Compresión y clustering de datos

Métodos Numéricos y Optimización - Ingeniería en Inteligencia Artificial - Universidad de San Andrés

Ignacio Schuemer (34575) Santiago Tomas Torres (34580)

8 de Junio de 2023

Resumen

Aplicación de métodos numéricos para obtener la descomposición en valores singulares de matrices que son representaciones de imágenes. Se utiliza SVD para realizar la compresión de imágenes reduciendo las dimensiones de la descomposición manteniendo la mayor información posible. Se analizan errores entre la imagen comprimida y la original, y se compara esa misma cantidad de dimensiones utilizadas con otras imágenes distintas.

Luego se comprenden técnicas de clustering de datos a partir de muestras aleatorias de datos en dimensión relativamente grande. Se reducen la muestras a 2 dimensiones utilizando la técnica de PCA. Luego se clasifican a los objetos muestrales en dos clusters mediante el método de *K-means*, *DBSCAN* y clasificación por centroides. Se comparan los resultados obtenidos entre los distintos métodos. Luego se repite este procedimiento para dimensiones de 4, 20 y 106, siendo esta última la dimensión inicial que presenta la muestra antes de realizar PCA. Por último se muestran las diferencias y limitaciones que presenta realizar el análisis para dimensiones más grandes a 2.

Palabras clave: Valores singulares, SVD, Clustering, PCA, Kmeans, DBSCAN.

1. Introducción

Además de su papel fundamental en la resolución de problemas, los métodos numéricos han demostrado ser especialmente eficaces en el análisis y procesamiento de datos complejos en el campo de la ciencia de datos. Estos métodos ofrecen herramientas poderosas para explorar y extraer información valiosa a partir de conjuntos de datos masivos y multidimensionales.

En este trabajo, nos enfocamos en dos aspectos clave de la ciencia de datos: la descomposición en valores singulares (SVD) de matrices que representan imágenes y las técnicas de clustering de datos utilizando muestras aleatorias en dimensiones relativamente grandes. Ambos aspectos son fundamentales en la exploración y comprensión de datos complejos, y nos permiten extraer patrones, identificar estructuras subyacentes y tomar decisiones informadas en función de estas análisis.

En primer lugar, abordamos la descomposición en valores singulares (SVD) aplicada a matrices que representan imágenes. La descomposición SVD es una técnica poderosa que permite descomponer una matriz en tres componentes: una matriz de vectores singulares izquierdos, una matriz diagonal de valores singulares y una matriz de vectores singulares derechos. Al aplicar la descomposición SVD a las matrices de representación de imágenes, podemos realizar una compresión reduciendo las dimensiones mientras se mantiene la mayor cantidad posible de información. Analizaremos los errores entre las imágenes comprimidas y las originales, evaluando así la calidad de la compresión y su impacto en la preservación de los detalles y características esenciales de las imágenes.

En segundo lugar, dado un dataset con muestras aleatoria se buscara encontrar grupos de alta similaridades entre si, para ello, comenzaremos por reducir la dimensionalidad de las muestras de datos utilizando la técnica de Análisis de Componentes Principales (PCA). Esta técnica nos permitirá proyectar los datos en un espacio de menor dimensionalidad, manteniendo la mayor cantidad posible de información relevante. A continuación, aplicaremos diferentes algoritmos de clustering, como K-means, DBSCAN y clasificación por centroides, para clasificar los objetos muestrales en grupos o clusters. Compararemos y analizaremos los resultados obtenidos con cada algoritmo, evaluando su capacidad para identificar estructuras y patrones ocultos en los datos, así como su sensibilidad a la dimensionalidad de las muestras. A continuación se detallan los métodos numéricos empleados.

2. Métodos numéricos utilizados

2.1. Compresión de imágenes

2.1.1. Descomposición en valores singulares (SVD)

El método de *Singular Value Decomposition* (véase [Str19] y [BFB17]) consiste en, dada una matriz, descomponerla mediante tres matrices o componentes principales.
Esta descomposición se define en general para una matriz rectangular $A \in \mathbb{R}^{mxn}$, y su representación por medio de SVD es:

$$A = U \cdot S \cdot V^T \tag{1}$$

donde: $U \in \mathbb{R}^{mxm}$ es una matriz ortogonal que contiene los autovectores izquierdos de A; $S \in \mathbb{R}^{mxn}$ matriz diagonal que contiene los valores singulares de A ordenados en forma descendente; y la matriz $V^T \in \mathbb{R}^{nxn}$ es la traspuesta de una matriz ortogonal V, y contiene los valores singulares derechos de A.

2.2. Clustering de datos

2.2.1. Análisis de Componentes Principales (PCA)

El método de *Principal Component Analysis* es una técnica que se utiliza para reducir la dimensionalidad de una conjunto de datos (dataset) manteniendo la mayor cantidad de información relevante posible. La idea principal es transformar un conjunto de variables correlacionadas en un nuevo conjunto de variables no correlacionadas (es decir, ortogonales entre sí) llamadas componentes principales (véase [Hol08]).

Este método consiste en, como primer paso, estandarizar los datos, lo que implica restar la media de cada variable y dividir por su desviación estándar. A partir de los datos estandarizados, se realiza la descomposición en valores singulares de esta matriz, que llamaremos X. Los valores singulares en S, obtenidos de la descomposición SVD, están ordenados en orden descendente. Luego se seleccionan los primeros k valores singulares y sus correspondientes columnas en V^T, donde k es el número de componentes principales deseados para realizar la reducción. Estas k columnas de V^T formarán la matriz de proyección. En otras palabras, esto permite reducir la dimensionalidad del conjunto de datos al elegir solo las primeras k componentes principales más importantes o deseados. Ahora bien, para proyectar los datos originales en un espacio de menor dimensión, se multiplica la matriz de datos X por la matriz de proyección obtenida en el paso anterior. Es decir, luego de esto los datos originales se proyectan sobre los nuevos componentes principales seleccionados.

Teniendo la matriz del dataset (a la que llamaremos X) y realizando la descomposición mediante SVD ([1]), la fórmula para poder aplicar PCA es realizar la siguiente multiplicación:

$$V_k^T \cdot X \tag{2}$$

2.2.2. Algoritmo *K-Means*

K-means es un método de clustering que se basa en agrupar un conjunto de datos en k grupos (o clusters) basados en sus características similares. Tiene como objetivo principal encontrar k centroides que representen los centros de cada grupo de datos de manera "óptima"(véase [Mac67]).

Lo que realiza este algoritmo es, primero seleccionar aleatoriamente k centroides iniciales dentro del espacio de datos (utilizando PCA). Estos centroides representarán los centros iniciales de los k grupos. A partir de eso, cada punto de los datos se asigna al centroide más cercano, creando así los diversos grupos. La cercanía se calcula utilizando como métrica de distancia a la distancia euclidiana. Posteriormente se calcula el nuevo centroide de cada grupo tomando el promedio de todas las instancias asignadas a ese grupo. Esto se hace para todos los k grupos, lo que resulta en k nuevos centroides. Luego, se ejecuta iterativamente los pasos anteriores de clasificación hasta que los centroides converjan y no haya cambios significativos en las asignaciones de puntos. El algoritmo converge cuando los centroides ya no cambian o cambian por debajo de un umbral predefinido. En este punto, se considera que los grupos están estables y se ha alcanzado una solución. El resultado una vez terminado el algoritmo es una partición de la representación de los datos en k grupos, donde cada uno presenta un centroide.

2.2.3. Algoritmo *DBSCAN*

El algoritmo de *DBSCAN* (del inglés *Density-Based Spatial Clustering of Applications with Noise*) es un método de clustering basado en la densidad de la muestra de puntos a dataset utilizados. Es decir, permite descubrir agrupaciones de puntos de similares características (véase [EKSX96]).

DBSCAN precisa de dos parámetros, ϵ (distancia o radio máximo) y "min-Pts"(mínimo de puntos); a continuación se explicará cómo se utilizan y cómo, en base a estos números *DBSCAN* define tres tipos de puntos en el conjunto de datos:

1. Puntos centrales (*core points*): son aquellos puntos que tienen al menos un número mínimo de puntos (*min-Pts*) dentro de una distancia máxima (*épsilon*). Estos puntos se encuentran en regiones densas del conjunto de datos y formarán el núcleo de los clusters.
2. Puntos de borde (*border points*): Son aquellos puntos que tienen menos puntos que el umbral *min-Pts* dentro de una distancia ϵ, pero se encuentran en la vecindad de un punto central. Estos puntos se consideran parte del cluster, pero no son tan densos como los puntos centrales.
3. Puntos de ruido (*noise points*): Son aquellos puntos que no son puntos centrales ni puntos de borde. Estos puntos no pertenecen a ningún cluster y se consideran ruido o valores atípicos.

Una vez dicho esto, el procedimiento que realiza este método es, inicialmente seleccionar un punto que no se visitó anteriormente. Luego se comprueba si el punto es un punto central calculando el número de puntos dentro de la distancia "epsilon" alrededor de él. Si el número es mayor o igual al umbral *min-Pts*, se marca como punto central y se forma un nuevo cluster. Posterior a eso se expande el cluster alrededor del punto central. Esto implica encontrar todos los puntos alcanzables dentro de una distancia "épsilon"del punto central y asignarlos al cluster. Se repiten los pasos anteriores hasta que se hayan visitado todos los puntos del dataset.

Nota: Este método a diferencia de *K-means* no requiere especificar previamente el número de clusters.

2.2.4. Clasificador mediante centroides

El procedimiento se resume en, dado k centroides iniciales (obtenidos utilizando otro método de clustering como por ejemplo *K-means*), donde k es la cantidad de clusters que sea desea, se calcula la distancia entre cada dato de la muestra y los centroides y los asigna al cluster cuyo centroide esté más cercano, es decir, que la distancia sea mínima.

3. Adaptación y decisiones de implementación

3.1. Compresión de imágenes

Primeramente, se dispone de 16 imágenes para realizar el experimento, cada una de ellas se puede representar como una matriz de píxeles de 28x28. Sin embargo, para el trabajo se buscará representar cada imagen como un vector $\mathbf{x} \in \mathbb{R}^{28*28}$. Con esto se busca definir una matriz de 28x28 = 784 filas y 16 columnas, es decir, cada columna es la representación en forma de vectores de cada imágenes. Por lo tanto, con esto se logra obtener una matriz con la información de todas las imágenes.
Una vez hecho esto, utilizamos la matriz anterior, que denotaremos como A, para realizar su descomposición mediante valores singulares ([1]). En este caso sus componentes principales quedan expresados como $U \in \mathbb{R}^{784\text{x}784}$, $S \in \mathbb{R}^{784\text{x}16}$, y $V^T \in \mathbb{R}^{16\text{x}16}$. Entonces $A = U \cdot S \cdot V^T$; con esta representación y sabiendo que los autovectores y los valores singulares están ordenados de manera decreciente, es decir, que los primeros son los que más información presentan acerca de cada imágen y los últimos los que menos. De esta manera, podemos realizar la compresión de las imágenes mediante la utilización de los vectores que más datos poseen.
En la siguiente subsección se mostrará visualmente la representación matricial de los primeros autovectores. Además, se buscará reducir la dimensionalidad de la representación de varias de las imágenes que se poseen. Y por último se quiere acotar el número mínimo de dimensiones (que denotaremos como d) a las que se puede reducir la dimensión de cada representación de las imágenes, teniendo como medida de error la norma de Frobenius. En otra palabras, la idea del experimento es iterar sobre diferentes valores de d (de manera creciente) reconstruyendo la imagen para cada caso y calcular el error bajo la norma de Frobenius hasta satisfacer que el error entre la imagen completa y la comprimida no exceda el 5 %.
La norma de Frobenius es una medida de la magnitud o tamaño de una matriz y se define como:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

donde a_{ij} representa el elemento en la fila i y columna j de la matriz A.

3.2. Clustering

En este caso se tiene con anterioridad una archivo de formato *.csv* que será el dataset empleado para esta segunda parte del trabajo. Este archivo contiene un conjunto de 2000 muestras, donde cada muestra la

representamos como $\mathbf{x_i} \in \mathbb{R}^{106}$. Utilizando este dataset y calculando la similaridad entre dos muestras x_i, x_j el objetivo será encontrar (si existen) grupos de alta similaridad (*clusters*) entre muestras del dataset y clasificar las muestras según pertenencia a uno de ellos. La similaridad entre dos muestras la podemos definir utilizando una función no-lineal de su distancia euclidiana (métrica de *K-means* [2.2.2]):

$$K_1(x_i, x_j) = \exp - \frac{||x_i - x_j||_2^2}{2\sigma^2}$$

Para realizar lo buscado, se empleará el método PCA ([2.2.1]) para lograr reducir la dimensionalidad del conjunto de datos y proyectarlos en distintas dimensiones para su respectivo análisis. Luego, por consiguiente, en las distintas dimensiones (reducidas) se emplearán distintos métodos o técnicas de clustering tales como *K-means* ([2.2.2]), *DBSCAN* ([2.2.3]) o clasificación mediante centroides ([2.2.4]). A partir de ello se analizarán y compararán los distintos métodos y se concluirán, primero, para qué dimensiones resulta más fácil (y lógico) hacer el experimento; y segundo, qué técnica converge a una mejor solución (dado la muestra que se tiene) en relación a cómo diferencia y encuentra los distintos clusters o grupos de similaridad.

En la siguiente sección se exponen los resultados obtenidos a través de las simulaciones de los métodos en diversas dimensiones y se llevará a cabo un análisis de lo observado.

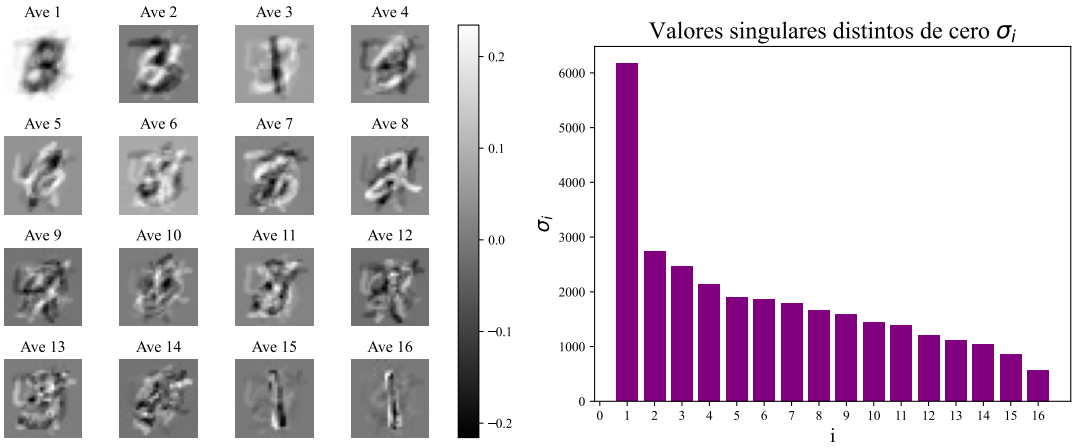
4. Análisis y resultados

4.1. Compresión de imágenes

Como se hizo mención en la sección anterior, representamos cada imagen como un vector fila, y por consiguiente obtenemos una matriz no cuadrada A donde en cada columna i se encuentra el vector (representación) correspondiente a la imagen i. Luego se procede a realizar la descomposición de la matriz que se obtuvo mediante la descomposición SVD. La matriz U tiene los autovectores de la matriz A, por lo que posee información acerca de la matriz, y por consiguiente, de las 16 imágenes. En otras palabras, la matriz U posee los autovectores (izquierdos) de A, los cuales representan las direcciones principales de variabilidad en los datos de cada imagen. Es decir, cada uno proporciona información sobre las estructuras o formas importantes presentes en las 16 imágenes. Además, cabe destacar que forman una base ortogonal del espacio al que pertenece.

Como se poseen 16 imágenes, los primeros 16 (de los 784) autovectores de la matriz U son los únicos que van a poseer datos de las figuras ya que cada autovector se relaciona con su respectiva imagen. Por otro lado, la matriz S presenta los valores singulares de la matriz A. Estos valores se pueden describir como los coeficientes que escalan los vectores singulares de esa misma matriz. Es decir, como están ordenados en orden descendente, significa que los primeros valores singulares tienen más influencia en la reconstrucción de la matriz, y por lo tanto, de las imágenes. Entonces, podemos concluir que los valores singulares más grandes van a representar las características más dominantes o importantes de las imágenes. Sabiendo esto, realizando la selección de los valores singulares más grandes, se puede lograr una aproximación de las imágenes originales con una pérdida mínima de información.

Para comprender en mejor medida este efecto, a continuación graficaremos los primeros 16 vectores que forman la matriz U. Además daremos a conocer los valores singulares de la matriz A para poder analizar lo que señalamos acerca de las direcciones dominantes o privilegiadas:



(a) Primeros 16 autovectores de la matriz U.

(b) Valores singulares σ_i asociados a cada autovector.

Figura 1: Primeros autovectores y valores singulares de la matriz A. Los vectores son las columnas de la matriz U: $u_1, u_2, .., u_{16}$ que fue obtenida por medio de SVD.

Observación 1 *Desde el autovector 17 al 784 solo hay ruido, por lo que al graficarlos en formato de matriz, la imagen es un recuadro negro. Los valores singulares asociados a esos autovectores son 0.*

Como se puede notar en la figura 1, el primer autovector es el que predomina sobre los demás. Esto se determina ya que su valor singular asociado es considerablemente mayor al resto; cuantitativamente se ve incrementado un 115 % con respecto al segundo valor singular. Como los valores se encuentran ordenados de forma decreciente, pierden relevancia a medida que se examinan las dimensiones más grandes. Además se puede corroborar que la tasa de cambio marginal (a partir del segundo valor singular) se reduce a un incremento no mayor al 20 % (sin tener en cuenta al último caso que la tasa es del 51 % aproximadamente). Se mencionó que el primer vector es el que mayor predominancia tiene, lo que se debe a el peso que tiene el primer valor singular, que es considerablemente mayor que los restantes 15 (y por obviedad, que los restantes 768 valores singulares).

Teniendo en cuenta las conclusiones anteriores acerca de la relación entre los datos o información de una imagen y los respectivos valores y vectores singulares de la matriz que la representa, ahora vamos a encontrar un valor d para conocer el número mínimo de dimensiones a las que podemos reducir la dimensionalidad de cada matriz. Como mencionamos en la sección anterior, vamos a encontrarlo por medio de un algoritmo iterativo que calcula el error entre la imagen original y la comprimida en d dimensiones para conseguir una tolerancia menor al 5 %. Cabe destacar que, resulta interesante evaluar la incidencia de comprimir una imagen con determinadas características a una cierta cantidad de dimensiones y considerar si la cota del error propuesta es válida para cualquier otra imagen del conjunto. A continuación se muestra una figura que da noción acerca de lo previo:

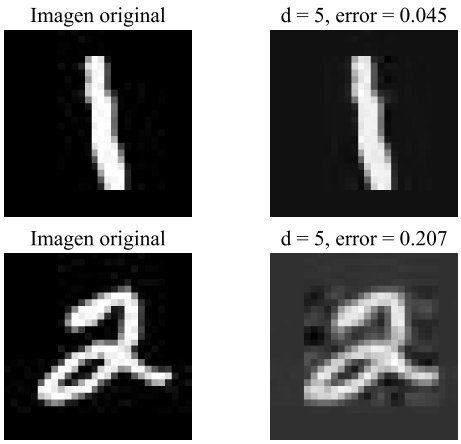


Figura 2: Valor de $d = 5$ hallado para la primera imagen y comparación (utilizando el mismo valor) con el error de la segunda imagen.

Como se observa en la figura 2, encontramos el valor de la mínima dimensión para una figura (elegida al azar de entre las 16), con el cual se llega a un error de aproximadamente 4,5 % en su representación mediante valores singulares. Utilizando ese mismo d se calculó el error que presenta otra imagen. En este caso el error alcanzó valores de casi 21 %; esto afirma lo que se mencionó anteriormente que mientras más información presenta la imagen (más píxeles en blanco haya), se necesitará tomar una mayor cantidad de dimensiones para reconstruirla y garantizar un cierto error.

Para comprender en mejor medida lo analizado, se utilizaron 4 imágenes distintas para graficar el error de reducción de dimensionalidad a medida que aumenta el valor d.

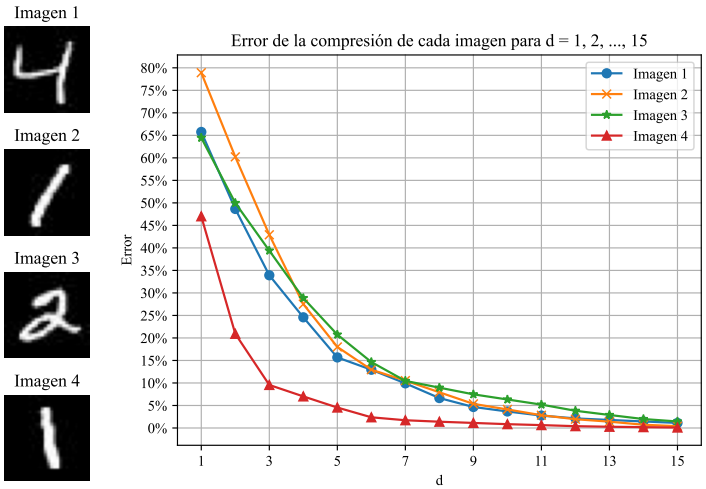


Figura 3: Comparación entre errores de compresión para distintas imágenes con valores de d desde 1 hasta 15.

Se observan claras diferencias en los errores entre varias imágenes. Por ejemplo, en la imagen 3 a simple vista parece ser que aquellas que presentan una mayor distribución de la información (píxeles en blanco más distribuidos) y además no cuentan con imágenes con cierta similitud en el dataset, requieren de tomar más dimensiones para su reconstrucción y asegurar cierto error. En este caso de estudio, fijando $d = 5$ la imagen alcanza un error menor al 5 %, mientras que las otras 3 imágenes superan el 15 %.

4.2. Clustering de datos

En esta sección del informe se procederá a mostrar los resultados obtenidos a partir del análisis del dataset de muestras. Para tener una idea gráfica de los datos se realizará PCA en 2 dimensiones y luego se graficará su resultado.

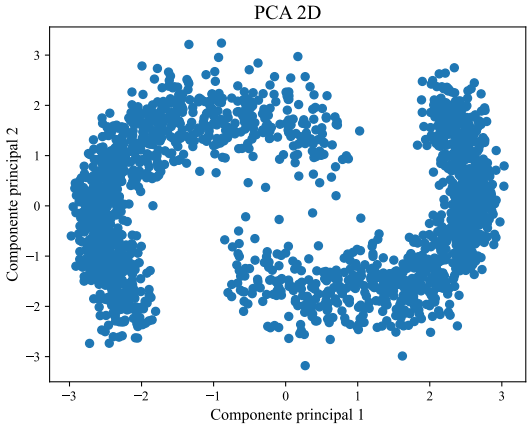
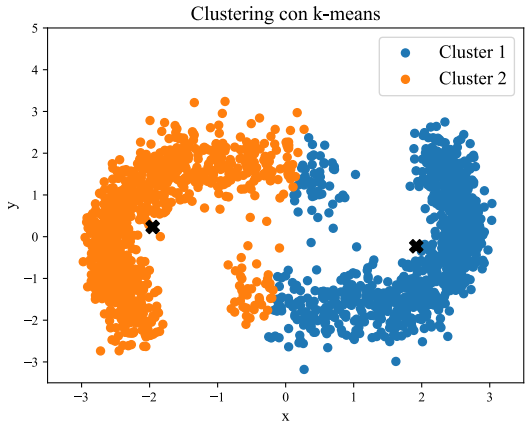
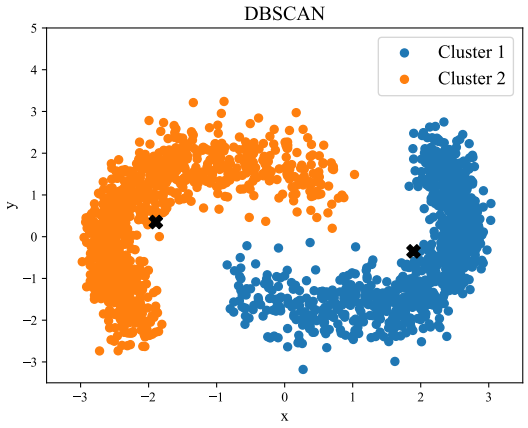


Figura 4: Proyección de los datos en 2 dimensiones mediante PCA.

En la figura 4, se logra diferenciar a simple vista 2 subgrupos de aglomeraciones de puntos. Realizaremos pruebas con los algoritmos mencionados (ver sección 3.2) para comprobar si realmente se pueden identificar 2 clusters como se supone.



(a) Separación en clusters con algoritmo K-means.



(b) Separación en clusters con algoritmo DBSCAN. Parámetros : $\epsilon = 0{,}67$, $min - Pts = 10$.

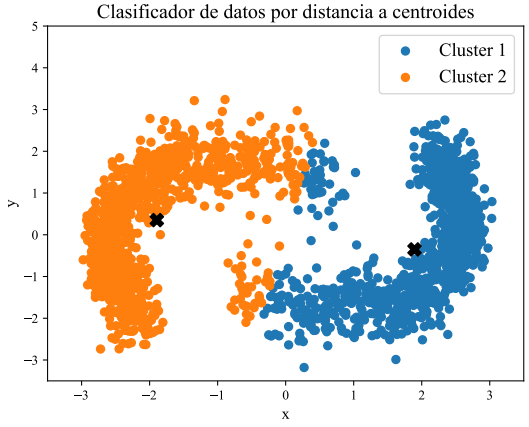
Figura 5: Clustering de los datos por medio de dos algoritmos de clasificación

Observación 2 *Los centroides calculados en ambos métodos se ubican en posiciones considerablemente cercanas.*

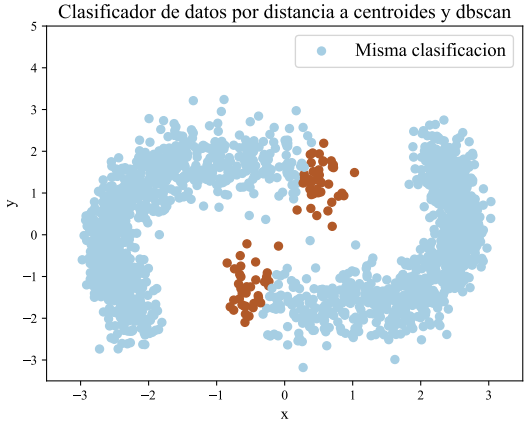
Observación 3 *Como se indicó previamente, al algoritmo de K-means ([2.2.2]) se le debe indicar un número de clusters. La prueba anterior fue realizada con 2.*

Entre los 2 métodos (fig. 5) se puede observar como los resultados obtenidos son diferentes entre sí, esto es producto de que los criterios de clasificación son distintos. Los clusters seleccionados por DBSCAN parecen ser más apropiados.

Se toma como referencia los centroides hallados en la figura 5b para hacer la próxima clasificación de los clusters, la cual consta de tomar la distancia de cada muestra al centroide y definir la pertenencia de cada una al respectivo cluster según su cercanía ([2.2.4]). Se hace la comparación con los clusters que resultan de aplicar DBSCAN.



(a) Separación en clusters con clasificador de distancia a centroides.



(b) Diferencia entre los clusters de DBSCAN y distancia a centroides.

Figura 6: Clasificador de distancias al centroide.

Observación 4 *La clasificación por distancia al centroide da una respuesta similar a la de K-means (fig. 5a).*

Observación 5 *Los métodos discrepan en 76 puntos (marcados con color marrón en la fig. 6b) de un total de 2000, lo que representa un 3,8 % del total.*

Por medio de las simulaciones realizadas podemos inducir de que hay 2 grupos de alta similaridad entre las muestras y que, además, visualmente DBSCAN parece ser un buen criterio para determinar estos subgrupos, mientras que, en este caso, *K-means* no parece tener una precisión adecuada al igual que el clasificador por distancias.

Como las simulaciones mostradas anteriormente se realizaron únicamente teniendo en cuenta la proyección de los datos a un espacio reducido de dimensión 2 y, como cada muestra pertenece a un espacio de dimensión 106, resulta de interés comprobar o verificar si en dimensiones más grandes aún se podrían diferenciar estos subgrupos.

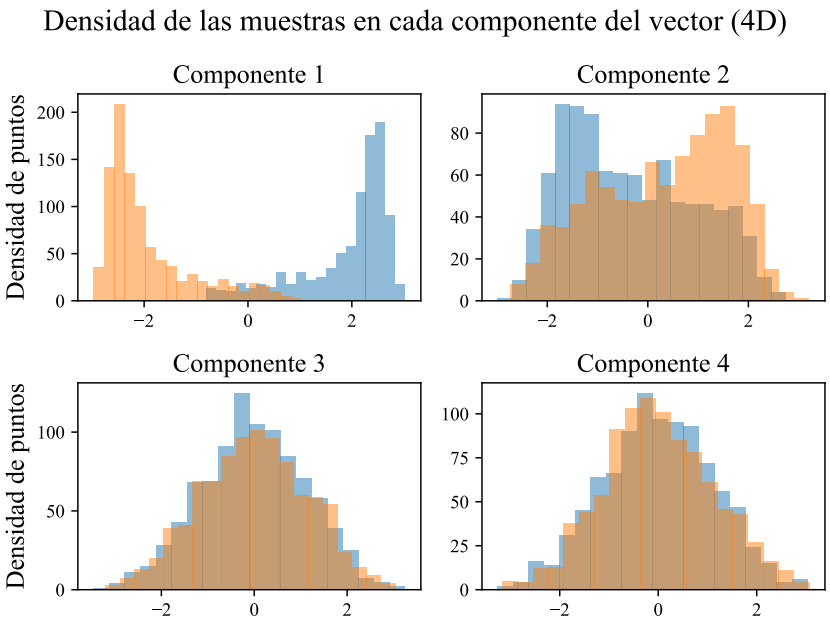


Figura 7: Histograma de la densidad de las muestras según cada componente de la proyección en 4 dimensiones. Clusters identificados por medio de DBSCAN.

Se puede apreciar como varía la densidad de puntos tomando cada una de las componentes de las muestras proyectadas a un espacio reducido de dimensión 4. En las primeras 2 componentes es notable una mayor densidad de los puntos en 2 intervalos distanciados, mientras que en las componentes 3 y 4 no se notan aglomeraciones de valores, por lo que no se pueden identificar subgrupos de muestras. Este efecto nos plantea la idea de que es posible que proyectar a dimensiones mayores a 2 no permita separar a las muestras en subconjuntos de datos, ya que no se perciben aglomeraciones. Para analizar lo anterior, se lleva a cabo la proyección a espacios de dimensión 20 y 106, tomando las últimas componentes de ambos e inspeccionando cómo es la densidad de las muestras.

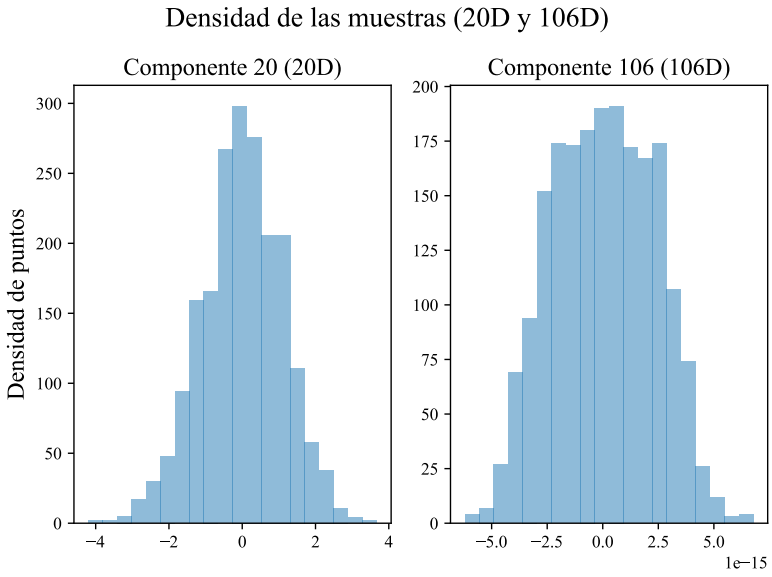
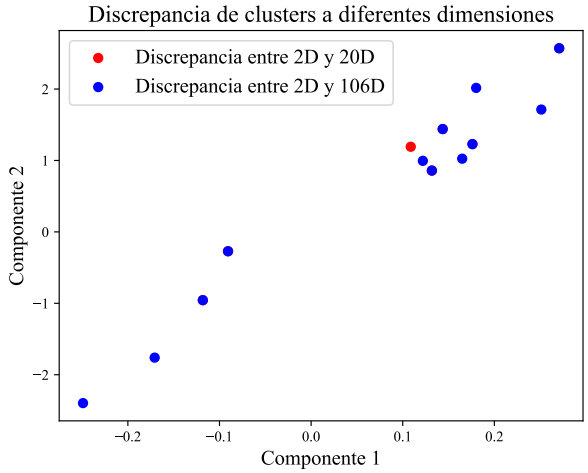


Figura 8: Histograma de la densidad de las muestras en la última componente de la proyección en 20D y en 106D. Hecho con DBSCAN.

Observación 6 *El algoritmo DBSCAN no converge a 2 clusters; por consiguiente, los gráficos se ven únicamente de un mismo color. El algoritmo deja de agrupar las muestras en clusters desde la proyección en 5D en adelante.*

Observación 7 *Los valores de la componente 106 varían muy poco (observar que la escala del eje es de órden de* $1e^{-15}$*). Los datos caen prácticamente en el mismo valor que es 0.*

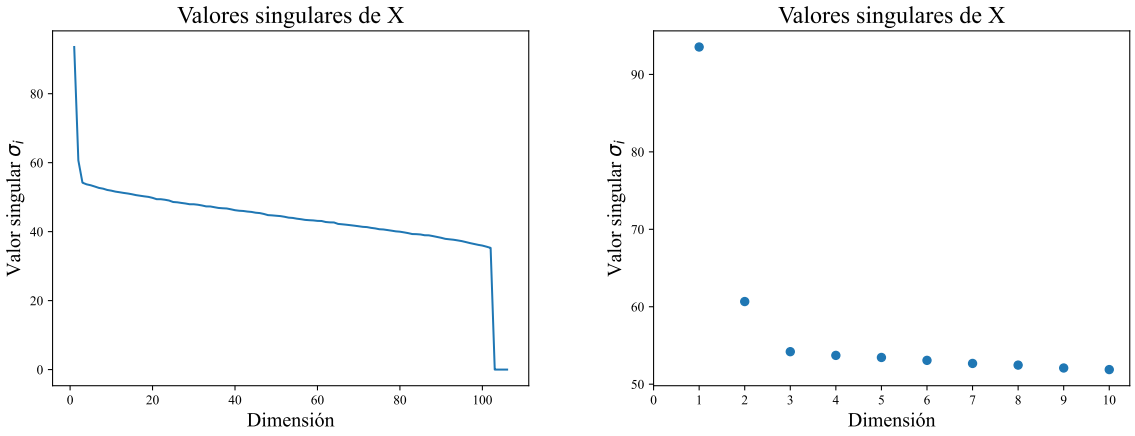
Los datos en estas dimensiones no muestran aglomeraciones de valores en diferentes rangos sino que se acumulan en un mismo intervalo, es decir, o no se ven clusters o hay uno único. Más aún se probó de graficar las 106 dimensiones proyectadas y los resultados fueron similares (parecidos a la gráfica de una campana de Gauss). Por lo tanto se puede concluir que a mayores dimensiones ($d \geq 3$) encontramos información repetida o que es redundante y que se puede considerar como ruido ya que no aporta información nueva para diferenciar datos. Para ver este fenómeno utilicemos el algoritmo de *K-means* (que separa siempre en la cantidad de clusters que se la pasa como parámetro) para diferentes dimensiones y analicemos si es significativo el cambio en el agrupamiento de subgrupos.



Dimensiones	Cantidad de elementos	
	Cluster 1	Cluster 2
2	1009	991
4	1009	991
20	1010	990
106	1005	995

Figura 9: Discrepancia en la agrupación de los clusters a diferentes dimensiones con el algoritmo *K-means*.

Agregar dimensiones no parece cambiar en gran medida el criterio de agrupación de los clusters, por lo que las primeras 2 dimensiones son las que muestran mayor relevancia para encontrar similitud entre las muestras. Lo exhibido motiva a analizar los valores singulares del dataset para determinar las direcciones privilegiadas de las muestras, que es consistente con ver las componentes con mayor varianza.



(a) Todos los valores singulares (1 a 106).

(b) Valores singulares de 1 a 10.

Figura 10: Valores singulares σ_i.

Observación 8 *El primer valor singular es* 72 % *más grande que el tercero; el segundo presenta un incremento de* 12 % *en comparación al tercero. Similarmente, la tercera supera en un* 0,8911 % *a la cuarta.*

Observación 9 *A partir del tercer valor singular del dataset, el "peso"asignado a cada dimensión (valor singular) comienza a ser semejante y, a su vez, decrece. Es decir, no varía significativamente.*

La mayor varianza se encuentra en las primeras dos direcciones y con estas se pueden diferenciar los datos en clusters.

4.3. Conclusiones

A modo de síntesis, se pudo ver y reconocer que mediante la representación matricial y la descomposición en valores singulares de imágenes se puede realizar la compresión de ellas utilizando la mínima información posible, pero usando los autovectores más importantes y dominantes. Por consiguiente, se puede lograr obtener un error relativamente chico en comparación a la representación original. Además se concluyó que, para varios casos, el mismo valor utilizado para reducir la dimensionalidad de la matriz de una imagen no se adecua a la compresión para otra imagen, ya que puede presentar más error dependiendo de que tanta información haya en cada una ([3]).

En la segunda parte del trabajo se pudo ver y reconocer los resultados que se obtienen con distintos métodos de clustering luego de proyectar la muestra en distintas dimensiones. Por un lado, se observó que el método de *K-means* y el clasificador por medio de centroides no fueron efectivos para encontrar los clusters adecuados. El nivel de certeza de estos algoritmos varía según el dataset del que se dispone, y, en este caso, como los datos no son linealmente separables, la respuesta no fue acertada. Por otro lado, el algoritmo *DBSCAN* logró identificar dos subgrupos de datos de manera más precisa.

Se observó que a medida que aumentaba la dimensionalidad del dataset, a partir de la tercera dimensión, se introducía ruido que comenzaba a dificultar la identificación de grupos con alta similitud entre las muestras. Esto se atribuye a la escasa variación presente en las dimensiones superiores, lo que hace que los datos sean más difíciles de agrupar en base a su similaridad y nos deriva en los valores singulares del dataset.

Como cierre, podemos afirmar que en cuanto a la compresión es determinante el número de dimensiones que se selecciona y que constituye a un *trade-off* entre la calidad y peso de la imagen comprimida. Y en el caso del clustering se concluye que es importante comprender la naturaleza del problema y llegar a una adaptación de los datos que se disponen para facilitar su análisis y obtener resultados contundentes.

Referencias

[BFB17] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis*. Cengage learning, 2017.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[Hol08] Steven M Holland. Principal components analysis (pca). *Department of Geology, University of Georgia, Athens, GA*, 2008.

[Mac67] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA, 1967.

[Str19] Gilbert Strang. *Linear algebra and learning from data*, volume 4. 2019.