Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Tarea 4

Marcelo Alberto Sanchez Zaragoza

20 de abril de 2021

1. Problema 1

Se nos porporciona un archivo que contiene datos en 2 dimensiones de 5 grupos, los datos se muestran en la figura 1.1, donde a simple vista podemos decir que es una representación muy cercana a los continentes.

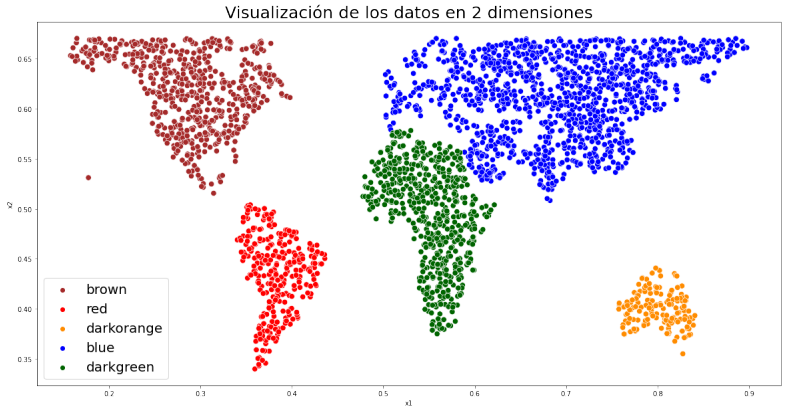

Figura 1.1: Visualización en 2 dimensiones.

Una vez que ya mostramos como están los datos en dos dimensiones se nos propone realizar una variedad no lineal en 3D, es decir, que por medio de cierta transformación llevemos nuestros datos a una dimensión más alta, en otras palabras intentar conservar la estructura pero ahora tomando en cuenta tres variables. En este caso podemos decir que vamos a ocupar las ya conocidas como X, Y y Z. No se va a presentar la transformación empleada en esta parte pero se anexa hasta el final por si existe curiosidad. En la figura 1.2 observamos que los datos proporcionados ya se encuentran en una dimensión más alta(3D).

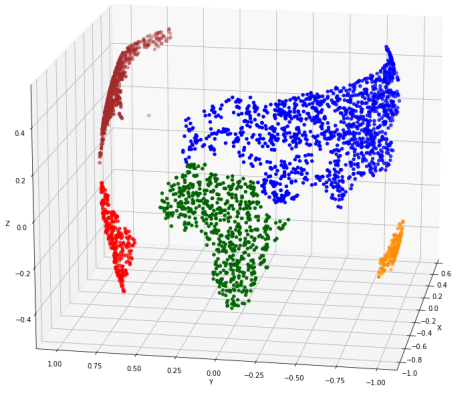

Figura 1.2: Visualización en 3 dimensiones.

Ya que nuestros datos cuentan con una nueva variable se nos solicita que por medio de los métodos de manifold learning basados en PCA, Kernel PCA, Spectral Embedding y t-SNE tratar de reconstruir los patrones encontrados en los datos 2D. Si la idea no es muy clara sobre los métodos y lo que hace cada uno, no se preocupe, la principal intención es solo llevar la nueva transformación de los datos a una representación lo más parecida a la que nos muestra la figura 1.1.

1.1. PCA

Con el primer método que vamos a trabajar es con el método PCA, en este caso lo que vamos a intentar es representar los datos transformados en 2 componentes principales, es decir, cada componente es una nueva representación de los datos transformados, al ocupar dos representaciones y

graficarlas obtenemos una grafica de 2 dimensiones. En este caso vamos a tener tres representaciones distintas pero que nos pueden ayudar a observar ciertos patrones.

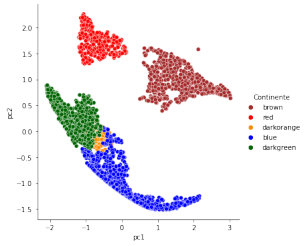

Figura 1.3: Visualización con PCA(componente 1 y 2).

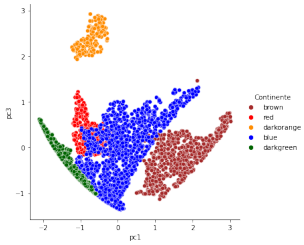

Figura 1.4: Visualización con PCA(componente 1 y 3).

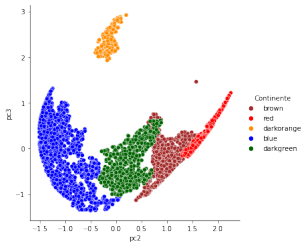

Figura 1.5: Visualización con PCA(componente 2 y 3).

Al parecer nuestros datos transformados en estas tres representaciones no nos ayudan mucho a encontrar patrones que se aproximen a la figura 1.1. Pero lo que si podemos observar que hacen algunos de ellos, es separar lo suficiente algunos grupos, por ejemplo, en la figura 1.3 tiene cierto parecido pero no muestra muy bien los datos en color amarillo ya que los pone en medio y detras de los datos azules y verdes, en las figuras 1.4 y 1.5 se logra rescatar muy bien los datos amarillos pero como no es nuestro objetivo separarlos sino encontrar patrones, lo dejaremos así por el momento.

1.2. Kernel PCA

Con el siguiente método se abren nuestro abanico de posibilidades ya que este método nos solicita ciertos parámetros como el tipo de kernel que vamos a ocupar, el valor de sigma y el número de componentes. Ya que podemos encontrar un mar de visualizaciones vamos a colocar aquellos que nos proporcionaron una buena representación de los datos y los que no nos dieron una buena representación solo los mencionaremos a manera de resumen.

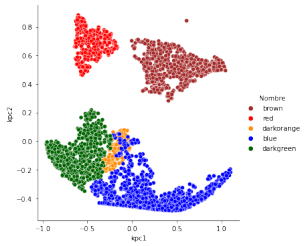

Figura 1.6: Visualización con Kernel-PCA(componente 1 y 2, *sigmoid*).

En la figura 1.6 se ocupo un kernel de tipo *sigmoid*, un sigma de 2.5 y se solicitaron 3 componentes principales. Se observa que si logra representar bien los colores café y rojo pero los demás los combina un poco. No es la mejor pero el parecido es bueno al menos con el color café y rojo. Se intento mejorar la representación proporcionando un valor de sigma más grande pero no se ve mucha diferencia a lo que nos proporciona la figura 1.6 y reduciendo el valor no nos da un buen patrón.

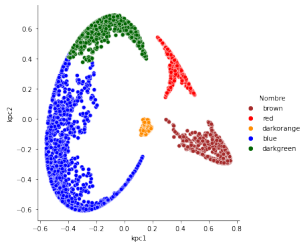

Figura 1.7: Visualización con Kernel-PCA(componente 1 y 2, rbf).

Solo a manera de ejemplo se muestra la figura 1.7 donde observamos quiza una buena separación pero no un buen patrón. Los parámetros que se ocuparon fue un kernel rbf(gaussiano), sigma de 2.5 y 3 componentes principales. Al igual variar el valor de sigma no vemos patrones buenos.

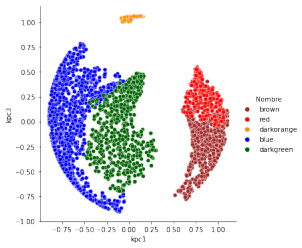

Figura 1.8: Visualización con Kernel-PCA(componente 1 y 3, *cosine*).

Con el kernel *cosine*(coseno) no nos ayuda mucho a encontrar patrones. En la figura 1.8 tomamos un kernel coseno, valor de sigma de 2.5 y 3 componentes principales. En la figura 1.8 tomamos el primer y tercer componente principal, a manera de ejemplo nos ayuda a ver que no siempre los primeros dos nos dan la mejor representación que estemos buscando.

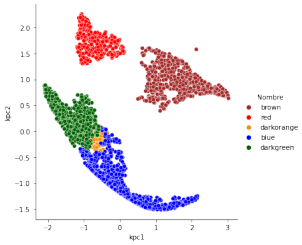

Figura 1.9: Visualización con Kernel-PCA(componente 1 y 2, *linear*).

En la figura 1.9 tomamos un kernel *linear*(lineal) con un sigma de 0.05 y tres componentes principales. En la figura 1.9 tomamos los dos primeros componentes ya que nos enseña un buen patrón pero con algunas fallas.

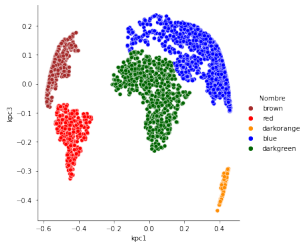

Figura 1.10: Visualización con Kernel-PCA(componente 1 y 3, *poly*).

Finalmente en la figura 1.10 tomamos un kernel *polynomial*, con un sigma de 5.5 y con 3 componentes principales. El resultado fue bueno para este tipo de kernel pero hay que mencionar que se cambio el signo de los valores para lograr tener como resultado el gráfico ya que si lo dejaramos así nuestra representación estaría de cabeza. El parecido es bueno principalmente por los valores verdes y casi ubica bien los datos.

1.3. Spectral Embeddings

Al igual que en el método anterior Spectral Embeddings cuenta con parámetros que nos solicita para poder trabajar por lo que vamos a mostrar algunos casos y explicar algunas gráficar que consideramos adecuadas para mostrar al menos una idea vaga de lo que realiza el método pero sin dejar de lado nuestra principal tarea que es encontrar una representación lo bastante parecida a la figura 1.1.
El método nos solicita el número de vecinos más cercanos, un valor de sigma y un tipo de kernel para encontrar la matriz de disimilaridad(algunos casos) y en ocasiones podemos proporcionar el tipo de laplaciano que queremos ocupar. Algunos de estos conceptos quiza confundan un poco pero dejemos de lado por un momento la explicación y concentremonos en las representaciones ya que solo son parámetros que se pueden ir cambiando e ir experimentando que conjunto de ellos nos da un mejor patron.

Como resultado de este método contamos con ciertas representaciones dadas por valores propios, que nos ayudan a separar los datos de cierta forma que encontremos patrones.
Solo para ilustrar la idea de esta sepación de valores en la figura 1.11 y 1.12 mostramos como funcionan estas representaciones de los datos en los valores propios, como la intención es encontrar patrones 'bonitos' primero debemos obtener valores propios que nos ayuden a separar estos datos.

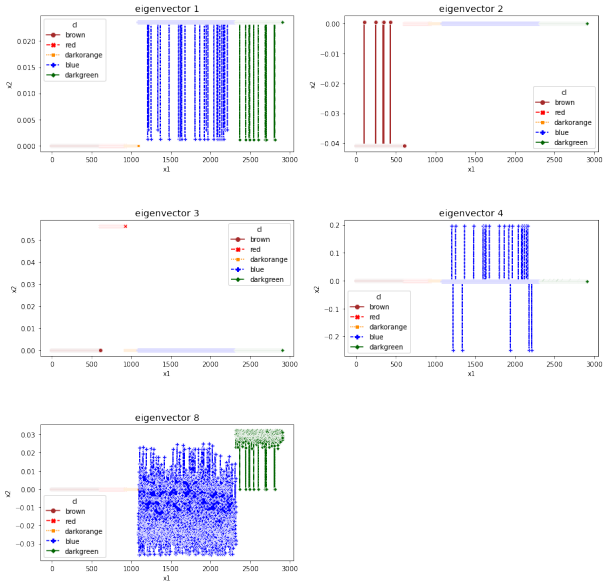

Figura 1.11: Representación en valores propios, no buena separación.

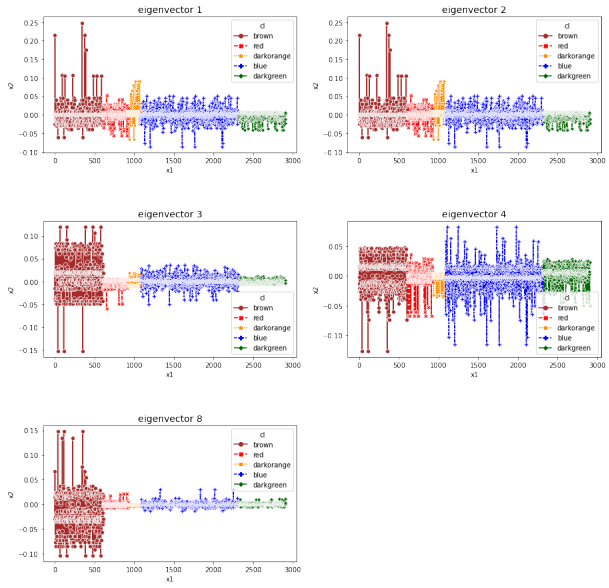

Figura 1.12: Representación en valores propios, buena separación.

En la figura 1.11 se ocupo un laplaciano rw, sigma igual a 0.8, un tipo de kernel rbf(gaussiano) y 5 vecinos cercanos y en la figura 1.12 se ocupo un laplaciano sym, sigma igual a 2.5, un tipo de kernel rbf(gaussiano) y 5 vecinos cercanos. Si prestamos atención en este primer gráfico no observamos bien los 5 grupos por lo que no nos ayuda mucho a separar estos datos y una representación tomando dos proyecciones de los datos en estos valores propios no nos seria de gran ayuda. En cambio en la figura 1.12 nos separa mejor los datos en cada una de las proyecciones y observamos por los menos datos de cada uno de los 5 grupos. Una representación en 2D tomando dos proyecciones nos ayudaria mucho y a lo mejor encontraremos los patrones que nos piden.
Una vez que aclaramos como vamos a seleccionar nuestras representaciones ahora solo se mostraran dichas representaciones y se mostraran los

parámetros que se ocuparon.

En la figura 1.13 observamos que nuestros datos no tienen una forma muy exacta a lo que nos muestra la figura 1.1 pero al menos recupera la forma de cada 'continente' solo que hay una rotación. El tipo de matriz de afinidad que ocupamos fue rbf(gaussiano), sigma 0.5 y 5 vecinos cercanos.

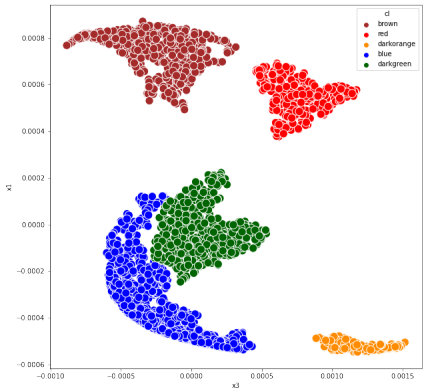

Figura 1.13: Visualización con Spectral Embeddings(componente 3 y 1).

Es una representación adecuada ya que las proyecciones en los valores propios nos logran separar los datos por lo que teniendo algo aproximadamente bueno podemos empezar a buscar combinaciones hasta encontrar un buen patrón.

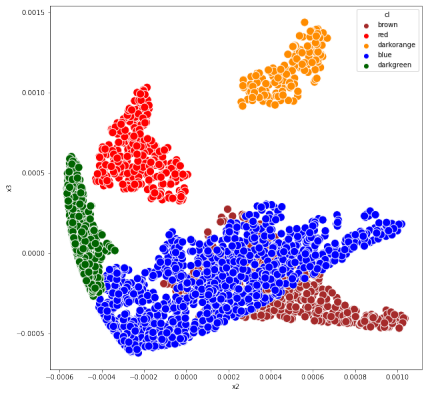

Figura 1.14: Visualización con Spectral Embeddings(componente 2 y 3).

En la figura 1.14 tomamos otras dos proyecciones como se observa en los nombres de los ejes del gráfico, con los mismos parámetros que ocupo la figura 1.13. Los datos no se encuentran muy pegados pero es muy dificil encontrar un patrón que nos ayude a acercanos a la figura 1.1.
Cabe recalcar que si nuestra intención fuera separar solo los grupos de colores la figura 1.13 logra una buena separación de los datos.

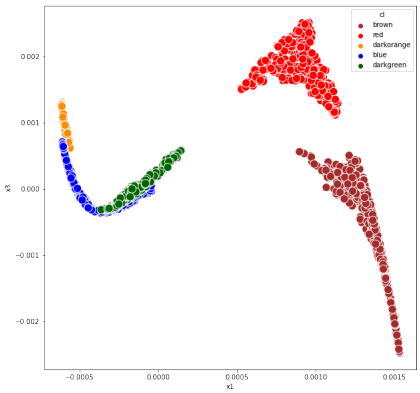

Figura 1.15: Visualización con Spectral Embeddings(componente 1 y 3).

En la figura 1.15 vemos que no nos da una buena representación y nos muestra que no siempre nos ayuda este método,cabe recalcar que intervienen mucho los parámetros que se proporcionen. En la figura 1.15 ocupamos los siguientes: El tipo de matriz de afinidad que ocupamos fue rbf(gaussiano), sigma 2.5 y 8 vecinos cercanos.

1.4. t-SNE

Finalmente tenemos el método de t-SNE donde lo que nos va a poder a ayudar a encontrar una buena representación de los datos transformados es la perplejidad que podemos proporcionar ya que a medida que vamos aumentando este parámetro vamos obteniendo un estructura similar a la que teniamos en un principio y que se represento en la figura 1.1.

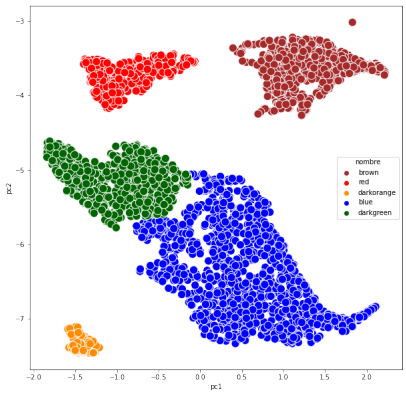

Figura 1.16: Visualización con t-SNE.

En la figura 1.16 propocionamos una perplejidad igual 1800(parametro) para poder obtener una buena representación, la perplejidad es muy alta pero con forme fuimos probando llegamos a ocupar una cantidad cada vez más alta y los resultados nos proporcionaban buenos patrones.

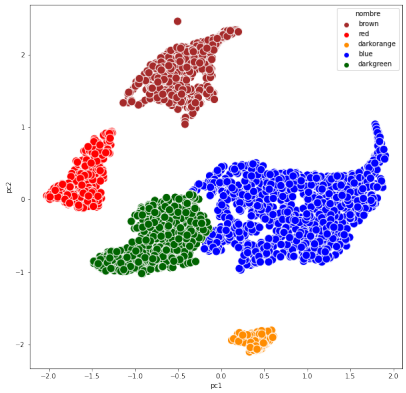

Figura 1.17: Visualización con t-SNE.

En la figura 1.17 proporcionamos una perplejidad igual a 2000 y nos da un buen resultado.

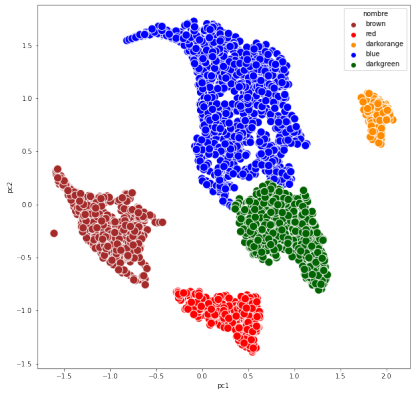

Figura 1.18: Visualización con t-SNE.

Finalmente en la figura 1.18 proporcionamos una perplejidad igual a 2300 y nos dio un resultado lo bastante bueno, es fácil notar que hay un parecido muy bueno a la figura 1.1. Al utilizar este método notamos que da buenos resultados pero necesitamos valores de perplejida muy altos para obtener los patrones que deseamos, despues de ocupar la perplejidad igual a 2300 nuestros gráficos empezaron a moverse mucho por lo que no se siguio aumentando. Lo anterior fue posible apoyandonos de un lenguaje de programación donde el unico parámetro que fuimos modificando fue la perplejidad.

1.5. Comentarios

A manera de conclusión despues de haber experimentado con los cuatro métodos solicitados observamos que con PCA no habia mucho que hacer

más que analizar los datos estandarizando y no estandarizados para finalmente gráficar y observar los patrones, en cambio en Kernel PCA si hubo mucho que hacer ya que desde este método nos pedian parámetros y automaticamente aumentaba mucho nuestros posibles resultados. En este método notamos que los kernel que mejor nos ayudo fue el kernel *sigmoid* y *poly*, en estos dos se observo mejor los patrones solicitados. Para Spectral Embeddings si se lograron buenos graficos solo que ligeramente rotados, pero se aproximan a la estructura de la figura 1.1.
Finalmente con t-SNE se vio una mejoria muy marcada comparada con las demás ya que nuestros datos nos los fue juntando y conforme ibamos aumentando el valor de perplejidad obtuvimos una buena representación y un patrón muy bueno comparado con los otros tres métodos. En lo personal me inclinaria más por el método de t-SNE por lo bueno que fue con estos datos, despues por Kernel PCA y finalmente con Spectral Embeddings, estos ultimos dos nos dieron muchos resultados y varios de ellos dieron patrones buenos pero hay que trabajar mucho con los parámetros.

2. Problema 2

Nos menciona que ahora tomemos los datos de *LFW* que previamente habiamos trabajado pero con la excepción de que tomemos todas las imagenes, un total de 1,288 imagenes. Teniendo el conjunto de dichas imagenes debemos ocupar los siguientes métodos de manifold learning basados en Kernel PCA, Spectral Embedding y t-SNE, en las imagenes y obtener una representación en 2D y compararlos con los que obtuvimos en PCA en la anterior actividad(Tarea 2).
En la figura 2.1 mostramos nuevamente el tipo de imagenes que vamos a trabajar y las dimensiones de dichas imagenes son de (125x94) pixeles.

Figura 2.1: Ejemplo de los rostros del dataset LFW.

Los nombres de los personas que estamos ocupando son las siguientes:

a) Ariel Sharon

b) Colin Powell

c) Donald Rumsfeld

d) George W Bush

e) Gerhard Schroeder

f) Hugo Chavez

g) Tony Blair

2.1. Kernel PCA

Al igual que en el ejercicio anterior nuestros resultados para este método son multiples por lo que vamos a mostrar aquellos que nos dieron una buena representación y proporcionar los parámetros que se emplearon en cada uno de ellos.

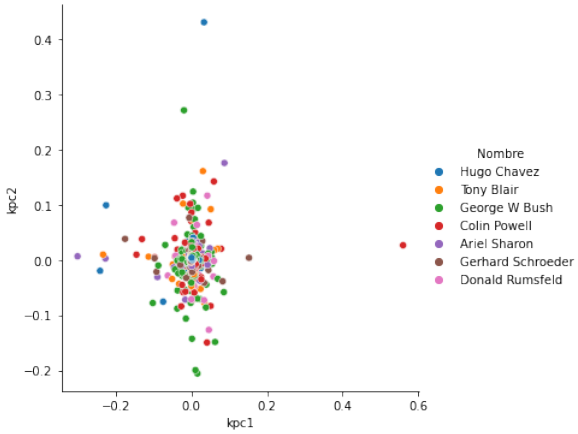

Figura 2.2: Visualización con Kernel PCA(kernel rbf, componentes 1 y 2).

En la figura 2.2 se muestra una primer gráfica utilizando Kernel PCA, observamos que muchos datos se encuentran muy pegados y otros algo lejanos pero por el momento no logramos una representación buena que nos separe bien los 7 grupos de imagenes. Los parámetros que ocupamos fueron los siguientes: un kernel rbf(gaussiano), sigma igual a 1.5 y solicitamos 3 componentes. En este método vamos a pedir tres componentes para realizar algunas combinaciones y escoger el mejor.

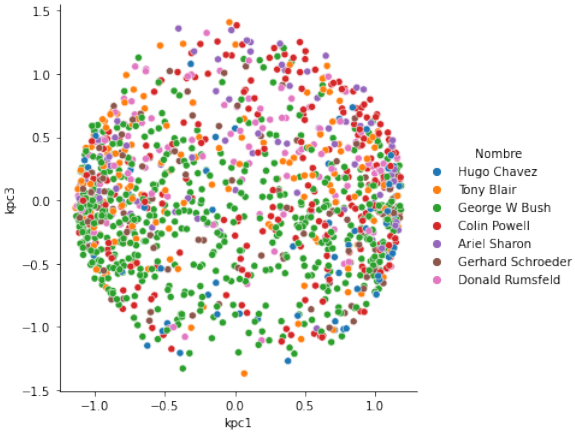

Figura 2.3: Visualización con Kernel PCA(*sigmoid*, componentes 1 y 3).

En la figura 2.3 tomamos un kernel *sigmoid* y nos dio una distribución de los datos muy diferente a la anterior, note que tomamos componentes diferentes ya que nos mostro una estructura interesante. Si obsevamos bien los datos color verde se ven un poco centrados en la parte inferior, además todos los datos forma como un circulo. Los parámetros que ocupamos fueron los siguientes: un kernel *sigmoid*, sigma igual a 2.5 y solicitamos 3 componentes.

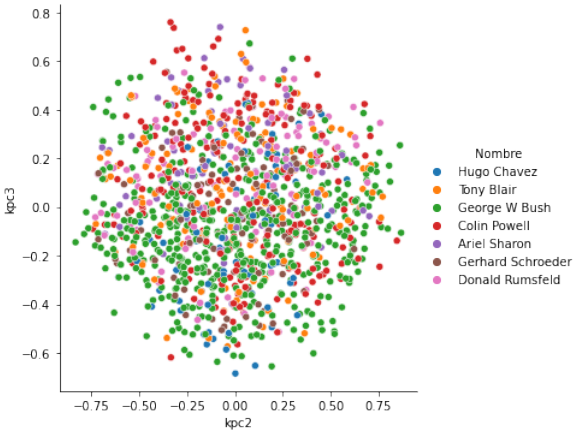

Figura 2.4: Visualización con Kernel PCA(*cosine*, componentes 2 y 3).

En la figura 2.4 tomamos un kernel *cosine* y hay cierto parecido con los datos color verde que vimos en el figura 2.3 ya que igual los coloca en la parte de abajo. También los datos los acomoda en forma de un circulo pero un poco más pegados y con menos espacios entre los datos. Los parámetros que ocupamos fueron los siguientes: un kernel *cosine*, sigma igual a 2.5 y solicitamos 3 componentes.

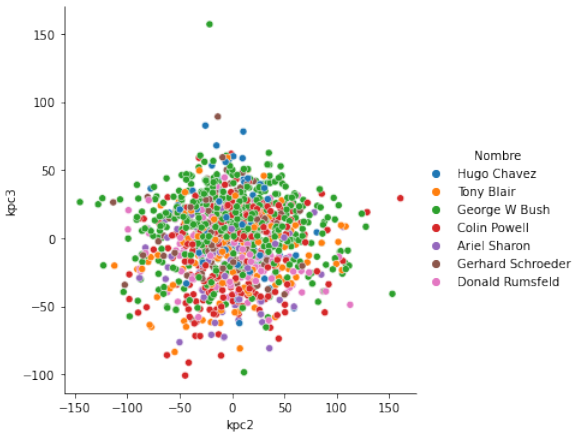

Figura 2.5: Visualización con Kernel PCA(*linear*, componentes 2 y 3).

Los datos en la figura 2.5 se encuentran más centrados y muy pegados pero ahora los datos de color verde los coloca en la parte de arriba pero muchos de ellos se encuentran alejados. Los parámetros que ocupamos fueron los siguientes: un kernel *linear*, sigma igual a 2.5 y solicitamos 3 componentes.

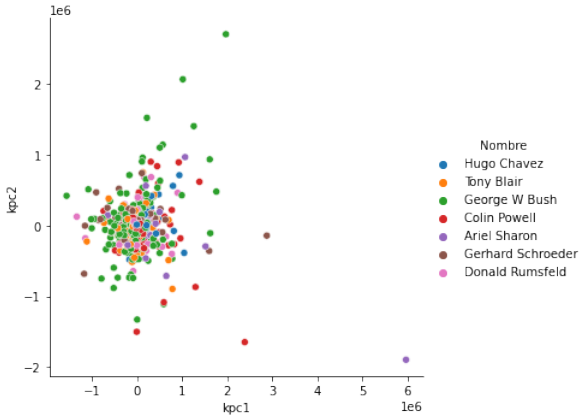

Figura 2.6: Visualización con Kernel PCA(*poly*, componentes 1 y 2).

En la figura 2.6 encontramos los datos muy centrados comparados con los anteriores gráficos, muchos de ellos están algo alejados y observemos que el conjunto de datos esta ligeramente recorrido a la izquierda. Los parámetros que ocupamos fueron los siguientes: un kernel *poly*, sigma igual a 1.5 y solicitamos 3 componentes.

2.2. Spectral Embedings

Al igual que con Kernel PCA nos piden parámetros con Spectral Embedings sucede lo mismo ya que debemos proporcionar el número de vecinos más cercanos, el valor de sigma, el tipo de laplaciano y el número de grupos.

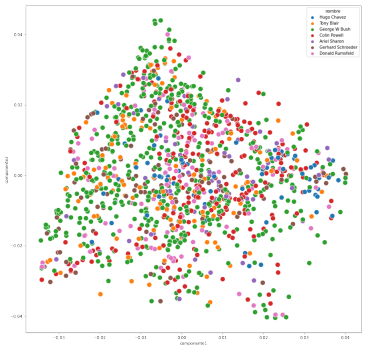

Figura 2.7: Visualización con Spectral Embedings(componentes 1 y 2).

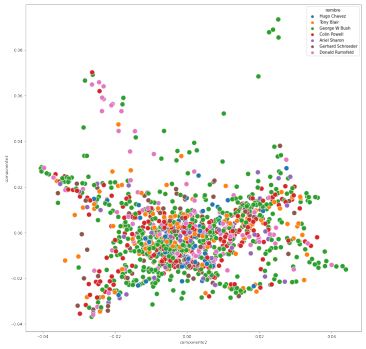

Figura 2.8: Visualización con Spectral Embedings(componentes 2 y 3).

En las figuras 2.7 y 2.8 notamos que nuestros datos se encuentran distribuidos de diferente forma. En la primer figura 2.7 no se observa un buen patrón mientras que en la figura 2.8 si hay cierto comportamiento de los datos, observamos que los datos están ubicados en la parte inferior y tienen una forma picuda, parecido a una estrella. Los parámetros para ambas figuras(2.7 y 2.8) que ocupamos fueron los siguientes: se ocupo un

laplaciano rw, sigma igual a 2.5 y 4 vecinos cercanos.

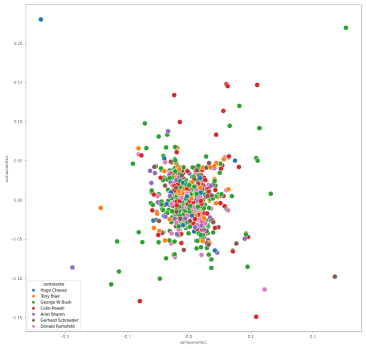

Figura 2.9: Visualización con Spectral Embedings(componentes 1 y 2).

En la figura 2.9 obsevamos que nuestros datos se encuentran muy cercanos unos con otros por lo que nos obtuvimos un buen patrón. Los parámetros que ocupamos fueron los siguientes: se ocupo un laplaciano rw, sigma igual a 0.5 y 6 vecinos cercanos.

2.3. T-SNE

Al igual que en el anterior ejercicio vamos a manejar distintos valores perplejidad donde observamos que a medida que aumentamos su valor va juntando los datos poco a poco.

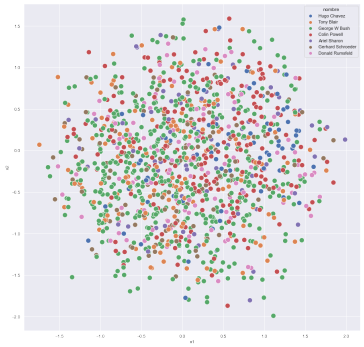

Figura 2.10: Visualización con t-SNE(Perplejidad de 1,000).

En la figura 2.10 observamos que los datos no nos proporcionan un buen patrón donde notemos que los datos se agrupen y nos ayude a separar los 7 grupos de imagenes.

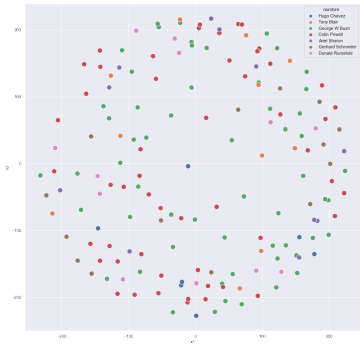

Figura 2.11: Visualización con t-SNE(Perplejidad de 2,000).

En la figura 2.11 obsevamos que nuestros datos están más alejados y hay muchos espacios entre ellos. Se ve muy bien pero todavía no logra juntar nuestros 7 grupos de imagenes como quisieramos.

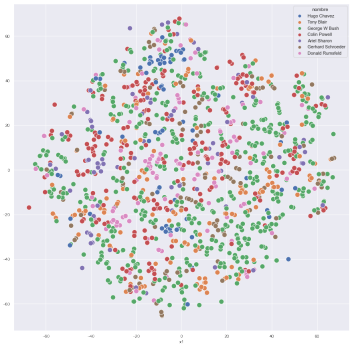

Figura 2.12: Visualización con t-SNE(Perplejidad de 10).

En la figura 2.12 observamos que nos junta los datos pero siguen estando distribuidos en todo el gráfico, a simple vista se observa que muchos si están bien relacionados y otros no. Note que el valor de perplejidad fue de 10, comparado con los que utilizamos en las primeras dos figura es bastante bajo.

2.4. Comentarios

Para poder dar una comparación entre lo expuesto en esta sección y lo que mostramos en la anterior tarea se coloca en la figura 2.13 el resultado de la tarea 2.

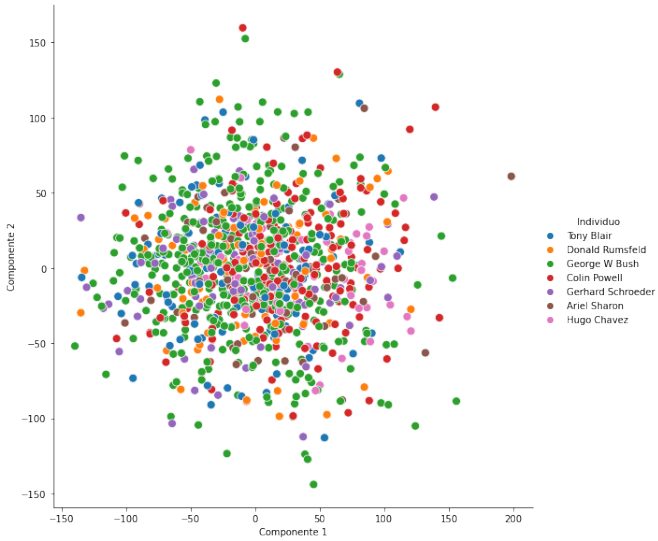

Figura 2.13: Visualización con PCA(Tarea2).

La intención de comparar estos métodos con lo que obtuvimos en la figura 2.13 es encontrar patrones, es decir,al aplicarle estos métodos ver comportamientos de los datos de forma que si es posible nos ayuden a separar los 7 grupos de imagenes y no se vean tan revueltos los puntos de colores, lo otro es encontrar buenos patrones que aunque no nos separen del todo los 7 grupos logren por lo menos agruparlos poco a poco.
La figura 2.13 nos muestra los datos muy agrupados pero no hay muchos patrones interesantes, mientras que con Kernel PCA si logramos patrones interesantes, en muchos de ellos logramos separar bastante los datos y en otros se observa que los datos se mandan a cierta parte del gráfico, el tipo de kernel que nos funciono bastante fue *sigmoid* y *cosine*. Para Spectral Embedings encontramos patrones interesantes en los datos, como la casi estrella que parece en la figura 2.8 pero no nos agrupa los datos en su respectivos grupos. Con t-SNE se logra un buen gráfico como en la figura 2.11, donde separa lo suficiente los datos y no se ven muy combinados.

Despues de observar los resultados al menos para los parámetros que proporcionamos podemos decir que los métodos que mejor nos dan patrones son Spectral Embedings y t-SNE, aunque entre estos dos t-SNE ayuda más ya que nos ubican mejor cada dato con su grupo correspondiente y no nos revuelven muchos los datos en el gráfico, también recodar que son métodos que requieren parámetros y esto podria ser de mucha ayuda ya que tenemos un mar de posibles resultados, depende mucho del objetivo que tengamos, de igual forma puede que no lleguemos a lo que necesitamos pero para este ejercicio nos ayudaron más que PCA por lo manipulable que fueron.