Tecnologías para Inteligencia Artificial (247101009)
Tema 4. Métodos de reducción de dimensionalidad

Javier Vales Alonso
Máster Universitario en Ingeniería Telemática
2020

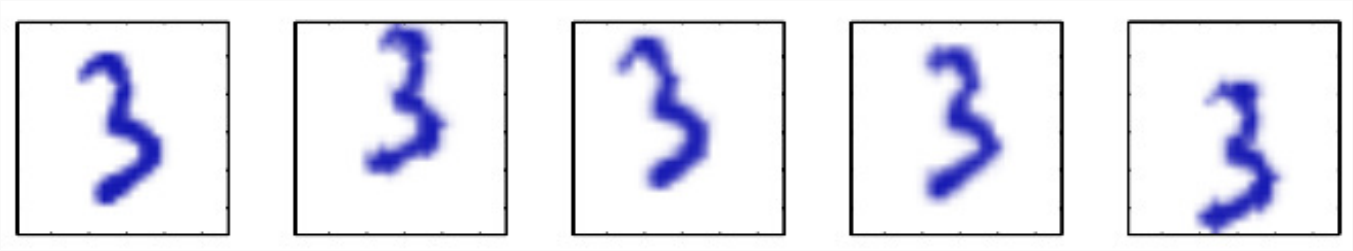
Universidad Politécnica de Cartagena

¿Cómo estudiar esta unidad?

1. Haga una primera lectura de las diapositivas de la unidad. Concéntrese en ver las ideas generales y hacer una primera revisión de las matemáticas.
2. Haga una revisión a fondo de las matemáticas con las diapositivas y resuelva en el notebook los ejercicios indicados. Intente comprender todos los desarrollos involucrados. En caso de dudas, lea las referencias sugeridas (ver referencias en Tema 0) o contacte con el profesor.
3. Finalmente, envíe el notebook a través de AV.

Introducción

¿Por qué reducción de dimensionalidad?

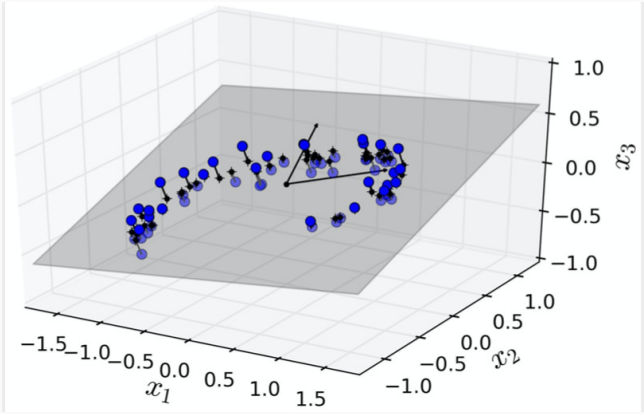


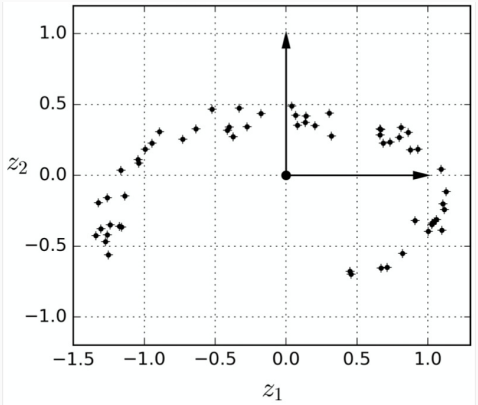
- En la figura de arriba, cada número es una imagen de tamaño 100×100, i.e., pertenece a un espacio 10.000-dimensional.
- Usar **una dimensión tan alta es ineficiente ya que todas las imágenes de cifras son muy similares**, simplemente escalados, rotaciones y traslaciones de la misma figura.
- Números escritos por diferentes personas tendrán grados adicionales de libertad, pero aún así mucho más bajo que considerando imágenes aleatorias.

¿Por qué reducción de dimensionalidad? (II)

- Lo mismo sucede con las caras en la figura izquierda (ver link).
- **El grado de libertad puede ser alto, pero aún es mucho más pequeño que la dimensión original de los datos.**

¿Por qué reducción de dimensionalidad? (III)





- Formalmente, los puntos de datos pertenecen (o están cerca) de una variedad de dimensión inferior al espacio original.
- Por ejemplo, podemos aproximar los puntos 3D que se muestran en la figura superior proyectándolos en el plano (figura inferior). De esta manera, obtenemos una representación 2D.

¿Por qué reducción de dimensionalidad? (IV)

- **Los espacios de alta dimensionalidad son contra-intuitivos**.
- Por ejemplo, en un hipercubo M-dimensional de lado unidad, se elige un punto aleatorio en el interior: a medida que M crece, es más y más probable que el punto está cerca de algún borde del hipercubo.
- Si piensa en personas, cuantas más características considere -e.g., altura, horas de trabajo, sueldo, ideologías, etc.- es más probable que todas se sitúen en algún extremo de alguna característica.

¿Por qué reducción de dimensionalidad? (V)

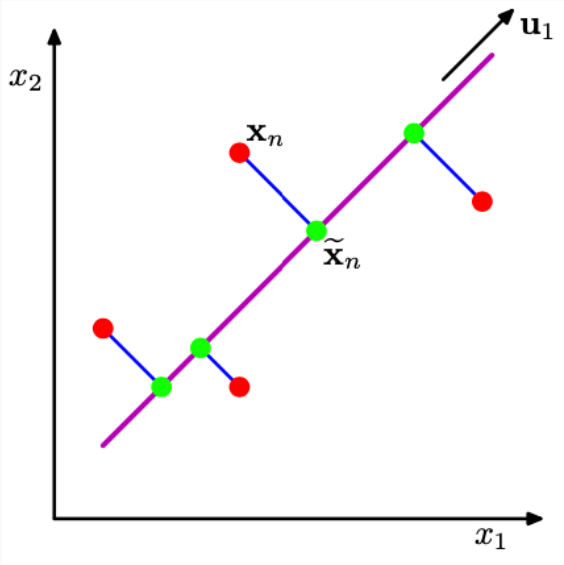
- Como otro ejemplo, la distancia entre puntos elegidos al azar dentro de un cuadrado de lado unidad es aproximadamente 0.52. Para un cubo, aproximadamente 0.66. Por un **hipercubo 1-millón-dimensional de lado unidad, es 408.25 (¡mucho más largo que la longitud del borde de 1 unidad!)**
- En pocas palabras, **los conjuntos de datos de espacios de alta dimensión son extremadamente dispersos. Reducir dimensiones es crítico para aumentar la fiabilidad de las predicciones** al evitar grandes extrapolaciones sobre los datos.

Análisis de componentes principales

Análisis de componentes principales

- PCA (también conocida como transformación de Karhunen-Loève) es una técnica "natural" utilizada para reducción de dimensionalidad.
- Diferentes aproximaciones dan lugar al mismo algoritmo.
- Fue desarrollado durante la primera mitad del siglo XX de forma independiente por Hotelling y Pearson.
- Sus aplicaciones incluyen compresión de datos (con pérdidas), extracción de características, visualización de datos, y su formulación probabilística permite la implementación de modelos generativos.

Formulación de máxima varianza



- Dado un conjunto de N observaciones $\{\boldsymbol{x}_n\}$ de dimensionalidad D nuestro objetivo es maximizar la varianza de los datos proyectados.
- Las proyecciones son los puntos M-dimensionales $\{\tilde{\boldsymbol{x}}_n\}$ siendo $M < D$.
- Maximizar la varianza permite preservar tanta información como sea posible.

Formulación de máxima varianza (II)

Suponga $M{=}1$ y sea $\boldsymbol{u}_1$ un vector unitario. El promedio de los datos proyectados es $\boldsymbol{u}_1^T\overline{\boldsymbol{x}}$, siendo $\overline{\boldsymbol{x}}$ el promedio del conjunto de datos:

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \tag{1}$$

y la varianza muestral:

$$\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{u}_1^T\boldsymbol{x}_n - \boldsymbol{u}_1^T\boldsymbol{x}_n)^2 = \boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1 \tag{2}$$

siendo $\boldsymbol{S}$ la matriz de covarianza de los datos:

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \overline{\boldsymbol{x}})(\boldsymbol{x}_n - \overline{\boldsymbol{x}})^T \tag{3}$$

Formulación de máxima varianza (III)

Por lo tanto, tenemos que maximizar $\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1$ con respecto a $\boldsymbol{u}_1$ y sujeto a $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$ (vector unitario). Introduciendo la restricción como un multiplicador de Lagrange el problema equivale a maximizar:

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1) \tag{4}$$

Al igualar la derivada igual a cero, obtenemos:

$$\boldsymbol{S} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{5}$$

Es decir, $\boldsymbol{u}_1$ es un autovector de $\boldsymbol{S}$ y la varianza es el autovalor $\lambda_1 = \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1$. Maximizarlo requiere seleccionar el mayor autovalor. El autovector asociado se llama **primer componente principal**.

Formulación de máxima varianza (IV)

Para $M>1$, se cogen los M mayores autovalores de $\boldsymbol{S}$ y sus autovectores asociados $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M$ (nótese que $\boldsymbol{u}_i^T \boldsymbol{u}_j=0$ para cada $i \neq j$).

El punto proyectado $\tilde{\boldsymbol{x}}_n$ viene dado (después veremos por qué) por:

$$\tilde{\boldsymbol{x}}_n = \overline{\boldsymbol{x}} + \sum_{i=1}^{M} [\boldsymbol{u}_i^T (\boldsymbol{x}_n - \overline{\boldsymbol{x}})] \boldsymbol{u}_i \tag{6}$$

Esta proyección tiene un error (pequeño si M es suficientemente grande) en comparación con el punto original.

Alternativamente, podríamos haber planteado **minimizar la suma de los errores de proyección**. Veremos ahora que **el resultado alcanzado no cambia**.

Formulación de mínimo error

Sea $\{\boldsymbol{u}_i\}$, i=1, . . . , D, una base de vectores ortonormales.
Cualquier punto $\boldsymbol{x}_n$ puede ser aproximado por:

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i\boldsymbol{u}_i, \tag{7}$$

donde **los coeficientes z_{ni} son particulares para cada punto x_n y los b_i son iguales para todos los datos**.

El error de aproximación conjunto es:

$$J = \sum_{n=1}^{N}(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^T(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n) \tag{8}$$

J depende de los coeficientes $\boldsymbol{z}_{ni}$, las constantes b_i, y la base $\{\boldsymbol{u}_i\}$.

Formulación de mínimo error (II)

Igualando las derivadas con respecto a $\boldsymbol{z}_{nj}$ a cero:

$$\frac{\partial J}{\partial z_{nj}} = 0 \implies -(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^T \frac{\partial \tilde{\boldsymbol{x}}_n}{\partial z_{nj}} = 0 \implies z_{nj} = \boldsymbol{x}_n^T \boldsymbol{u}_j \tag{9}$$

Y respecto a b_j a cero:

$$\frac{\partial J}{\partial b_j} = 0 \implies -\sum_{n=1}^{N} (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^T \frac{\partial \tilde{\boldsymbol{x}}_n}{\partial b_j} = 0 \implies$$

$$\sum_{n=1}^{N} \boldsymbol{x}_n^T \boldsymbol{u}_j - \sum_{n=1}^{N} \sum_{i=1}^{M} z_{ni} \boldsymbol{u}_i^T \boldsymbol{u}_j - \sum_{n=1}^{N} \sum_{i=M+1}^{D} b_j \boldsymbol{u}_i^T \boldsymbol{u}_j \implies \tag{10}$$

$$Nb_j = \sum_{n=1}^{N} \boldsymbol{x}_n^T \boldsymbol{u}_j \implies b_j = \overline{\boldsymbol{x}}^T \boldsymbol{u}_j$$

Formulación de mínimo error (III)

Entonces, el error de proyección viene dado por:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} [(\boldsymbol{x}_n - \overline{\boldsymbol{x}})^T \boldsymbol{u}_i]\boldsymbol{u}_i \tag{11}$$

Y el error acumulativo es:

$$J = \sum_{n=1}^{N} \sum_{i=M+1}^{D} (\boldsymbol{x}_n^T \boldsymbol{u}_i - \overline{\boldsymbol{x}}^T \boldsymbol{u}_i)^2 = \sum_{i=M+1}^{D} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i \tag{12}$$

Formulación de mínimo error (IV)

J todavía tiene que ser minimizado con respecto a la base $\{\boldsymbol{u}_i\}$. Si D=2 y M=1, la minimización de J sujeta a tener una base unitaria puede expresarse nuevamente con un multiplicador de Lagrange:

$$\frac{\partial \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2 + \lambda_2(1 - \boldsymbol{u}_2^T \boldsymbol{u}_2)}{\partial \boldsymbol{u}_2} = 0 \implies \boldsymbol{S}\boldsymbol{u}_2 = \lambda_2 \boldsymbol{u}_2 \qquad (13)$$

Por lo tanto, J=λ_2 es minimizado seleccionando como λ_2 el menor autovalor de S y siendo $\boldsymbol{u}_2$ su autovector asociado.

Para valores arbitrarios M, D, se escogen los $D - M$ menores autovalores y resulta J=$\sum_{i=M+1}^{D} \lambda_i$. Es decir, **el error mínimo está determinado por el conjunto complementario a los autovalores que nos maximizaban la varianza**.

Formulación de mínimo error (V)

Habíamos visto que la proyección del punto $\boldsymbol{x}_n$ es:

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i\boldsymbol{u}_i,$$

Es decir,

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M}(\boldsymbol{u}_i^T\boldsymbol{x}_n)\boldsymbol{u}_i + \sum_{i=M+1}^{D}(\boldsymbol{u}_i^T\overline{\boldsymbol{x}})\boldsymbol{u}_i,$$

Puesto que $\overline{\boldsymbol{x}} = \sum_{i=1}^{M}(\boldsymbol{u}_i^T\overline{\boldsymbol{x}})\boldsymbol{u}_i + \sum_{i=M+1}^{D}(\boldsymbol{u}_i^T\overline{\boldsymbol{x}})\boldsymbol{u}_i$, llegamos a:

$$\tilde{\boldsymbol{x}}_n = \overline{\boldsymbol{x}} + \sum_{i=1}^{M}[\boldsymbol{u}_i^T(\boldsymbol{x}_n - \overline{\boldsymbol{x}})]\boldsymbol{u}_i$$

Autoencoders

Autoencoders

Un autoencoder es una estructura paramétrica (típicamente una ANN) que transforma las entradas de datos $\boldsymbol{x}_n$ en una forma codificada $c(\boldsymbol{x}, \boldsymbol{w_c})$, y luego la decodifica de vuelta al espacio de entrada original $\widetilde{\boldsymbol{x}}_n = d(c(\boldsymbol{x}_n, \boldsymbol{w_c}), \boldsymbol{w_d})$.

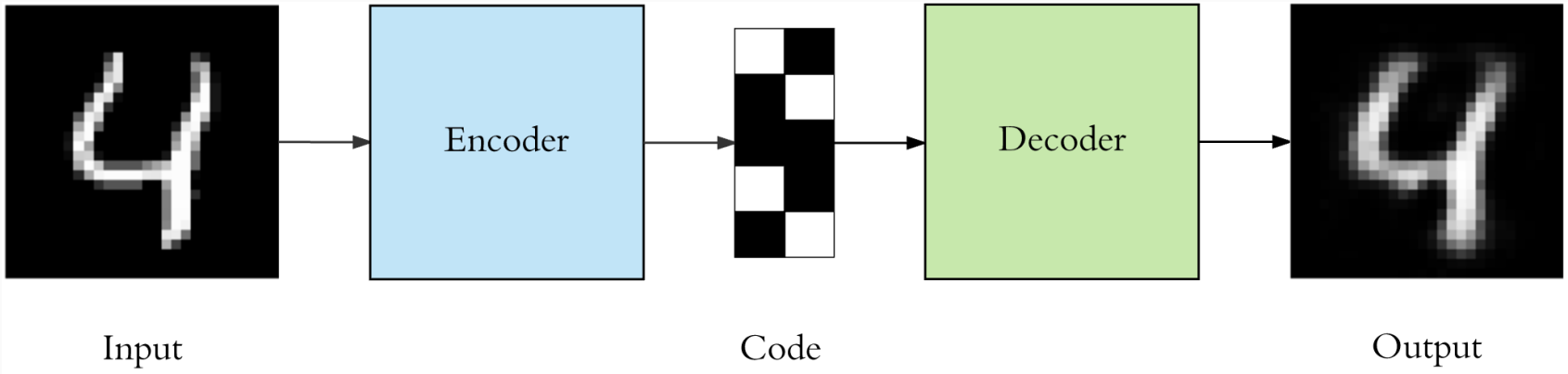
Dado que la codificación es de (mucho) menor dimensionalidad que el espacio de entrada, aparece algún error de reconstrucción $J_n = \|\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n\|^2$.

Los parámetros del autoencoder $\boldsymbol{w_c}$ y $\boldsymbol{w_d}$ se encuentran minimizando el error de reconstrucción agregado (J):

$$\min_{\boldsymbol{w_c}, \boldsymbol{w_d}} \frac{1}{2} \sum_{n=1}^{N} J_n \tag{14}$$

Autoencoders (II)

Estructura del autoencoder:

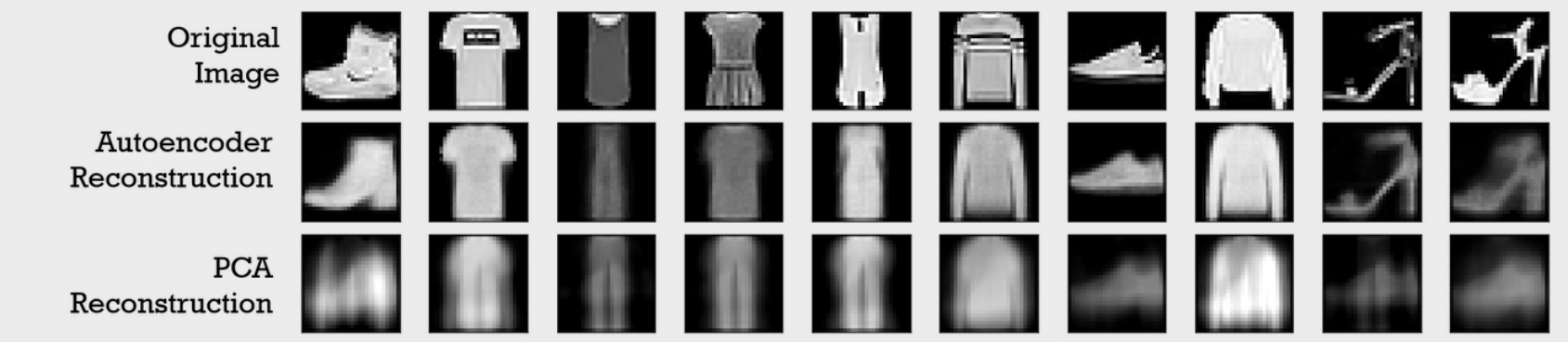


Otros usos del autoencoder incluyen modelos de eliminación de ruido o modelos generativos (ver link, y link).

Aplicaciones

Extracción de características

Los sistemas de reducción de dimensionalidad pueden usarse para obtener una **representación de los datos de baja dimensión (codificación) que se puede usar como un espacio de características, para compresión de datos (con pérdidas) u otras aplicaciones**. Como ejemplo, la siguiente figura muestra cómo se reconstruyen los datos a partir de PCA y de un autoencoder con la misma dimensionalidad del código.



Visualización de datos

M se selecciona de modo que $(\sum_{i=1}^{M} \lambda_i)/(\sum_{i=1}^{D} \lambda_i)$ sea mayor que un ratio elevato, e.g., 0.9 (i.e., la codificación conserva el 90 % de la varianza del conjunto de datos).

También es posible fijar $M=$ 2 o 3, para visualizar los datos. La siguiente figura muestra un ejemplo relacionado con proteínas.