

Reducción de dimensionalidad

1. El concepto de reducción de dimensionalidad.
2. El algoritmo PCA.

El concepto de
reducción de
dimensionalidad.

Reducción de dimensionalidad

- » Los set de datos con los que trabajamos están compuestos por filas y columnas.
- » En machine learning, las filas son objetos, individuos o fenómenos de estudio. Las columnas son características o variables que los describen.
- » Hoy en día, muchos set de datos están sobrecargados con numerosas características que dan como resultado un ajuste excesivo, y aumentan los enormes costos tanto de almacenamiento como de tiempo de respuesta.

Reducción de dimensionalidad

Los algoritmos desarrollados a lo largo del tiempo tienen como objetivo resolver algunos de los problemas básicos, incluyendo:

- Reducir la dimensión del conjunto de datos restableciendo la varianza y manteniendo intacta la información relevante.
- Reducir el tiempo y el costo de almacenamiento.
- Estructurar formas de visualización efectivas.

El problema surge ante la tendencia de obtener más y más variables (o características) del fenómeno a estudiar. Llega un punto en el cual el incorporar más variables empeora la situación en vez de mejorarla. A esto se le llama:

¡¡La maldición de la dimensionalidad!!


Reducción de dimensionalidad

La maldición de la dimensionalidad,

ocurre porque la densidad de los datos disminuye exponencialmente con el aumento de la dimensionalidad.

Cuando seguimos añadiendo características sin aumentar el número de muestras, la dimensionalidad del espacio de características crece y se vuelve más y más dispersa.

Debido a esta escasez, resulta mucho más fácil encontrar una solución perfecta para el modelo de Machine Learning, lo que muy probablemente conduce a un **sobreajuste**.

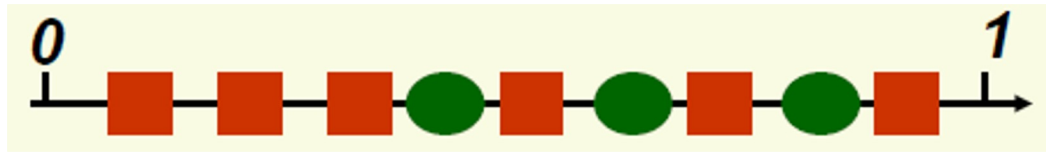


Lo veremos más adelante, pero en palabras simples significa que el algoritmo de ML memoriza en vez de aprender de los datos.

Reducción de dimensionalidad

Veamos la situación descrita con un ejemplo:

Supongamos que tenemos solo una variable para describir 9 objetos:

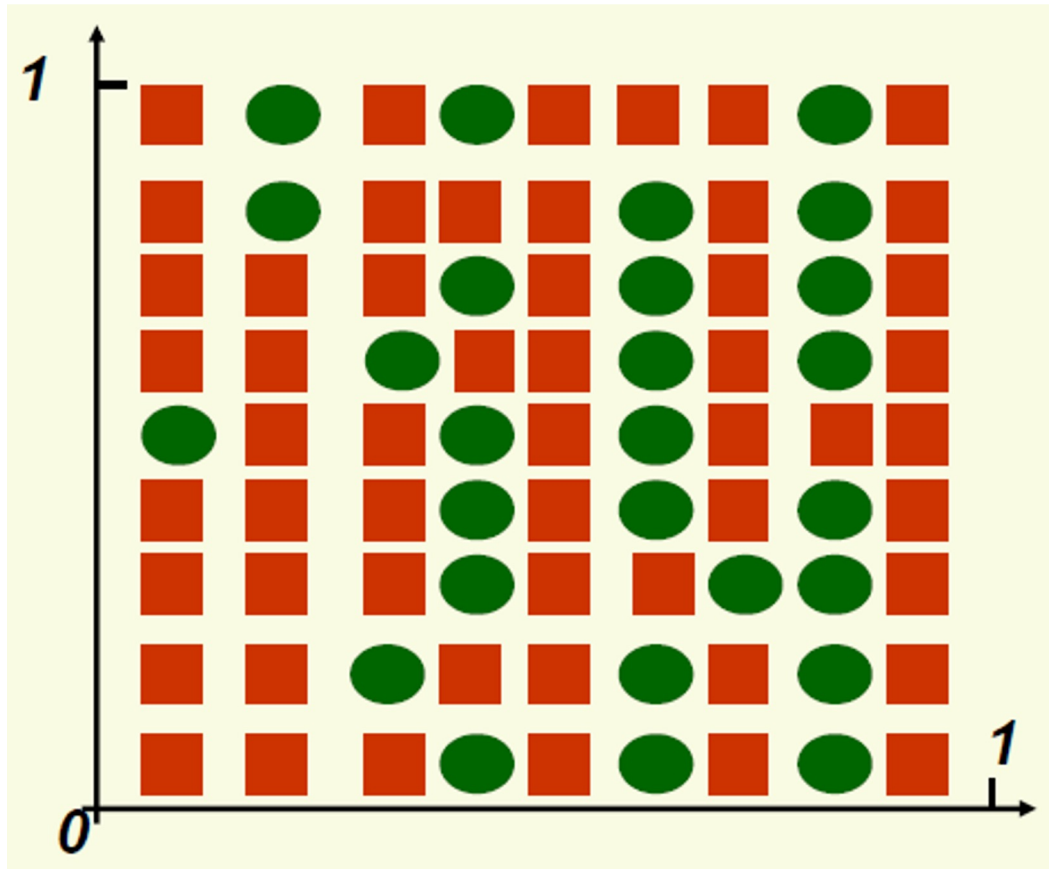


Los cuadrados y círculos están mezclados, por lo tanto solo 1 variable no describe bien estos datos para que, de alguna manera, pudieramos discriminarlos.

Ojo con la densidad de estos datos...

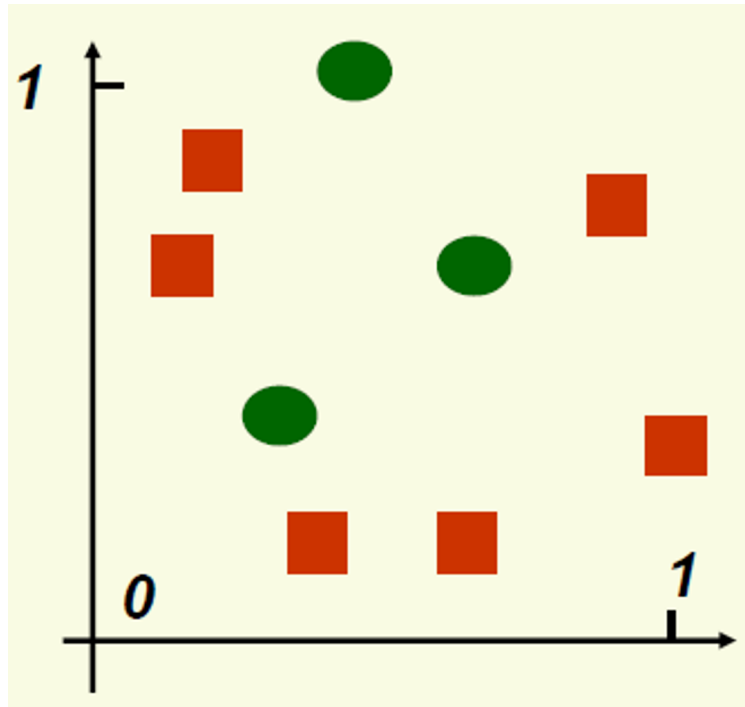
Reducción de dimensionalidad

Entonces necesitamos 9^2 ejemplos para mantener la misma densidad:



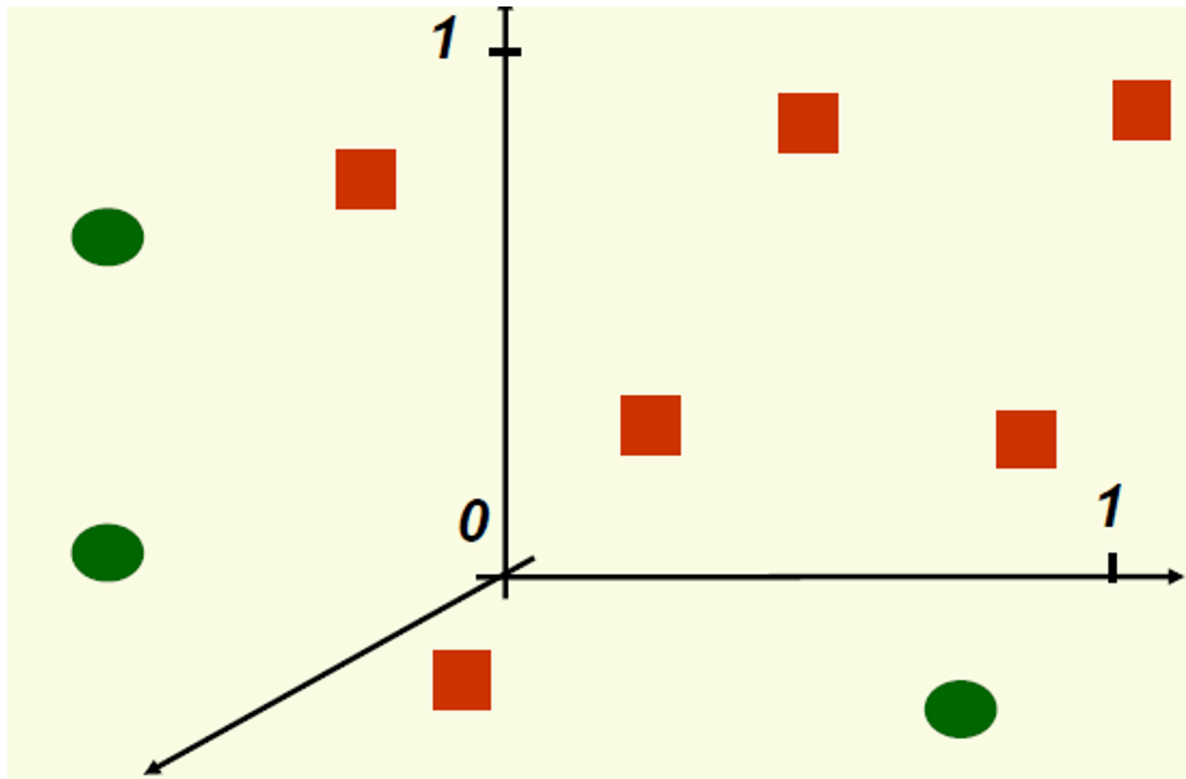
Reducción de dimensionalidad

Pero cuando aumentamos las dimensiones, no se aumentan los ejemplos. Entonces, obtenemos esto:



Reducción de dimensionalidad

Y en 3 dimensiones, es aun peor. Suma y sigue...



Se necesita 9^3 (729) ejemplos para conservar la densidad.

Entonces:

- » Si tenemos d dimensiones, necesitamos n^d ejemplos.
- » Esta cantidad crece extremadamente rápido en función de d .

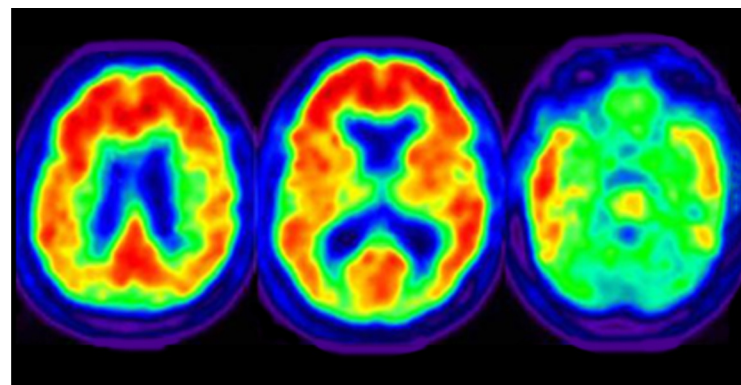
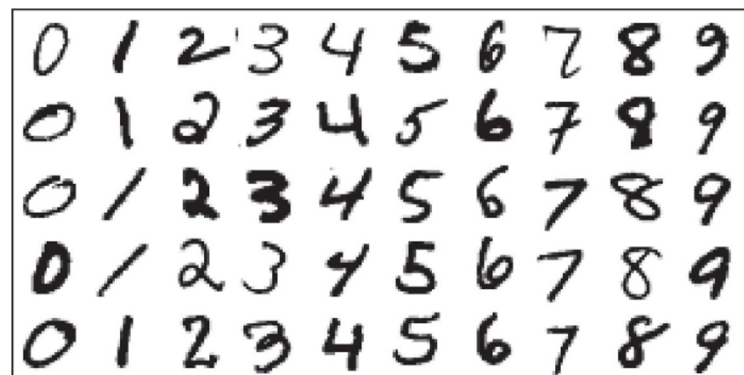
Dificultades:

- » Si no se puede resolver el problema con pocas características, es lógico pensar que agregando más es una buena idea.
- » Sin embargo, el número de ejemplos no cambia.
- » El agregar más características no necesariamente mejora el resultado.

- » Lo ideal es tener muchos ejemplos y varias características.

$$n \gg D$$

- » Pero pocas veces tenemos pleno control de estas cantidades. Obtener ejemplos puede ser costoso y las características pueden ser muchas por defecto.
- » Ejemplo: Imágenes!!



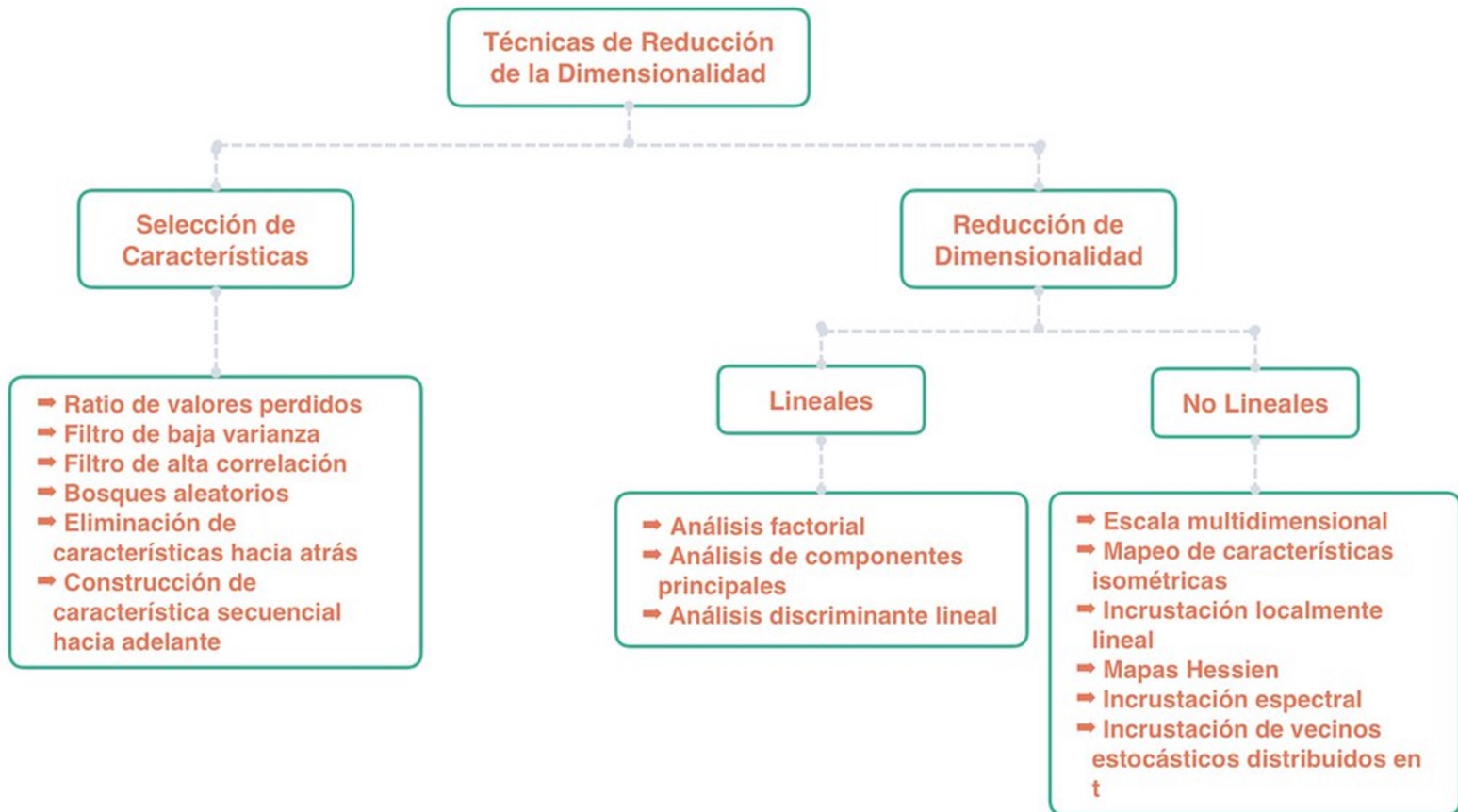
- » La reducción de dimensionalidad busca reducir el número de características de una matriz X de datos.
- » Esto quiere decir que si la matriz X tiene dimensiones $n \times D$ pasa a tener dimensiones $n \times d$ donde $d \ll D$.
- » Al reducir dimensiones, sin duda existe pérdida de información, pero: ¿esta información es valiosa?
- » Cuando estamos en presencia de la maldición de la dimensionalidad, entonces una solución es aplicar técnicas de reducción de dimensionalidad.

En términos generales, la reducción de la dimensionalidad tiene dos clases: eliminación de características y la extracción de características.

Eliminación de la característica: es la eliminación de algunas variables completamente si son redundantes con alguna otra variable o si no están proporcionando ninguna información nueva sobre el conjunto de datos. La ventaja de la eliminación de características es que es fácil de implementar y hace que nuestro conjunto de datos sea pequeño, incluyendo sólo las variables en las que estamos interesados. Pero como desventaja, podríamos perder algo de información de las variables que dejamos de evaluar.

Extracción de variables: es la formación de nuevas variables a partir de las antiguas (y las antiguas son eliminadas). Esto realiza una transformación de los datos, lo cual hace perder interpretación de las variables (desventaja), pero permite resumir la información importante de las variables usando menos variables (ventaja). Es decir, reduce la dimensión tratando de perder la menor cantidad de información valiosa posible.

La taxonomía de técnicas de reducción de dimensionalidad no está definida 100%



El algoritmo PCA

PCA

PCA: Principal Component Analysis

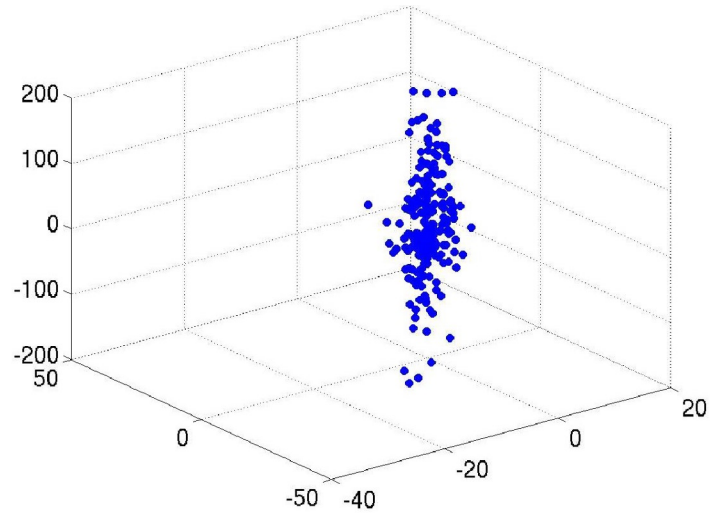
Es la técnica más usada para reducir dimensionalidad.

Realiza una transformación de los datos basada en la varianza. Intenta colocar los puntos en un nuevo espacio manteniendo la máxima varianza posible.

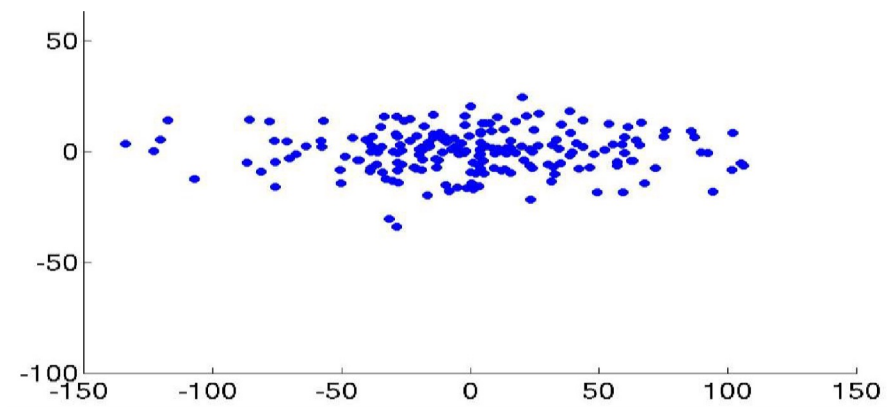
Un resultado en la figura siguiente:

PCA

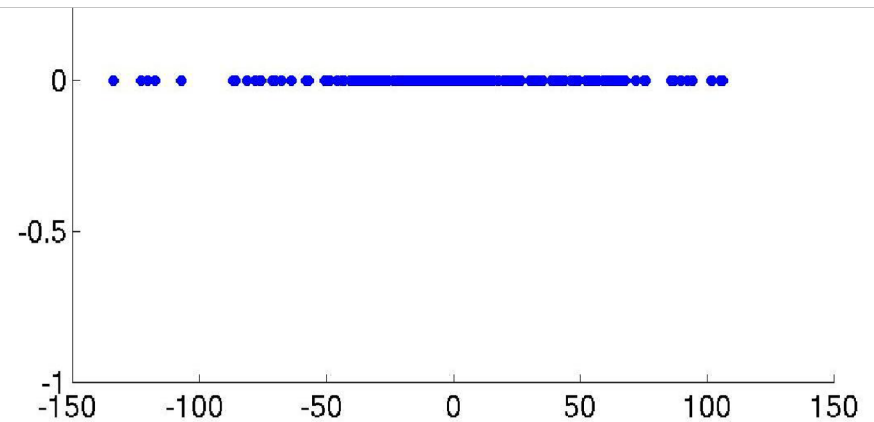
Datos en 3-D



Datos en 2-D

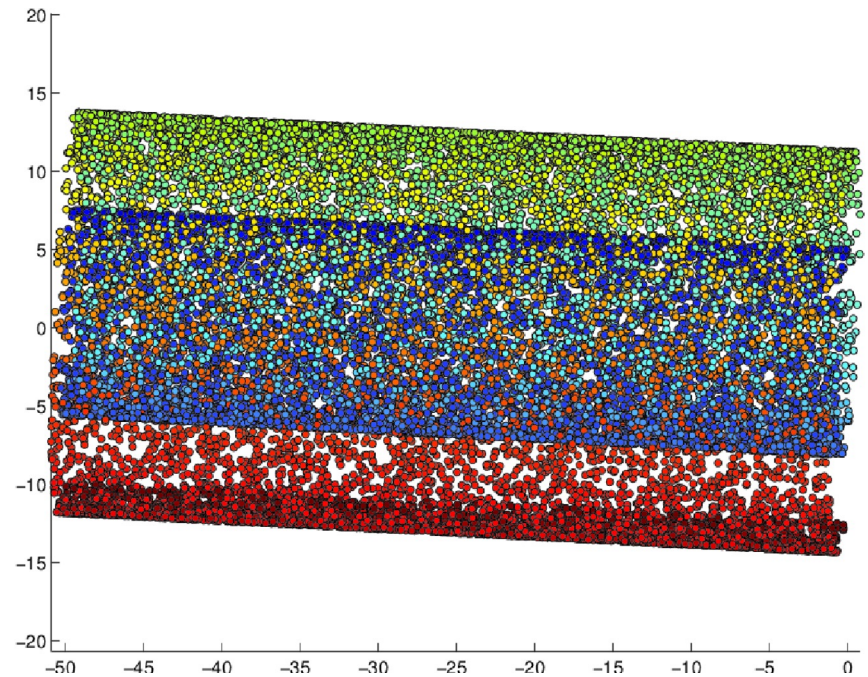
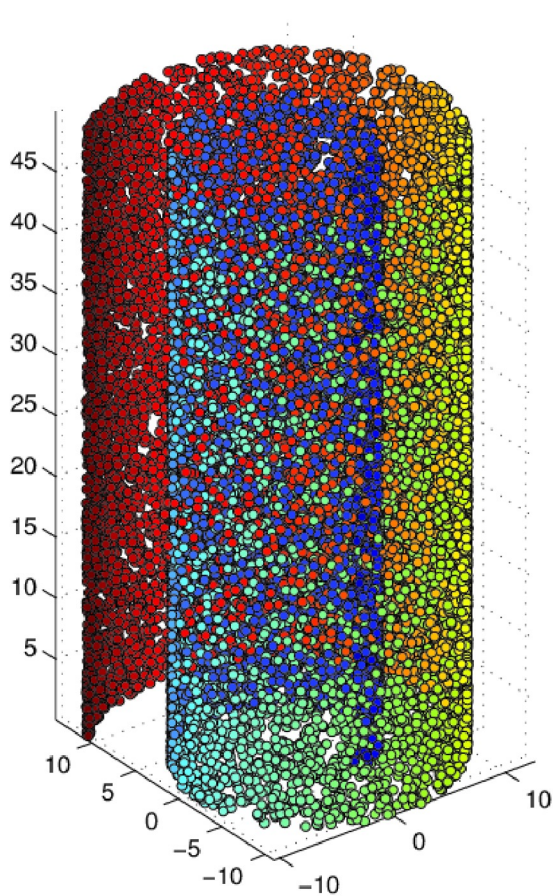


Datos en 1-D



PCA

Otro ejemplo:



PCA

Para aplicar PCA se deben realizar 3 pasos:

- 1) Centralizar los datos a media cero (se puede estandarizar los datos también).

$$x'_{ij} = x_{ij} - \bar{x}_i \qquad \bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ij}$$

- 1) Calcular la matriz de covarianza.

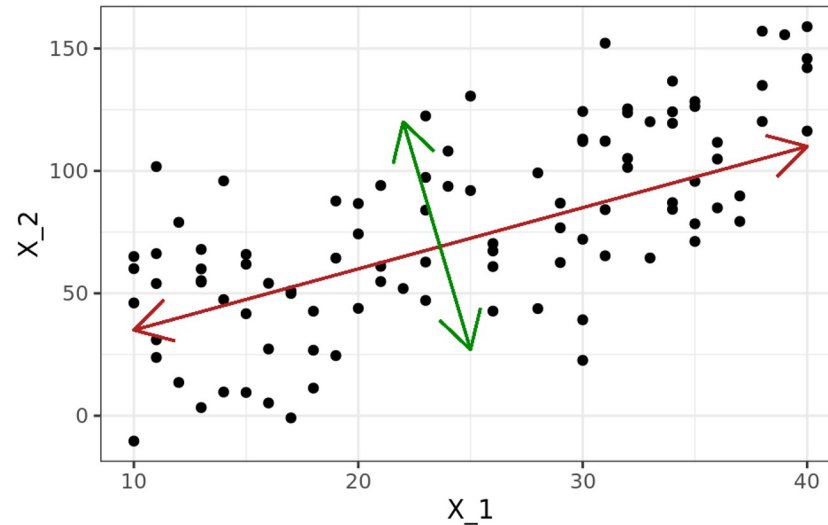
$$C = \frac{1}{n} X X^T$$

- 1) Calcular los eigenvalues y los eigenvectors de la matriz de covarianza. Como ejemplo, se puede usar la descomposición espectral:

$$C = U \Lambda U^T \qquad \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_D)$$

PCA

La obtención de los valores y vectores propios permiten lo siguiente:



Esas flechas indican las direcciones donde hay mayor variabilidad.

PCA encuentra esas direcciones para mantener la mayor variabilidad de los datos. Esas direcciones son los llamados:

Componentes Principales

PCA

Entonces, se escogen los componentes principales como los vectores singulares asociados a los valores singulares mayores.

Estos componentes principales construirán la matriz que transforma los datos originales a un espacio nuevo de menor dimensión.

Uno elige cuantos componentes principales usará para la transformación.

$$Y_{n \times d} = X_{n \times D} W_{D \times d}$$

The diagram illustrates the PCA transformation equation $Y_{n \times d} = X_{n \times D} W_{D \times d}$. Three red arrows point from descriptive labels to the corresponding matrices in the equation:

- An arrow points from the label "Datos originales transformados" to the matrix $Y_{n \times d}$.
- An arrow points from the label "Datos originales" to the matrix $X_{n \times D}$.
- An arrow points from the label "Matriz de transformación" to the matrix $W_{D \times d}$.

PCA

¿Cuántos componente elegir?

PCA tiene una propiedad muy buena:

- » La varianza de los datos está dada por cada eigenvalue y eigenvector.
- » La siguiente expresión entrega la proporción de la varianza expresada con los primeros d componentes principales.

$$\frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^D \lambda_i}$$

Lo normal es entregar como parámetro el porcentaje mínimo de varianza explicada.

¿Donde estamos
en CRISP-DM?

CRISP-DM



AQUI