Categoría.

Aprendizaje no supervisado.

Analísis de componentes principales PCA (Jolliffe, 2010) Pág 64.

Objetivo: Reducir la dimensionalidad de un dataset formado por numerosas variables interrelacionadas, manteniendo tanto como sea posible la varianza presente en el dataset.

Puede ser considerado modelo más que un sólo algoritmo (es ampliamente aplicable), es una herramienta que nos permite ver de un conjunto de variables cuales son aquellas que tienen máxima varianza y las ordena de mayor a menor.

Entre mayor sea la varianza mejor podrémos hacer proyecciones sobre esa variable o eje.

También este modelo nos permite cuantificar que tan buena es una proyección de un espacio a otra de menor dimensión (10 dimensiones a 3 dimensiones). Es un conjunto de numeros que muestran las variables más relevantes y las ordena. Si tomamos las últimas componentes (menos relevantes) nos sirve para detectar outliers *valores atípicos*.

Se usa aquí el método del cálculo de eigenvectores y eigenvalores de la **matriz de covarianzas**.

Variables interrelacionadas.

Si guardan correlación un par de variables tal que Correlación = 1 y Magnitud = (->,->) ó -1 (->,<-) estas son descartables para representar nuestras identidades, *podemos decidir entre una u otra*.

Carecterizando con 2 medidas el siguiente dataset (imagen plano cartesiano) **PCA encontrará la dirección de máxima varianza** (aquí la diagonal) Si rotamos el dataset de manera que quede sobre la horizontal, reducimos la caracteriación o rasgos del dataset de 2 dimensiones a 1. Si hay un grupo de datos que caen sobre el mismo punto en la recta se **pierde la información**, esto es usual y se busca estar a un porcentaje de **tolerancia** de información para usarce como **aproximación**.

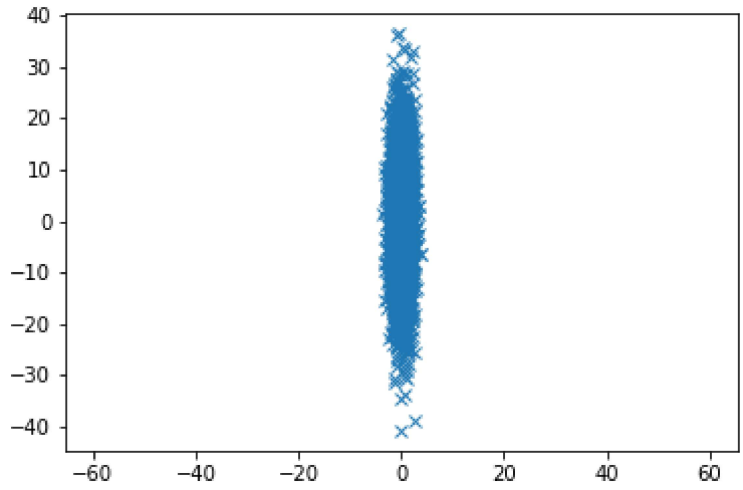
La maldición de la dimensionalidad.

Cuando la dimensionalidad incrementa, el volumen del espacio incrementa tan rápido tal que la información disponible se esparce o dispersa. Ejemplos:

- **Singularidad de matrices:** que la matriz crezca tanto que se vuelva singular (por lo que su determinante es cero) y ya no es invertible, pierde muchas propiedades del algebra líneal.

- **Efecto de aniquilación y explosión:** cuando los valores se hacen tan pequeños hasta que numericamente desaparecen (potencias con datos normalizados. i.e. $x^n | x \in (0,1)$). O que crecen tan rápido que tienden a infinito incluso con valores pequeños. i.e. 1.2
- *Distribución de baja redundancia:** La varianza en cualquiera de las direcciones es aproximadamente la misma. i.e. un dataset distribuido en un circulo. estilo de simetría.
- *Distribución de alta redundancia:** El caso de una distribución sobre una linea recta. El método es aplicable en despreciar una dimension después de su proyección

```
mean = [0, 0]
cov = [[1, 0], [0, 100]]
import matplotlib.pyplot as plt
import numpy as np
x, y = np.random.multivariate_normal(mean, cov, 5000).T
plt.plot(x, y, 'x')
plt.axis('equal')
plt.show()
```



+ Código + Texto

Si bajamos la varianza de 100 de la segunda variable la distribución es más esférica. Con varianza 100 es más redundante que con varianzas similares. Nos quedamos con la de mayor varianza por que la proyección es muy fiel, proyecta los puntos sin tanta perdida de información. Al perder dimension no pierde información, es redundante.

```
def ave(array):
  return np.sum(array, axis = 0)/len(array) #row = len(array), col = len(array[0])

def cov(array1, array2, dim, sample_no):
  ave1, ave2 = ave(array1), ave(array2)
  diff = [(array1[i] - ave1)*(array2[i] - ave2) for i in range(sample_no)]
  return np.sum(diff, axis = 0)/(sample_no) #poblational version

def covM(data, dim, sample_no):
  M = []
  M = [[cov(data[:,i], data[:,j], dim, sample_no) for j in range(dim)] for i in range(dim)
```

```
  return np.asarray(M)

data = np.asarray([[90, 60, 90],
        [90, 90, 30],
        [60, 60, 60],
        [60, 60, 90],
        [30, 30, 30]])

covM(data, len(data[0]), len(data))
```

```
array([[504., 360., 180.],
       [360., 360.,   0.],
       [180.,   0., 720.]])
```

Autovalores

Se dice que $\boldsymbol{a}$ es el autovector del operador Ω *representable como matriz* con el autovalor ω

$$\Omega\boldsymbol{a} = \omega\boldsymbol{a} \rightarrow (\Omega - \omega I)\boldsymbol{a} = 0,$$

$$e_i(\Omega - \omega I)\sum_{j=1}^{n} a_j e_j = 0,$$

$$\sum_{j=1}^{n} a_j e_i(\Omega - \omega I)e_j = 0,$$

$$\sum_{j=1}^{n}(\Omega_{ij} - \omega\delta_{ij})a_j = 0$$

Este es un grupo de ecuaciones lineales homogeneneas *con* v_j *desconocidas* conocida como la ecuacion característica. Una solución no trivial, en este caso, existe si el determinante del coeficiente matricial se desvanece, i.e.,

$$det(\Omega_{ij} - \omega\delta_{ij}) = 0$$

Claramente, si estamos trabajando en un espacio vectorial n-dimensional, este polinomio es de orden n en ω y por lo tanto posee n soluciones para ω, que corresponde a todos los autovalores de Ω. Estas raíces no deben ser necesariamente distintas o reales. Sin embargo, una vez que se obtienen los autovalores, los autovectores se derivan de la ecuación característica de forma simple.

Ejercicio. La implementación de la matriz de covarianzas en excel se obtiene:

					VARIANZAS			COVARIANZAS		
	A	B	C		AA	BB	CC	AB	AC	CB
1	71	35	96		900	17.64	2787.84	-126	1584	-221.76
2	53	48	26		144	77.44	295.84	105.6	-206.4	-151.36
3	15	39	14		676	0.04	852.64	5.2	759.2	5.84
4	51	49	1		100	96.04	1780.84	98	-422	-413.56
5	15	25	79		676	201.64	1281.64	369.2	-930.8	-508.36
PROMEDIO	41	39.2	43.2		499.2	78.56	1399.76	90.4	156.8	-257.84
	30	-4.2	52.8		MATRIZ DE COVARIANZA					
	12	8.8	-17.2		X	A	B	C		
	-26	-0.2	-29.2		A	499.2	90.4	156.8		
	10	9.8	-42.2		B	90.4	78.56	-257.84		
	-26	-14.2	35.8		C	156.8	-257.84	1399.76		

$$\Omega = \begin{bmatrix} 499.2 & 90.4 & 156.8 \\ 90.4 & 78.56 & -257.84 \\ 156.8 & -257.84 & 1399.76 \end{bmatrix}$$

$$\sum_{j=1}^{3} (\Omega_{ij} - \omega\delta_{ij})v_j = 0$$

Tendremos una solucion no trivial siendo que el determinante del coeficiente de la matriz se desvanece.

$$det \begin{bmatrix} 499.2 - \omega & 90.4 & 156.8 \\ 90.4 & 78.56 - \omega & -257.84 \\ 156.8 & -257.84 & 1399.76 - \omega \end{bmatrix} = 0$$

Se resuelve el polinomio cubico

$$-\omega^3 + 1977.52\omega^2 - 748702.624\omega + 1026875.751 = 0$$

Para obtener los autovalores siguientes

$$\omega = 1.37654, 508.1724, 1467.97105$$

```
Matrix = np.asarray([[499.2, 90.4, 156.8],
         [90.4, 78.56, -257.84],
         [156.8, -257.84, 1399.76]])

w, v = np.linalg.eig(Matrix)
w
```

```
array([1.46797105e+03, 5.08172405e+02, 1.37654156e+00])
```

$$\Omega = \begin{bmatrix} 497.8 & 90.4 & 156.8 \\ 90.4 & 77.18 & -257.84 \\ 156.8 & -257.84 & 1398.38 \end{bmatrix}$$

$$\Omega = \begin{bmatrix} -8.9 & 90.4 & 156.8 \\ 90.4 & -429.61 & -257.84 \\ 156.8 & -257.84 & 891.59 \end{bmatrix}$$

$$\Omega = \begin{bmatrix} -968.8 & 90.4 & 156.8 \\ 90.4 & -1389.41 & -257.84 \\ 156.8 & -257.84 & -68.21 \end{bmatrix}$$

```
1 v
```

```
array([[-0.14177009, -0.96131831,  0.23615323],
       [ 0.17169236, -0.25882631, -0.9505423 ],
       [-0.97489639,  0.09421277, -0.20174484]])
```

```
1
```