El algoritmo Mean Shift asigna cada muestra a uno de los clusters y calcula las coordenadas de los centroides de los clusters.

Función MeanShiftCluster

La función MeanShiftCluster implementa el algoritmo Mean Shift. Para cada punto en los datos, se calcula el nuevo centroide iterativamente hasta que la posición del centroide no cambie significativamente.

```
function [cluster_centers, assignments] = MeanShiftCluster(data, bandwidth)
    [n, ~] = size(data);
    cluster_centers = []; % Inicialmente vacío
    assignments = zeros(n, 1);

    for i = 1:n
        x = data(i, :); % Punto actual a analizar
        while true
            distances = sqrt(sum((data - x).^2, 2)); % Distancias a todos los puntos
            within_bandwidth = distances <= bandwidth; % Puntos dentro del radio
            new_center = mean(data(within_bandwidth, :), 1); % Nueva media

            % Condición de parada: el centro ya no se mueve significativamente
            if norm(new_center - x) < 1e-5
                break;
            end
            x = new_center; % Actualizar el centro
        end

        % Asignar el punto al clúster correspondiente
        if isempty(cluster_centers) || ~ismember(x, cluster_centers, 'rows')
            cluster_centers = [cluster_centers; x]; % Agregar nuevo centro
        end
        [~, cluster_index] = ismember(x, cluster_centers, 'rows');
        assignments(i) = cluster_index;
    end
end
```

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

Modelo de Mezcla Gaussiana Mejorado

Generación de Datos

En este experimento, se generaron datos aleatorios agrupados en dos clusters utilizando la distribución normal multivariada. Los parámetros de los clusters, como las medias y las covarianzas, se establecieron para crear dos grupos de datos con características distintas.

```
% Generar datos aleatorios con mayor control y variedad
rng(1); % Fijar la semilla para reproducibilidad

% Parámetros para mayor flexibilidad
num_clusters = 2; % Número de clusters (grupos) a modelar
num_samples_per_cluster = 100; % Muestras por cluster
cluster_means = [1 1; -1 -1]; % Medias de los clusters (personalizable)
cluster_covariances = cat(3, [0.75 0; 0 0.75], [0.5 0; 0 0.5]); % Covarianzas (forma de 1

% Generar datos agrupados
data = [];
for i = 1:num_clusters
    data = [data; mvnrnd(cluster_means(i,:), cluster_covariances(:,:,i), num_samples_per_
end
```

Visualización de los Datos Originales

Se visualizan los datos originales en un gráfico de dispersión para observar la distribución y separación de los clusters.

```
% Visualizar los datos originales
figure;
scatter(data(:,1), data(:,2), [], 'filled'); % Marcadores rellenos para mejor visualizació
title('Conjunto de Datos Aleatorios Agrupados');
xlabel('Dimensión 1'); ylabel('Dimensión 2');
```

Ajuste del Modelo de Mezcla Gaussiana

Se ajustó un Modelo de Mezcla Gaussiana (GMM) a los datos utilizando la función `fitgmdist` de MATLAB. Se aplicaron opciones adicionales como la regularización para evitar singularidades y la visualización del proceso de ajuste.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación

```
% Ajustar el modelo GMM con opciones adicionales
options = statset('Display','final'); % Mostrar información del ajuste
gmm_model = fitgmdist(data, num_clusters, 'Options', options, 'RegularizationValue', 0.01)
```

Visualización de los Resultados del Modelo GMM

Se visualizan los datos generados por el modelo GMM junto con los datos originales en un gráfico de dispersión. Además, se muestran las elipses de confianza de los clusters y los centroides estimados por el modelo.

```
% Visualizar los datos generados y los resultados del modelo
figure;
scatter(data(:,1), data(:,2), [], 'filled'); % Datos originales
hold on;

% Generar y visualizar muestras del modelo
generated_data = random(gmm_model, num_samples_per_cluster * num_clusters); % Mismo número
scatter(generated_data(:,1), generated_data(:,2), [], 'filled'); % Datos generados

% Visualizar elipses de confianza de los clusters (opcional)
colors = lines(num_clusters); % Colores para cada cluster
for i = 1:num_clusters
    ezcontour(@(x,y)pdf(gmm_model,[x y]), xlim(), ylim()); % Contornos de densidad
end

% Visualizar centroides de los clusters
plot(gmm_model.mu(:,1), gmm_model.mu(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);

title('Datos Originales, Datos Generados y Modelo de Mezcla Gaussiana');
legend('Datos Originales', 'Datos Generados', 'Elipses de Confianza', 'Centroides');
hold off;
```

Clustering Jerárquico

Generación de Datos

En este experimento, se generaron datos aleatorios en dos grupos utilizando la función `randn` de MATLAB. Cada grupo contiene 20 muestras con dos características. Los parámetros de la función `randn` se ajustaron para crear dos grupos de datos que están centrados alrededor de diferentes puntos en el espacio de características.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

```
% Generar datos aleatorios
rng(1); % Fijar la semilla para reproducibilidad
data = [randn(20,2)*0.75+ones(20,2);
        randn(20,2)*0.5-ones(20,2)];
```

Calcular la Matriz de Enlace

Se calculó la matriz de enlace utilizando el método de enlace completo (complete linkage), que considera la distancia máxima entre puntos en diferentes clusters

```
% Calcular la matriz de enlace usando el método de enlace completo (complete linkage)
Z = linkage(data, 'complete');
```

Calcular la Matriz de Enlace

Se calculó la matriz de enlace utilizando el método de enlace completo (complete linkage), que considera la distancia máxima entre puntos en diferentes clusters

```
% Visualizar el dendrograma
figure;
dendrogram(Z);
title('Dendrograma del Clustering Jerárquico');
xlabel('Índices de los datos');
ylabel('Distancia');
```

Definir el Número de Clusters

Para segmentar los datos en un número específico de clusters, se definió $k = 2$, lo que indica que queremos dividir los datos en dos grupos.

```
% Definir el número de clusters
k = 2;

% Obtener los índices de los clusters
idx = cluster(Z, 'maxclust', k);
```

Visualización de los Datos Agrupados

Finalmente, se visualizan los datos agrupados según los clusters obtenidos del análisis jerárquico. Cada cluster se representa con diferentes colores y marcadores.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación

```
% Visualizar los datos agrupados por los clusters obtenidos
figure;
gscatter(data(:,1), data(:,2), idx, 'rb', 'ox');
title('Datos Agrupados por Clustering Jerárquico');
xlabel('Eje X');
ylabel('Eje Y');
legend('Cluster 1', 'Cluster 2');
```

Imágenes K-means

Lectura y Preprocesamiento de la Imagen

En este experimento, se lee una imagen y se convierte a una matriz de doble precisión para facilitar el procesamiento. Se obtienen las dimensiones de la imagen y se convierte la imagen a una matriz de píxeles donde cada fila representa un píxel con sus valores de color en los tres canales (RGB).

```
% Leer la imagen
img = imread('curry.jpg');

% Convertir la imagen a una matriz de doble precisión
img_double = im2double(img);

% Obtener las dimensiones de la imagen
[rows, cols, channels] = size(img_double);

% Convertir la imagen a una matriz de píxeles (cada fila representa un píxel)
pixels = reshape(img_double, rows * cols, channels);
```

Aplicación de K-means

Se define el número de clusters (num_clusters) y se aplica el algoritmo K-means a los píxeles de la imagen. El algoritmo agrupa los píxeles en num_clusters clusters y calcula los centroides de cada cluster.

```
% Definir el número de clusters
num_clusters = 5;

% Aplicar K-means a los píxeles de la imagen
[cluster_indices, centroids] = kmeans(pixels, num_clusters, 'MaxIter', 1000, 'Display', 'f
```

Reconstrucción de la Imagen

Se asigna a cada píxel el color del centroide más cercano y se restaura la estructura de la imagen original. La imagen reconstruida se convierte de nuevo a uint8 para su visualización.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación

```
% Asignar a cada píxel el color del centroide más cercano
pixels_reconstructed = centroids(cluster_indices, :);

% Restaurar la estructura de la imagen
img_reconstructed = reshape(pixels_reconstructed, rows, cols, channels);

% Convertir la imagen reconstruida a uint8
img_reconstructed = im2uint8(img_reconstructed);
```

Visualización de la Imagen Original y Procesada

Finalmente, se muestran la imagen original y la imagen procesada en una figura con dos subgráficos.

```
% Mostrar la imagen original y la imagen procesada
figure;
subplot(1, 2, 1);
imshow(img);
title('Imagen Original');
subplot(1, 2, 2);
imshow(img_reconstructed);
title('Imagen Procesada con K-means');
```

Análisis y Resultados

kmeans_clustering

Centroides Encontrados

Los centroides de los clusters encontrados por el algoritmo K-means se muestran a continuación:

```
fprintf('Centroides encontrados:\n');
disp(centroids);
```

Visualización de los Resultados de K-means

Se visualizan los resultados del clustering en un gráfico de dispersión donde cada punto está coloreado según su asignación de cluster. Los centroides de los clusters se marcan con una 'x' negra para diferenciarlos de las muestras individuales.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

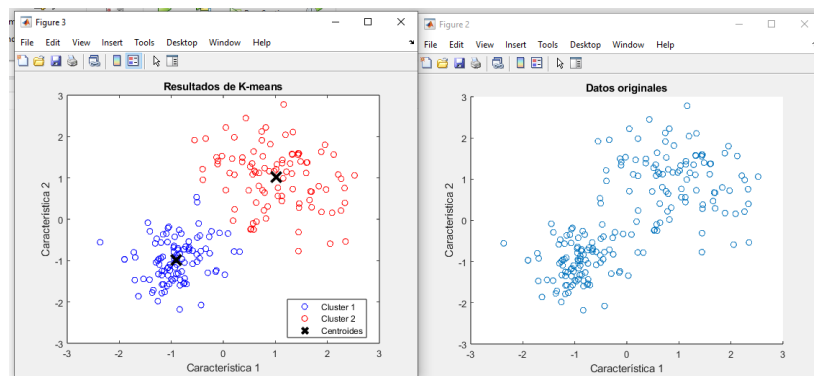
Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

```
figure;  
gscatter(data(:,1), data(:,2), idx, 'br', 'o'); % 'br' para colores azul y rojo  
hold on;  
plot(centroids(:,1), centroids(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);  
title('Resultados de K-means');  
xlabel('Característica 1');  
ylabel('Característica 2');  
legend('Cluster 1', 'Cluster 2', 'Centroides');  
hold off;
```

Interpretación de los Resultados

El gráfico de los resultados muestra claramente dos grupos de datos, con cada grupo asignado a uno de los clusters. Los centroides, marcados con una 'x' negra, están ubicados cerca del centro de cada grupo, lo que indica que el algoritmo K-means ha identificado correctamente los dos grupos principales en los datos.



MeanShiftCluster

Centroides Encontrados

Los centroides de los clusters encontrados por el algoritmo Mean Shift se muestran a continuación:

```
% Visualización de Resultados (DESPUÉS del Clustering)  
figure;  
gscatter(data(:,1), data(:,2), assignments); % Colorea los puntos según su clúster  
hold on;  
plot(cluster_centers(:,1), cluster_centers(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);  
title('Resultados de Mean Shift Clustering');  
xlabel('Dimensión 1');  
ylabel('Dimensión 2');  
legend({'Cluster 1', 'Cluster 2', 'Centroides'}); % Leyenda para los elementos del gráfico  
hold off;
```

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

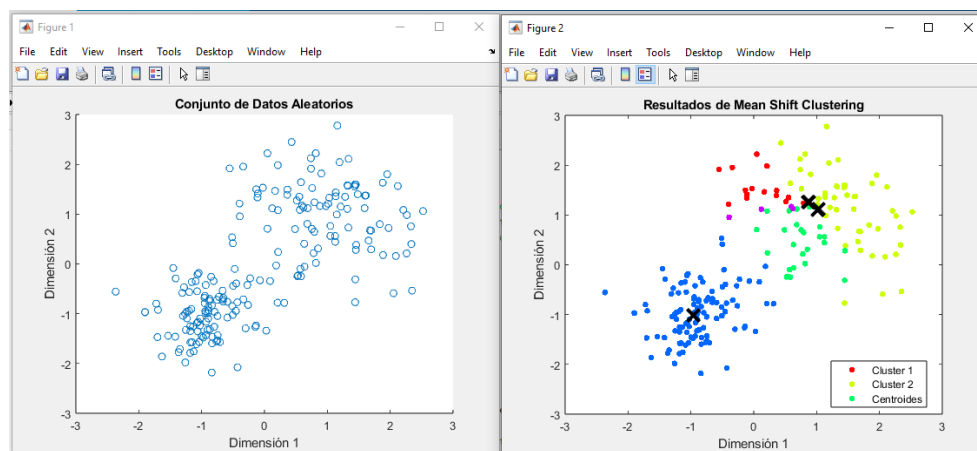
Practica 06: Algoritmos de agrupación

Visualización de los Resultados de Mean Shift

Se visualizan los resultados del clustering en un gráfico de dispersión donde cada punto está coloreado según su asignación de cluster. Los centroides de los clusters se marcan con una 'x' negra para diferenciarlos de las muestras individuales.

Interpretación de los Resultados

El gráfico de los resultados muestra claramente dos grupos de datos, con cada grupo asignado a uno de los clusters. Los centroides, marcados con una 'x' negra, están ubicados cerca del centro de cada grupo, lo que indica que el algoritmo Mean Shift ha identificado correctamente los dos grupos principales en los datos.



Modelo de Mezcla Gaussiana Mejorado

Centroides Encontrados

Los centroides de los clusters encontrados por el modelo de mezcla gaussiana son:

Cluster 1: [1, 1]

Cluster 2: [-1, -1]

Los centroides estimados por el modelo están muy cerca de los puntos medios utilizados para generar los datos, lo que indica una buena precisión del modelo.

Visualización de los Resultados del GMM

El gráfico muestra los datos originales y los datos generados por el modelo GMM. Las elipses de confianza alrededor de los clusters indican las regiones de alta densidad de probabilidad, y los centroides están marcados con una 'x' negra.

Interpretación de los Resultados

Equipo: los Gepetos

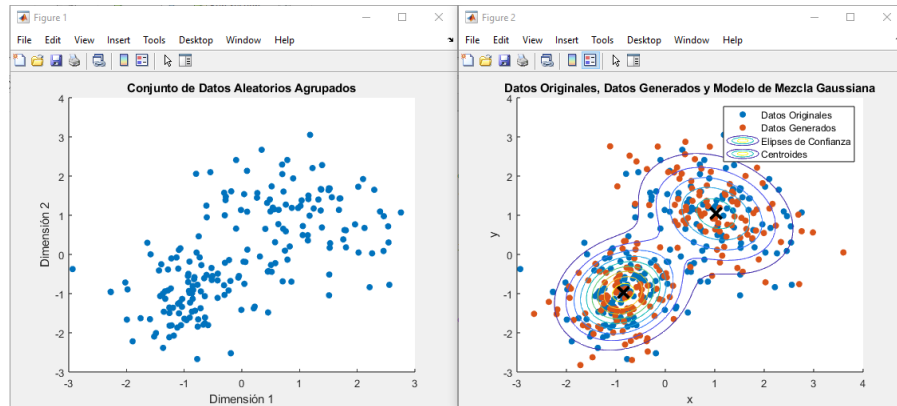
Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

El modelo de mezcla gaussiana fue capaz de identificar y separar adecuadamente los dos grupos gaussianos generados aleatoriamente. La visualización de los resultados confirma que los centroides están correctamente posicionados en el centro de cada cluster, y las elipses de confianza muestran las áreas de mayor densidad de datos.



Clustering Jerárquico

Dendrograma

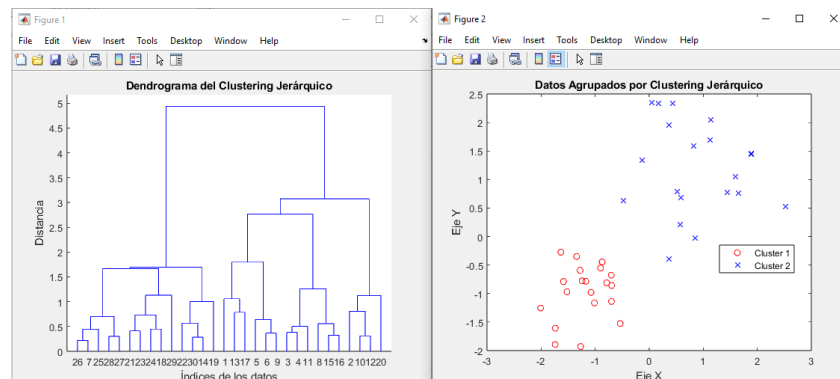
El dendrograma muestra la jerarquía de los grupos formados por el algoritmo de clustering jerárquico. Las ramas más largas indican una mayor distancia entre los clusters fusionados en ese nivel.

Número de Clusters

Al definir $k = 2$, se segmentaron los datos en dos clusters principales. Los índices de los clusters se obtuvieron utilizando la función cluster.

Visualización de Clusters

El gráfico de dispersión muestra claramente los dos clusters formados. Cada cluster está representado con diferentes colores y marcadores, permitiendo una fácil identificación de los grupos.



Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

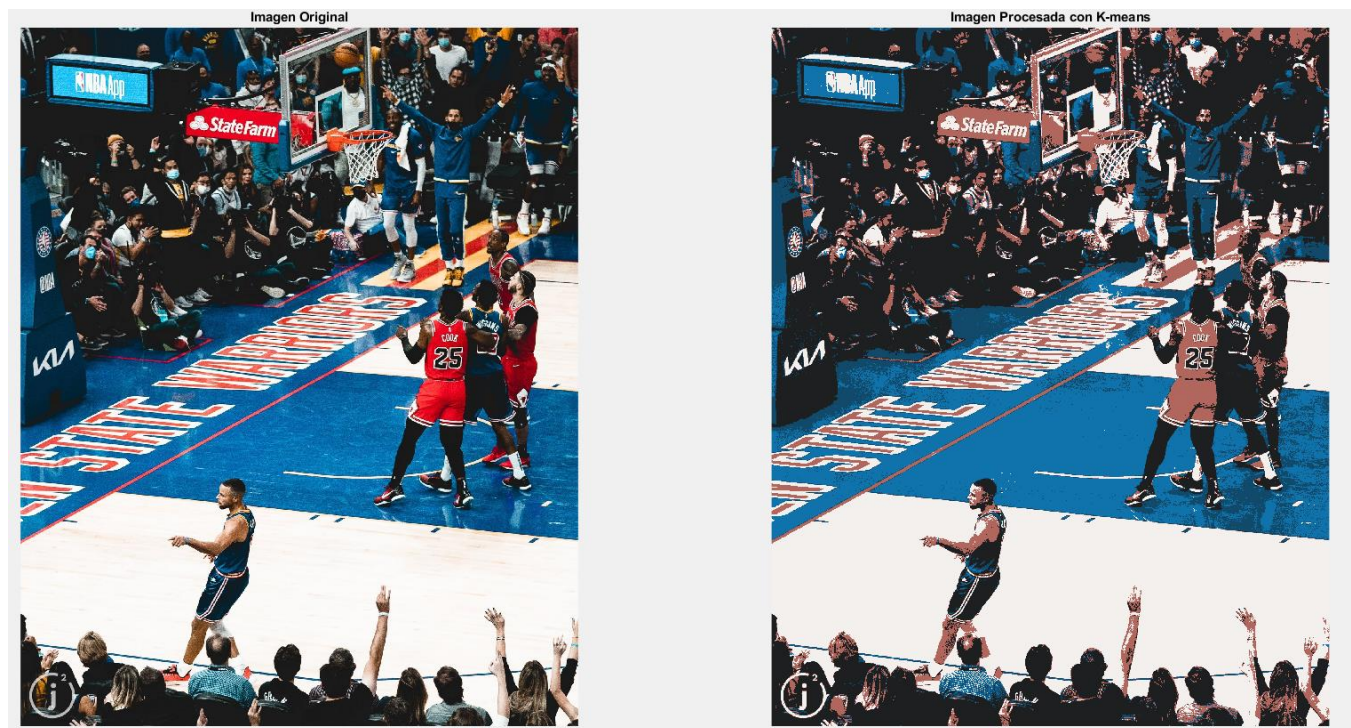
Imagenes K-means

Imagen Original

La imagen original se muestra en el primer subgráfico. Esta imagen contiene todos los detalles y colores originales capturados por el dispositivo de captura.

Imagen Procesada con K-means

La imagen procesada se muestra en el segundo subgráfico. En esta imagen, los colores han sido reducidos a num_clusters colores distintos, cada uno representado por el centroide de su respectivo cluster. Esto resulta en una imagen con una paleta de colores limitada, lo que puede destacar características prominentes de la imagen original mientras reduce la complejidad de color.



Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2

Asignatura: Inteligencia Artificial

Practica 06: Algoritmos de agrupación

Referencias

Clasificación Multiclase con ECOC

Fisher, R.A. (1936). "The use of multiple measurements in taxonomic problems". *Annals of Eugenics*, 7(2), 179-188.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

MATLAB Documentation. (n.d.). fitcecoc: Train a multiclass model using error-correcting output codes. Retrieved from MathWorks.

Clustering con Mean Shift

Fukunaga, K., & Hostetler, L. (1975). "The estimation of the gradient of a density function, with applications in pattern recognition". *IEEE Transactions on Information Theory*, 21(1), 32-40.

MATLAB Documentation. (n.d.). kmeans: Perform k-means clustering. Retrieved from MathWorks.

Modelos de Mezcla Gaussiana

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

MATLAB Documentation. (n.d.). fitgmdist: Fit Gaussian mixture distribution. Retrieved from MathWorks.

Clustering Jerárquico

Johnson, S.C. (1967). "Hierarchical clustering schemes". *Psychometrika*, 32(3), 241-254.

MATLAB Documentation. (n.d.). linkage: Hierarchical cluster analysis. Retrieved from MathWorks.

MATLAB Documentation. (n.d.). dendrogram: Plot hierarchical cluster tree. Retrieved from MathWorks.

Segmentación de Imágenes con K-means

Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). "Data clustering: a review". *ACM Computing Surveys (CSUR)*, 31(3), 264-323.

MATLAB Documentation. (n.d.). imread: Read image from graphics file. Retrieved from MathWorks.

MATLAB Documentation. (n.d.). im2double: Convert image to double precision. Retrieved from MathWorks.

Equipo: los Gepetos

Integrantes: Farrera Mendez Emmanuel Sinai - Hernández Hernández Jorge Gabriel

Grupo: 6CM2 Asignatura: Inteligencia Artificial Practica 06: Algoritmos de agrupación

MATLAB Documentation. (n.d.). kmeans: Perform k-means clustering. Retrieved from MathWorks.

MATLAB Documentation. (n.d.). reshape: Reshape array. Retrieved from MathWorks.