Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Tarea 4

Enrique Santibáñez Cortés

21 de abril de 2021

1. Proyección de la superficie de los continentes de 2D a 3D

El conjunto de datos `data_world.csv` representan puntos aleatorios dentro de los cincos continentes respectivamente (ver Figura 1.1).

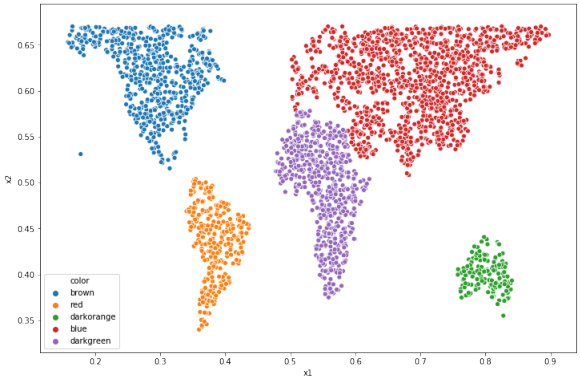

Figura 1.1: Data world scatterplot

Ahora considerando el embedding de los datos en una variedad no lineal en 3D (en una esfera, ver Figura 1.2), es decir, se considera la siguiente parametrización

$$p = x * (2\pi - 0{,}55), \quad t = y * \pi$$

y por lo tanto los datos proyectamos en la variedad no lineal en 3d se representan con las nuevas variables transformadas

$$\begin{aligned} x_{esfera} &= \sin(t) * \cos(p) \\ y_{esfera} &= \sin(t) * \sin(p) \\ x_{esfera} &= \cos(t) \end{aligned}$$

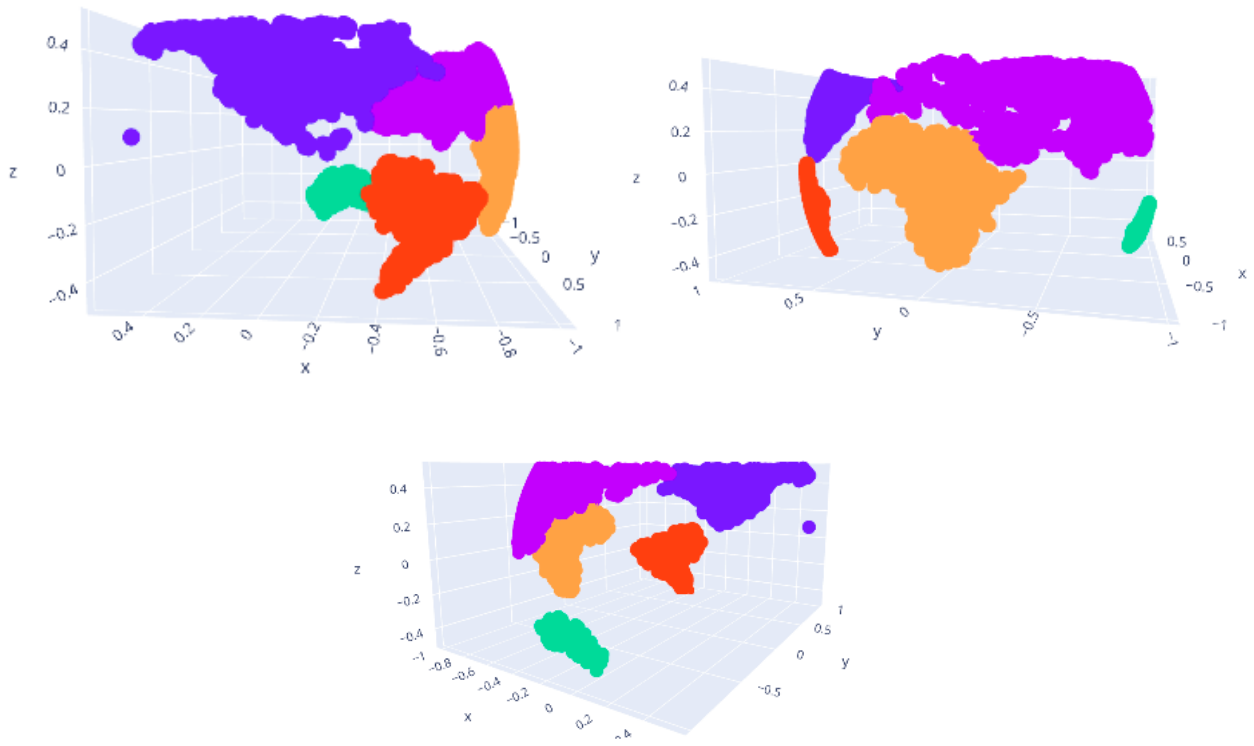

Figura 1.2: Data world proyectados en un esfera, scatterplots 3D

Entonces el objetivo **de este trabajo es tratar de reconstruir los patrones encontrados en los datos 2D, a partir de los datos 3D**. Para ello consideraremos los métodos de manifold learning basados en PCA, Kernel PCA, Spectral Embenddings y t−SNE.

1.1. Ajuste de hiperparámetros

Considerando los datos 2D, se pueden observar dos tipos de patrones interesantes a buscar: división de los continentes y relación entre los continentes. Es decir, **en el primero nos interesaría que podemos poder clasificar los puntos en los 5 continentes y el segundo sería como se *relacionan* los continentes entre sí (que tan alejados están los continentes entre sí)**. Es decir, se puede interpretar como la información dentro de los continentes y fuera de los continentes.

Procedemos a buscar en cada uno de los métodos de reducción de la dimensionalidad, la mejor combinación de parámetros en donde se pueda observar los patrones anteriores. Para ello el criterio se selección de los parámetros será a ***ojo*** ya que no tenemos una medida cuantitativa para saber que combinación de parámetros es ***mejor***. Se presentan los efectos de cada uno de los parámetros en los distintos métodos, **para todas las combinaciones se ocupo la semilla igual a "19970808"**. No se presentan todas las combinaciones probadas, solo las más relevantes.

PCA

Para este método de reducción de dimensión solo consideramos la diferencias del método cuando se aplica al conjuntos de datos sin estandarizar y estandarizado.

Observando la Figura 1.3, notamos que este método no funciona adecuadamente para encontrar los patrones descritos anteriormente. Esto se justifica claramente debido a que el embedding de los datos considera una variedad no lineal y, si recordamos, PCA no sirve para *patrones lineales*.

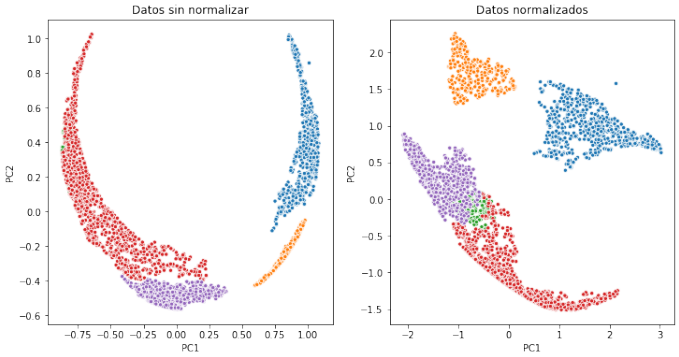

Figura 1.3: Efecto de la estandarización de los datos.

Kernel PCA

En este método existen dos hiperparámetros que afectan la reducción de dimensionalidad: ***kernel a ocupar y coeficiente del kernel***. Los kerneles que se probaron son *rbf, sigmoid, poly, linear, cosine* (ver [2] para mas detalles de las expresiones especificas), y para el coeficiente del kernel probamos los siguientes valores: 0.005, 0.1, 0.2, 0.5, 0.9, 1.4. Este último coeficiente, solo se usa para algunos kerneles (rbf, poly y sigmoid).

En total se probaron 17 combinaciones de parámetros distintos, las cuales se probaron en los datos estandarizados y no estandarizados. Primero observemos el efecto que tiene el coeficiente del kernel en un kernel rbf (ver Figura 1.4), es fácil ver que entre más grande sea este coeficiente los patrones buscados son difíciles de encontrar. De hecho, para valores mayores a 0.5 se puede observar el problema de crowding.

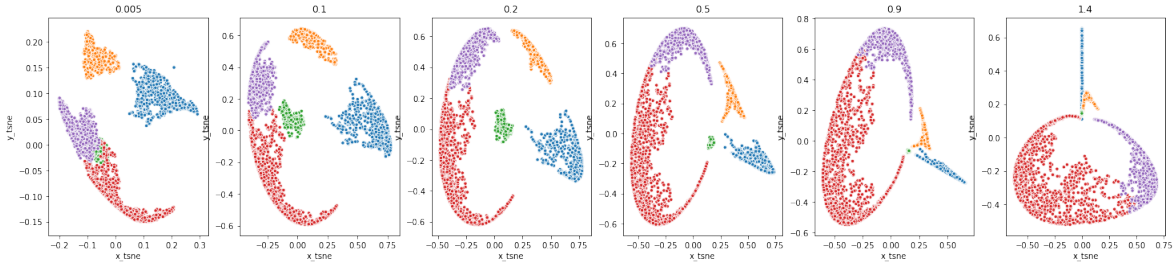

Figura 1.4: Efecto de coeficiente del kernel usando kernel rbf.

Ahora, **veamos que el uso del kernel que ocupemos presenta un mayor impacto en las proyecciones que el coeficiente del kernel** (ver Figura 1.5). En las proyecciones de los kerneles **rbf y cosine** vemos que se puede identificar los puntos en sus respectivos continentes, pero en las proyecciones del kernel rbf no se observa tan claro la relación de los datos entre los continentes. **Por lo que podemos concluir que la mejor combinación es utilizando el kernel cosine**.

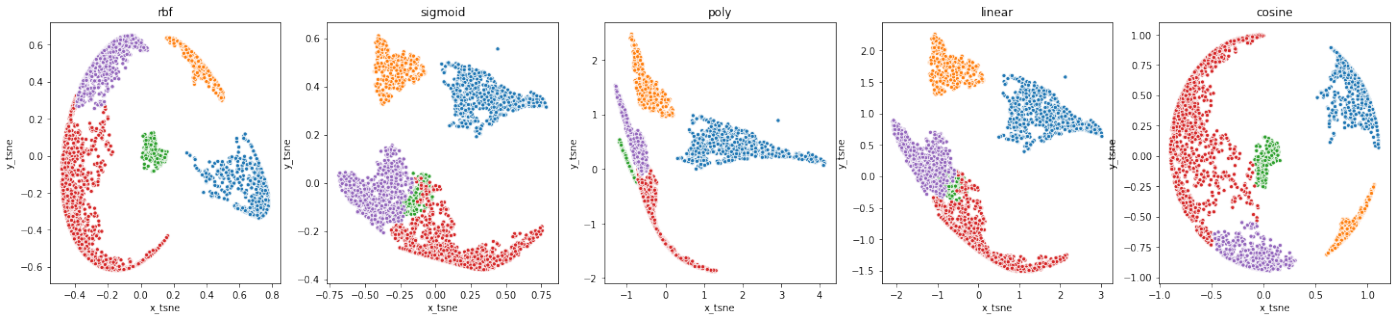

Figura 1.5: Efecto de los diferentes kerneles, (coeficiente kernel = 0,2)

Por último, veamos el efecto que tiene estandarizar los datos antes de ocupar Kernel PCA usando el kernel cosine (ver Figura 1.6). Observemos que cuando no estandarizamos los datos no se observar los patrones originales de los datos en 2D. **Por lo tanto, la mejor combinación para este método es usando el kernel cosine en los datos estandarizados** (la proyección de la derecha en la Figura 1.6).

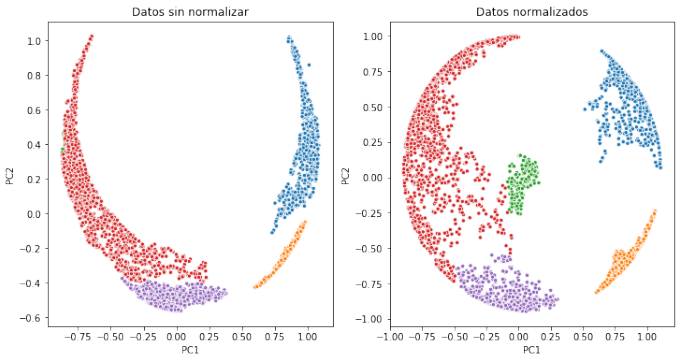

Figura 1.6: Efecto de la estandarización de los datos.

Spectral Embenddings (SE)

Para este método, los parámetros que afectan el ajuste son: **construcción de la matriz de afinidad, coeficiente del kernel y número de vecinos más cercanos**. Para la construcción de la matriz de afinidad existen dos maneras de hacerlo: calculando un gráfico de vecinos más cercanos y calculando un núcleo de función de base radial (rbf). Cuando se utiliza vecinos más cercanos se necesita determinar el número de vecinos para el gráfico y cuando se utiliza un núcleo de función de base radial se necesita determinar el coeficiente de kernel.

Es decir, probamos SE calculando la matriz de afinidad considerando $(2, 3, 4, 5, 10)$ vecinos más cercanos y calculando la matriz de afinidad usando rbf considerando gamma = (,005, ,1, ,4, ,42, ,6). Se ocuparon los datos normalizados y sin normalizar, pero aquí solo se presentan los datos normalizados.

El efecto de coeficiente del kernel usando rbf en SE es muy notorio, entre más grande sea (menor a 1) se observan más claros los diferentes cluster (ver Figura 1.7). Ahora, considerando SE cuando se calcula la matriz usando diferentes números de vecinos más cercanos

(ver Figura 1.8) se observa claramente el efecto, pero este enfoque es erróneo ya que no podemos apreciar ningún patrón. **Por lo que la mejor combinación para este metodo es considerando rbf y usando el valor de 0.4 como valor del coeficiente de l kernel.**

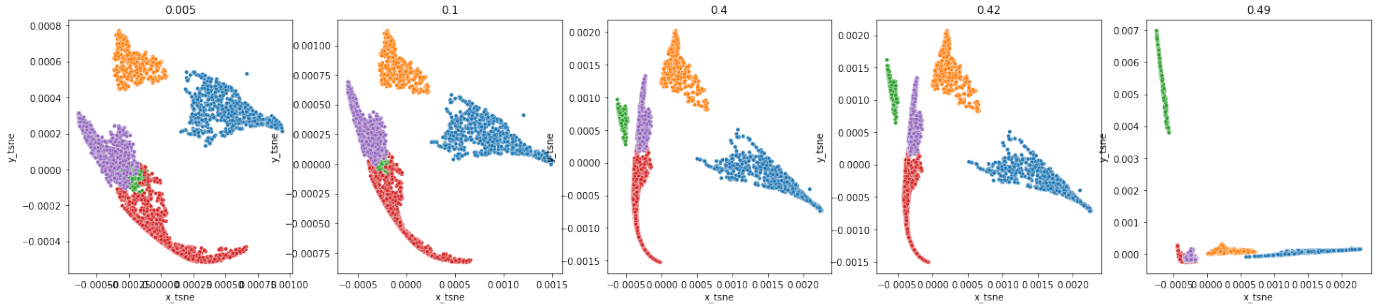

Figura 1.7: Efecto del coeficiente de kernel usando rbf en SE.

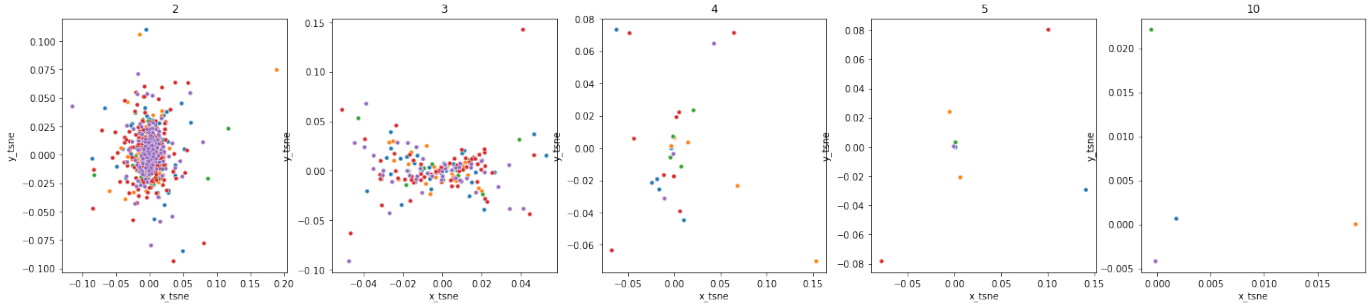

Figura 1.8: Efecto del número de vecinos más cercanos en SE.

T−SNE

Y por último, para este método existen diversos parámetros a considerar: **perplexity, tamaño de aprendizaje y verbose.** Considerar diferentes tamaños de aprendizaje y el verbose distintos no implico diferencias grandes resultados, ya que como tenemos un conjunto de datos pequeño no existen muchas soluciones óptima por lo que cada combinación convergió a la global, por lo cual se omiten los resultados. Entonces se ocuparon los valores de perplexity: $5, 10, 20, 40, 60$. Entre más grande se la perplexity se observan más claro la separación de distintos tipos de grupos, es decir, se hace más claro el patrón de los continentes. **La mejor combinación sería considerar una perplexity igual a 40.**

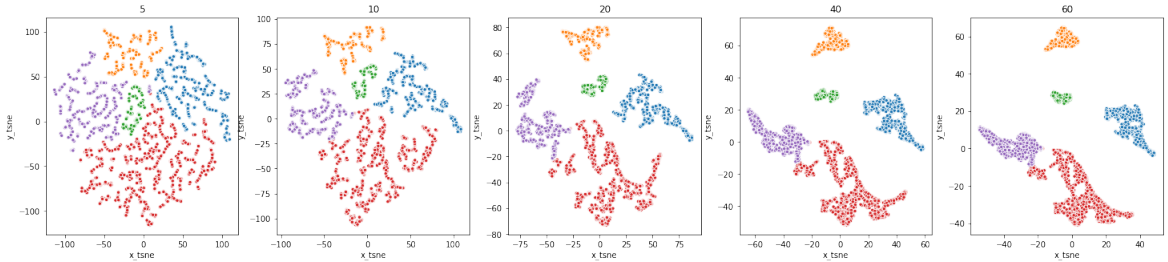

Figura 1.9: Efecto de la perplexity en T−SNE

1.2. Resumen y Conclusiones

Por último, compararemos todas las mejores combinaciones de cada método explicado anteriormente (ver Figura 1.10). Recordemos que estamos interesados en tener la información de los continentes dentro y fuera de ellos.

PCA no logra capturar capturar ninguno de los patrones de los datos originales, por lo que en este caso es el método con los peores resultados.

T−SNE logra capturar el patrón dentro de los continentes, es decir, puede separar claramente los puntos por continente. Pero la interpretación fuera de los continentes no es clara, ya que sería un error interpretar la distancia que existen entre estos continentes, por ejemplo, considerar que los puntos naranjas están más alejados de los puntos rojos esto sería un gran error por definición de T−SNE.

En mi parecer los mejores resultados se obtienen cuando se considero Kernel PCA y SE. Utilizando kernel PCA se puede apreciar los dos patrones del conjunto de datos en 2D, **solo que desde una perspectiva distinta**. Es decir, en la figura 1.1 representa un mapa tradicional del mundo y las proyecciones utilizando Kernel PCA representan la proyección del mundo en un plano perpendicular a los polos (proyección polar (ver Figura 1.11), [1]).

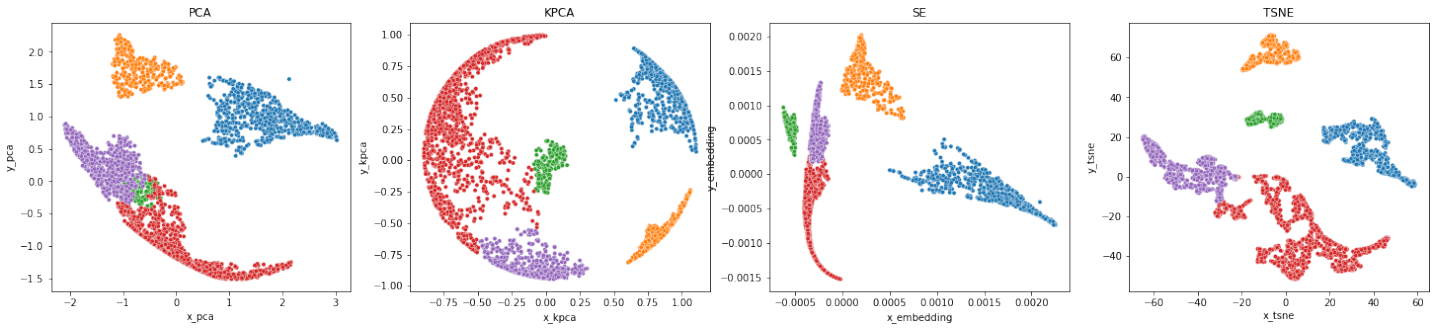

Figura 1.10: Mejores proyecciones de cada métodos.

En cambio SE, presenta una proyección de los datos que se puede apreciar la información dentro y fuera de cada continente. Es decir, **podemos decir que esta proyección es más apegada a la original de los datos en 2D.**

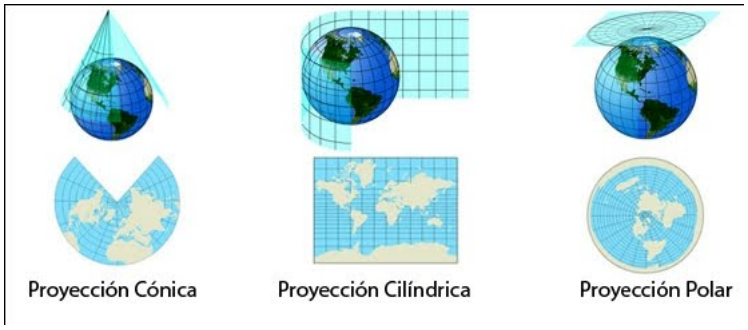

Figura 1.11: Proyecciónes geograficas.

2. Extensión del problema de los Eigenfaces

En este ejercicio nos enfocamos al conjunto de datos Labelled Faces in the Wild, que consiste en fotografías de rostros recolectados de internet y contenido en `sklearn`. Solo se consideraron aquellas personas que tienen al menos 70 fotografías de su rostro en su tamaño original de la imagen (125×94).

El número total de imágenes analizadas fueron 1288 (ver Cuadro 2.1).

Nombre	Número de Fotografia
George W Bush	530
Colin Powell	236
Tony Blair	144
Donald Rumsfeld	121
Gerhard Schroeder	109
Ariel Sharon	77
Hugo Chavez	71

Cuadro 2.1: Número de fotografías por personaje en la muestra.

Si ocupamos en el enfoque tradicional de PCA para reducir el número de variables (en este caso, cada pixel), para poder representar cada fotografía de los rostros en 2D (ver Figura 2.1). Se esperaría que la representación en 2D permitiera identificar los grupos de fotografías por cada uno de los 7 personas que están en todas las fotografías, pero usando PCA esto no es posible. **Por lo que utilizaremos otros métodos de reducción de dimensionalidad para buscar una representación en donde se observe los 7 grupos esperados.**

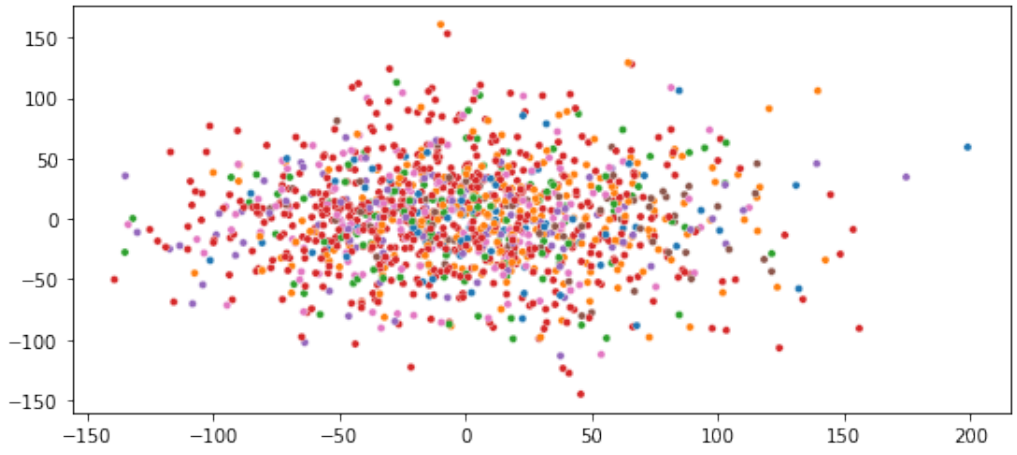

Figura 2.1: Representación de los rostros en 2D.

Kernel PCA, SE y T−SNE.

Consideramos los métodos de manifold learning basado en Kernel PCA, SE y $t-$SNE para obtener las representaciones en 2D de los rostros. De el Sección 1, nos dimos cuenta de la importancia que tienen los hiperparámetros en estos métodos.

Para **Kernel PCA** se comparo el efecto de: **kernel a ocupar y coeficiente del kernel**. Se compararon las mismas combinaciones que en la sección anterior (un total de 34 combinaciones), **y la mejor combinación fue considerando un kernel coseno**.

Ahora, para **SE** los hiperpárametros a comparar fueron: **construcción de la matriz de afinidad, número de vecinos más cercanos y coeficiente del kernel.** De igual manera, se compararon las mismas combinaciones que en la sección anterior, y **la mejor combinación fue utilizando los vecinos más cercanos para construir la matriz de afinidad con 7 vecinos más cernanos.**

Y por último, para $t-$**SNE** a diferencia de la sección anterior la tasa de aprendizaje tuvo un efecto notorio, por lo que los hiperpárametros que se compararon son: **perplexity, tasa de aprendizaje y verbose.** Para este método se probaron alrededor de 50 combinaciones de parámetros, y **la mejor combinación fue considerando un valor de 40 de perplexity, usando un valor de 15 en la tasa de aprendizaje y verbose igual a 1.**

Para no hacer más extenso este trabajo omitimos los efectos que tienen cada uno de los hiperpárametros ya que en la sección 1 ya se mostró la importancia de estos, pero para más detalle puede consultar las visualización del archivo `eigenfaces_plus.ipynb`.

2.1. Comparación de los métodos

Considerando las mejores combinaciones descriptas en la subsección anterior, procedemos a proyectar los datos de las imagenes de los rostros a dos dimensiones (ver Figura 2.2). **Cabe resaltar que con ninguno de los cuatros métodos pudo representar los datos de tal manera que se puedan dividir el espacio en el número de personajes que aparecen en las fotografías**, es decir, no es posible representar la información obtenida de todos los pixeles de las fotografías en solo dos dimensiones. Pero si existe una mejoría contra las proyecciones usando PCA.

Si comparamos las proyecciones usando PCA (Figura 2.1) con las proyecciones usando Kernel PCA y SE podemos concluir que **ningún de estos métodos presenta mejorías para poder clasificar los datos por cada uno de los personajes**, parece que los datos están de forma aleatoria en todo el espacio.

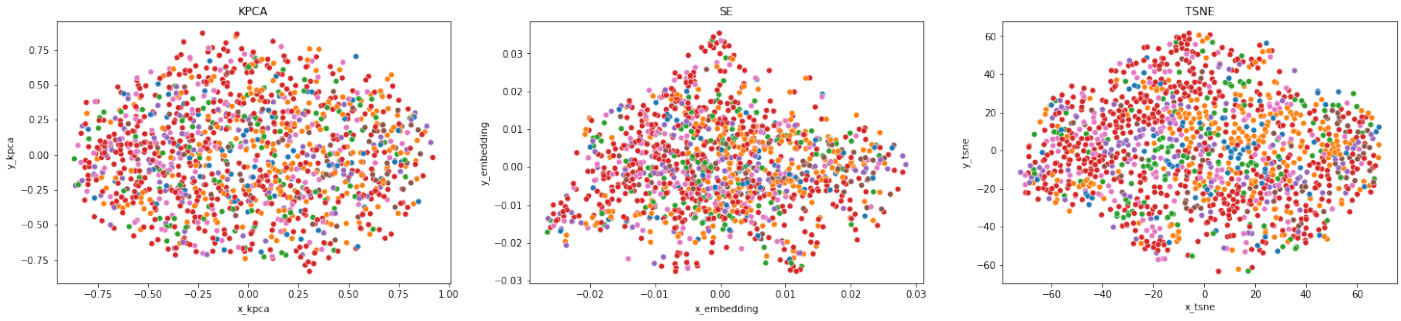

Figura 2.2: Mejores proyecciones de los rostros en 2D

En cambio, si observamos las proyecciones **usando $t-$SNE se pueden notar pequeñas agrupaciones de fotografías del mismo personaje en todo el espacio** (ver Figura 2.3), es decir, este método no logra separar todas las fotografías de un mismo personajes pero si lograr agrupar

algunas fotografías de un mismo personaje, por lo que se considera que este método es mejor que usar PCA, Kernel PCA y SE.

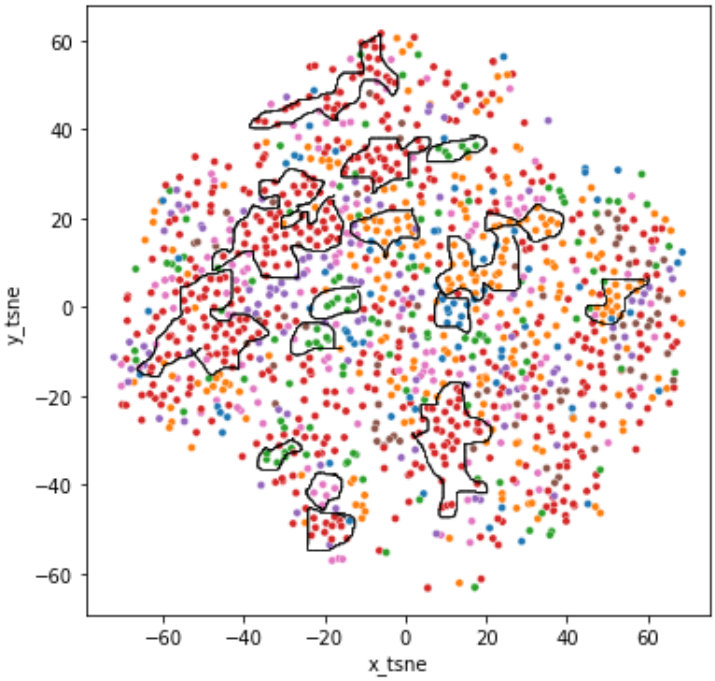

Figura 2.3: Proyecciones de las fotografías usando $t-$SNE.

Observamos la Figura 2.4 podemos notar que existe una concentración de fotografías de Hugo Chavez en el espacio, pero en esta concentración no están todas las fotografías (los puntos cafes representan las fotografías de Hugo Chavez).

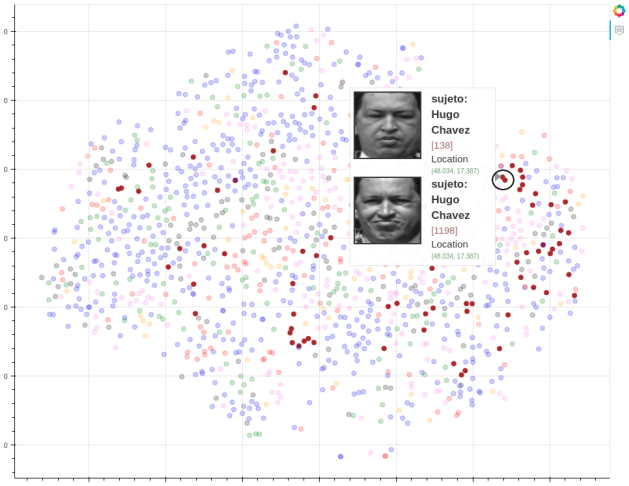

Figura 2.4: Fotografías de Hugo Sanchez.

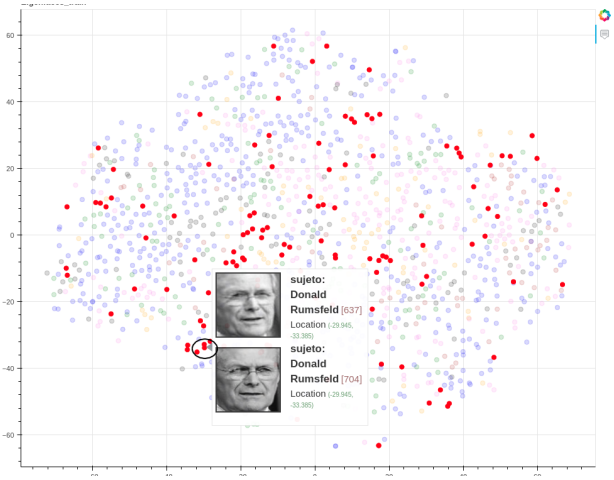

Figura 2.5: Fotografías de Donald Rumsfeld

De igual manera, si observamos la Figura 2.5 podemos observar algunas concentraciones de las fotografías de Donald Runsfeld (los puntos rojos representan las fotografías de Donald Runsfeld).

Además, se puede observar que no existe alguna agrupación que tenga todas las fotografías de Donald Runsfeld, pero si pequeñas agrupaciones con estas.

Ahora, intente entender como se forman las agrupaciones (ver Figura 2.6), es decir, **no solo considerar quien aparece en las fotografías si no si la persona esta sonriendo, con la boca abierta, con lentes, con vista frontal derecha/izquierda, la edad de la persona, etc,** pero como son múltiples combinaciones implica que no sea posible realizarlo de manera manual.

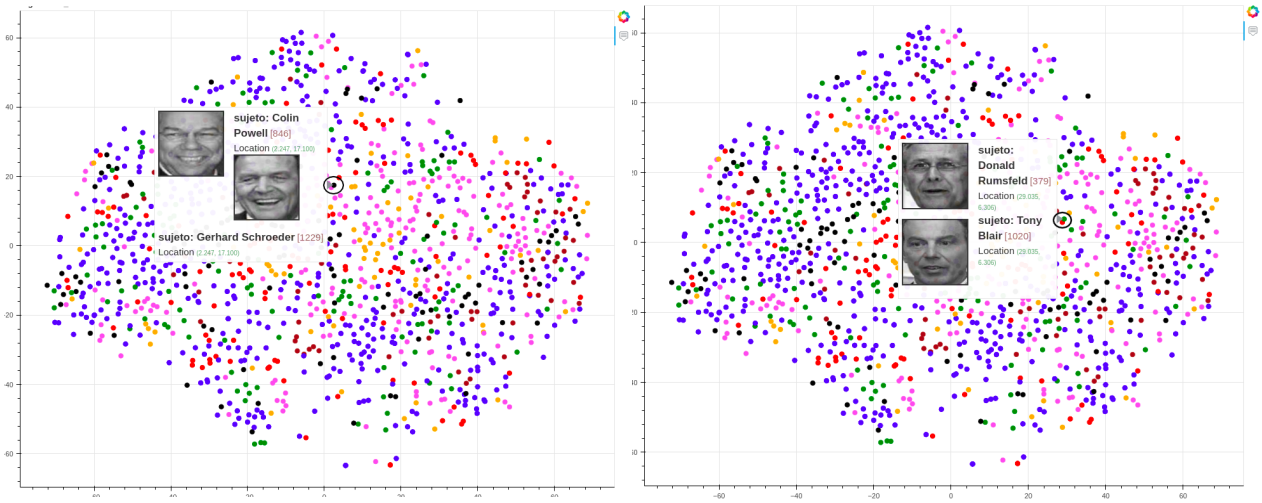

Figura 2.6: Agrupación errónea.

Conclusiones

En conclusión, **con ningún método fue posible de proyectar las fotografías de tal manera que se pudieran identificar cada una de los personajes**. La razón por lo cual no se puede es debido a que estamos reducir demasiadas dimensiones para llegar solo a 2, lo que hace que perdamos bastante información. Además, podemos decir que existe una mejor versión de las proyecciones de las fotografías en 2D, en comparación con las proyecciones de PCA.

Recordando un ejemplo muy parecido visto en clase, de las proyecciones de las fotografías de los dígitos usando PCA y cuando se usa $t-$SNE. En este ejemplo vimos que si se puede identificar la mayoría de los 10 dígitos en el espacio por separado, pero en comparación con el ejercicio analizado en este trabajo las fotografías de los dígitos presentan menor información (varianza en sus variables) que las fotografías de los rostros. Por lo que, esta diferencia de información hace posible o no identificar más fácilmente grupos de fotografías en una dimensión más pequeña.

Anexo

Todos los códigos utilizados para estos resultados se pueden encontrar en mi página personal de Gitgub: Enriquesec. En el repositorio Ciencia de Datos/Tareas/Tareas4/. El notebook `world_2d.ipynb` contiene los resultados de la primera sección, y el notebook `eigenfaces_plus.ipyng` contiene la segunda sección de este trabajo.

Referencias

[1] ArcMap. *Sistemas de coordenadas proyectadas, tipos de proyección.* URL: `https://desktop.arcgis.com/es/arcmap/10.3/guide-books/map-projections/projection-types.htm`.

[2] F. Pedregosa y col. "Scikit-learn: Machine Learning in Python". En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.