Aprendizaje automático

Práctica 6. Análisis de componentes principales

Universidad de Zaragoza, curso 2022/2023

Juan Eizaguerri Serrano 816079

Reducción de la dimensionalidad en MNIST

En esta práctica se va a resolver el problema de clasificación multiclase de números manuscritos. Se va a utilizar el dataset MNIST, de imágenes de 20x20 píxeles. Los atributos de los datos será la intensidad de cada uno de los píxeles. Este dataset consta de 5000 datos, que se van a partir en 4000 para entrenamiento y 1000 para test. Además, se quiere utilizar el algoritmo PCA para reducir la dimensión de los datos.

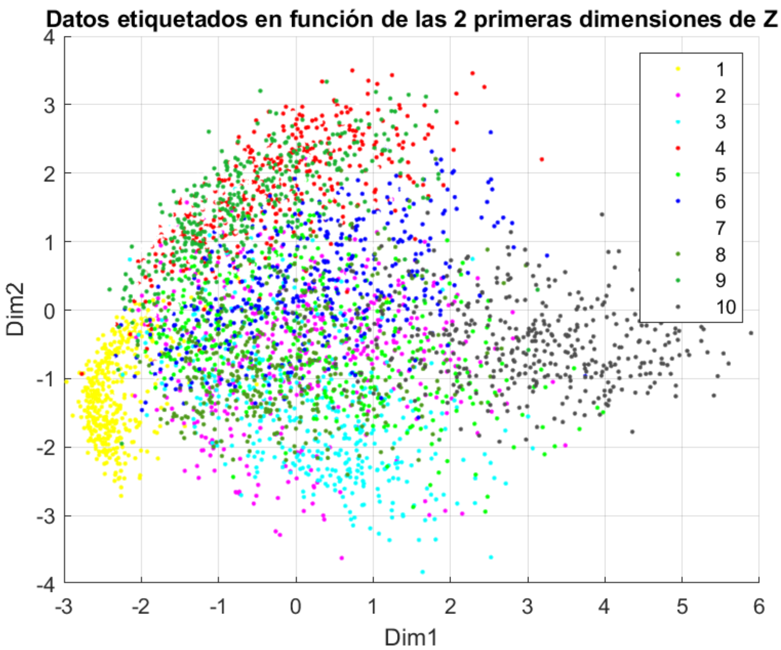
Aplicación de PCA para el aprendizaje supervisado

En primer lugar se estandarizan los datos restándoles la media de cada dimensión para centrarlos en el origen. A continuación se calcula la matriz de covarianzas Σ de los atributos, que muestra en qué medida varían los atributos conjuntamente.

A partir de Σ se obtienen su matriz de vectores propios U y su matriz diagonal de valores propios Λ. A continuación se ordenan los vectores propios de U según los valores propios de Λ.

Los vectores de U permiten aplicar una transformación a los datos X:
Z=XU

Se puede visualizar los datos etiquetados respecto a las 2 dimensiones más relevantes.

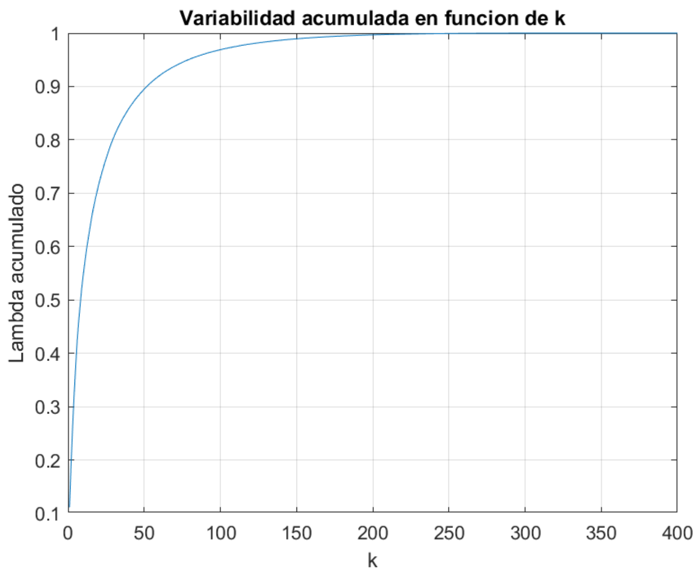


Se observa que, tan sólo con estas dos dimensiones, ya se puede realizar cierta separación entre los datos etiquetados. Por ejemplo, se ve que los datos de clase 1 están claramente separados del resto.

Medidas en función de las dimensiones

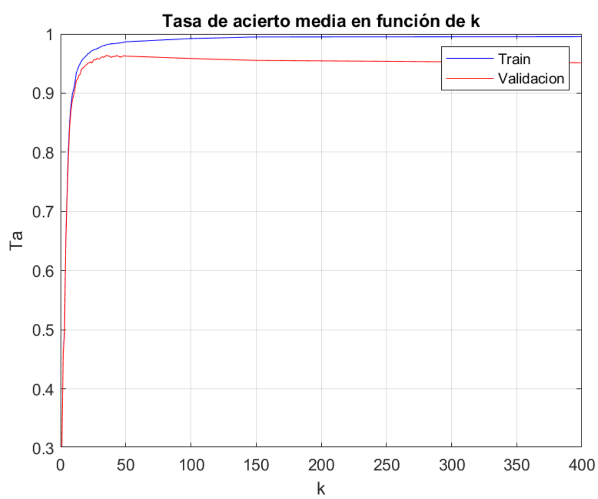
Se quiere entrenar un modelo de clasificación Bayesiana de covarianzas completas utilizando como entrada Z_k, que utiliza los k primeros vectores propios de U.

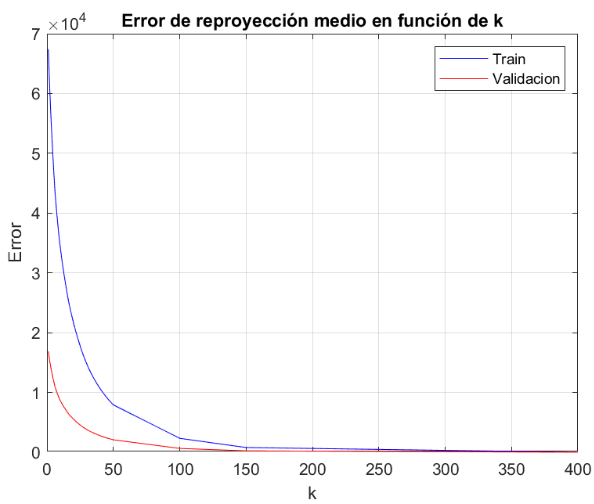
Es interesante comprobar la variabilidad para cada k, resultante de la suma acumulada de los valores propios desde 1 hasta k, resultante de sumar los k primeros valores propios de Λ y dividirlos por la suma de todos los valores de Λ.



Se observa que la variabilidad es del 90% con tan sólo las primeras 50 dimensiones, y se alcanza el 99% con 153 dimensiones.

Para ver el impacto de la elección del parámetro k sobre la tasa de acierto del modelo, se va a entrenar el modelo Bayesiano de covarianzas completas y parámetro de regularización 10^{-2} desarrollado en la práctica anterior con distintos valores de k, utilizando k-fold cross-validation y calculando la media de tasa de acierto y el error de reproyección medio para los datos de validación y entrenamiento con cada uno de los modelos.





Se puede comprobar que para valores muy bajos de k la tasa de acierto es cercana a 0 tanto para los datos de entrenamiento como para los de validación, es decir, se produce sub-ajuste. Ambas tasas de acierto aumentan rápidamente. Hay un codo en torno a k=30, y a partir de ahí la tasa de acierto aumenta muy lentamente para datos de entrenamiento e incluso disminuye para lo de validación, produciéndose un ligero sobre-ajuste.

El punto óptimo se encuentra en k = 35, para el cual la tasa de acierto media para los datos de validación es 0.9637.

Como es de esperar, el error de reproyección es siempre decreciente conforme aumenta k, disminuyendo muy rápido al principio. Llegando a 0 cuando la k es igual al número de dimensiones

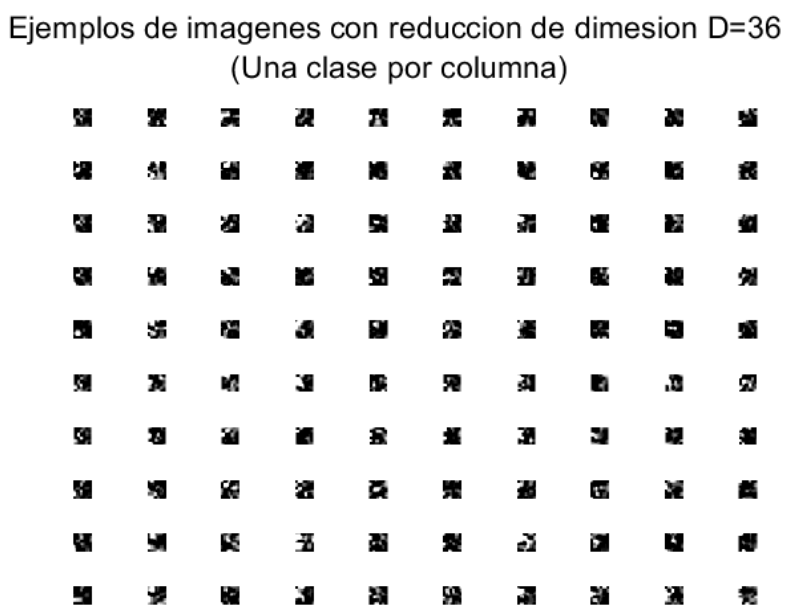
Modelo final y análisis de resultados

Se entrena un modelo con todos los datos de entrenamiento reducidos con U_{35} y se calcula la tasa de acierto tanto para los datos de entrenamiento como para los de test.

	Ta train	**Ta test**
PCA k=36 Covarianzas completas	0.9797	0.9630

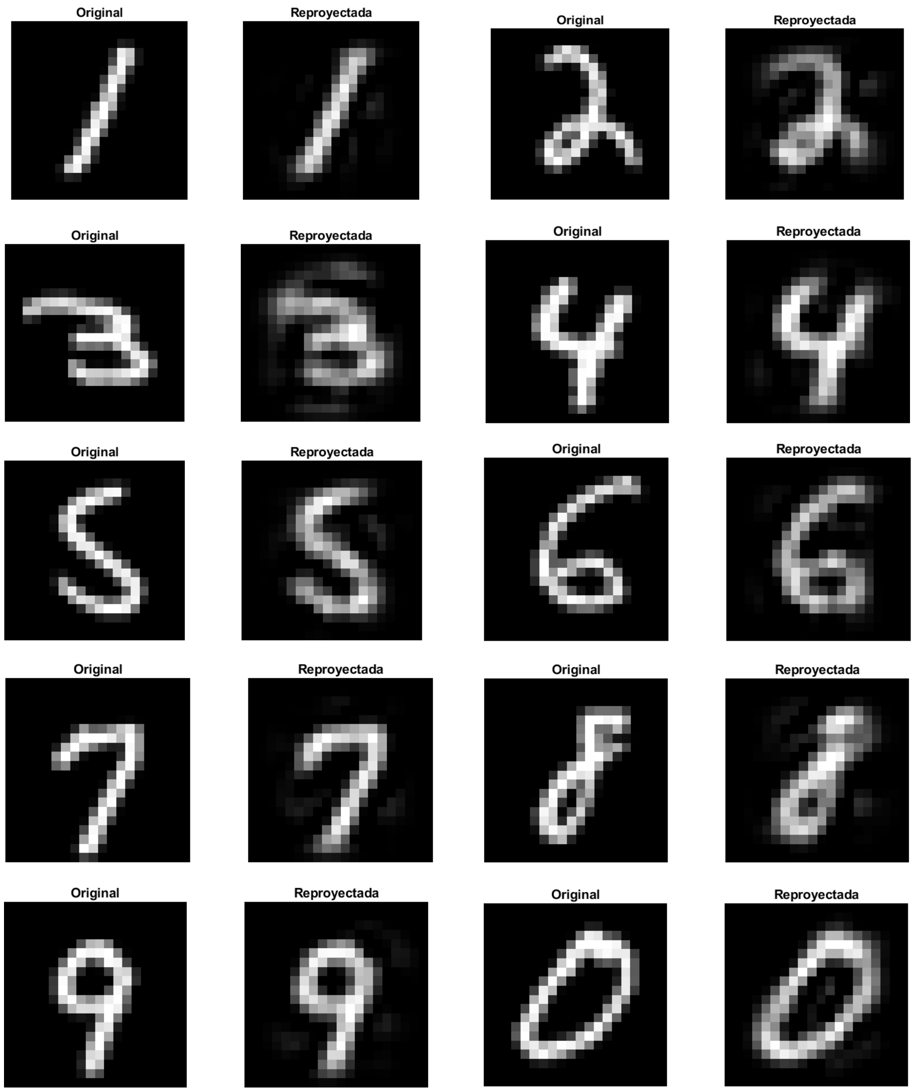
El modelo entrando con datos a las que se le ha aplicado reducción de la dimensión da lugar a muy buenos resultados, casi idénticos que al utilizar los datos originales.

Para ver qué información contienen los datos con su nueva dimensionalidad, se toman algunos ejemplos de cada clase, se reestructuran en forma de matrices cuadradas y se muestran como imagen.



Con los datos de pocas dimensiones, no se ven las figuras de los números, pero sí que se aprecia que datos de una misma clase tienden a tener valores altos en los mismos atributos.

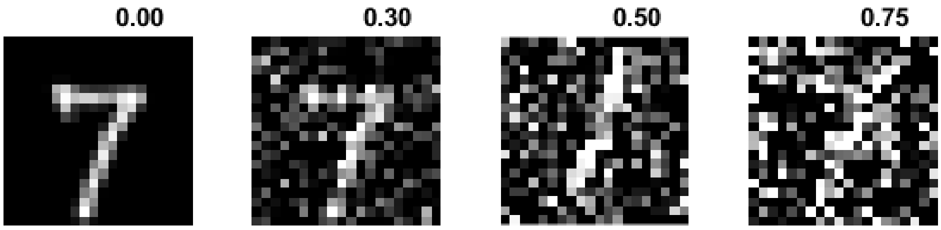
Por otro lado, también es interesante reproyectar los datos de dimensionalidad baja para comprobar la pérdida de información que supone reducir las dimensiones. Se realiza el siguiente cálculo: Xreproyectado = $Z_{36} * U^{T}_{36}$. Esto da lugar a una matriz de las mismas dimensiones que los datos originales, por lo que se puede mostrar en forma de imagen de igual forma. A continuación se muestran algunos ejemplos de datos junto a su equivalente reproyectado.



Se observa que todas los dibujos tienen casi exáctamente la misma forma que el dato original. Las imágenes reproyectadas pierden algo de intensidad y nitidez, pero claramente mantienen la suficiente información como para ser distinguidas. Demostrando de nuevo que una cantidad tan baja como 36 dimensiones es más que suficiente para entrenar el modelo.

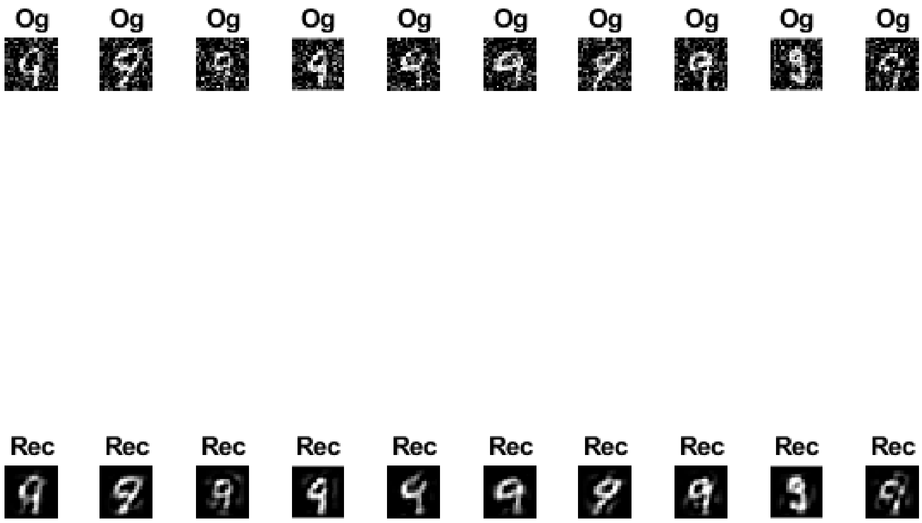
Filtrado de ruido en imágenes

En este apartado se quiere utilizar PCA para eliminar el ruido de imágenes de distintos dígitos para después clasificarlos. Este ruido se genera añadiendo valores en atributos aleatorios de los datos con una intensidad controlada por el parámetro std_noise. Cuanto mayor es este valor, mayor es el ruido de la imágen.



Eliminación de ruido

Para eliminar el ruido, se elige un valor de k, y se proyectan los datos estandarizados X sobre los primeros k vectores propios para obtener una versión reducida Z. Después se hace lo inverso, calculando una imagen X reproyectada utilizando los primeros k vectores propios sobre Z. La cantidad de ruido eliminado dependerá de la cantidad de componentes principales seleccionados. A continuación se muestra un ejemplo del resultado utilizando k=35.



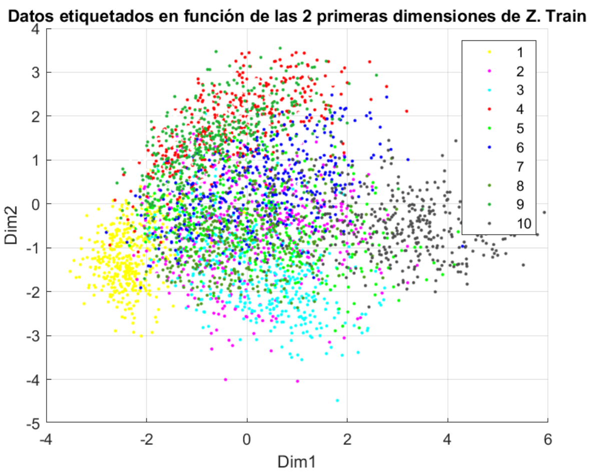
Clasificadores con imágenes ruidosas y en espacio PCA

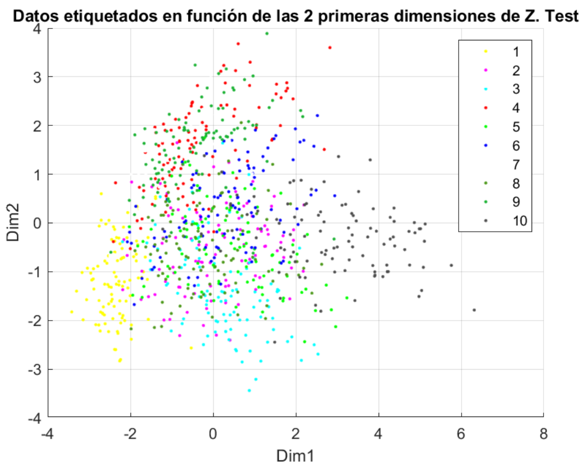
Se utilizan los modelos desarrollados en la práctica anterior y en el apartado 1 para entrenar utilizando los datos con ruido y en el espacio PCA.

En ambos casos se utiliza un clasificador Bayesiano multiclase de covarianzas completas y parámetro de regularización 10^{-2}. Para transformar los datos a la nueva dimensión se utilizan las primeras 35 componentes principales. Se entrenan los modelos con todos los datos de entrenamiento y se obtienen predicciones tanto para los datos de entrenamiento como para los de test, calculando después las tasas de acierto. Estos son los resultados.

	Ta train	**Ta test**
Datos con ruido	1	0.6330
Datos en espacio PCA	0.9665	0.9430

Para los datos con ruido, el modelo "memoriza" los patrones del ruido dando lugar a una tasa de acierto de 1 para los datos de entrenamiento, pero tan sólo acierta un 63% de las veces con los datos de test. Al reducir la dimensión de los datos, se elimina el ruido, permitiendo entrenar un modelo que obtiene buenos resultados tanto para los datos de entrenamiento como para los de test, cercanos a los que se obtendrían si se entrena el modelo con imágenes sin ruido.

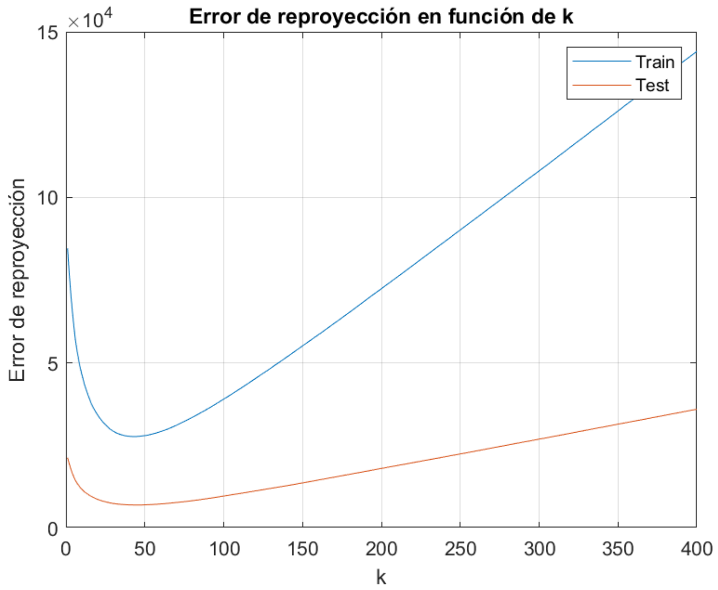




Se puede ver en las dos primeras dimensiones más significativas de Z que el resultado obtenido aplicando PCA a los datos con ruido es muy similar al que se obtenía al reducir los datos sin ruido.

Selección de número de componentes para el denoising

Para el problema de la eliminación del ruido, interesa minimizar el error de reconstrucción frente a las imágenes sin ruido. Se pueden probar distintos valores de k y calcular este error para cada uno.



El error de reproyección para los datos de test alcanza un mínimo absoluto en k= 46, punto en el cual el error es de 0.007.

Conclusiones

Se ha utilizado reducción de la dimensionalidad para mejorar el entrenamiento de modelos de aprendizaje supervisado, más específicamente de clasificación Bayesiana multiclase. Se puede utilizar una cantidad de dimensiones que las de los datos originales mediante selección de componentes principales sin empeorar los resultados, dando lugar a modelos incluso más precisos. Además, se ha demostrado que el PCA puede servir para eliminar el ruido de los datos e incluso entrenar modelos con entradas de datos de entrenamiento con ruido.

A continuación se muestra un resumen de las tasas de acierto de los modelos desarrollados.

	Ta train	**Ta test**
Sin PCA	0.9927	0.9640
PCA datos originales k=35	0.9797	0.9630
Datos con ruido k=35	1	0.6330
PCA Datos con ruido k=35	0.9665	0.9430