
**PRÁCTICA 1B: RECONOCIMIENTO DE
LETRAS Y DÍGITOS MEDIANTE EL
ANÁLISIS DE COMPONENTES
PRINCIPALES (PCA)**

**Ingeniería del Conocimiento 3º curso
Curso académico 2023-2024**

Yerson Caleb Yarhui Sarate
17/11/2023

Contents

1	Introducción	3
2	Explicación del problema	3
3	Resultados	4
3.1	Binarias Conflitivas	4
3.2	Clasificación múlticase de MNIST	4
3.3	Clasificación múlticase de EMNIST	5
4	Conclusión final	6
4.1	Clasificaciones Binarias Conflitivas	6
4.2	Clasificación Multicase de MNIST	6
4.3	Clasificación Multicase de EMNIST	6

1 Introducción

Evaluaremos el algoritmo de las componentes principales (PCA) para el reconocimiento de imágenes para diferentes conjuntos de datos. Para resolver estos problemas debemos realizar un proceso de aprendizaje que permita obtener una función, a partir de los ejemplos de entrenamiento disponibles (de los que conocemos su clase), que nos permita clasificar nuevos ejemplos que debamos clasificar en el futuro.

Con la técnica aplicada del PCA obtenemos en general de beneficios la reducción de dimensionalidad ya que ayuda a reducir la complejidad de las imágenes al proyectarlas en un espacio de menor dimensión, ya que puede preservar la información esencial mientras reduce la redundancia, pero se asume que los datos observados son combinación lineal de una cierta base.

Tenemos dos conjuntos de datos $dataSet = \{MNIST, EMNIST\}$ donde MNIST es un dataset que consta de $28 \times 28 = 784$ pixeles de imágenes de dígitos en el rango de 0-9, y EMNIST es un conjunto de imágenes de números del 0 al 9, letras mayúsculas de la 'A' a la 'Z' y letras minúsculas de la 'a' a la 'z', donde solo consideremos las mayúsculas.

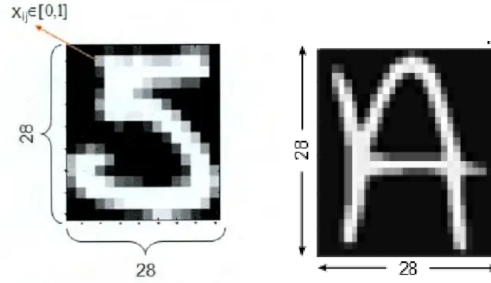


Figure 1: Dígito representarlo en 28×28 pixeles. La imagen del '5': $X_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{i784}\}$, $y_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, y la imagen de 'A': Dígito representarlo en 28×28 pixeles. Además: $X_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{i784}\}$, $y_i \in \{1, \dots, 26\} \approx \{'A', \dots, 'Z'\}$

2 Explicación del problema

Para comprobar el funcionamiento y evaluación de el modelo PCA para clasificar imágenes con nuestro primer conjunto del dataSet, MNIST, donde solo contiene dígitos del 0 al 9 nos surge el problema de que, por ejemplo, la forma de escribir los números $\{1, 7\}$, $\{2, 7\}$, $\{3, 8\}$, $\{5, 6\}$, $\{4, 9\}$ o $\{0, 6\}$ puede que se parezcan por las formas de escritura y, a simple vista, si la escritura es mala también nos costaría identificar que dígito es realmente. Nos centraremos especialmente en la clasificación de nuestro algoritmo para estos dígitos conflictivos, y también para ver lo bueno que es clasificando entre todos los dígitos $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, 10 clases en total. A esto le combinaremos el cambio de tamaño para los datos de entrenamiento y los datos de test para identificar si con un aumento de tamaño en estos conjuntos concluye en mejores resultados.

Con nuestro dataSet EMNIST, que contiene dígitos del 0 al 9, letras mayúsculas de la 'A' a 'Z' y minúsculas de la 'a' a 'z', nuestro objetivo será clasificar correctamente letras mayúsculas (26 clases).

3 Resultados

3.1 Binarias Conflitivas

Para estudiar la clasificación con $\{\text{sizeTrain} = 80, \text{sizeTest} = 20\}$ de imágenes. Cuando probamos a detectar entre los dígitos 1 y 7 el $\text{accuracy} = 100\%$, cuando probamos con 2 y 7 el $\text{accuracy} = 90\%$, y cuando probamos con 4 y 9 el $\text{accuracy} = 90\%$.

PCA funciona mejor cuando hay menos clases que clasificar o diferenciar, en estos casos que buscará las dos direcciones principales de variación en el espacio de características que maximizan la varianza. Si solo tienes dos clases, las componentes principales (vectores propios) de PCA representarán las direcciones en las que las dos clases difieren más entre sí. Al proyectar nuevas imágenes en este espacio reducido, se podría observar una separación más clara entre las dos clases en términos de las características más importantes.

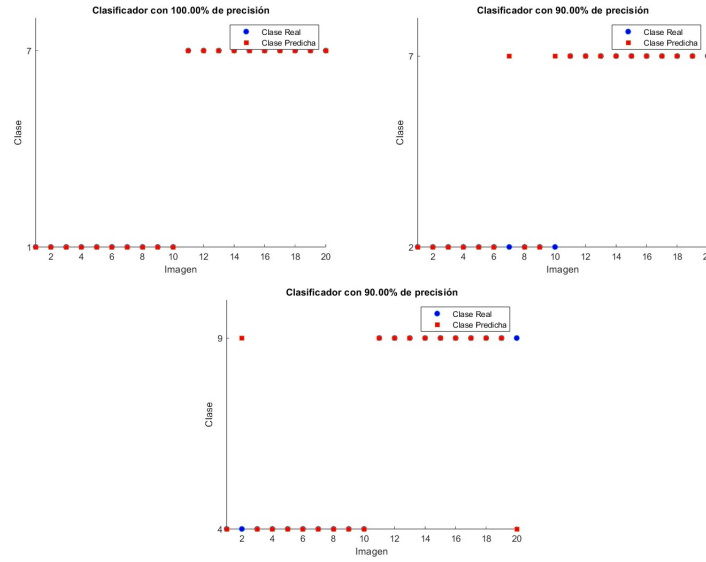


Figure 2: Clasificación de dígitos $\{1, 7\}$, $\{2, 7\}$ y $\{4, 9\}$

3.2 Clasificación múlticlasa de MNIST

Vamos a utilizar todas las clases de nuestro conjunto de imágenes MNIST, es decir, dígitos del 0 al 9, y se varía el tamaño de número imágenes para el entrenamiento y para el test. En el primer caso con $\{\text{sizeTrain} = 80, \text{sizeTest} = 20\}$ de imágenes obtenemos un $\text{accuracy} = 70\%$, y en el segundo caso con $\{\text{sizeTrain} = 160, \text{sizeTest} = 40\}$ de imágenes obtenemos un $\text{accuracy} = 67.5\%$.

En este caso con más clases para aplicar PCA, la capacidad de distinguir entre clases específicas disminuye a medida que aumenta el número de clases. Además a medida que aumentábamos el número de imágenes para el entrenamiento y el test, el accuracy es menor ya que tendría que considerar más espacios y poder abstraer y generalizar más características.

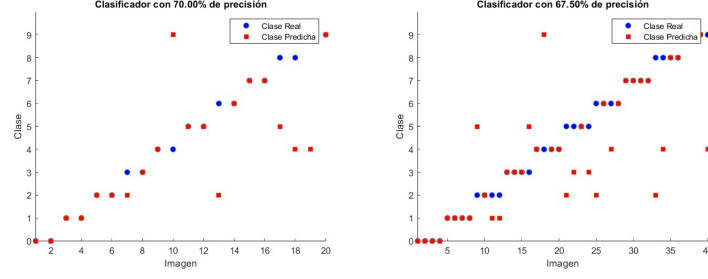


Figure 3: Clasificación de dígitos del 0 al 9 con $\{\text{sizeTrain} = 80, \text{sizeTest} = 20\}$ y $\{\text{sizeTrain} = 160, \text{sizeTest} = 40\}$

3.3 Clasificación múlticlase de EMNIST

Vamos a utilizar todas las clases de nuestro conjunto de imágenes EMNIST, es decir, caracteres de la 'A' a la 'Z' pero los generalizaremos como del '1' al '26', y se varía el tamaño de número imágenes para el entrenamiento y para el test.

En el primer caso con $\{\text{sizeTrain} = 1000, \text{sizeTest} = 104\}$ de imágenes obtenemos un $\text{accuracy} = 32\%$, en el segundo caso, con $\{\text{sizeTrain} = 2000, \text{sizeTest} = 104\}$ de imágenes obtenemos un $\text{accuracy} = 42\%$, con $\{\text{sizeTrain} = 5000, \text{sizeTest} = 104\}$ de imágenes obtenemos un $\text{accuracy} = 45\%$ y con $\{\text{sizeTrain} = 10000, \text{sizeTest} = 104\}$ de imágenes obtenemos un $\text{accuracy} = 44\%$.

En este caso al aumentar el tamaño del conjunto de entrenamiento parece tener un impacto positivo en la precisión del modelo, hasta cierto punto. Es posible que haya una saturación en la mejora de la precisión al continuar aumentando el tamaño del conjunto de entrenamiento.

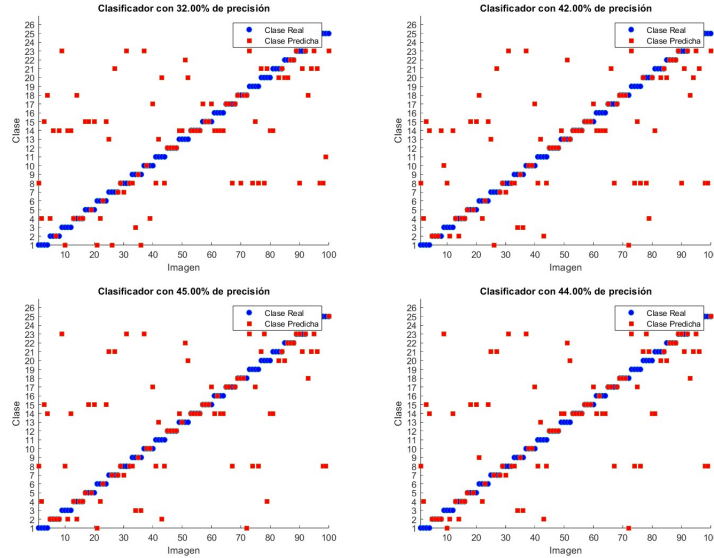


Figure 4: Clasificación de caracteres de la 'A' a la 'Z' con $\{\text{sizeTrain} = 1000, \text{sizeTest} = 104\}$, $\{\text{sizeTrain} = 2000, \text{sizeTest} = 104\}$, $\{\text{sizeTrain} = 5000, \text{sizeTest} = 104\}$ y $\{\text{sizeTrain} = 10000, \text{sizeTest} = 104\}$

4 Conclusión final

Se llevaron a cabo experimentos con conjuntos de datos como MNIST y EMNIST, evaluando la capacidad de PCA para clasificaciones binarias y multiclase. PCA ha resultado eficaz para la clasificación binaria y en problemas multiclase, destacándose en la reducción de dimensionalidad y búsqueda de direcciones principales de variación.

Para aplicar PCA, primero calculamos la matriz de covarianza de este conjunto de datos. La covarianza entre dos variables nos indica cómo varían juntas. En este contexto, la covarianza entre dos píxeles nos dirá cómo varían conjuntamente en todas las imágenes. La matriz de covarianza C para un conjunto de datos X se define como: $C_{N_{\text{imágenes}} \times N_{\text{imágenes}}} = (X - \bar{X})^T(X - \bar{X})$, luego se calculan los vectores propios y valores propios de la matriz de covarianza $\{\lambda, V\}$. Los vectores propios representan las direcciones principales de variación, y los valores propios indican la cantidad de variabilidad en esas direcciones. Ahora, finalmente proyectaremos las imágenes en una nueva base donde se trataran estos datos $W_{N_{\text{imágenes}} \times N_{\text{imágenes}}} = (\frac{(X - \bar{X})V}{\sqrt{\lambda}})^T(X - \bar{X})$. Con esto podremos generalizar las imágenes de cada clase como una media de imágenes de los de W correspondiente a cada clase, es decir por cada clase haremos una media de cada imagen para situar un centroide y poder saber la cercanía o lejanía de una imagen a cada clase: $\text{ImagenGeneral}[clase]_{imagen_i} = \text{mean}(W[clase], imagen_i) | imagen_i = 1, \dots, \text{card}(X)$

En resumen, PCA abstrae características importantes al encontrar las direcciones en las que los datos tienen la mayor variabilidad y proyectar los datos originales en ese espacio de menor dimensión. Las primeras dimensiones capturan la mayor parte de la variabilidad en los datos, permitiendo una representación más compacta y preservando las características más importantes. La elección del tamaño del conjunto de entrenamiento es crítica y puede influir significativamente en la precisión del modelo. En problemas más complejos, como la clasificación de letras, se evidenció la necesidad de explorar técnicas más avanzadas y conjuntos de datos más grandes.

4.1 Clasificaciones Binarias Conflitivas

PCA demostró un buen rendimiento en la clasificación de dígitos conflictivos como 1, 7, 2, 7, y 4, 9. La capacidad de PCA para buscar las direcciones principales de variación en el espacio de características resultó eficaz en la distinción de clases similares.

4.2 Clasificación Multiclase de MNIST

Con el conjunto de datos MNIST (dígitos del 0 al 9), PCA mostró resultados positivos.

La precisión del modelo fue evaluada con diferentes tamaños de conjuntos de entrenamiento y prueba, observándose una disminución en la precisión al aumentar la complejidad del problema.

4.3 Clasificación Multiclase de EMNIST

EMNIST, que incluye letras mayúsculas, minúsculas y los datos de MNIST, aunque solo utilizamos las letras mayúsculas, presentó un desafío adicional para PCA.

Se observó un impacto positivo al aumentar el tamaño del conjunto de entrenamiento, aunque la mejora se saturó después de cierto punto.