Reducción de dimensión

Aprendizaje Automático

Juan David Martínez
jdmartinev@eafit.edu.co

Agenda

- Reducción de dimensión
- Análisis de componentes principales (PCA)
- Otras formas de reducción de dimensión

Reducción de dimensión

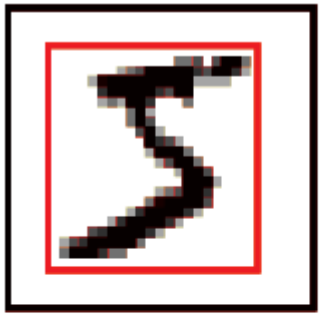
Muchos problemas de **Machine Learning** implican **millones de características** para cada ejemplo (instancia de entrenamiento)

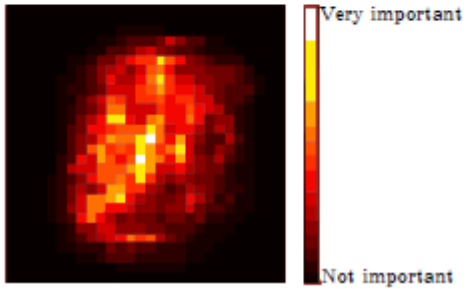
- Extremadamente lento
- Difícil de encontrar una buena solución

Este problema se conoce como la maldición de la dimensionalidad

Reducción de dimensión

Es posible reducir considerablemente el número de características

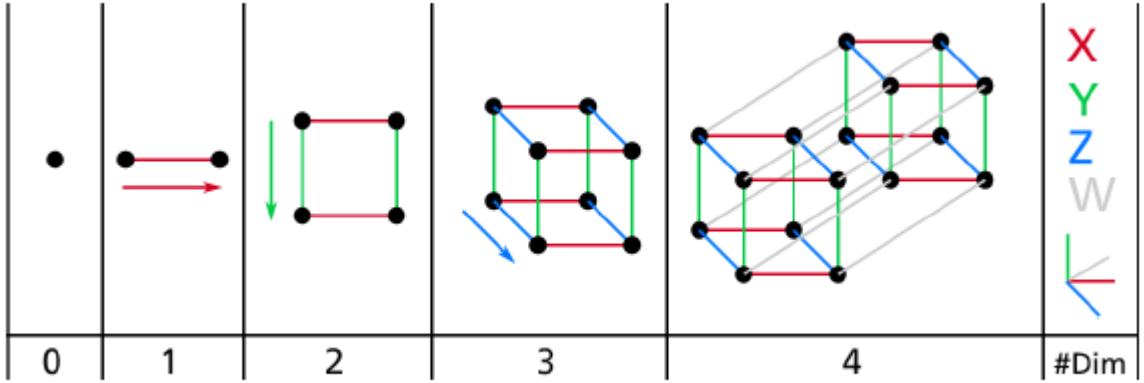




- Los píxeles en el borde generalmente son blancos
- Se pueden eliminar sin perder mucha información
- En el proceso de reducción de dimensión **se pierde algo de información**
- Filtra ruidos y detalles innecesarios y puede mejorar el rendimiento
- Acelera el entrenamiento y facilita la visualización

La maldición de la dimensión

Estamos acostumbrados a vivir en **tres dimensiones,** la intuición nos falla cuando intentamos imaginarnos un espacio de alta dimensión

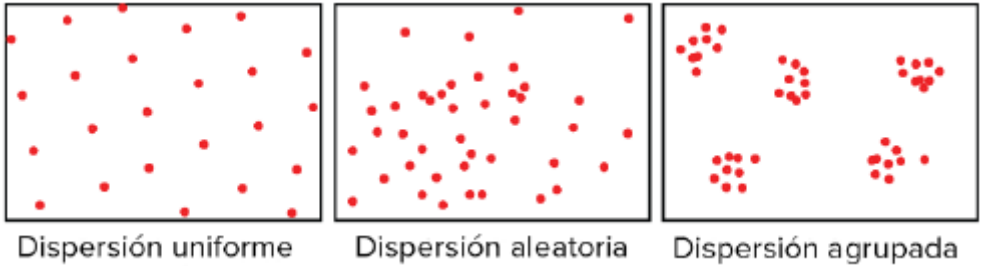


La maldición de la dimensión

- En un cuadrado de lado 1, la distancia entre dos puntos será en promedio 0,52.
- En un cubo de lado 1, la distancia promedio será 0,66
- En un hipercubo de lado 1 de 1 millón de dimensiones, la distancia promedio será de 408,25. Es decir, la mayoría de muestras están muy lejos una de la otra
- ¡Entre más dimensiones, mayor riesgo de sobre-entrenamiento!
- Solución: Aumenta el tamaño de la muestra.
- Problema: El número de muestras requeridas para alcanzar una densidad dada crece exponencialmente con el número de dimensiones.
- Con 100 características, se necesitarían más instancias que átomos en el universo
- Solución: Reducir el número de características

Enfoques principales: Proyección

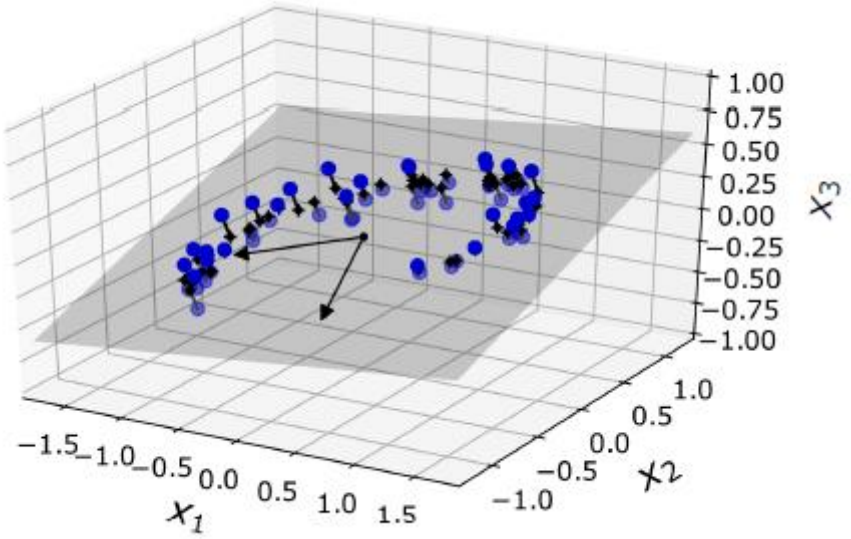
- Generalmente los datos no se distribuyen uniformemente en todas las dimensiones

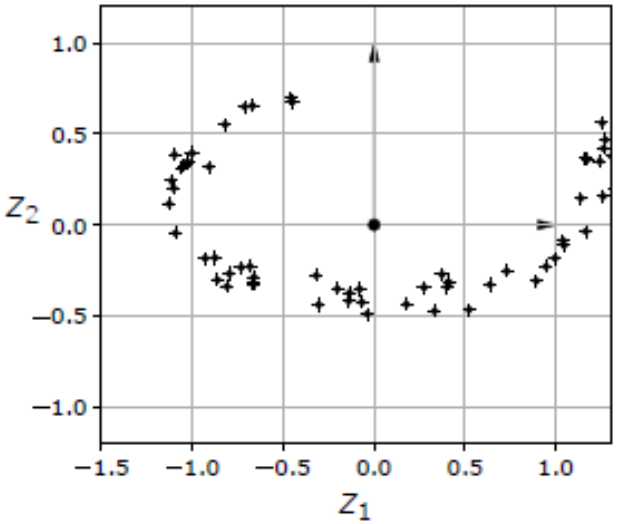


- Muchas características son casi constantes mientras que otras están altamente correlacionadas
- Las muestras se encuentran dentro o cerca de un subespacio de dimensiones mucho más bajas

Proyección: Ejemplo 1

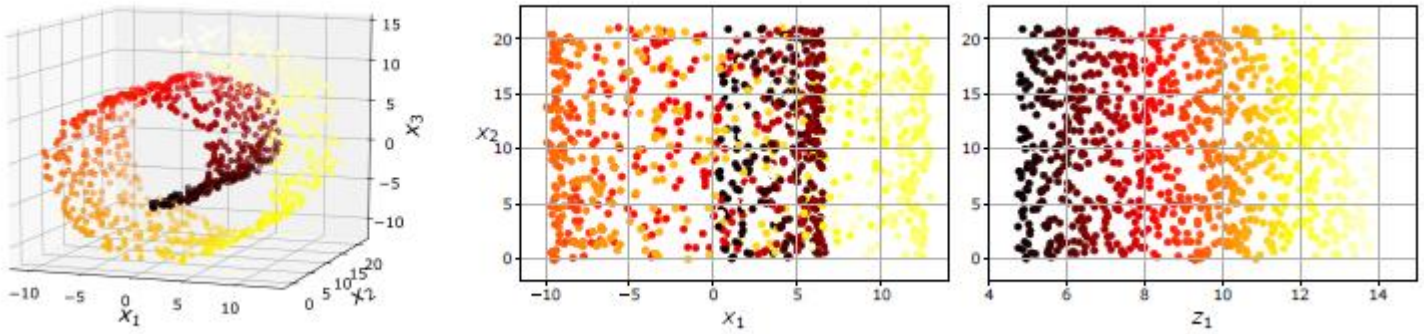
Todas las muestras se encuentran cerca de un plano de menor dimensión (2D)





Proyectamos todas las muestras al nuevo espacio de representación z_1 y z_2

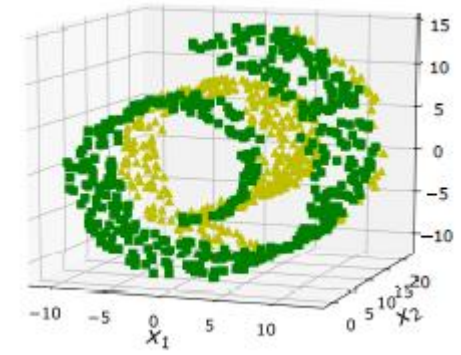
Proyección: Ejemplo 2

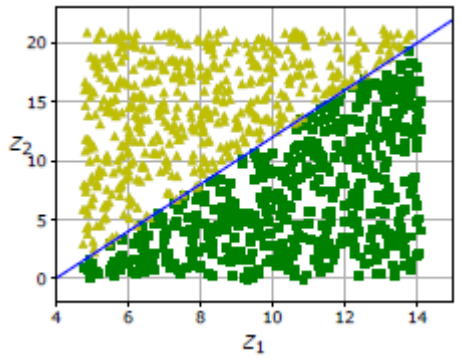


- La proyección en un plano (por ejemplo quitando x_3) aplastaría capas del rollo suizo (centro)
- Lo que deseamos es desenrollar el rollo (derecha)

Enfoques principales: Manifold learning

- El rollo suizo es un ejemplo de una variedad 2D
- Una variedad d −dimensional es una parte de un espacio n −dimensional ($d << n$) que localmente se parece a un hiperplano d −dimensional
- Manifold assumption: La mayoría de los conjuntos de datos de alta dimensión se encuentran cerca de un manifold de mucha más baja dimensión
- La tarea a resolver (clasificación, regresión) será más simple en este nuevo espacio

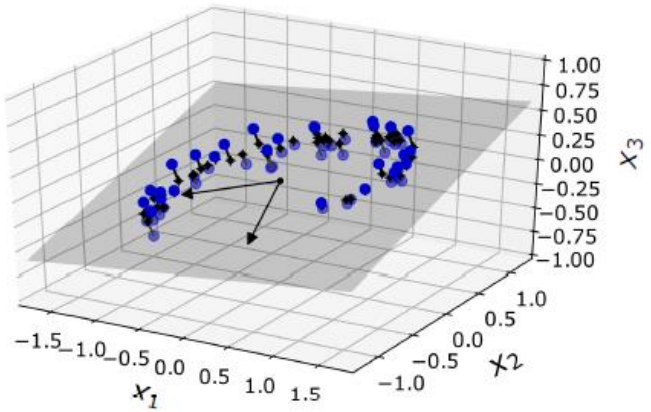




Proyección: PCA

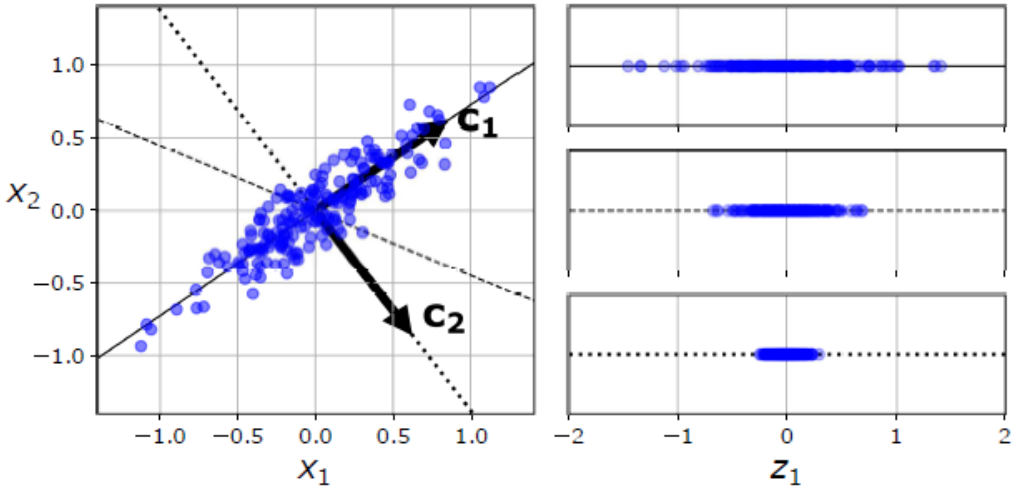
Principal Component Analysis (PCA) es el algoritmo de reducción de dimensionalidad más popular. ¿Cómo funciona?

- Identifica el hiperplano que se encuentra más cerca de los datos
- Proyecta los datos sobre el



PCA: Preservando la varianza

Para proyectar el conjunto de entrenamiento en un hiperplano de dimensiones inferiores, debemos elegir el hiperplano correcto.



- PCA selecciona el eje que preserva la cantidad máxima de varianza de los datos
- También encuentra un segundo eje (ortogonal al primero) que representa la mayor cantidad de varianza restante

PCA: Some math

- Partimos de los datos $\mathbf{x}^{(i)} \in \Re^m, i = 1, \ldots, n$
- La media muestral se define como:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$$

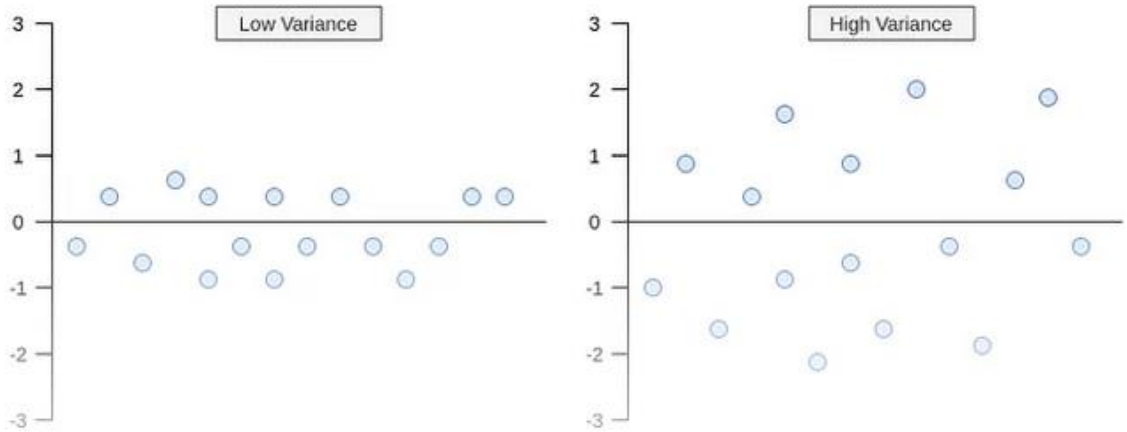
- La covarianza muestral se define como:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^\top$$

Variance

- La varianza de la $j-$ésima característica se define como:

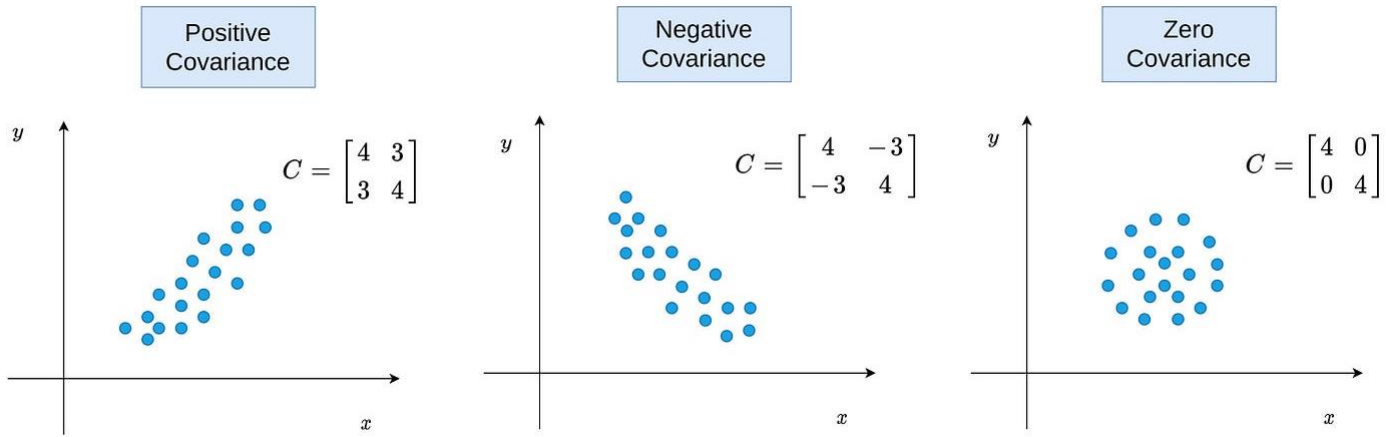
$$s_j = \frac{1}{N}\sum_{i=1}^{n}(x_j^{(i)} - \bar{\mathbf{x}}_j)^2$$



Covarianza

- La covarianza entre las características j y k se define como:

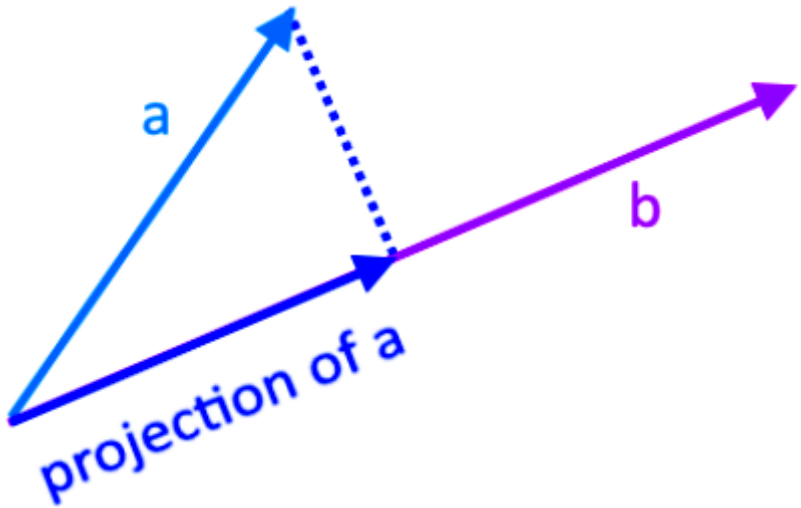
$$s_{jk} = \frac{1}{N} \sum_{i=1}^{n} (x_j^{(i)} - \bar{\mathbf{x}}_j)(x_k^{(i)} - \bar{\mathbf{x}}_k)$$



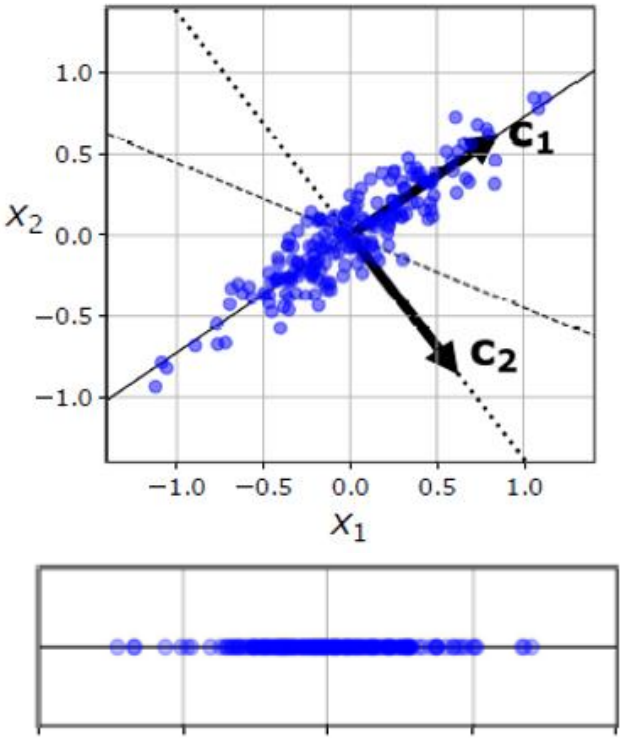
Proyección

- La proyección de **a** en **b** se define como:

$$\text{Proj}_{\mathbf{b}}\mathbf{a} = \frac{< \mathbf{a}, \mathbf{b} >}{||\mathbf{b}||^2}\mathbf{b}$$



Proyección



- La proyección de todos los datos se realiza:

$$\mathbf{z} = \mathbf{X}\mathbf{w}$$

PCA: Some math

- Queremos seleccionar el vector $\mathbf{w}$ que maximice la varianza de los datos proyectados z

$$\max_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}^{(i)}$$

- Dado que los datos tienen media 0, y que $\mathbf{w}$ hace una transformación lineal de los datos, la media de $\mathbf{z}$ también es 0

$$\max_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{w}^\top \mathbf{x}^{(i)})^2$$

$$\max_{\mathbf{w}} \mathbf{w}^\top \left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} \mathbf{x}^{(i)\top} \right) \mathbf{w}$$

$$\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{S} \mathbf{w}$$

PCA: Some math

- Para evitar ambiguedades, definimos que $\mathbf{w}$ sea un vector unitario $||\mathbf{w}||^2 = 1$

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{S} \mathbf{w} \\ \text{s.t.} \quad & ||\mathbf{w}||^2 = 1 \end{aligned}$$

- Utilizando multiplicadores de Lagrange:

$$\max_{\mathbf{w}} \quad \mathbf{w}^\top \mathbf{S} \mathbf{w} + \lambda(1 - \mathbf{w}^\top \mathbf{w})$$

- Igualando la derivada con respecto a $\mathbf{w}$ a 0, tenemos:

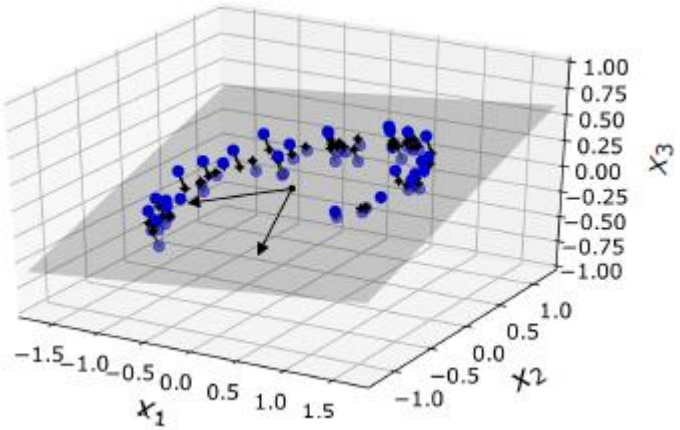
$$\mathbf{S}\mathbf{w} = \lambda \mathbf{w}$$

- Por lo tanto $\mathbf{w}$ y λ son el primer eigenvector y eigenvalor de $\mathbf{S}$

PCA: Proyección

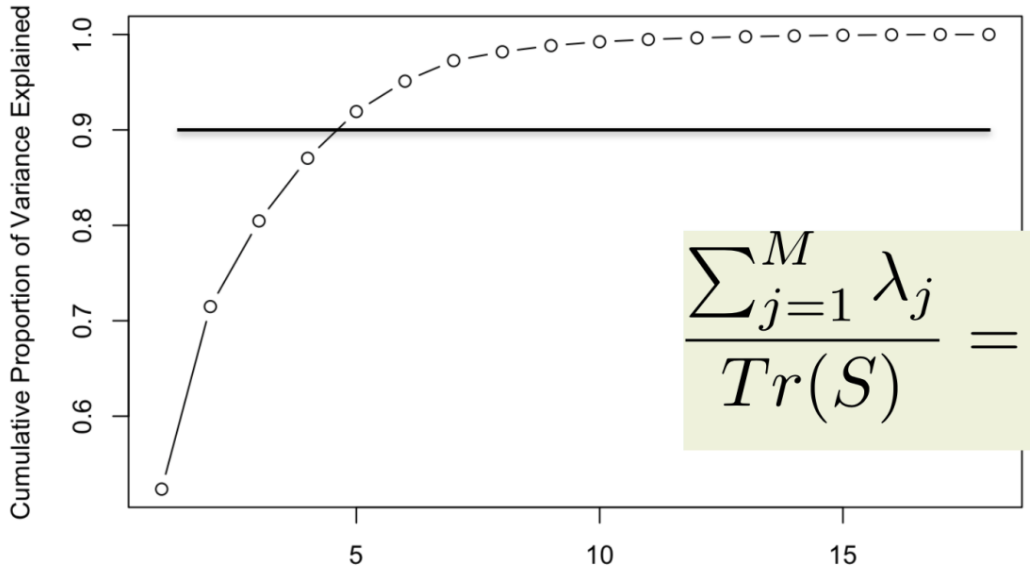
- La siguiente dirección $\mathbf{w}_2$ se puede encontrar maximizando la varianza proyectada en el espacio 1-dimensional ortogonal a $\mathbf{w}$. Esto se logra con el segundo eigenvector de $\mathbf{S}$.
- El hiperplano de d dimensiones se puede formar creando una matriz $\mathbf{W}$ donde cada columna corresponde a los eigenvectores de la matriz $\mathbf{S}$ y proyectando los datos sobre dicha matriz:

$$\mathbf{Z} = \mathbf{XW}$$



PCA: Número de componentes

- Generalmente se escoge un número de diemensiones que alcancen una proporción suficientemente alta de varianza (95%)



The proportion of explained variance

$$\frac{\sum_{j=1}^{M} \lambda_j}{Tr(S)} = \frac{\sum_{j=1}^{M} \lambda_j}{\sum_{j=1}^{D} \lambda_i} > 0.9$$

PCA: Compresión de datos

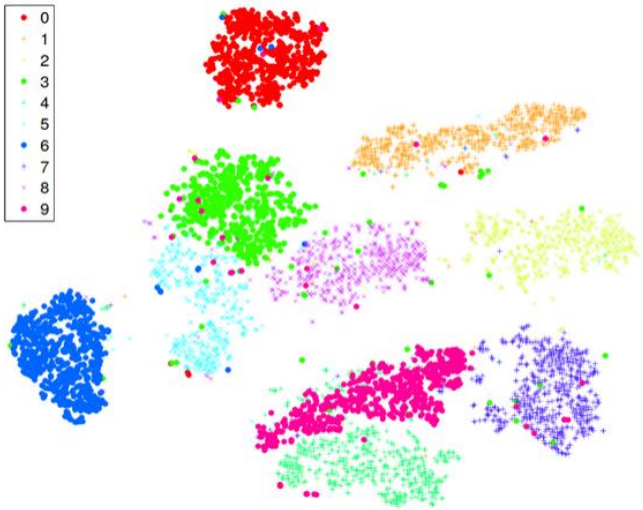
- Después de la reducción de dimensionalidad, el conjunto de datos ocupa mucho menos espacio
- Es posible descomprimir el conjunto de datos reducido de nuevo a las diemnsiones originales aplicando la transformación inversa de la proyección. Esta reconstrucción perderá algo de información.

$$\hat{\mathbf{X}} = \mathbf{Z}\mathbf{W}_d^\top$$

$$e = ||\mathbf{X} - \hat{\mathbf{X}}||^2$$

t-Distributed Stochastic Neighbor Embedding (t-SNE)

- Reduce la dimensionalidad al tratar de mantener instancias similares cerca e instancias diferentes lejos
- Se utiliza principalmente para visualizar datos ya que no genera una función de mapeo



Visualizing Data using t-SNE

Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

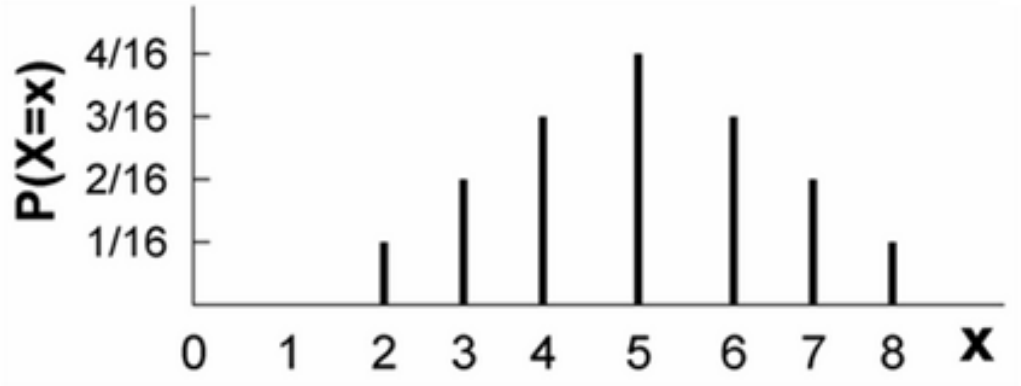
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Editor: Yoshua Bengio

Softmax

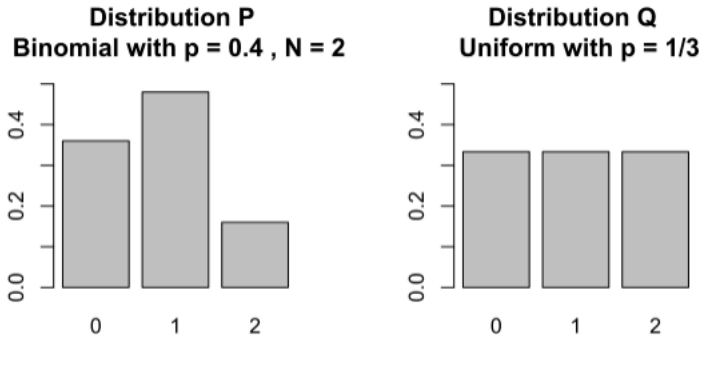
−0.1	0.02
3.8	0.91
1.1	0.06
−0.3	0.01

$$z_i = \frac{e^{z_i}}{\sum_{i=1}^{N} e^{z_i}}$$



Divergencia de Kullback–Leibler

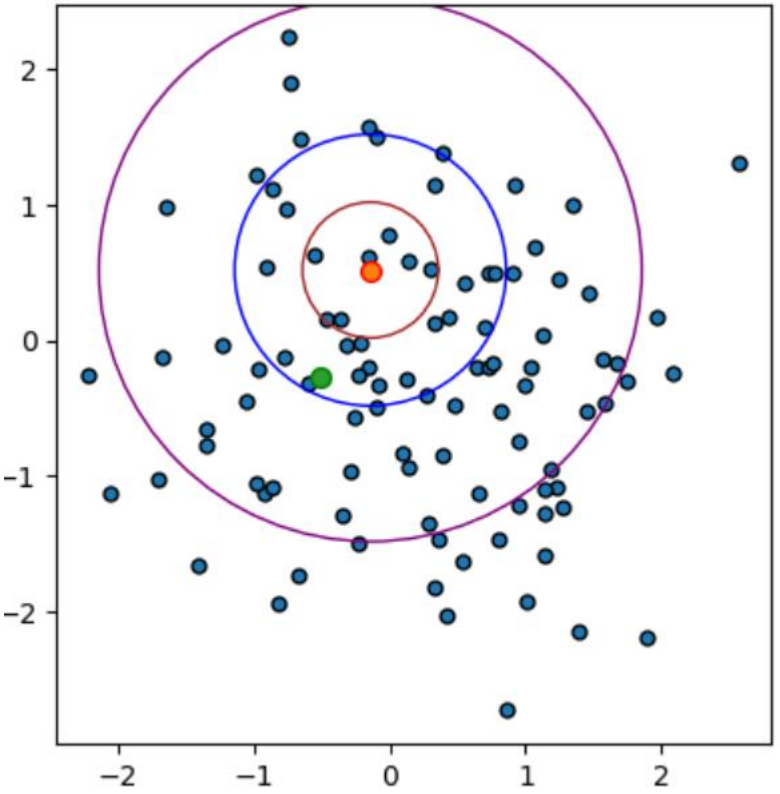
$$D_{KL}(P||Q) = \sum_{i=1}^{N} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

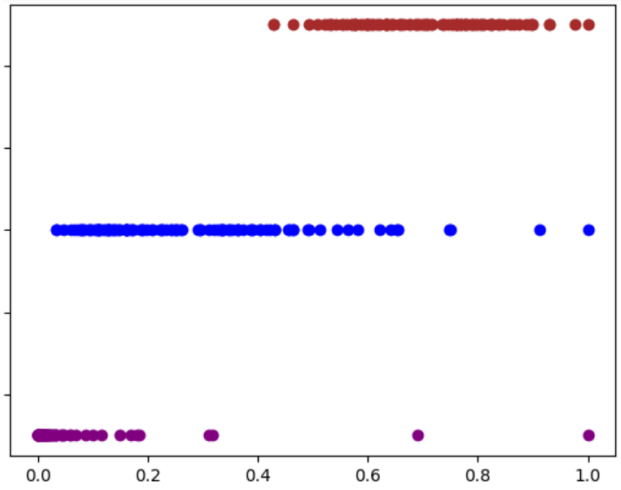


$$D_{KL}(P||Q) \approx 0.085$$

$$D_{KL}(Q||P) \approx 0.097$$

Distancia





$$\text{sim}_{i,j} = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2\right)$$

```
s^2 = 1, d = 0.8780828489462678
s^2 = 2, d = 0.21952071223656694
s^2 = 0.5, d = 3.512331395785071
```

tSNE

- Dado un conjunto de n objetos de alta dimensión, tSNE calcula las probabilidades $p_{i|j}$ que son proporcionales a la similitud de los objetos i y j:

$$p_{i|j} = \exp\left(\frac{-\frac{1}{2\sigma_j^2}||\mathbf{x}_i - \mathbf{x}_j||^2}{\sum_{k\neq i} -\frac{1}{2\sigma_j^2}||\mathbf{x}_i - \mathbf{x}_k||^2}\right)$$

- La similitud $p_{i|j}$ es la probabilidad condicional de que $\mathbf{x}_i$ eligiría a $\mathbf{x}_j$ como vecino en ese espacio de alta dimensión
- Para facilitar la optimización, volvemos simétrica la medida

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

- Y las probabilidades cuando $i = j$ se establecen como 0

tSNE

- tSNE tiene como objetivo aprender un mapa $d-$dimensional $\{\mathbf{y}_i \in \Re^d\}$ (generalmente d=2) que refleja las similitudes p_{ij} lo mejor posible
- Con este fin, mide similitudes q_{ij} entre dos puntos en el nuevo espacio de representación de la siguiente forma:

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq i}(1 + ||\mathbf{y}_i - \mathbf{y}_k||^2)^{-1}}$$

- Se usa la distribución Student-t de cola pesada para medir similitudes entre puntos de baja dimensión. También $q_{ii} = 0$

tSNE

- Las ubicaciones de los puntos $\mathbf{y}_i$ se determinan minimizando la divergencia KL entre las distribuciones P y Q:

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

- La minimización se hace usando gradiente descendente, calculando la derivada de la divergencia con respecto a cada uno de los puntos $\mathbf{y}_i$