ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN PROFESIONAL

Fundamentos Machine & Deep Learning

Diplomado Big Data y Analítica de Datos 2022-2023

Profesor:
Rodrigo Sandoval U.

Clases Restantes

HOY	• Reducción dimensional • Visión Computacional con DL
Clase 6	• Introducción NLP • Clasificación y extracción de información de texto
Clase 7	• Clasificación No-Supervisada
Clase 8	• Secuencia y Redes Neuronales • NLU y modelos avanzados • Proyectos Machine Learning y estado del arte

En el ejemplo de datos de Marketing bancario

```
      Edad                Ocupación       EstadoCivil
 Min.   :17.00   admin.      :10422   divorced: 4612
 1st Qu.:32.00   blue-collar: 9254   married :24928
 Median :38.00   technician : 6743   single  :11568
 Mean   :40.02   services   : 3969   NA's    :   80
 3rd Qu.:47.00   management : 2924
 Max.   :98.00   (Other)    : 7546
                 NA's       :  330
              Educación     Default      Hipotecario  Consumo
 university.degree  :12168   no  :32588   no  :18622   no  :33950
 high.school        : 9515   yes :    3   yes :21576   yes : 6248
 basic.9y           : 6045   NA's: 8597   NA's:  990   NA's:  990
 professional.course: 5243
 basic.4y           : 4176
 (Other)            : 2310
 NA's               : 1731
```

¿Cuáles atributos (*features*) numéricos aparentan ser poco relevantes?

En el ejemplo de datos de Marketing bancario

```
     Contacto           Mes          Día          Duración       NumContactos
cellular :26144   may    :13769   fri:7827   Min.   :   0.0   Min.   : 1.000
telephone:15044   jul    : 7174   mon:8514   1st Qu.: 102.0   1st Qu.: 1.000
                  aug    : 6178   thu:8623   Median : 180.0   Median : 2.000
                  jun    : 5318   tue:8090   Mean   : 258.3   Mean   : 2.568
                  nov    : 4101   wed:8134   3rd Qu.: 319.0   3rd Qu.: 3.000
                  apr    : 2632              Max.   :4918.0   Max.   :56.000
                  (Other): 2016
  DíasAtrás          Previo        ResultadoPrevio    EmpTasaVar
Min.   :  0.0   Min.   :0.000   failure    : 4252   Min.   :-3.40000
1st Qu.:999.0   1st Qu.:0.000   nonexistent:35563   1st Qu.:-1.80000
Median :999.0   Median :0.000   success    : 1373   Median : 1.10000
Mean   :962.5   Mean   :0.173                       Mean   : 0.08189
3rd Qu.:999.0   3rd Qu.:0.000                       3rd Qu.: 1.40000
Max.   :999.0   Max.   :7.000                       Max.   : 1.40000

     IPC             ICC          Euribor3m      NumEmpleados    OK
Min.   :92.20   Min.   :-50.8   Min.   :0.634   Min.   :4964   no :36548
1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099   yes: 4640
Median :93.75   Median :-41.8   Median :4.857   Median :5191
Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5167
3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228
Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228
```

¿Cuáles atributos (*features*) numéricos aparentan ser poco relevantes?

Un primer análisis

¿Podría haber una relación entre estos atributos?

```
                Educación
university.degree  :12168
high.school        : 9515
basic.9y           : 6045
professional.course: 5243
basic.4y           : 4176
(Other)            : 2310
NA's               : 1731
```

```
          Ocupación
admin.      :10422
blue-collar: 9254
technician  : 6743
services    : 3969
management  : 2924
(Other)     : 7546
NA's        :  330
```

```
      Edad
Min.   :17.00
1st Qu.:32.00
Median :38.00
Mean   :40.02
3rd Qu.:47.00
Max.   :98.00
```

```
   EstadoCivil
divorced: 4612
married :24928
single  :11568
NA's    :   80
```

Para discutir

- ¿Por qué podría interesar determinar cuáles atributos son menos relevantes? (¿Qué se gana con descartarlos?)
- ¿Qué se podría ganar si se reduce la cantidad de atributos, combinando algunos que se representan por uno nuevo?
  - Por ej: "Nivel Académico-Profesional", que combina "Educación", con "Ocupación" y ese se relaciona con la variable de interés "OK".
- Si hay menos atributos ¿es menor el esfuerzo de entrenamiento? ¿mejora el rendimiento del modelo?
- Más importante: en lugar de enfocarnos en un problema con 20 variables o atributos, ¿tendrá mayor valor esta reducción de atributos en un problema de alta dimensionalidad, como análisis de imágenes?

6. Reducción Dimensional

Fuentes: Ismini Lourentzou & Boris Ginsburg

Reducción Dimensional

Cualquiera que ha trabajado con datasets con muchos atributos conoce lo difícil que es explorar sus relaciones. No poner atención puede resultar fácilmente en modelos de pobre generalización, sobreajustes o ignorar algunas restricciones y suposiciones.

Aquí es donde la reducción dimensional entra en juego.

En Machine Learning, la reducción dimensional es el proceso de reducción de variables o atributos del problema, bajo la idea de quedarse con las variables más relevantes.

Al reducir la dimensión del espacio de atributos, se logra analizar con más facilidad las pocas variables que quedan, minimizando la posibilidad de sobre ajuste.

Motivación

- Simplificación de datos de dimensionalidad muy compleja
- Agrupar o totalizar datos de un vector de alta dimensionalidad en una menor dimensionalidad.
  - Datos puntos de datos en *d* dimensiones
  - Convertir esos mismos datos en $r < d$ dimensiones
  - Con ninguna o mínima pérdida de información.
- Eso tiene el potencial de:
  - Reducir el esfuerzo de entrenamiento de un modelo de clasificación. (Compresión de Datos)
  - Mejorar el desempeño, al reconocer atributos relevantes, en lugar de correr el riesgo de overfitting con algunos atributos poco significativos, pero que por su distribución de valores, terminan siendo más influyentes de lo necesario.

Formas de Reducción Dimensional

Eliminación de Características

- Se reduce el espacio de características, simplemente eliminando algunas de ellas.
- El desafío está evitar eliminar inadvertidamente las más determinantes.

Selección de Características

- Se aplican test estadísticos para ordenar los atributos o características en relación a su importancia, para seleccionar un subconjunto más relevante.
- El desafío está en la pérdida de información y estabilidad, según el tipo de test seleccionado, ya que diferentes pueden resultar en puntajes diferentes para los atributos.

Extracción de Características

- Se combinan algunos atributos para crear unos nuevos, que los representen (reduciendo así la dimensión).
- Estas técnicas pueden ser divididas entre técnicas de reducción dimensional lineal y no-lineal.

Técnicas y Conceptos en Reducción Dimensional

PCA (Principal Component Analysis)

t-SNE (T-Distributed Stochastic Embedding)

LDA (Linear Discriminant Analysis)

ICA (Independent Component Analysis)

Multidimensional Scaling

Autoencoders

Reducción Dimensional

6.1. PRINCIPAL COMPONENT ANALYSIS - PCA

Fuente: Fereshteh Sadeghi – UW CSE

Principal Component Analysis

Goal: Find r-dim projection that best preserves variance

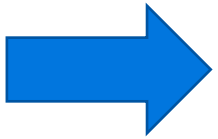
1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one, and the rows of A are the eigenvectors

Covarianza

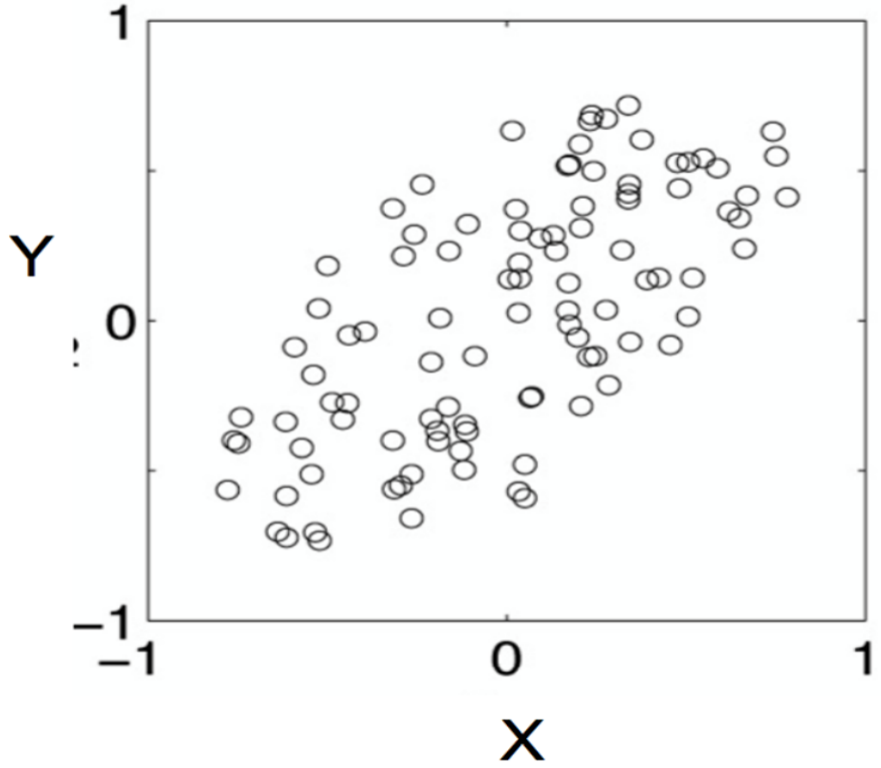
- Varianza y Covarianza
  - La medida de la dispersion de un conjunto de puntos alrededor de su centro de masa (promedio)
- Varianza:
  - Medida de la desviación desde el promedio para puntos 1D
- Covarianza:
  - Medida de cuanto varía cada una de las dimensiones desde el promedio en relación a las otras



- **Covarianza se mide entre 2 dimensiones**
- **Covarianza ve si existe una relación entre 2 dimensiones**
- **Covarianza entre una dimension es la Varianza**

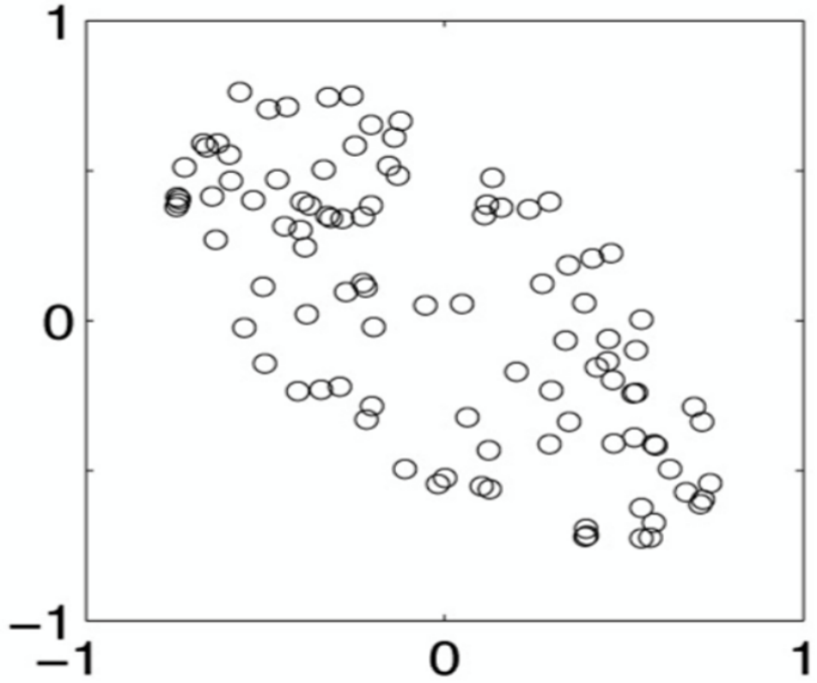
Covarianza Positiva

Ambas dimensiones aumentan o disminuyen juntas



Covarianza Negativa

Mientras una dimensión aumenta, la otra disminuye



Covarianza

Se usa para encontrar relaciones entre dimensiones de conjuntos de datos de alta dimensionalidad

$$q_{jk} = \frac{1}{N} \sum_{i=1}^{N} \left(X_{ij} - E(X_j)\right)\left(X_{ik} - E(X_k)\right)$$

El promedio de la muestra

Vectores y Valores Propios

$$Ax = \lambda x$$

A: Matriz Cuadrada
X: Vector propio o vector característico
λ: Valor propio o valor característico

- ***El vector cero NO puede ser un vector propio***
- ***El valor cero o nulo SI puede ser un valor propio***

Ejemplo

$$Show\ x = \begin{bmatrix} 2 \\ 1 \end{bmatrix} is\ an\ eigenvector\ for\ A = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}$$

$$Solution: Ax = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$But\ \ for\ \ \lambda = 0,\ \ \lambda x = 0\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$Thus, x\ is\ an\ eigenvector\ of\ A, and\ \lambda = 0\ is\ an\ eigenvalue.$$

Vector propio y valor propio

$$Ax = \lambda x \longrightarrow Ax - \lambda x = 0$$
$$(A - \lambda I)x = 0$$

Si se define una nueva matriz B: → $B = A - \lambda I$

$$Bx = 0$$

Si B tiene inversa: → $x = B^{-1}0 = 0$ ✖

¡¡Pero un vector propio no puede ser nulo!!

X será un vector propio de A, si y solo si, B no tiene inversa o, equivalentemente, su determinante es 0 det(B)=0:

$$\det(A - \lambda I) = 0$$

Vectores y Valores Propios

Ejemplo1: Encontrar los valores propios de $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$

$$|\lambda I - A| = \begin{vmatrix} \lambda - 2 & 12 \\ -1 & \lambda + 5 \end{vmatrix} = (\lambda - 2)(\lambda + 5) + 12$$

$$= \lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2)$$

Dos valores propios: –1, – 2

NOTA: las raíces de la ecuación característica se pueden repetir.
Es decir, $\lambda_1 = \lambda_2 = \ldots = \lambda_k$. Si eso sucede, se dice que el valor propio es de multiplicidad k.

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Ejemplo 2: Encontrar los valores propios de

$$|\lambda I - A| = \begin{vmatrix} \lambda - 2 & -1 & 0 \\ 0 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 2 \end{vmatrix} = (\lambda - 2)^3 = 0$$

$\lambda = 2$ es in valor propio de multiplicidad 3.

Principal Component Analysis

Entrada: $\mathbf{x} \in \mathbb{R}^D$: $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

Vectores básicos $\mathbf{u}_1, \ldots, \mathbf{u}_K$

Resumir un vector X de *D* dimensiones con un vector característico *h(x) de K* dimensiones

$$h(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \cdots \\ \mathbf{u}_K \cdot \mathbf{x} \end{bmatrix}$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$$

Vectores base son ortonormales

$$||\mathbf{u}_j|| = 1$$

Nueva representación de datos *h(x)*

$$z_j = \mathbf{u}_j \cdot \mathbf{x}$$

$$h(\mathbf{x}) = [z_1, \ldots, z_K]^T$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$$

Nueva representación de datos *h(x)*

$$h(\mathbf{x}) = \mathbf{U}^T \mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{U}^T (\mathbf{x} - \mu_0)$$

Promedio empírico de los datos $\mu_0 = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$

El espacio de todas las imágenes de caras

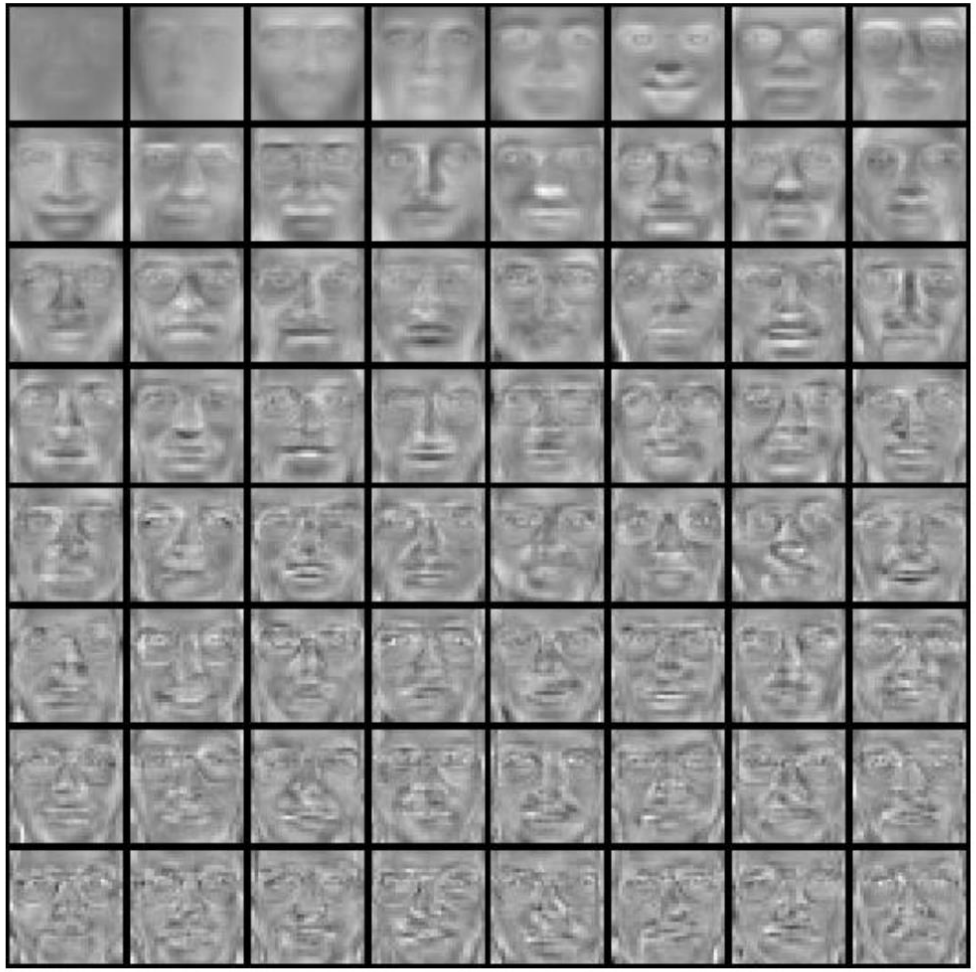
- Cuando se las mira como vectores de valores de pixeles, las caras son de muy alta dimensión:
  - 100x100 pixeles = 10.000 dimensiones
  - Lento de procesar y ocupa mucho almacenamiento
- Pero muy pocos pixeles de 10.000 dimensiones son imágenes válidas de caras.
- Se quiere modelar el subespacio de las imágenes de caras

Fuente: Derek Hoiem

Ejemplo de "Eigenfaces"

Top eigenvectors: $u_1,\ldots u_k$

Mean: μ



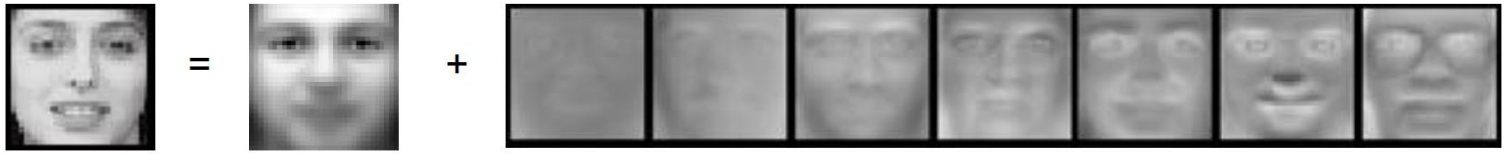
slide by Derek Hoiem

Representación y Reconstrucción

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:



$\hat{x} = \mu + w_1u_1+w_2u_2+w_3u_3+w_4u_4+ \ldots$

slide by Derek Hoiem

Reconstrucción

ORL: La base de datos de caras
https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

P = 4

P = 200

P = 400

After computing eigenfaces using 400 face images from ORL face database

slide by Derek Hoiem

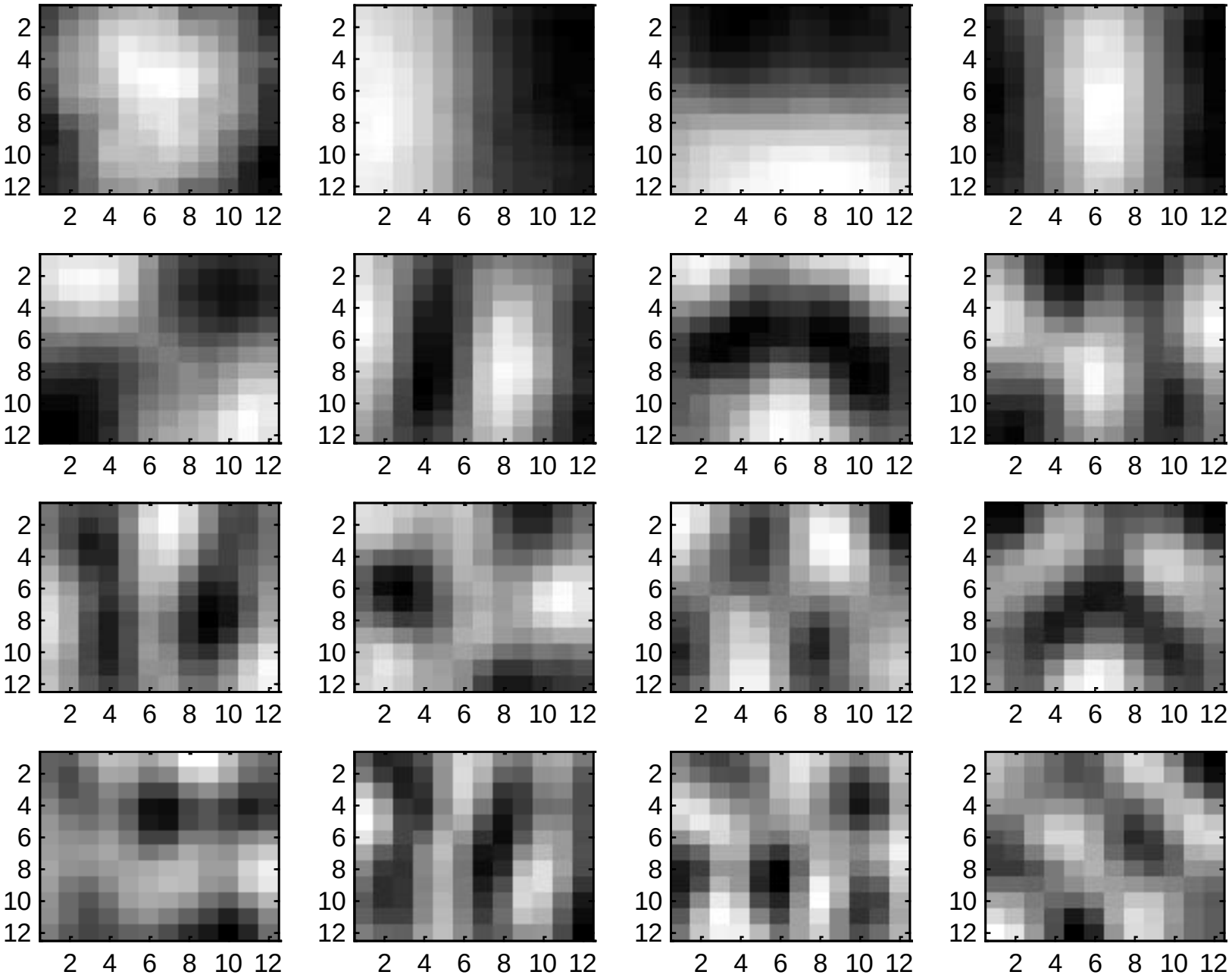
Aplicación: Compresión de Imágenes

- Imagen de 372x492
- Dividirla en "parches":
  - Cada parche es una instancia que contiene 12x12 pixeles en una grilla o cuadriculado
- Se puede definer como un vector de 144 dimensiones

Compresión PCA
144D → 60D

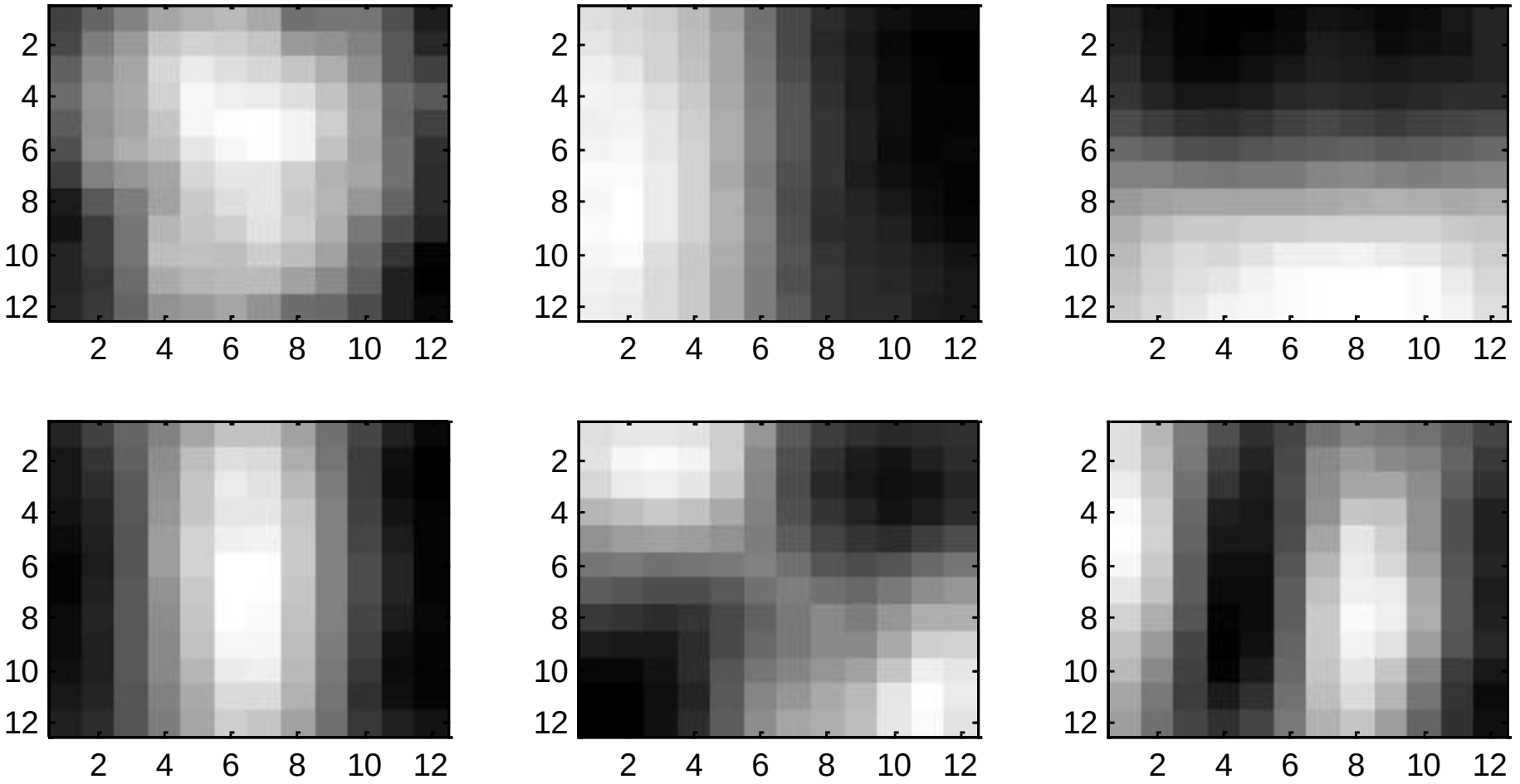
Compresión PCA
144D → 16D

16 vectores propios más relevantes



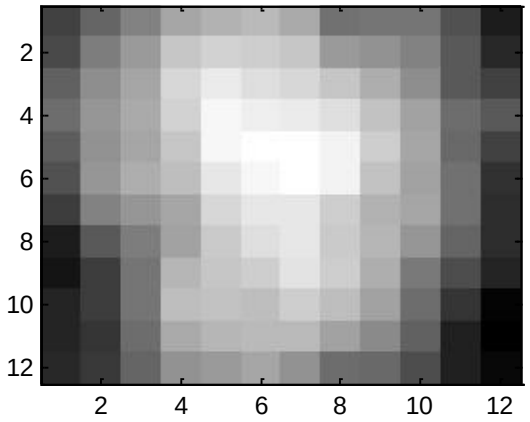
Compresión PCA
144D → 6D

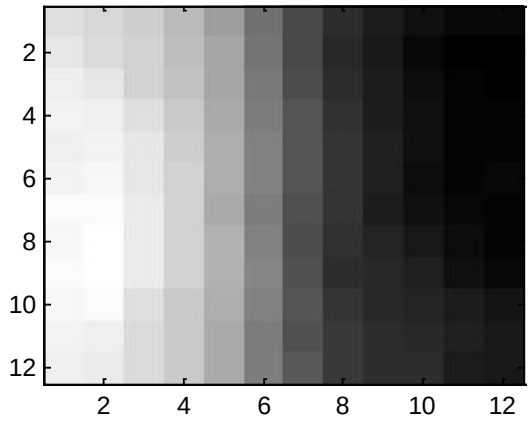
6 vectores propios más importantes

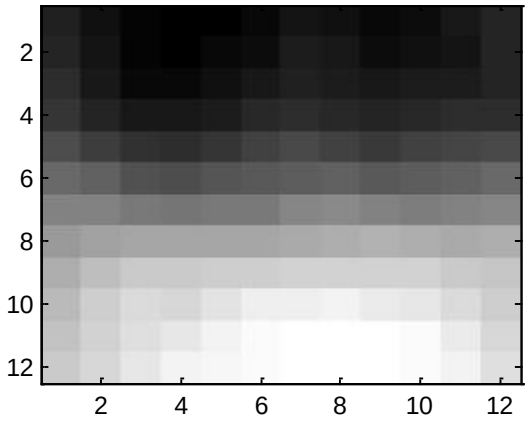


Compresión PCA
144D → 3D

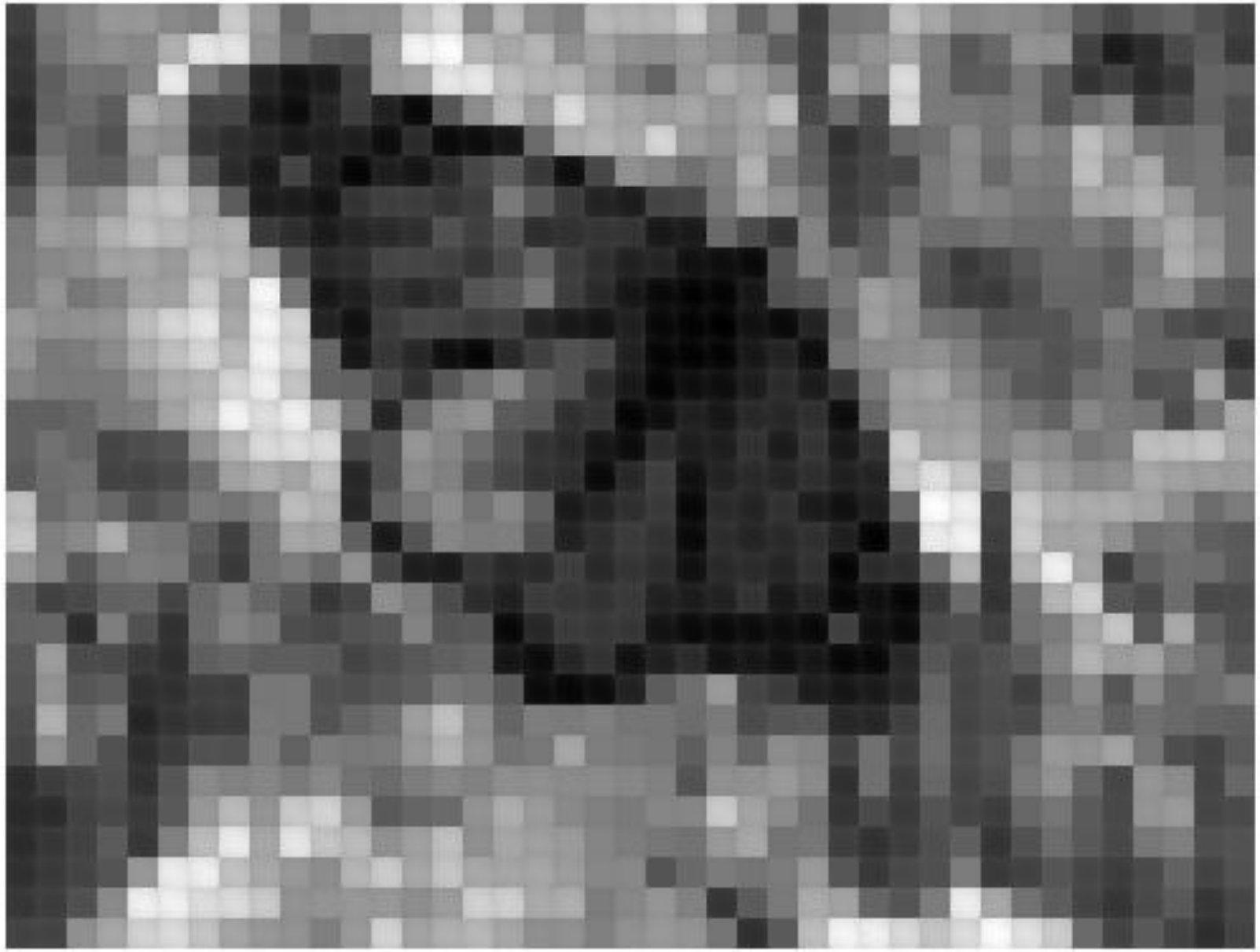
3 vectores propios más importantes





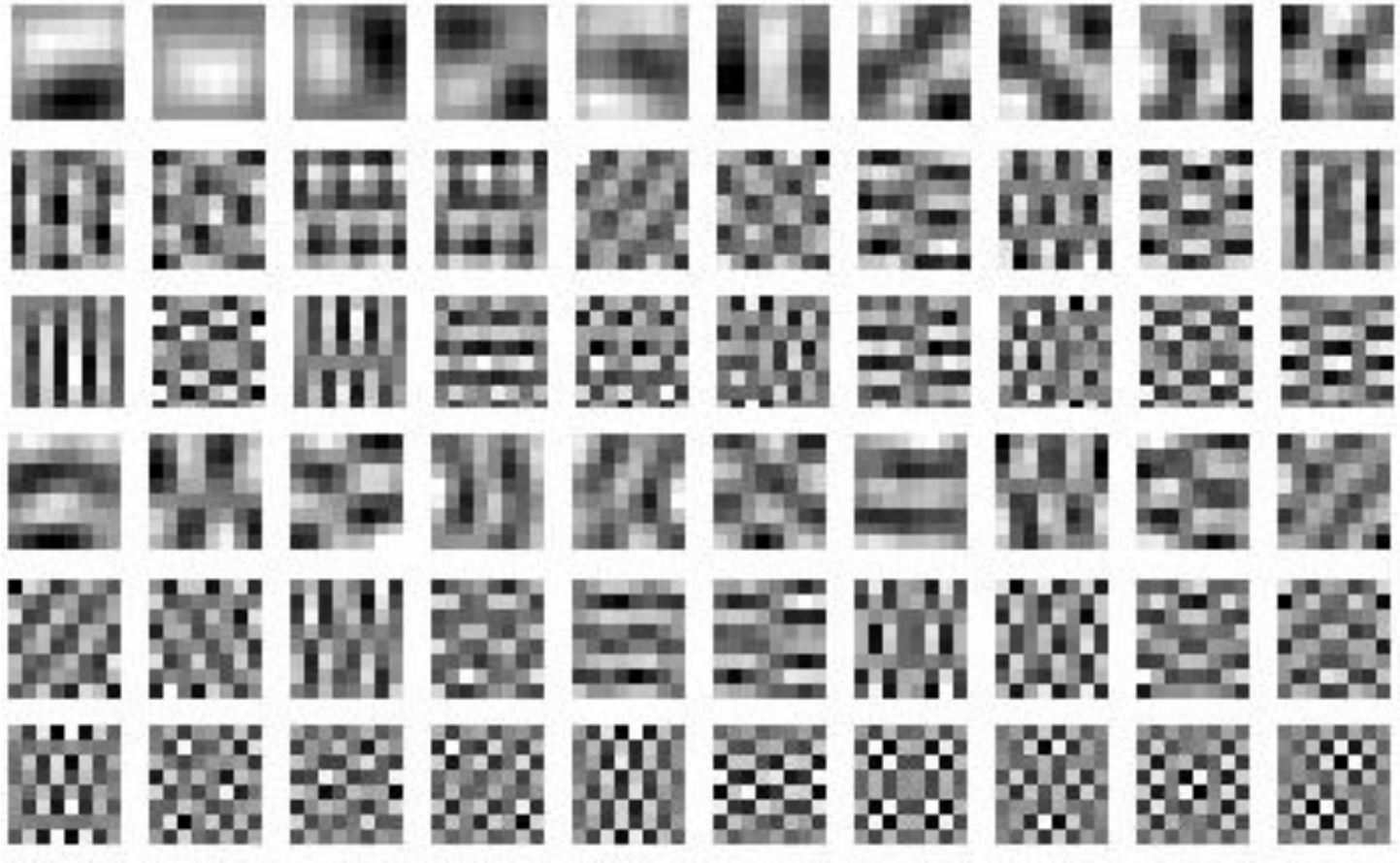


Compresión PCA 144D → 1D

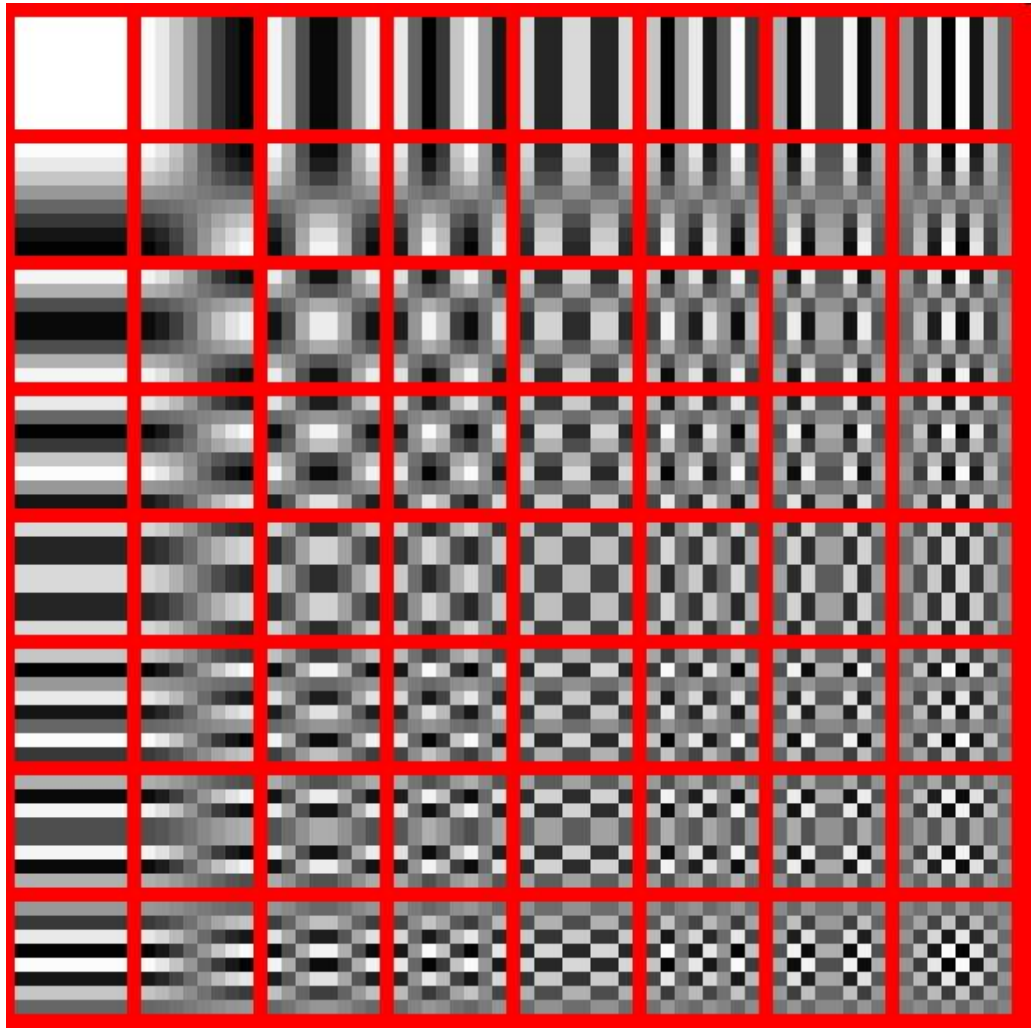


60 vectores propios más importantes

El coseno discreto es la base del formato de compresión de imágenes JPEG



2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform

Preguntas:

- ¿PCA se enfoca eliminación, selección, o extracción de características?

  EXTRACCIÓN,
  particularmente extracción lineal (mapeo lineal o directo)

- ¿En qué contexto de datos podría dar un gran aporte en reducción dimensional?

  DATOS MASIVOS/NO-ESTRUCTURADOS

6.2. OTROS MÉTODOS

Otras Técnicas (además de PCA)

t-SNE (T-Distributed Stochastic Embedding)	• Reducción dimensional con la facilidad de visualizar grandes datasets.
LDA (Linear Discriminant Analysis)	• Maximizar los ejes de componentes para separación de clases.
ICA (Independent Component Analysis)	• Similar a PCA, excepto que asume características no-Gaussianas.
Multidimensional Scaling	• Encontrar la proyección que mejor conserva las distancias entre puntos.
Autoencoders	• Método que logra codificar capas de entrada en menor # de nodos, minimizando pérdida de información.

Gracias

rsandova@ing.puc.cl
rodrigo@RSolver.com
@RSandovalSolver
/in/RodrigoSandoval
www.RodrigoSandoval.net
www.RSolver.com

ANEXOS: DETALLES DE OTROS MÉTODOS

t-SNE

T-Distributed Stochastic Neighbor Embedding

Es una técnica no-lineal para reducción dimensional, especialmente adecuada para la visualización de datasets de alta dimensión.

Se usa extensivamente en procesamiento de imágenes, procesamiento de lenguaje natural (NLP), datos genómicos y procesamiento de voz.

Se basa en cálculos de probabilidades y de semejanza.

Algoritmo t-SNE

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for $t=1$ **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

Algoritmo t-SNE

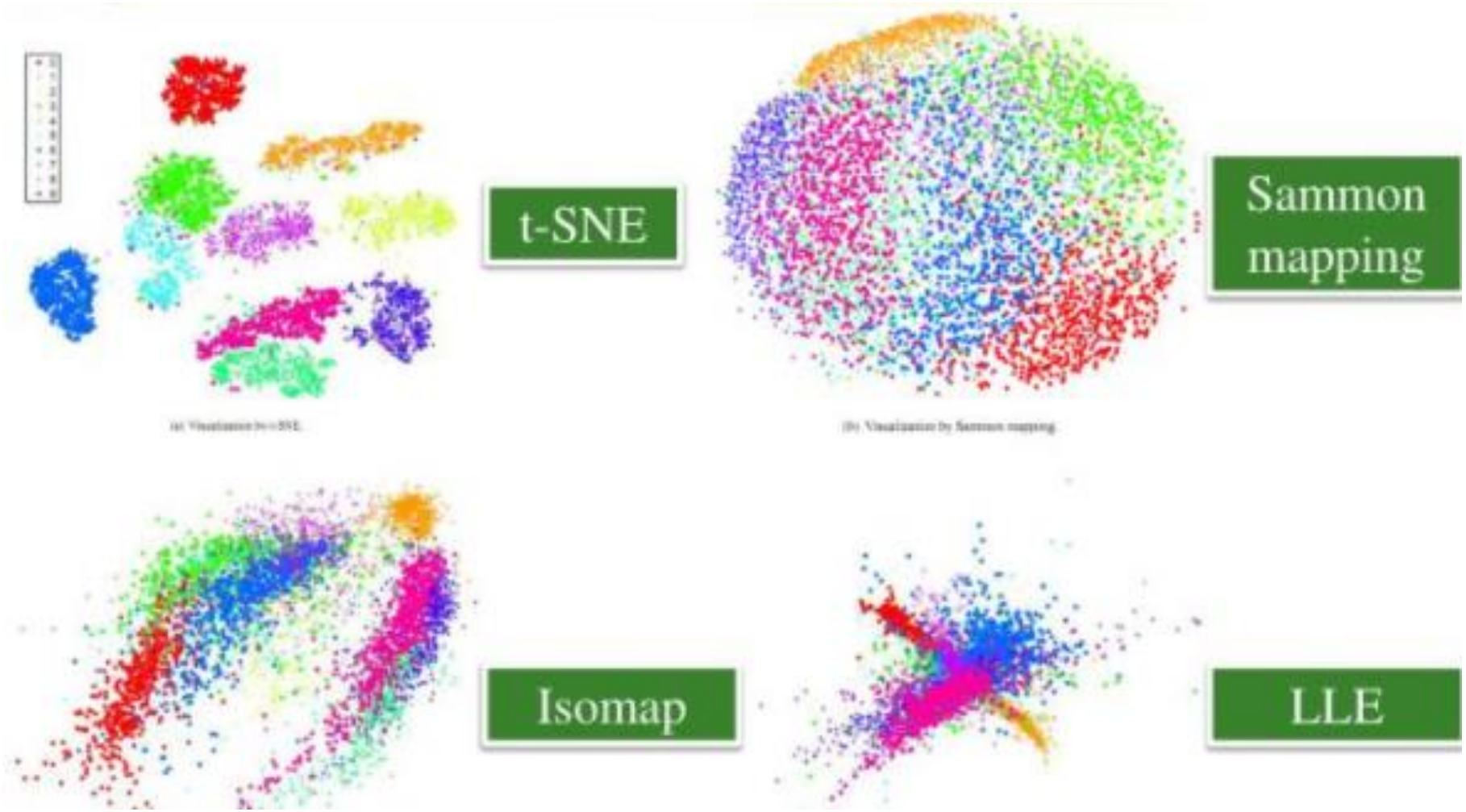
1. Se calcula la probabilidad de similitud de puntos en espacio de alta-dimensión, y la probabilidad de similitud de puntos en el correspondiente espacio de menor dimensión.
   - Esta similitud es la probabilidad condicional de que un punto A elegirá un punto B como su vecino si los vecinos se eligen en proporción a su densidad proporcional, bajo una curva de Gauss centrada en A.
2. A continuación se minimiza la diferencia entre estas probabilidades condicionales (o similitudes) en los espacios de alta-dimensión y baja-dimensión, para una representación perfecta de los puntos de datos del espacio de baja-dimensión.
3. Se mide esta minimización de la suma de diferencias de probabilidad condicional por medio de la minimización de la suma de la divergencia de Kullback-Leibler sobre todos los puntos usando un método de descenso de gradiente (GD).

t-SNE aspectos significativos

- Pone énfasis en el modelamiento de puntos disímiles por medio de distancias grandes entre pares.
- Dado su enfoque en distancias/cercanías, es un enfoque ideal para poder visualizar similitudes entre los datos, describiendo la forma de agrupar los datos, llegando así a reducir dimensionalmente estos datos.

MNIST database (Modified National Institute of Standards and Technology database) es un set abierto de ejemplos de dígitos manuscritos que se ha utilizado por años para el entrenamiento de sistemas de Procesamiento de imágenes, particularmente texto manuscrito.

http://yann.lecun.com/exdb/mnist/



En el análisis de las imágenes MNIST se reconoce que t-SNE logra diferenciar (separar) de mejor manera los diferentes clusters (dígitos diferentes), a diferencia de otros esquemas, en que no se logra la misma separación. Esto se debe al esquema de cálculo de semejanza/distancia entre los datos.

t-SNE

Referencia importante

- Blog de Laurens van der Maaten, uno de los autores de t-SNE junto a G. Hinton: https://lvdmaaten.github.io/tsne/
- Un mini-tutorial para R https://www.displayr.com/using-t-sne-to-visualize-data-before-prediction/

LDA – Linear Discriminant Analysis (LDA)

Se utiliza para resolver reducción dimensional con atributos masivos, pero también es una técnica que se puede usar como clasificador (Agente de predicción "supervisado").

Tiene un paso de pre-procesamiento para clasificación de patrones y aplicaciones de Machine Learning.

Se enfoca en la extracción de características (*feature extraction*)

Se basa en una transformación lineal que maximiza la separación entre múltiples clases.

(Sub)Espacio de características

Reducir las dimensiones de un dataset de D dimensiones, proyectándolo a un subespacio de K dimensiones, donde K < D.

Se debe validar que la representación del espacio de características es correcta.

Computar los eigen vectors del dataset	Recolectarlos en una Scatter Matrix.	Generar datos en K dimensiones desde el espacio de D dimensiones.

Scatter Matrix / Matriz de Dispersión

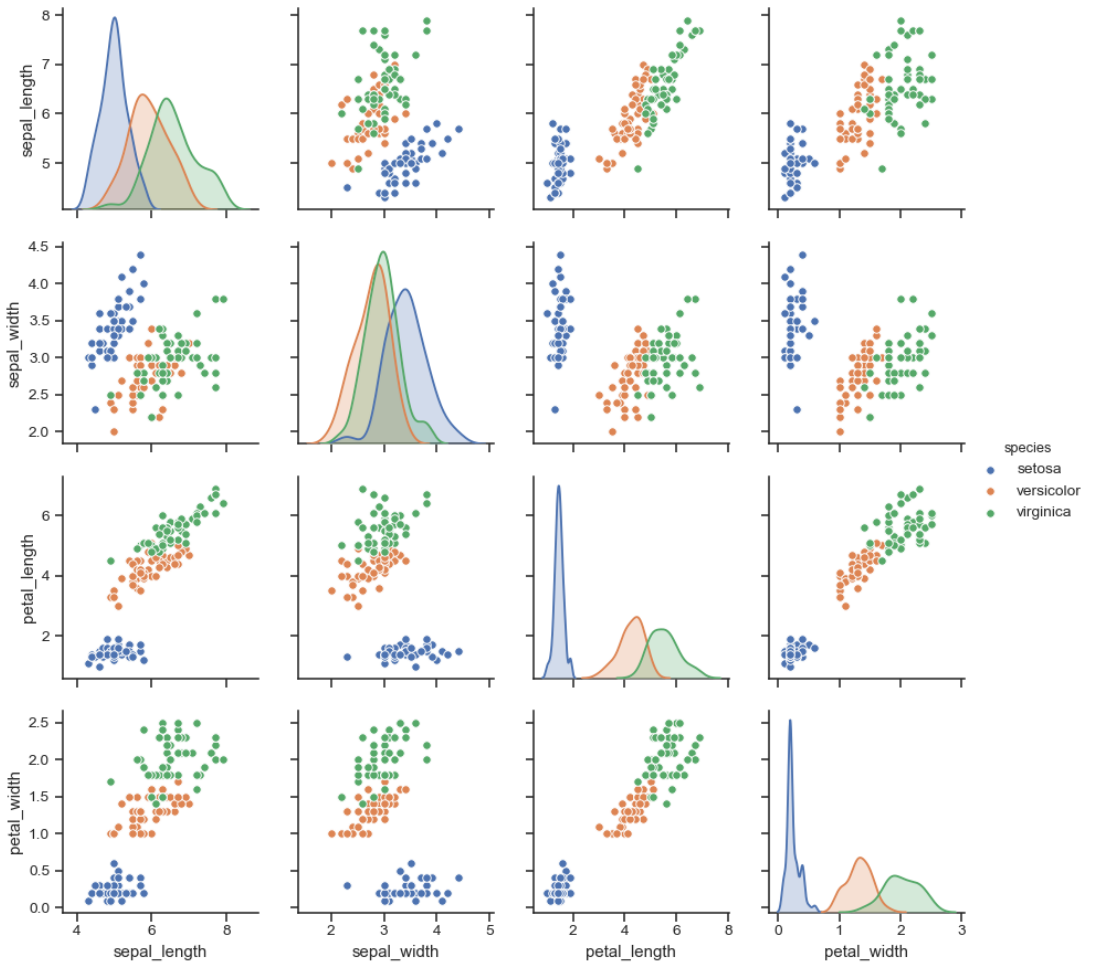
- Matriz de dispersión dentro de la clase

$$S_W = \sum_{i=1}^{c} S_i$$

- Matriz de dispersión entre clases

$$S_B = \sum_{i=1}^{c} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

La idea es maximizar la medición entre clases y minimizar la medición dentro de la clase (distancia intra clase).

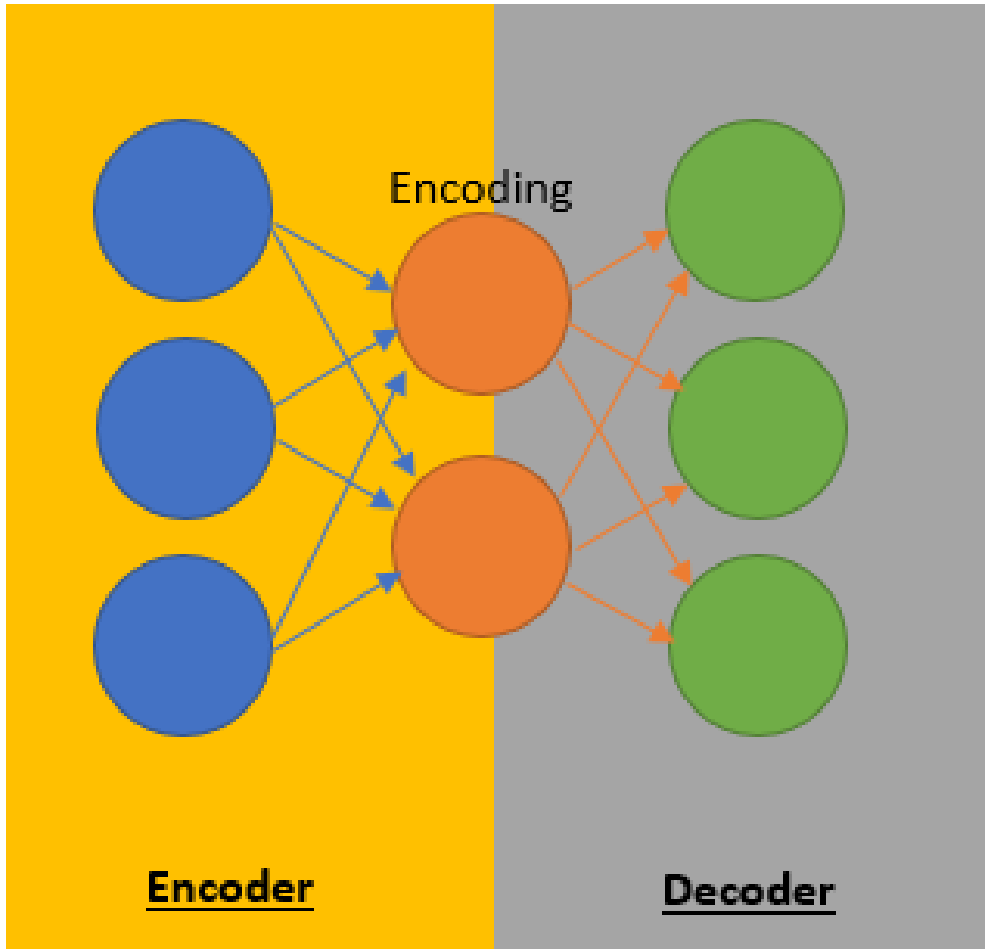


Algoritmo LDA

1. Computar los vectores medios en el espacio de D dimensiones.
2. Computar las matrices de dispersión (Scatter Matrix)
3. Computar los eigenvectors y los correspondientes eigenvalues (vectores y valores propios) para las matrices de dispersión.
4. Ordenar los eigenvalues y seleccionar aquellos con los mayores eigenvalues para forma una matriz de D x K dimensiones.
5. Transformar los ejemplos (datos) al nuevo subespacio.

- ICA: Independent Component Analysis: creación de variables independientes desde las originales.
- Ejemplo: el coctel.
  Mucha gente en un salón teniendo conversaciones diferentes y el objetivo es separar las diferentes conversaciones. Si se tienen muchos micrófonos, cada uno de ellos tendrá una combinación lineal de todas las conversaciones, pero con ruido. Entonces, ICA debería poder diferenciar estas conversaciones del ruido.
  Esto se puede ver como un problema de reducción dimensional, si es que se tienen 200 micrófonos y sólo 10 conversaciones: por lo que se logran representar las 10 variables independientes con ICA.

- Es un método, basado en conceptos de redes neuronales, en el que por medio de codificaciones de los datos y sus variables, se recodifica en menos dimensiones (reduciendo así las dimensiones).
- La red neuronal puede ser una Feed-Forward Network, u otras arquitecturas complejas, donde la idea es que de los datos originales, se pasa a una codificación automática (autoencoder), validando que esa codificación es correcta, al decodificarla y tener el mismo resultado (o con mínimas variaciones).
- Se utiliza como un método de clustering.



Algunas referencias complementarias

- Selección de Características
  https://www.datacamp.com/community/tutorials/feature-selection-R-boruta
- Machine Learning Explained: Dimensionality Reduction
  https://www.r-bloggers.com/machine-learning-explained-dimensionality-reduction/
- Visualizing data using t-SNE
  https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
- Introduction to PCA and t-SNE
  https://www.datacamp.com/community/tutorials/introduction-t-sne