Introducción a Machine Learning

Sesión 4: Aprendizaje No Supervisado

Ronald Cárdenas Acosta

Agosto, 2016

Aprendizaje No Supervisado

- Data de entrenamiento: $x^1, x^2, ..., x^N$
- Objetivo: encontrar agrupaciones o estructuras abstractas en la data
- Forma probabilística: $p(x|parametro)$
- Aplicaciones
  - Clustering
  - Aprendizaje de Hiperplanos (Manifold Learning)
  - Descomposición de señales
  - Reducción de dimensionalidad
  - Detección de outliers
  - entre otros

Reducción de Dimensionalidad

- Objetivo: inferir $Z = T(X)$, donde $[Z] = NxK$ y $[X] = NxM$, $K << M$
- Usado en
  - Compresión de data
  - Visualización de data ($K = 1, 2, 3$)
  - Optimización en aprendizaje supervisado
- Algoritmos mas usados
  - Principal Component Analysis (PCA)
  - Singular Value Decomposition (SVD)
  - Factorización por matrices no negativas (NNMF)
  - Independent Component Analysis (ICA)

Outline

Analisis de Componentes Principales

Definición

Consiste en proyectar la data linealmente a un espacio vectorial que conserve la mayor cantidad de varianza posible.

- Se minimiza el error de reconstrucción: $\sum_{i=1}^{N}(x^i - z^i)^2$
- Algoritmo requiere que la data este centrada ($\hat{x} = 0$) y escalada.

Ejemplo: 2D a 1D

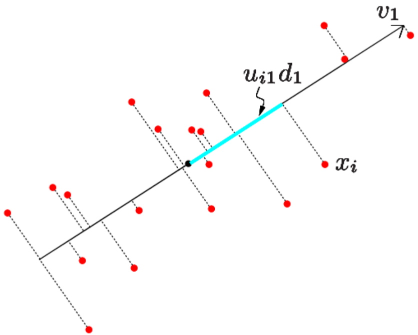
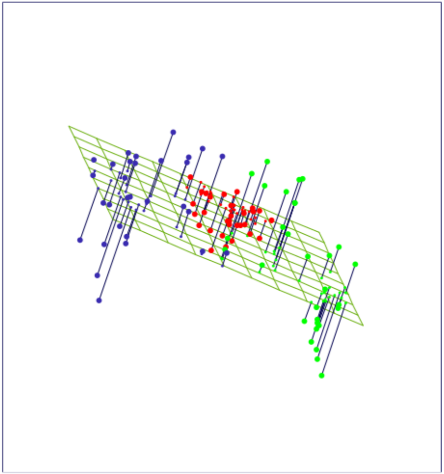


Figure: PCA: Proyección de dos dimensiones a una

Ejemplo: 3D a 2D



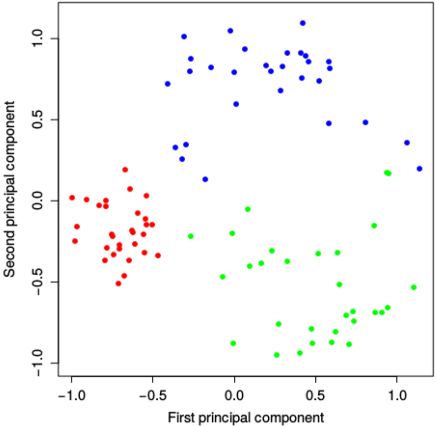


Figure: PCA: Proyección de tres dimensiones a dos

PCA: Algoritmo

- Preprocesamiento
  - Centrar data: $x_j = x_j - \mu_j$,μ_j : media de característica j
  - Si las características estan en diferente escala: $x_j = x_j/\sigma_j$, σ_j: desviación estándar de característica j
- Calcular matriz de covarianza
  $\Sigma = \frac{1}{N}X^T.X = \frac{1}{N}\sum_{i=1}^{N}(x^i)^T.x^i$
- Calcular los eigen-vectores (via SVD)
  $\Sigma \approx U.D.V$,
  $U[MxM]$ contiene un eigen-vector por columna
- $z^i = U_K.x^i$, U_K contiene las primeras K columnas de U
  $Z = U_K.X$

Cómo escoger K

$$\frac{\frac{1}{N}\sum_{i=1}^{N} \|x^i - U_K^T.z^i\|^2}{\frac{1}{N}\sum_{i=1}^{N} \|x^i\|^2} \leq \eta \tag{1}$$

- η: porcentaje de varianza perdida en aproximación
- Escoger menor K que cumpla con desigualdad

Outline

Clustering

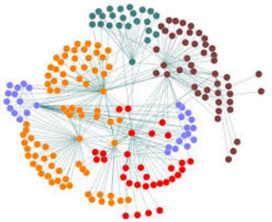
Objetivo

Segmentar la data en grupos o "clusters" de tal forma que un dato esté más relacionado a los de su mismo cluster que a los de otro.

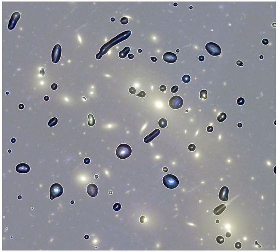
- La agrupación se basa en la definición de "similaridad" usada, por ejemplo
  - Para redes o grafos, cantidad de nodos en el camino que los conecta
  - Para distribuciones de frecuencias (conteo de palabras), metricas de Teorías de Información (KLD, MI)
  - Para casos generales, distancia euclideana, coseno, entre otros.

Clustering: Aplicaciones

(a) Segmentación de mercado



(b) Analisis de redes sociales



(c) Astronomía. Analisis de estrellas y galaxias.

Outline

1. Aprendizaje No Supervisado
2. Reducción de Dimensionalidad
   - Principal Component Analysis
3. Clustering
   - Planteamiento
   - Tipos de clustering
   - Métodos de Evaluación
   - Algoritmo KMeans

Tipos de Clustering

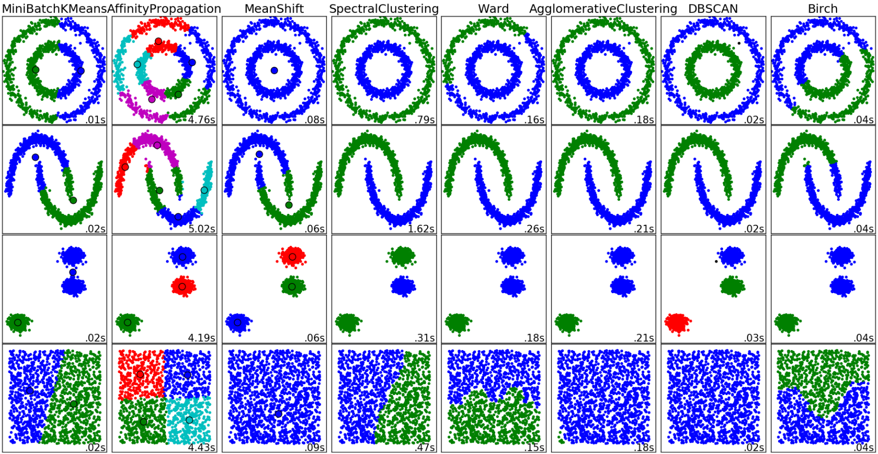


Figure: Tipos de clustering

Outline

Métodos de Evaluación

- Si se conoce el verdadero grupo al que pertenece cada muestra
  - Rand Index
  - Información Mutua (MI, KLD)
  - Homogeneidad y Completividad
- Si no
  - Suma de distancias al centroide en cada cluster

Métricas de evaluación: [Adjusted] Rand Index

Sea C la asignación conocida de grupos y K la del clustering

$$RI = \frac{a + b}{C_2^N}$$

Donde:

- a: numero de pares que estan en el mismo cluster en C y en K
- b: numero de pares que estan en diferentes clusters en C y en K

Normalizando por chance:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

Métricas basadas en Información Mutua

Miden la similitud entre los dos grupos de asignaciones de cluster (real y estimado) mediante Información Mutua.

$$MI(C,K) = \sum_{i=1}^{|C|}\sum_{j=1}^{|K|} P(i,j) log(\frac{P(i,j)}{P(i).P(j)})$$

Donde:

- $P(i) = |C|/N$
- $P(j) = |K|/N$
- $P(i,j) = |C \cap K|/N$

Métricas basadas en pertenencia

- Homogeneidad: grado en el que cada cluster contiene solo miembros de una clase

$$h = 1 - \frac{H(C|K)}{HC)}$$

Donde:
  - $H(C|K) = -\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{N} log(\frac{n_{c,k}}{N})$
  - $H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{N} log(\frac{n_c}{N})$
- Completividad: grado en el que todos los miembros de una clase son asignados a un mismo cluster

$$c = 1 - \frac{H(K|C)}{H(K)} \quad (2)$$

- V-measure: media armónica entre h y c:

$$v = 2\frac{h.c}{h + c} \quad (3)$$

Outline

Algoritmo KMeans

- Separa la data en K grupos disjuntos de igual varianza
- Minimiza criterio de *Inercia* o *Suma de Cuadrados dentro del cluster*
- Cada cluster esta descrito por su centroide μ_j, de la forma:

$$\sum_{i=0}^{N} \min_{\mu_j \in C} (\|x_i - \mu_j)$$

- La inercia asume que los clusters son convexos e isotrópicos, lo cual no siempre es el caso

KMeans: algoritmo

- Incializar aleatoriamente los K centroides $\mu_1, \mu_2, ...\mu_K \in \Re^M$
- Iterar por *num iteraciones*
    - for $i = 1 -> N$
      c^i = index (de 1 a K) del centroide mas cercano a x^i
    - for $k = 1 -> K$
      μ_k = promedio de puntos asignados a cluster k