Reducción de Dimensiones

75.06 - Organización de Datos
Septiembre 2019

Motivación

- Combatir la maldición de la dimensionalidad.
    - Para poder estudiar si un algoritmo mejora o empeora cambiando la dimensión de los datos.
- Performance
    - Poder usar un algoritmo inviable con nuestro set de datos por la cantidad de dimensiones que tiene
    - Adaptarlo por limitaciones de recursos (disco, memoria) facilitando el almacenamiento y el procesamiento de los datos.

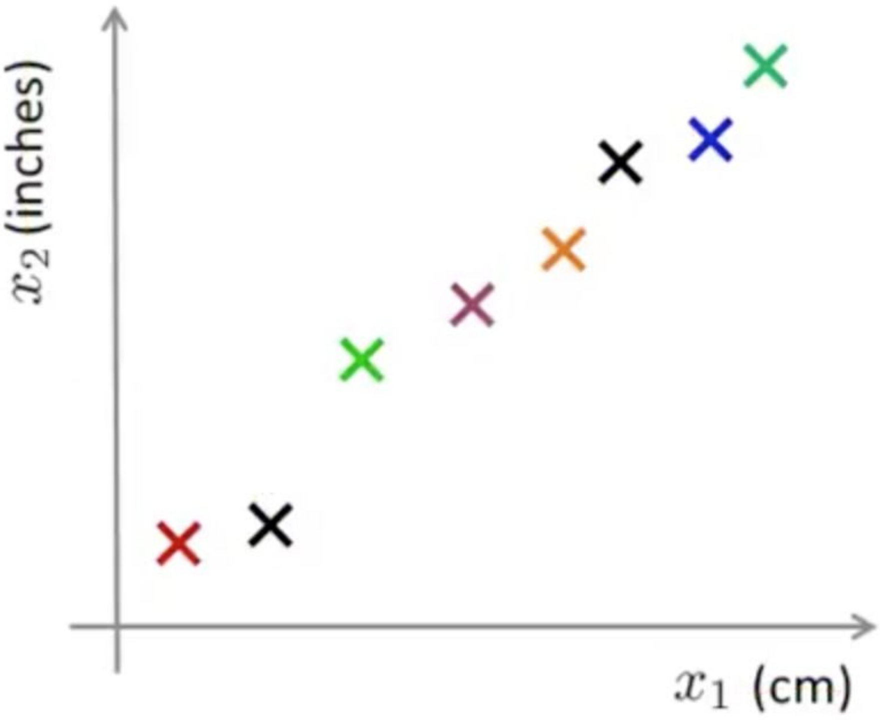
Motivación

- Mecanismo de Feature Engineering
  - Reducir el nuevo set de datos filtrando features ruidosos y devolviendo un set con una menor cantidad de ruido.
  - Estos features pueden agregarse a los otros, aumentando el nivel de señal de los datos.
  - Descubrir correlaciones/tópicos ocultos en los datos
    - Por ejemplo palabras que ocurren comúnmente con otras.

Motivación

- Interpretación y Visualización de Datos
  - Cerebro Humano entrenado para poder entender datos en dos dimensiones y tres (en menor medida).
  - Uso de algoritmos para llevarlos a estar representaciones
    - Manifold Learning: Aprender la “superficie” de cualquier figura (de multiples dimensiones).

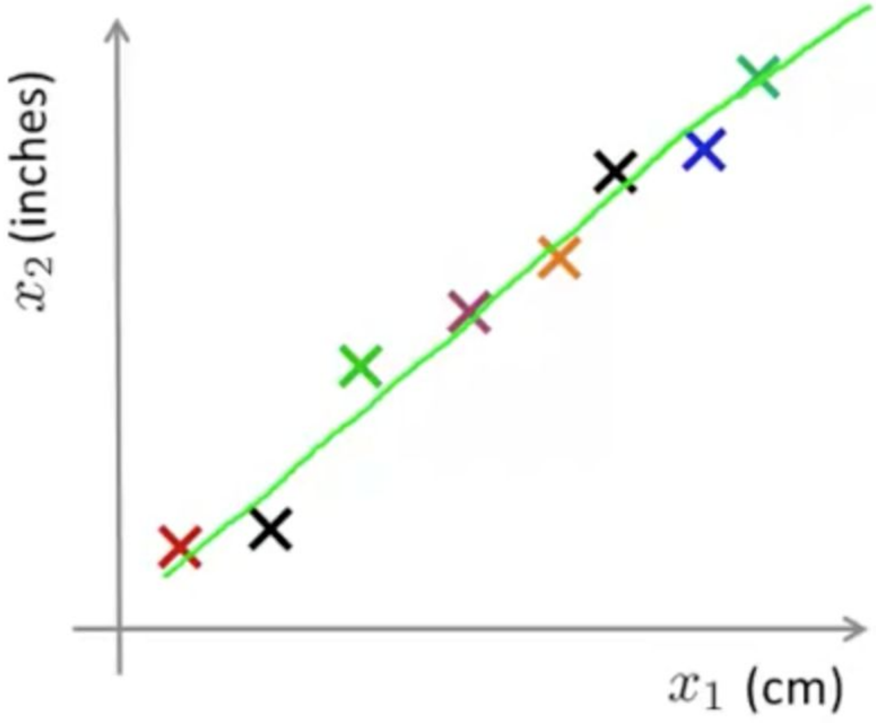
Intuición: ¿Qué es reducir dimensiones?



De 2D a 1D.

Dado un set de datos o features en los que x1 y x2 representan cm y pulgadas.

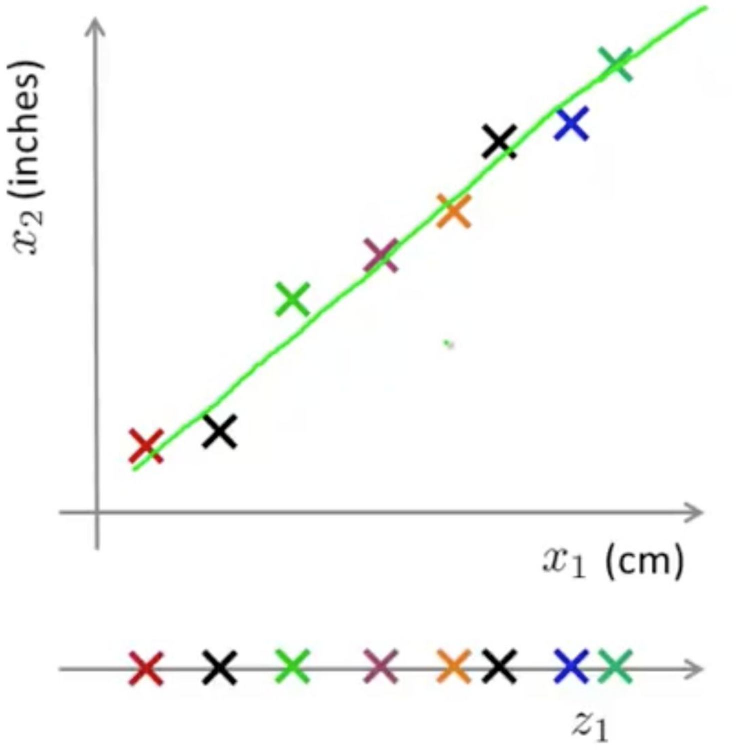
Intuición: ¿Qué es reducir dimensiones?



De 2D a 1D.

Nos gustaria encontrar esta linea/recta a la que todos los datos parecen estar muy cerca y proyectar los mismos sobre ella.

Intuición: ¿Qué es reducir dimensiones?

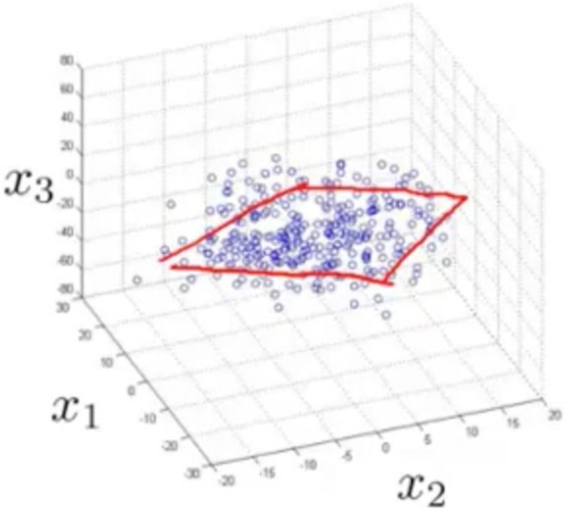


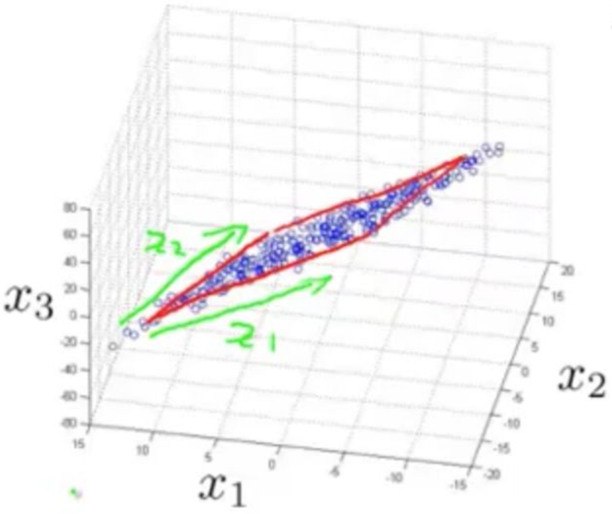
De 2D a 1D.

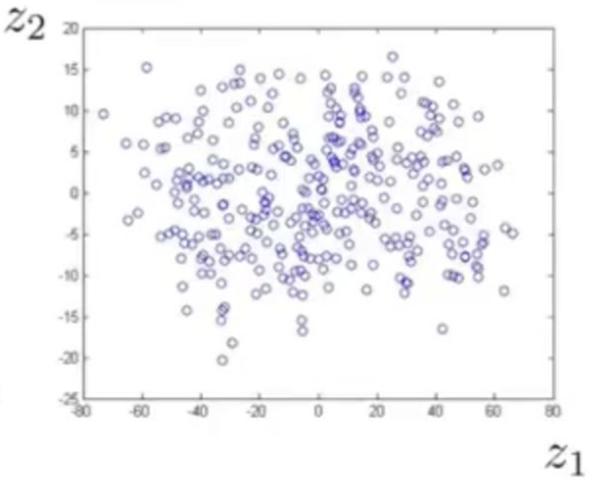
Al realizar la proyección sobre esa recta, obtendremos un nuevo feature z1, que estaría representado por un único valor (la posición en la recta), reduciendo las dimensiones de 2D a 1D.

Descubrimos el Eje de los datos.
Al hacer estoy incurrimos en un poco d error ya que los puntos no están exactamente en la linea.

Intuición: ¿Qué es reducir dimensiones?







De 3D a 2D

Conceptualmente es similar proyectando sobre un plano.

SVD (Singular Value Decomposition)

- Toda matriz se puede descomponer como el producto de tres matrices:

$$A_i = U_i * \Sigma * V^T$$

V es una base de r vectores en n dimensiones

∑ es una matriz diagonal con los valores singulares ordenados de mayor a menor

U indica el coeficiente por el cual multiplicar cada vector de V para reconstruir los datos originales

SVD: Aproximación de Rango k

- Sabemos que los vectores de V están ordenados según su importancia.
- Podemos tomar los k primeros vectores para representar los datos.

La SVD nos da la mejor aproximación de rango k posible a la matriz original

Dimensión intrínseca de los Datos

- La clave es Σ que contiene los valores singulares ordenados de mayor a menor.
- Podemos calcular la energía que conservamos usando los k valores singulares más significativos:

$$\sum (\text{valores singulares})^2$$

Dimensión intrínseca de los Datos

- Visualmente también podemos graficar los valores singulares e identificar para qué valor de k hace un codo.
- En el ejemplo tendríamos que probar con k=3 a 5

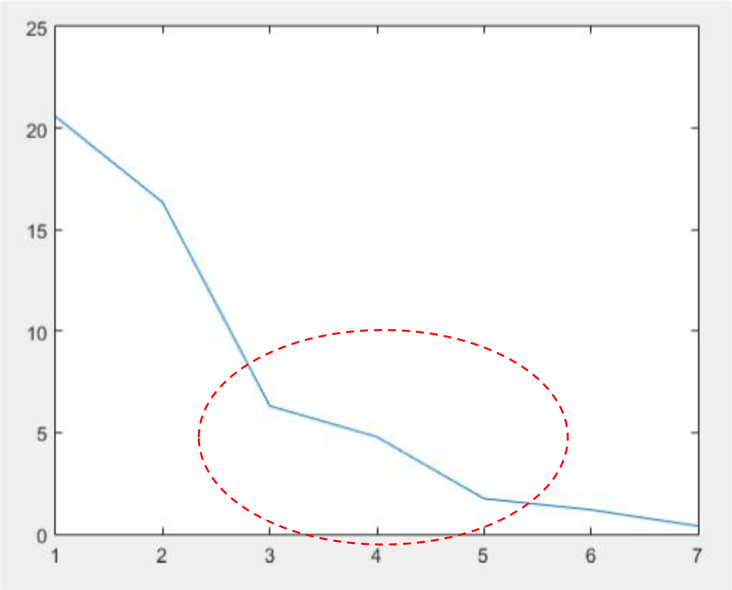
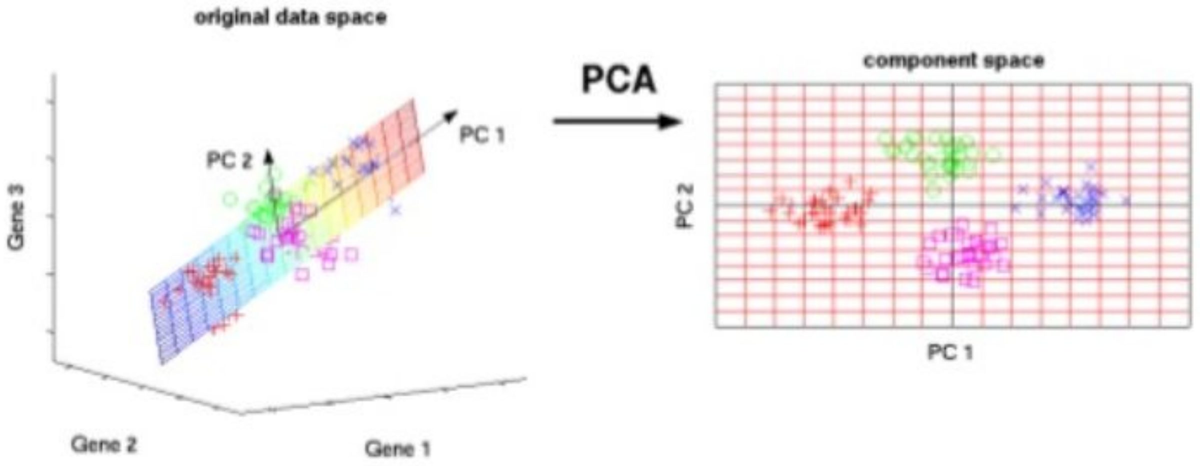


Figure 9.3: Valores Singulares de A

PCA

- Puede ser vista como una rotación del espacio, para encontrar un eje que mejor exprese la variabilidad de los datos.



PCA

PCA y SVD buscan preservar la varianza de los datos

En qué consiste el método:

- Normalizar los datos para centrarlos
- Calcular la matriz de Covarianza
- Calcular autovalores y Autovectores de la matriz de Covarianza

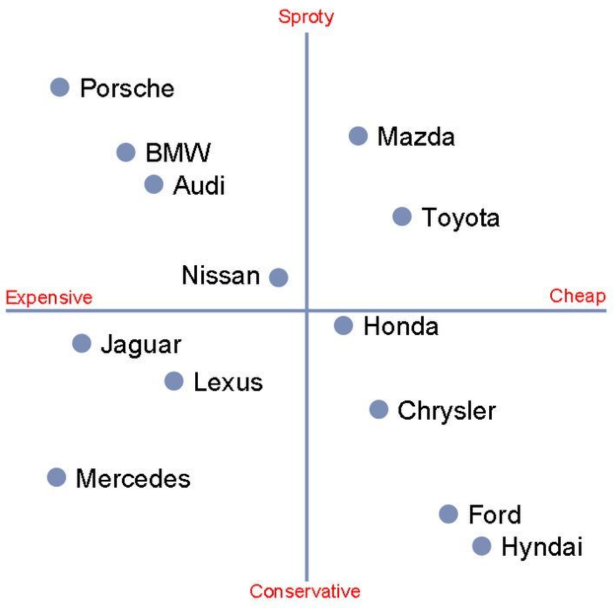
PCA = SVD (normalizada)

MDS (Multidimensional Scaling)

- Contamos con una matriz de distancias
- Buscamos obtener las coordenadas que respeten esas distancias

D -> Distancias

X -> puntos que queremos conocer



MDS

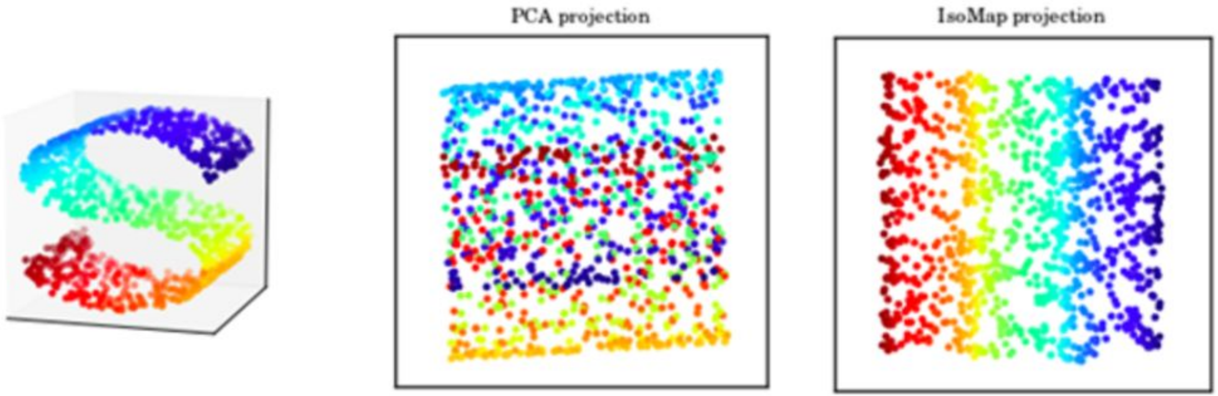
En qué consiste el método:

- Partimos de la matriz de distancias
- Elevamos la matriz de distancias al cuadrado
- Centramos la matriz para que tanto filas como columnas tengan promedio cero
- Calculamos la SVD de la matriz
- Las primeras q columnas de U nos dan las coordenadas de nuestros puntos (en q dimensiones)

ISOMAP

NO LINEAL

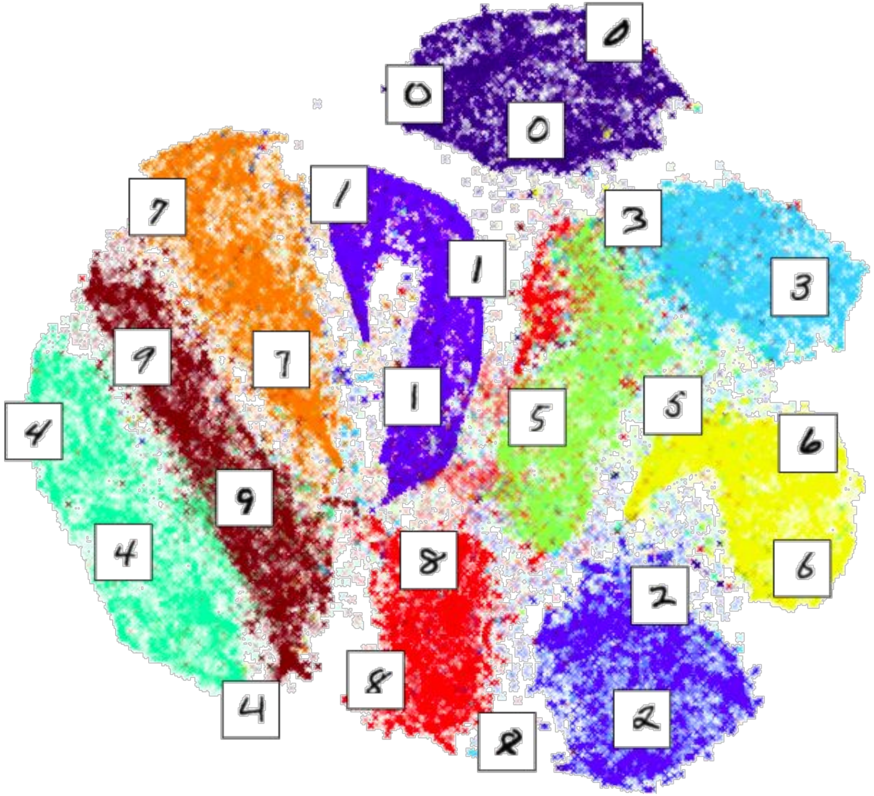
- Partimos de la construcción de un grafo no dirigido
    - Cada punto estará conectado con los k vecinos mas cercanos (KNN)
- Calculamos las distancias (Floyd-Warshall)
- Utilizamos MDS para obtener una representación en dos (o tres) dimensiones



T-SNE

- Partimos calculando la probabilidad de que un punto sea vecino de otro

- Buscaremos que, si dos puntos son cercanos en el espacio original, lo sean también en el reducido



T-SNE

- Probabilidad de que un punto sea cercano a otro:

$$p_{i|j} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k|| / 2\sigma_i^2)}$$

- Probabilidad de que dos puntos en el espacio original sean cercanos:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$$

- Probabilidad de que dos puntos mapeados en el nuevo espacio sean cercanos:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2) - 1}{\sum_{k \neq i} (1 + ||y_k - y_i||^2) - 1}$$

T-SNE

LE y T-SNE buscan que los puntos que estaban cerca, sigan cerca.

- Buscamos que puntos que originalmente estaban cerca, sigan cerca en el nuevo espacio (si pij es un valor cercano a 1 entonces qij sea un valor cercano a 1 y si pij es un valor cercano a cero entonces qij puede quedar libre)
- Utilizaremos la divergencia de Kullback-Leibler:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Buscamos minimizarla.

Teorema fundamental de la dimensionalidad

No siempre es conveniente reducir la dimensionalidad de un set de datos