

# Mini Proyecto

## Nombre del estudiante:

César Augusto Núñez Medina 12141019

Luis Carlos Flores 12011299

Ronal Josué Zúñiga Gallegos 12011247

## Sede de estudio:

Tegucigalpa

## Docente:

Ing. Claudia Patricia Cortés Pavón

## Sección:

866 Lenguajes de programación

## Fecha de entrega:

2023-08-26

## Propuesta de Proyecto

Semana: 2

## Descripción del Problema

Se nos pidió implementar el algoritmo para el análisis de componentes principales en dos lenguajes de programación diferentes. Dado un cuadrante según el tipado del lenguaje, ya sea estático o dinámico, o fuerte o débil, decidimos usar JavaScript y Go, ya que son dos lenguajes de diferentes cuadrantes.

El propósito principal del análisis de componentes principales es que se utiliza para el análisis de datos. Usualmente, si tenemos dos componentes, es muy fácil compararlos y ver si sus vectores o valores (puntos) apuntan en la misma dirección, ya sea negativa o positiva. Sin embargo, este problema se vuelve más complejo al introducir diferentes componentes en los cuales se utiliza el ACP. En este caso, el análisis de componentes principales toma todos los componentes que conforman nuestros datos y los puede analizar en un plano con múltiples componentes principales. Esto puede mostrar similitudes, es decir, hacia qué coordenadas apuntan diferentes componentes de vectores.

## Justificación y Explicación

Decidimos comenzar con el lenguaje de programación Go. La elección de Go se basó en varias características clave que creímos que serían beneficiosas para implementar el análisis de componentes principales (PCA). Go encaja en la categoría de lenguajes de tipado fuerte y estático, lo cual es un enfoque riguroso pero seguro. Al ser de tipado fuerte, Go nos garantiza precisión al nivel de hacer los cálculos matemáticos necesarios.

El hecho de que Go sea de tipado fuerte significa que no permite operaciones matemáticas de diferentes tipos sin conversiones explícitas. Esto es fundamental en nuestro caso, ya que queremos asegurarnos de que los cálculos se realicen de manera coherente y sin ambigüedades.

Otro factor que consideramos es el tipado estático de Go. En el contexto de la implementación de PCA, a menudo necesitamos trabajar con vectores y matrices, e incluso definir variables con precisión científica debido al gran tamaño de los datos. Esto es una parte esencial de nuestra elección, ya que queríamos asegurar que al declarar un tipo de variable no cambiaría en tiempo de ejecución.

En resumen, la elección de Go se basó en su tipado fuerte y estático, que promete precisión y coherencia en nuestros cálculos matemáticos. Esto es crucial en la implementación exitosa de PCA, donde la exactitud es fundamental para obtener resultados confiables y significativos.

# Documentación código Golang

## 1 Normalizar Matriz

Dado que el análisis PCA es afectado por la escala de las variables, es vital normalizar la matriz inicial. Esto se realiza para que las variables o valores dentro de un rango más limitado no alteren indebidamente los cálculos.

### 1.1 Calcular Medias de Columnas

Utilizando la matriz inicial, se recorren las filas y luego las columnas para obtener y acumular valores en un arreglo. Al finalizar, cada valor del arreglo se divide por el tamaño de la matriz.

#### Mean Formula

$$\text{Mean} = \frac{\text{Sum of All Data Points}}{\text{Number of Data Points}}$$

### 1.2 Calcular desviación estándar

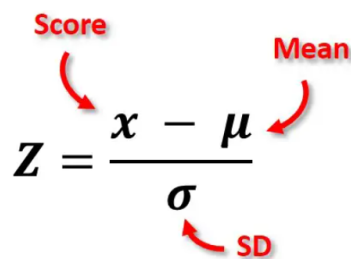
Basándose en la matriz inicial, se declara un arreglo que utiliza la función de medias del paso anterior. Luego, se recorre la matriz y se aplica la fórmula de desviación estándar, evaluando numerador y denominador separadamente.

#### Calculating Standard Deviation

$$s_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

### 1.3 Aplicar Z-Score normalization

Con la matriz inicial y los arreglos previos, se recorre la matriz aplicando la fórmula de normalización, asegurando valores entre -1, 0 y 1 para los datos.

$$Z = \frac{x - \mu}{\sigma}$$


## 2 Cálculo de Matriz de Correlaciones

Esta matriz evalúa las relaciones lineales entre diferentes variables.

### 2.1 Calcular el valor de correlación

Se recorre la matriz ya normalizada, llenando cada celda de la nueva matriz de correlaciones con un valor que se obtiene a través de una función dedicada.

## 2.2 Función de Valor de correlación

Al recibir dos vectores o arreglos, se implementa la fórmula de correlación. Dado que la matriz ha sido previamente normalizada, se aplica una versión simplificada de esta fórmula (sin obtener la media).

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

## 3 Calcular los Vectores y Valores Propios

Los vectores propios representan direcciones de variabilidad en datos, y los valores propios su magnitud.

### 3.1 Calcular valores y vectores

Usando la matriz de correlación R, primero se inicializa y se normaliza (mediante la media) un vector aleatorio. Posteriormente, al multiplicar la matriz R con el vector, se usa la función producto punto para determinar el valor entre el vector resultante y R.

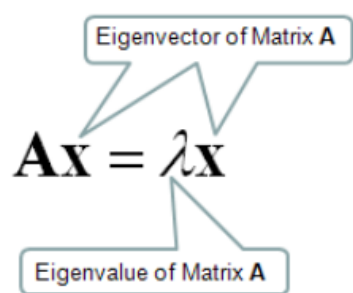


Diagrama de la ecuación  $Ax = \lambda x$ . La etiqueta "Eigenvector of Matrix A" apunta al vector  $x$ . La etiqueta "Eigenvalue of Matrix A" apunta al escalar  $\lambda$ .

### 3.2 Ordenar y optimizar la matriz

Para optimizar, se garantiza que el mayor valor propio y su vector correspondiente se agrupen. Se realiza restando de R el producto del vector propio por su valor propio.

## 4 Cálculo de vectores propios

Tras ordenar valores propios y vectores en el paso anterior, se empleó una estructura para replicar las "tuples" de Python, garantizando la correspondencia adecuada.

## 5 Cálculo de matriz de Componentes principales

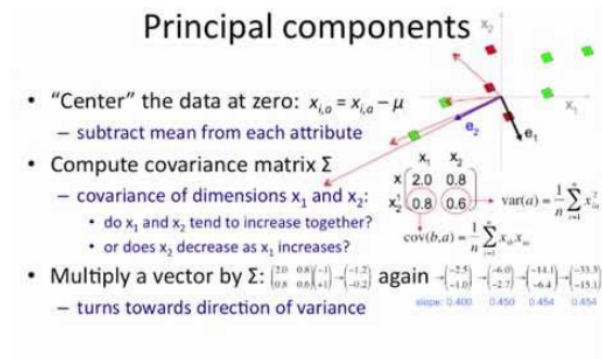
Ofrece una representación reducida de los datos originales.

### 5.1 Función para calcular la matriz

Usando la matriz normalizada y la matriz de vectores propios, se calcula cada valor para la matriz principal a través de una sumatoria, multiplicando valores de la matriz normalizada por su vector propio correspondiente.

## 6 Cálculo de Matriz de calidades de individuos

Mide la precisión de cada dato en función del componente principal.



## 6.1 Función para calcular matriz

Una función toma la matriz de componentes principales y la matriz normalizada. Posteriormente, para el cálculo, se suman los valores elevados al cuadrado de la matriz X (denominador) y se contrastan con los valores elevados al cuadrado de la matriz C (numerador).

$$Q_{ir} = \frac{(C_{i,r})^2}{\sum_{j=1}^m (X_{ij})^2} \quad \text{para } i = 1, 2, \dots, n; \quad r = 1, 2, \dots, m.$$

# 7 Cálculo de la matriz de coordenadas de las variables

## 7.1 Calcular valores y vectores propios

El algoritmo es capaz de calcular los valores y los vectores propios asociados a la matriz de correlaciones resultante del conjunto de datos proporcionado, y a partir de estos valores, calcular la matriz de coordenadas que representa las proyecciones de los datos en el espacio de componentes principales. Primero, dentro de la función, se calcula la matriz de correlaciones. Luego, se obtienen los valores y los vectores propios utilizando la matriz de correlaciones previamente calculada.

## 7.2 Calcular matriz de coordenadas

La función se llama con las matrices de componentes principales y la matriz de vectores propios. En esta función, se inicializan variables para el número de filas y columnas en ambas matrices. Luego, se crea una matriz vacía para almacenar las coordenadas calculadas. Para asignar el valor correspondiente a la matriz de coordenadas, mediante dos ciclos for anidados, se realiza lo siguiente:

- Se obtiene la columna  $j$  de la matriz de vectores propios.
- Se calcula el producto punto entre la fila  $i$  de la matriz de componentes principales y la columna  $j$  de la matriz de vectores propios. El resultado se asigna a la celda correspondiente en la matriz de coordenadas.

# 8 Calcular matriz de calidades de las variables

Esta matriz refleja la proporción de varianza explicada por cada componente principal y sus respectivas coordenadas en el espacio de componentes. La función recibe una lista de vectores propios y una matriz de coordenadas como entrada. Dentro de la función, se realiza un recorrido por las filas y columnas de la matriz de coordenadas. Se inicializan las dimensiones de la matriz de calidad según el número de filas y columnas en la matriz de coordenadas. Luego, se utiliza una fórmula para calcular el valor de cada celda en la matriz de calidad. En el bucle exterior que recorre las filas, se crea una fila correspondiente en la matriz de coordenadas. Dentro del bucle anidado que recorre las columnas, se calcula el valor de calidad para cada celda utilizando la fórmula:

$$(coordenada^2) / valorpropio \times 100$$

Esta fórmula representa la proporción de la varianza explicada por la coordenada en relación con su valor propio correspondiente.

## 9 Calcular el vector de inercias

El vector de inercia refleja la cantidad total de varianza explicada por cada componente principal. La función recibe una lista de vectores propios como entrada. Dentro de la función, se obtiene la longitud del vector y se reserva memoria para el vector de inercia. Luego, se recorre el vector, y en cada iteración se calcula el valor de inercia utilizando la fórmula:

$$(\text{valorpropio}/m) \times 100$$

donde  $m$  es la longitud del vector.