Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Tarea 3

Victor Manuel Gómez Espinosa

23 de marzo de 2020

1. Problema 1

Para el problema de clustering y mezclas de Gaussianas se tomaron los datos sintéticos de la Figura 1.1.

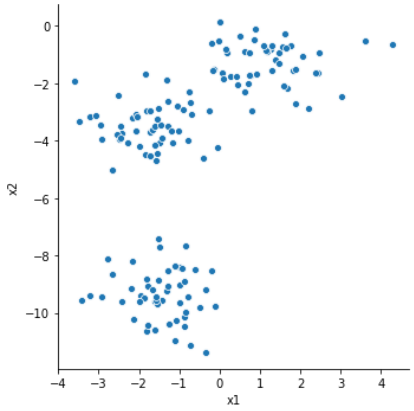

Figura 1.1: Datos sintéticos en dos dimensiones.

Para estos datos sus estimadores de máxima verosimilitud y log-verosimilitud respectivamente son:

$$\bar{X} = [-0.69, -4.72], S = \begin{bmatrix} 2.48 & 2.81 \\ 2.81 & 12.25 \end{bmatrix}, \ell = -822.93$$

Posteriormente se implementó un método de clustering usando el algoritmo **MMG-EM** en Python (se agrega el código, así como ejemplos de uso en el Anexo A) y se probó en el conjunto de datos de la Figura 1.1 y se comparó contra *fuzzy k-means* (Figura 1.2).

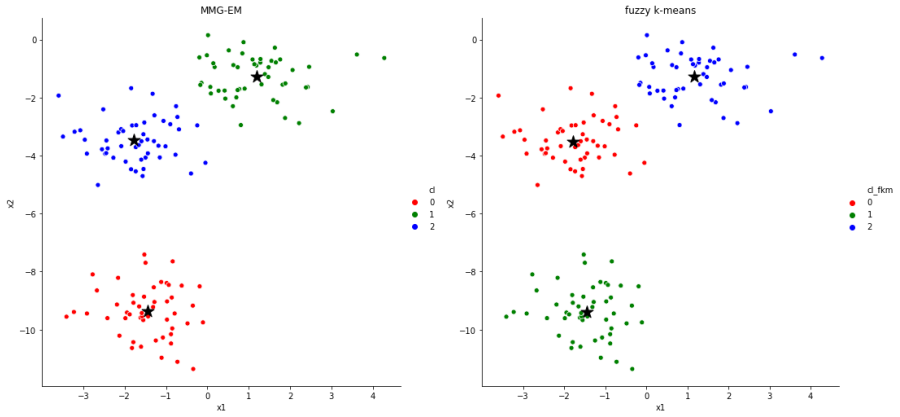

Figura 1.2: a) clustering MMG-EM, b) clustering fuzzy k-means.

Observe la Figura 1.2 y note que los centroides de ambos son prácticamente los mismos, sin embargo, los clusters no lo son para cada grupo, es decir, se separa bien los 3 grupos, pero la 'etiqueta' es diferente en cada método.

Ahora si, por ejemplo, consideramos otro conjunto de datos sintéticos en el cual la matriz de covarianza esférica respectiva a cada grupo sea la misma (o prácticamente la misma) y sus varianzas son muy pequeñas tal que tienden a 0, o, dicho de otra forma, que las nubes o grupos de puntos están prácticamente juntos sin mucha dispersión (Observe la Figura 1.3).

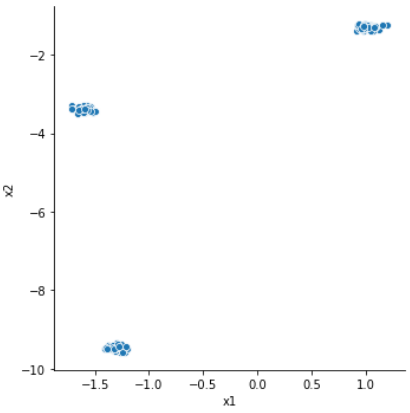

Figura 1.3: Datos sintéticos con poca dispersión ($\sigma^2 \to 0$)

Ahora volvimos a probar el algoritmo **MMG-EM** en el conjunto de datos y se comparó contra *k-means,* observe la Figura 1.4 y note que en este caso las etiquetas sí coinciden, o, dicho de otra forma, bajo estas condiciones el algoritmo **MMG-EM** converge a *k-means.*

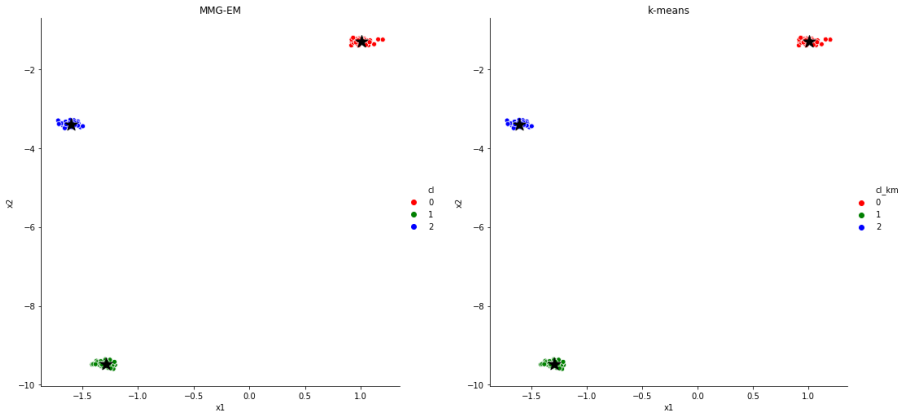

Figura 1.4: a) clustering MMG-EM, b) clustering k-means.

Como podemos observar en ambos casos fue bueno el método de clustering MMG-EM, pero ahora consideremos otros conjuntos de datos (Figura 1.5).

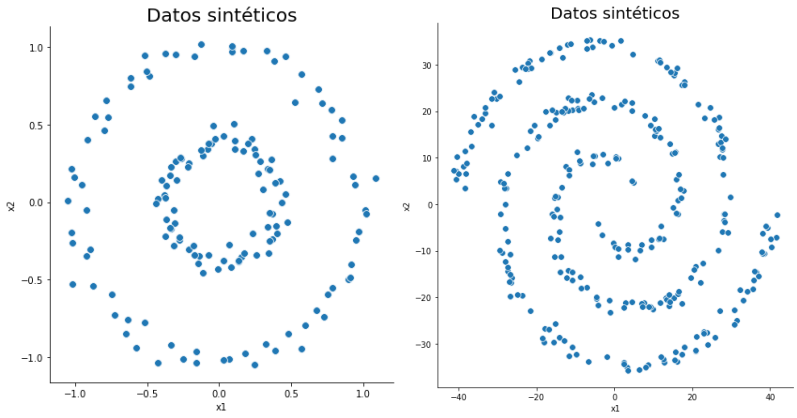

Figura 1.5: Dos conjuntos de datos sintéticos donde hay 2 grupos diferentes.

Posteriormente volvimos a probar el algoritmo **MMG-EM** en ambos conjuntos de datos y se comparó contra *k-means* y *fuzzy k-means* (Figura 1.6).

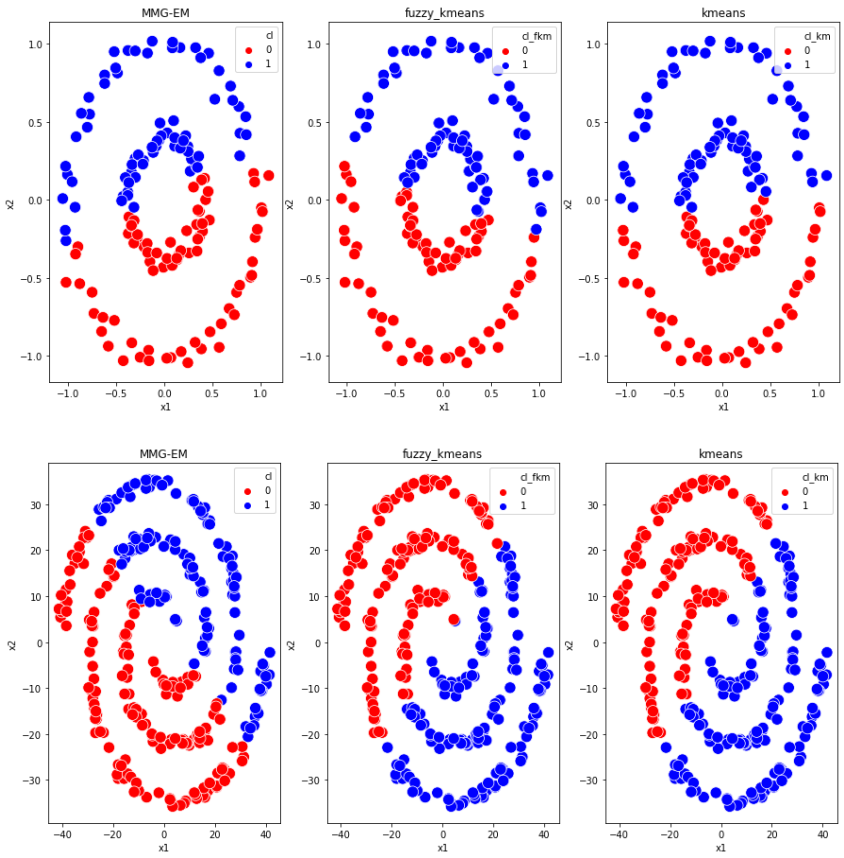

Figura 1.6: a) clustering MMG-EM, b) clustering fuzzy k-means, c) clustering k-means.

Observe la Figura 1.6 y note que en estos datos ninguno de los métodos anteriores resultó adecuado, mostrando resultados muy similares en algunos casos.

Como comentarios adicionales sobre el algoritmo **MMG-EM**, se pudo observar que el desempeño del algoritmo (es decir la convergencia), depende mucho de cómo se inicialice y puede en algunos casos tener problemas por valores o muy grandes o pequeños en los estimadores de máxima verosimilitud y por consecuencia también problemas de matrices singulares que hacen que falle, pero en cambio cuando se inicializa correctamente este puede converger satisfactoriamente durante las primeras iteraciones.

2. Problema 2

Para este problema se tomaron los mismos 3 tipos de conjuntos de datos sintéticos del problema 1 (Figuras 1.1 y 1.5).

Posteriormente se implementó un método de clustering usando el algoritmo de **kernel k-means** en Python (se agrega el código, así como ejemplos de uso en el Anexo A) y se probó en los 3 conjuntos de datos para 3 tipos de kernel (Gaussiano (1.2),Polinomial (1.1), Sigmoide (1.3)).

$$k\left(a_i, a_j\right) = \left(a_i \cdot a_j + c\right)^d \tag{1.1}$$

$$k\left(a_i, a_j\right) = \exp\left(-\left\|a_i - a_j\right\|^2 \Big/ 2\alpha^2\right) \tag{1.2}$$

$$k\left(a_i, a_j\right) = \tanh\left(c\left(a_i \cdot a_j\right) + \theta\right) \tag{1.3}$$

Con parámetros por kernel para cada conjunto de conjunto de datos:

$$\alpha_1 = 100, (c_1 = 1000, d_1 = 2), (c_1 = -100, \theta_1 = {}^{3\pi}\!/_{4})$$
$$\alpha_2 = 0.2121, (c_2 = 0, d_2 = 2), (c_2 = -100, \theta_2 = {}^{\pi}\!/_{4})$$
$$\alpha_3 = 100, (c_3 = 1000, d_3 = 2), (c_3 = 1000, \theta_3 = {}^{4\pi}\!/_{4})$$

Los parámetros sólo se variaron hasta que a propia consideración optimizaran la asignación de clusters.

Observe la Figura 2.1 y note que, para el primer conjunto de datos, los kernels Gaussiano y Polinomial tienen el mismo resultado y asignan correctamente el cluster mientras que para el Sigmoid no. Para el segundo conjunto de datos el que lo hace correctamente es el kernel Polinomial, mientras que el Gaussiano y sigmoide lo hace incorrecto. Por último, para el tercer conjunto de datos los resultados son similares, pero ninguno lo hizo correctamente.

Por lo tanto podemos concluir que, el tipo de método de clustering se ve fuertemente influenciado por el conjunto de datos sobre lo vamos a utilizar, es decir de este depende la selección, segundo el resultado del algoritmo depende mucho también de como se inicialice y de los parámetros de cada modelo (por ejemplo tipo de kernel y sus parámetros), por ejemplo en este caso para el segundo conjunto de datos que se observa claramente que cada cluster no es lineal, lo mejor es una transformación con kernel polinómica de segundo grado.

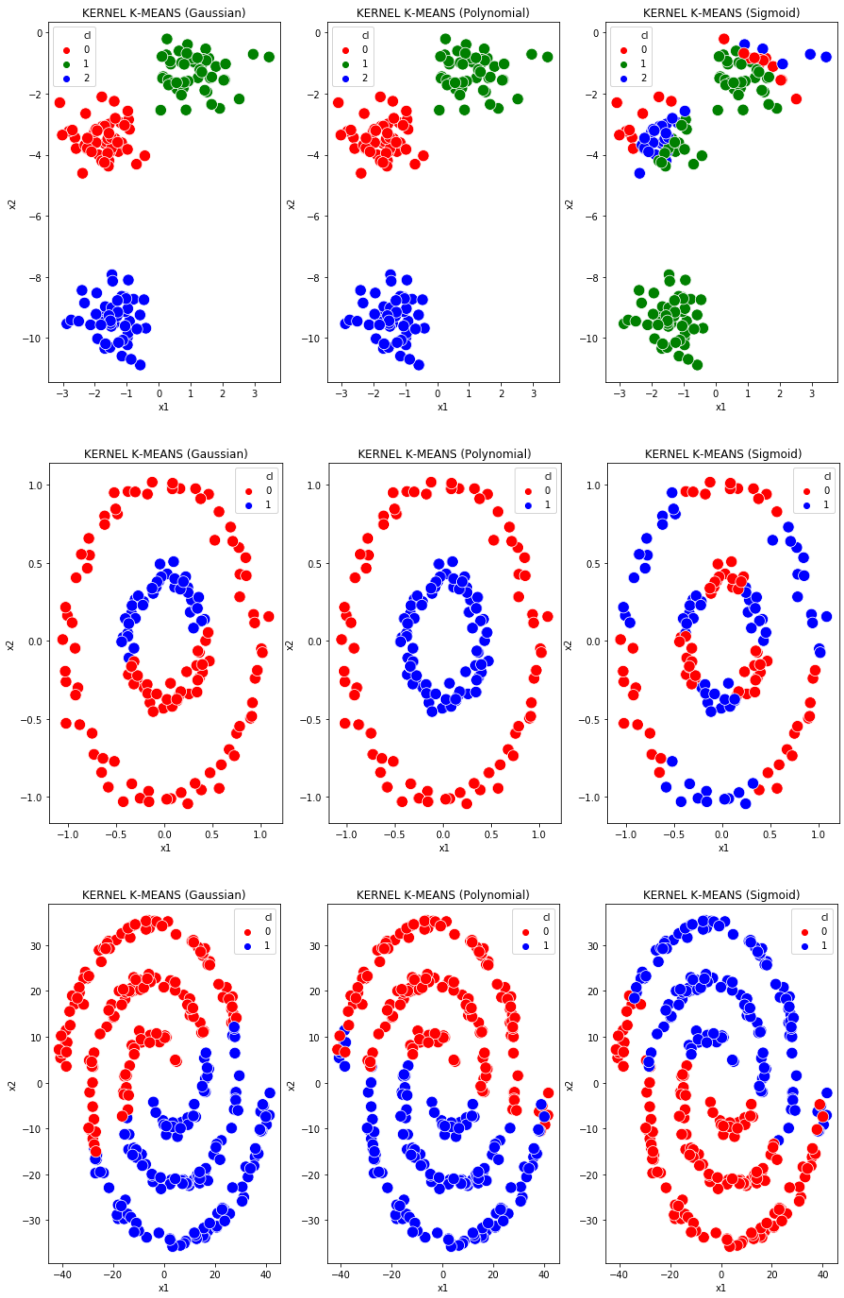

Figura 2.1: Columnas: tipo de kernel a) Gaussiano, b) Polinomial, c) Sigmoid, renglones: conjunto de datos 1,2 y 3.

3. Problema 3

En este problema se utilizaron como datos 1300 imágenes de 11 tipos de frutas (entre ellas 3 tipos de manzanas) tomadas en diferentes orientaciones con diferentes características de forma y maduración. El objetivo en este problema es tratar de identificar las frutas a partir de representaciones de estas imágenes.

El procedimiento fue el siguiente, primero se obtuvieron 2 conjuntos de datos a los cuales se les aplicó de forma separada el mismo análisis, uno (1300x3) al que se obtuvo la mediana de cada canal en el espacio RGB para las 1300 imágenes y otro (1300x9) al que se obtuvo los 3 cuantiles centrales (0.25,0.5,0.75) por cada canal en el espacio HSV.

En la Figura 3.1 puede observar el primer conjunto de datos sobre las medianas de cada canal por tipo de fruta y puede notar que en algunas de estas representaciones las cerezas y los arándanos (rojos) claramente se separan del resto, esta última además es una de las principales frecuencias (o la más grande) en los canales 1 y 3. También podemos observar que los aguacates y la carambula (verdes) siguen un comportamiento lineal en los canales 1 y 2.

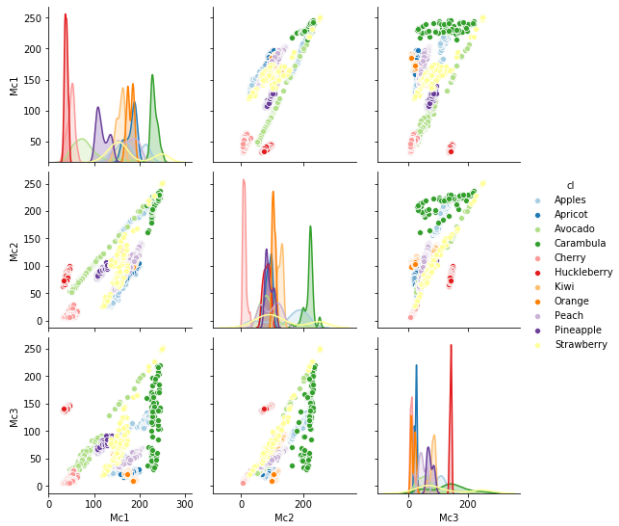

Figura 3.1: Matriz de gráficos de dispersión de las medianas de cada canal RGB (Mc1, Mc2, Mc3) por tipo de fruta.

Posterior a obtener los dos conjuntos de datos se realizó PCA sobre ambos conjuntos y se aplicaron 2 algoritmos de clustering (K-means y Kernel K-means) para tratar de identificar los diferentes grupos de frutas.

Para el primer conjunto en el espacio RGB observa la parte superior de la Figura 3.2 y nota que los arándanos se separan por completo de las demás frutas en la parte superior (color azul), las cerezas y las piñas forman también pequeños grupitos bien definidos, pero no separados, la carambula y el aguacate parecen seguir la misma tendencia lineal. También algo interesante es que parecen estar agrupadas o seguir tendencias por tonalidades de color (Rojo, Verde, Azul) por ejemplo las fresas, naranjas, las manzanas rojas y cerezas están más a la parte izquierda, los azules a la parte superior.

La parte inferior de la Figura 3.2, el clustering, podemos notar que son muy similares los resultados pero con diferente 'etiqueta' de fruta y bajo este criterio ninguno de los métodos lo hizo bien, pero si dejamos de un lado las etiquetas podríamos decir que en algunos grupos concuerdan ambos, es decir los grupos de los colores azules y naranjas son muy similares, dentro de estos tenemos a los arándanos, a los aguacates y piñas, y las manzanas rojas, naranjas, fresas y duraznos, es decir formó pequeños grupos por tonalidades.

Ahora para el segundo conjunto de datos en el espacio HSV, observe la Figura 3.3 y note que nuevamente los arándanos (azules) están muy separados del resto de frutas, pero ahora el resto se ven mas juntos, pero se alcanza a distinguir más cerca los que tienen una tonalidad de color similar, por ejemplo, los que son más amarillos como las carambula, o naranjas, rojos, manzanas, fresas, verdes piñas, aguacate, manzanas.

Para la parte inferior la etiqueta no fue acertada, pero si la dejamos a un lado, nuevamente hay grupos similares, pero no es suficiente para identificar las frutas, es decir no lo hace correctamente.

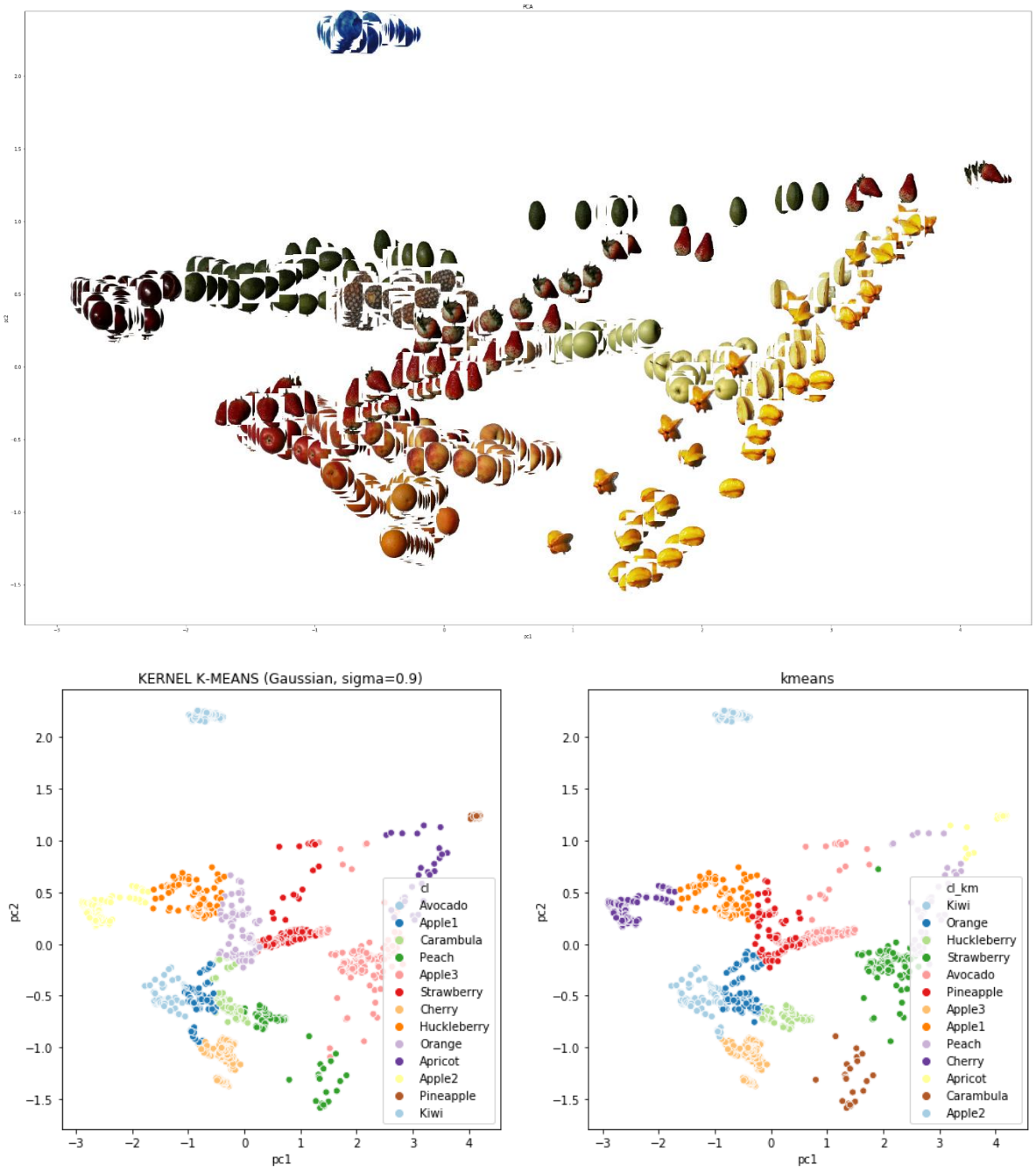

Figura 3.2: Representación de las imágenes frutas (usando la mediana en cada canal RGB) sobre los dos primeros componentes principales (pc1, pc2) utilizando PCA, parte inferior se aplicó dos diferentes algoritmos de clustering (Kernel k-means y k-means) para intentar identificar los diferentes grupos de frutas.

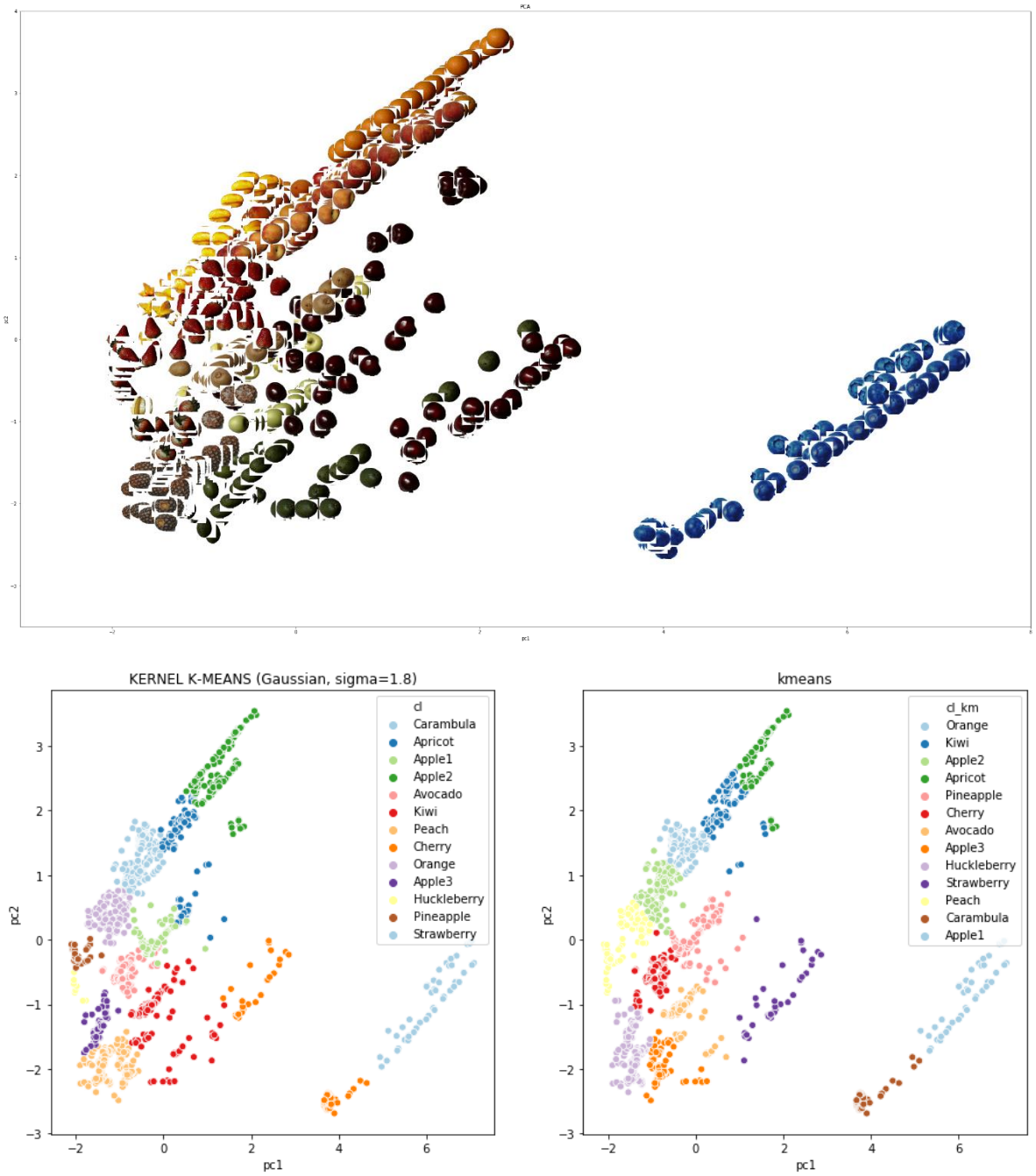

Figura 3.3: Representación de las imágenes frutas (usando los cuantiles 0.25, 0.5 y 0.75 en cada canal HSV) sobre los dos primeros componentes principales (pc1, pc2) utilizando PCA, parte inferior se aplicó dos diferentes algoritmos de clustering (Kernel k-means y k-means) para intentar identificar los diferentes grupos de frutas.

Ahora después de haber aplicado Kernel PCA con kernel Gaussiano a ambos conjuntos de datos, se repite el mismo procedimiento.

Observe la parte superior de la Figura 3.4, para el primer conjunto de datos podemos observar nuevamente como hay una tendencia sobre las tonalidades de color, pero ahora esta es no lineal, podemos ver como en la parte central (cercanos al origen) hay más verdes obscuros, mientras que del lado derecho son tonalidades rojas y del lado izquierdo verdes claras o amarillas, y hacia la parte inferior azul. Esto, por un lado, por otro también podemos observar que la tendencia no lineal también parece seguir la orientación de las frutas.

En la parte inferior de la Figura 3.4 nuevamente aplicando los 2 métodos de clustering, de nuevo hay semejanzas entre ambos métodos, pero no se etiqueta correctamente (ni por color) grupos de frutas, quizá si es mejor respecto a otras características como las tonalidades, pero no sobre la fruta, es decir el clustering no es correcto en asignar que tipo de fruta es.

Ahora observe la Figura 3.5 para el segundo conjunto de datos sobre el espacio HSV y Kernel PCA, nuevamente podemos observar un comportamiento no lineal, pero en esta ocasión parece que mejora la separación de las frutas, ya sea por tonalidad y por orientación. Podemos observar en la parte inferior grupos de aguacates y piñas (y algunas fresas) dentro de este grupito se puede ver que parece que las que están mas maduras se encuentran mas alejadas del conjunto principal (lo mismo para las fresas). También podemos ver las manzanas verdes y amarrillas por la parte superior izquierda y en el extremo a los kiwis. Del lado derecho las frutas con tonalidades rojas, peo se puede diferenciar entre las fresas, naranjas y manzanas, mientras que en la parte inferior están los arándanos y cerezas.

Nuevamente el clustering, no parece ser correcto para identificar los tipos de frutas, pero si grupos con tonalidades similares, pero en general para este problema parece que el haber agregado más información (1300x9) y aplicar este tipo de kernel deja apreciar mejor las características de las imágenes de las frutas y quizá el cluster podría mejorar si no se realiza por fruta, sino por colores o tonalidad de estos, por ejemplo buscar 3 o 6 clusters en lugar de uno por cada tipo de fruta y esto podría ayudar a encontrar las frutas por tipo de color o por maduración, es decir hacer subgrupos con características similares.

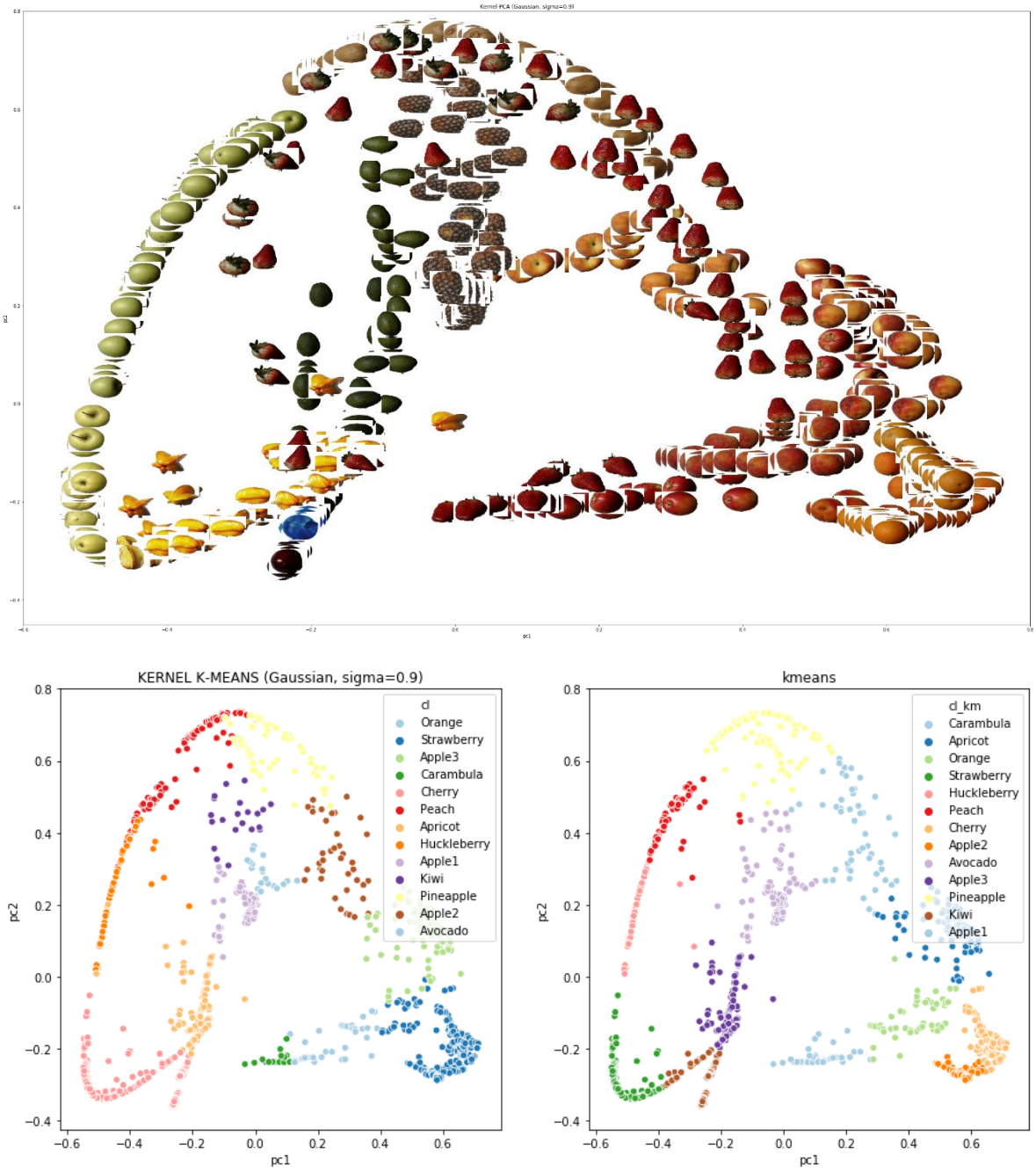

Figura 3.4: Representación de las imágenes frutas (usando la mediana en cada canal RGB) sobre los dos primeros componentes principales (pc1, pc2) utilizando Kernel PCA con kernel Gaussiano $(\sigma = 0.9)$, parte inferior se aplicó dos diferentes algoritmos de clustering (Kernel k-means y k-means) para intentar identificar los diferentes grupos de frutas.

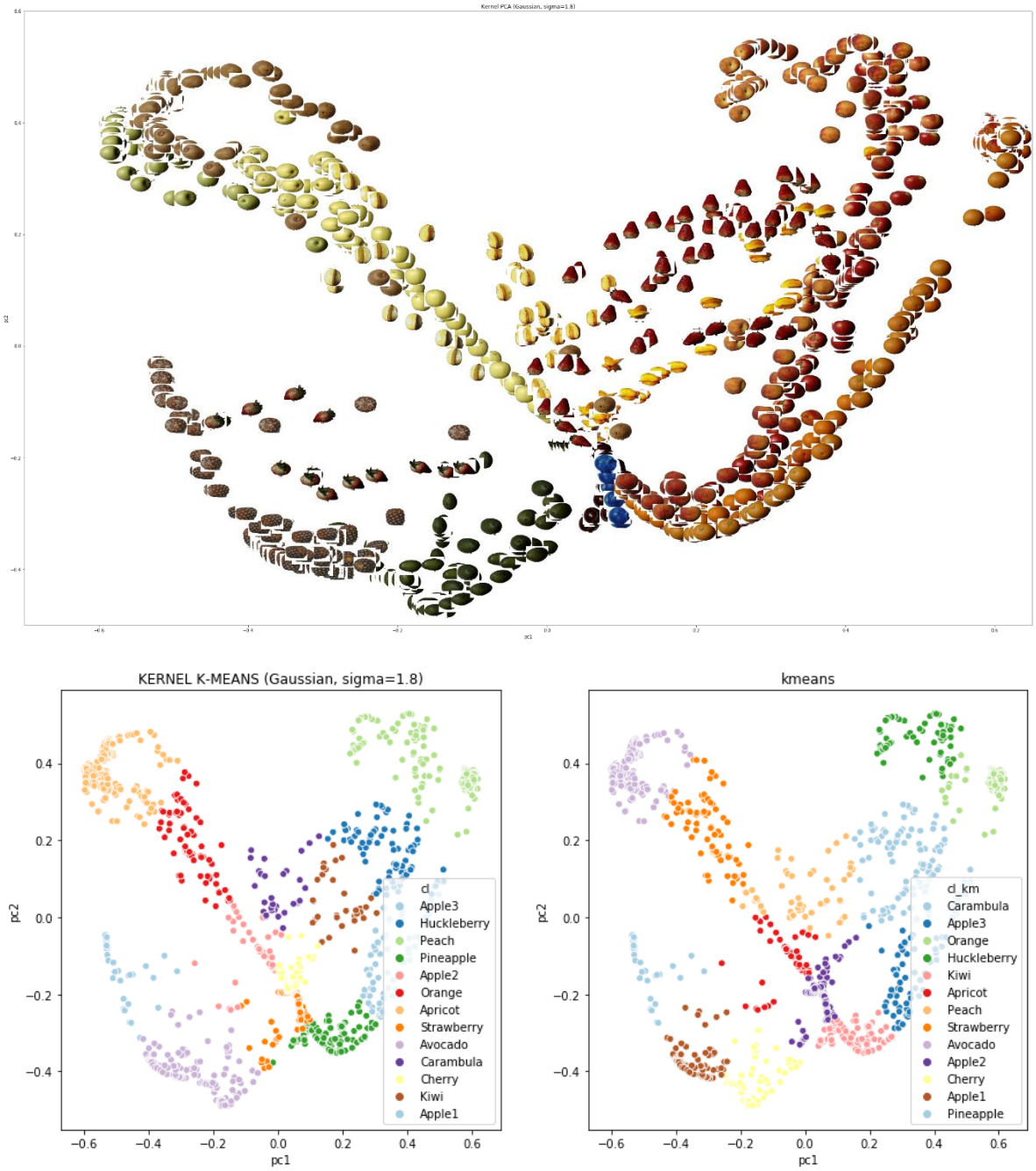

Figura 3.5: Representación de las imágenes frutas (usando los cuantiles 0.25, 0.5 y 0.75 en cada canal HSV) sobre los dos primeros componentes principales (pc1, pc2) utilizando Kernel PCA con kernel Gaussiano $(\sigma = 1.8)$, parte inferior se aplicó dos diferentes algoritmos de clustering (Kernel k-means y k-means) para intentar identificar los diferentes grupos de frutas.