Análisis de componentes principales

Práctica 6

César Borja Moreno

03/05/2022

Aprendizaje Automático

Índice

1. Objetivo

Utilizar técnicas de reducción de dimensión basadas en el análisis de componentes principales. En concreto se usarán PCA para la reconstrucción de dígitos y su clasificación y SVD para comprimir una imagen.

2. Reducción de la dimensión en MNIST

En este apartado se pide reducir la dimensión de los datos del conjunto de datos MNIST con el algoritmo PCA, clasificar dichos datos con el algoritmo de entrenamiento de Bayes Completo (utilizado en la práctica 5) y comparar los resultados de la clasificación con los obtenidos en la práctica 5.

Para utilizar PCA, primero se han estandarizado los datos de entrada respecto de la media con:

```
normalize(X, 'center');
```

Después se calcula la matriz de co-varianzas Σ junto con sus vectores propios y los valores propios:

```
Sigma = 1/(nimages-1)*(Xn'*Xn);
[U,A] = eig(Sigma);
```

La matriz Δ es una matriz diagonal en cuya diagonal se encuentran los valores propios de Σ, mientras que la matriz U contiene los vectores propios de Σ.

A continuación, se ordenan los valores propios de Δ en orden decreciente para después ordenar los vectores propios según dichos valores propios. De esta manera, los vectores propios con mayor valor propio están los primeros. Esto se hace así para garantizar que los vectores que proporcionan más información se utilicen antes que otros que proporcionen menos al utilizar una porción reducida de la matriz U.

Tras esto, se elige el valor de k y se reduce la dimensión de los datos estandarizados a k dimensiones.

```
Z = Xn*Uk;
```

siendo X_n los datos estandarizados y U_k la matriz compuesta por las primeras k columnas de U.

2.1. ¿Con cuántas de las 400 componentes te puedes quedar sin alterar los resultados de la clasificación?

Se ha seleccionado el valor de k tal que se cumpla lo siguiente:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > 0{,}99 \tag{1}$$

siendo λ_i el valor propio de la componente i.

Esto da como resultado k = 153. Es decir, reduciendo la dimensión de los datos de 400 a 153 componentes, se mantiene el 99 % de la información de los datos originales.

2.2. ¿Qué tan buenos son los resultados obtenidos al mantener el 99 % de la variabilidad?

Para comparar los resultados obtenidos se ha calculado la matriz de confusión, así como los valores de precisión y recall para cada dígito para los datos reducidos con PCA, y se han comparado con los resultados de estas métricas obtenidos en la práctica 5.

En las Figuras 1 y 2 se muestran las matrices de confusión de ambos modelos. Las filas representan las clases predichas mientras que en las columnas se representan las clases reales (siendo la primera el 1, ascendiendo hasta 9, y la última el 0). Se puede observar como las matrices son prácticamente idénticas, con pequeñas variaciones asumibles (se mantiene una variabilidad del 99 %).

La comparación de los valores de precisión y recall para cada dígito se muestran en las Figuras 1 y 2.

Además, en la Figura 3 se puede ver la comparación del resultado de la función verConfusiones(), que muestra algunos de los errores producidos en la clasificación. Como se puede ver, los resultados son muy similares.

Al disminuir la dimensión de los datos, el rendimiento en cuanto a tiempo de la clasificación mejora considerablemente. Todo esto igualando los resultados de la clasificación con respecto a la misma pero sin aplicar la reducción de dimensión. En esencia, se pasan por alto aquellas componentes

que no proporcionan información útil para la clasificación (los píxeles de las esquinas siempre son 0, por tanto, no proporcionan información a tener en cuenta).

$$\begin{pmatrix} 99 & 3 & 1 & 1 & 0 & 0 & 2 & 2 & 1 & 0 \\ 1 & 94 & 2 & 1 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 94 & 0 & 1 & 0 & 0 & 3 & 2 & 0 \\ 0 & 0 & 0 & 98 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 95 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 99 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 96 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 3 & 0 & 0 & 91 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 94 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 100 \end{pmatrix}$$

Figura 1: Matriz de confusión de la clasificación de Bayes Completo sin utilizar PCA (práctica 5)

$$\begin{pmatrix} 99 & 1 & 0 & 1 & 0 & 0 & 2 & 2 & 1 & 0 \\ 1 & 97 & 3 & 1 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 94 & 0 & 1 & 0 & 0 & 3 & 2 & 0 \\ 0 & 0 & 0 & 96 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 93 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 99 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 96 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 4 & 0 & 0 & 91 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 94 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix}$$

Figura 2: Matriz de confusión de la clasificación de Bayes Completo utilizando PCA (práctica 6)

Dígito	Precisión	Recall
1	0.9083	0.9900
2	0.9400	0.9400
3	0.9307	0.9400
4	1.0000	0.9800
5	1.0000	0.9500
6	0.9802	0.9900
7	0.9796	0.9600
8	0.9286	0.9100
9	0.9592	0.9400
0	0.9804	1.0000

Cuadro 1: Precisión y recall para cada dígito de la clasificación de Bayes Completo sin utilizar PCA (práctica 5)

Dígito	Precisión	Recall
1	0.9340	0.9900
2	0.9327	0.9700
3	0.9307	0.9400
4	1.0000	0.9600
5	0.9688	0.9300
6	0.9802	0.9900
7	0.9897	0.9600
8	0.9192	0.9100
9	0.9495	0.9400
0	0.9901	1.0000

Cuadro 2: Precisión y recall para cada dígito de la clasificación de Bayes Completo utilizando PCA (práctica 6)

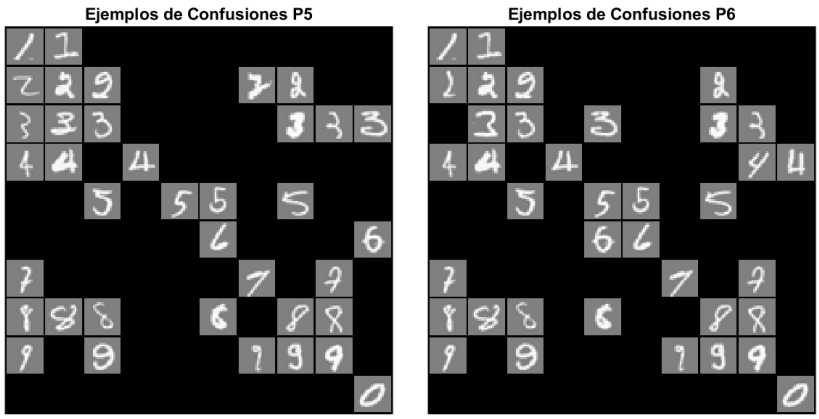

Figura 3: Comparación del resultado de la función verConfusiones() entre la clasificación de Bayes Completo sin utilizar PCA (práctica 5) y la clasificación de Bayes Completo utilizando PCA (practica 6)

.

3. Compresión de imágenes

En este segundo apartado se pide utilizar el algoritmo de Descomposición en Valores Singulares (SVD) para comprimir una imagen en en escala de grises.

Para ello, primero se descompone la imagen en las matrices U, S y V con:

$$[\mathrm{U}, \mathrm{S}, \mathrm{V}] \;=\; \mathbf{svd}(\mathrm{X})\,;$$

y después se elige el valor de k, que serán el número de componentes a sumar para generar la nueva imagen.

Se pide encontrar el valor de k que mantenga el 90 % de la variabilidad. Este valor se elige de la misma manera que en el apartado anterior:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > 0{,}90 \tag{2}$$

El valor resultante de k es 29.

Figura 4: Imagen original

Figura 5: Imagen reconstruida con k = 29

Una vez seleccionado k, se reconstruye la imagen con las k primeras componentes. En las Figuras 4 y 5 se muestra la comparación entre la imagen original y la imagen reconstruida.

En cuanto a las métricas de espacio ahorrado, la imagen original es de dimensiones 305x219, que se traducen en 66795 componentes. Tras hacer la descomposición y obtener las k primeras componentes, el total de componentes utilizadas tras usar SVD son:

$$(n + m + 1) * k = (305 + 219 + 1) * 29 = 15225 \text{ componentes} \tag{3}$$

Con esto, el porcentaje de espacio ahorrado con respecto a la imagen original es:

$$\frac{66795 - 15225}{66795} * 100 = 77{,}21\,\% \tag{4}$$

4. Conclusiones

En esta práctica se ha comprobado que las técnicas de reducción de dimensión suponen una importante mejora en el rendimiento, tanto en es-

pacio como en tiempo, de los programas, manteniendo unos resultados muy similares a los que se obtendrían sin usar estas técnicas.