Ingeniería del Conocimiento CIF-8458

Introducción al Aprendizaje No Supervisado

Carlos Valle Vidal

Segundo Semestre 2023

Clustering

- El analisis de clústeres es la busqueda de distintos grupos (clúster) en los datos, en los cuales datos del mismo clúster deben ser **similares** y marcadamente **diferentes** de los datos de las otras clases.
- El propósito de ello es descubrir algún patrón que permita discriminar entre los distintos grupos encontrados.
- En general, la similaridad entre los grupos se basa en la distancia.
- Los datos normalmente no tienen etiqueta alguna, pero también se pueden utilizar este tipo de algoritmos para datos etiquetados.

Similaridad

- ¿Cómo saber si dos objetos son similares?
- La distancia entre las características de los objetos permite determinar el grado de similaridad que existe entre ellos.
- Acá es importante tener en cuenta que los atributos pueden ser numéricos o categóricos.

Medidas de distancia para atributos numéricos

- Sean $x = (x_1, \ldots, x_I)^T$ y $y = (y_1, \ldots, y_I)^T$ vectores con I atributos.
- Distancia Euclideana: Es la distancia clásica, y corresponde a la longitud de la recta que une dos puntos en el espacio euclideano. Se conoce también como la métrica L2.

$$d(x, y) = \sqrt{\sum_{i=1}^{I} (x_i - y_i)^2}$$

Medidas de distancia para atributos numéricos (2)

- Distancia de Manhattan: Su nombre hace referencia a lo equivalente de ir recorriendo cuadras para llegar de un punto a otro. Se conoce tambien como la métrica L1.

$$d(x, y) = \sum_{i=1}^{I} |x_i - y_i|$$

Medidas de distancia para atributos numéricos (3)

- Distancia de Minkowski: Corresponde a una generalizacion de las distancias euclideanas (p = 2) y de manhattan (d = 1). Se conoce también como la metrica Lp.

$$d(x,y) = \left(\sum_{i=1}^{I} (x_i - y_i)^p \right)^{1/p}$$

- Distancia de Chebyshev: Es la distancia del máximo y corresponde a la mayor diferencia en valor absoluto de los atributos. Se conoce también como la métrica $L\infty$.

Medidas de distancia para atributos categóricos

- Distancia de Hamming: Compara dos vectores de igual largo y contabiliza la cantidad de atributos distintos.
- Distancia Atributos Multivaluados o distancia de Jaccard: Sean X e Y conjuntos cuyos elementos corresponde a los valores posible de la variable.

$$d(x, y) = \left(\#(X \cup Y) - \#(X \cap Y)\right)$$

Calculando distancias totales

- Combinacion Lineal Convexa de Distancias: La combinación lineal convexa de distancias es también una distancia:

$$d(x,y) = \sum_t \alpha_t d_t\,(x,y),$$

- donde $\sum_t \alpha_t = 1$.

Distancia normalizada

- Distancia Normalizada: Esta expresión se utiliza para escalar el resultado de una distancia al intervalo [0, 1] cuando el máximo es desconocido:

$$d^*(x,y) = \frac{d(x,y)}{1 + d^*(x,y)}$$

- Esta distancia tiene las siguientes propiedades:

1. $0 \leq d(x,y) \leq 1$
2. Si $d^*(x,y) \to \infty$ entonces $d(x,y) \to 1$.

K-means

- Entrada: Número de clusters K.

1. Inicializar los centroides $\mu_1, \mu_2, \ldots, \mu_K \in R^I$
2. Repetir
   1. for $m = 1$ to m do
      1. $c_m \leftarrow \text{argmin}_j \|x_m - \mu_j\|^2$
   2. end for
   3. for $j = 1$ to K do

$$\mu_j \leftarrow \frac{\sum_{x_m : c_m = j} x_m}{\sum_{x_m : c_m = j} 1}$$

   4. End for
3. Hasta convergencia

K-means (2)

- Se debe escoger el número de clusters K
- Al principio los centroides se inicializan con algún criterio, por ejemplo, elegir ejemplos aleatoriamente.
- En cada iteración:
  - Cada ejemplo de entrenamiento se asigna al cluster más cercano comparando la distancia con cada centroide del clúster.
  - Y, los centroides se recalculan como la media de los puntos asignados a ese cluster.
- Como podemos ver cada ejemplo pertenece a un solo clúster (clustering duro), por lo que el algoritmo funciona bien cuando los datos están claramente separados en clusters bien definidos.

Mezcla de Gaussianas

- Un modelo de mezcla de gaussianas (GMM) intenta encontrar una mezcla de distribuciones de probabilidad gaussianas multidimensionales que mejor modelen cualquier conjunto de datos de entrada.
- En el caso más simple, los GMM se pueden usar para encontrar clústeres de la misma manera que k-means.
- Cada cluster es modelado por una gaussiana. Para describirlo formalmente, se define z como una variable latente que puede tomar los valores $1,2 \ldots K$. Entonces:

$$x_m | z = k \sim N(\mu_k, \Sigma_k).$$

Mezcla de Gaussianas (2)

- Como GMM contiene un modelo probabilístico, cada ejemplo tiene probabilidad de pertenencia a cada cluster (clustering blando):

$$p(x_m, z) = p(x_m|z)p(z)$$

- Notemos que el primer factor es una gaussiana, por lo que depende de su promedio y varianza.

Donde $p(z = k) = \phi_k$

- Entonces usando la regla de Bayes:

$$p(z = k\,|x_m) = \frac{p(x_m|z = k)\phi_k}{\sum_{k=1}^{K} p(x_m|z = k)\phi_k}$$

Mezcla de Gaussianas (3)

1. Escoger μ_k, Σ_k y ϕ_k, $\forall k = 1,2, \ldots, K$ (para cada cluster)
2. Repetir
   1. Para cada ejemplo computar $\mathrm{w}_{\mathrm{m}}^{\mathrm{k}} = p(z = k \,|x_m)$.
   2. Para cada cluster actualizar μ_k, Σ_k y ϕ_k.
3. Hasta convergencia

Mezcla de Gaussianas (4)

- Las actualizaciones se computan:

$$\widehat{\phi_k} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}}^{\mathrm{k}}, k = 1, \ldots, K$$

$$\widehat{\mu_k} = \frac{\sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}}^{\mathrm{k}} x_m}{\sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}}^{\mathrm{k}}}, k = 1, \ldots, K$$

$$\widehat{\Sigma_k} = \frac{\sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}}^{\mathrm{k}} (x_m - \mu_k)(x_m - \mu_k)^T}{\sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}}^{\mathrm{k}}}, k = 1, \ldots, K$$

Transformación de características

- Una herramienta que se puede usar, tanto en aprendizaje no supervisado, como supervisado, es transformar los atributos originales de entrada del problema, de manera que los nuevos atributos hagan que el problema sea más fácil de resolver.
- Esto se conoce como extracción de características o **feature extraction**.
- En general feature extraction busca reducir la dimensión del espacio de entrada, y así poder evitar el sobreajuste.
- Veremos ds técnicas para extraer características:
- PCA y t-SNE

Principal component analysis (PCA)

- En español es análisis de componentes principales.
- Es un método estadístico que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información.
- Originalmente tenemos M individuos con I atributos cada uno, es decir el espacio de entrada tiene I dimensiones.
- La idea de PCA es representar cada vector de entrada con z variables o atributos, donde $z < I$.

Valores y vectores propios

- Los vectores propios o eigenvectors son un caso particular de multiplicación entre una matriz y un vector.
- Obsérvese la siguiente multiplicación:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} * \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 * \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

- El vector resultante de la multiplicación es un múltiplo entero del vector original.
- Los vectores propios de una matriz son todos aquellos vectores que, al multiplicarlos por dicha matriz, resultan en el mismo vector o en un múltiplo entero del mismo.

Valores y vectores propios (2)

- Los vectores propios tienen las siguientes propiedades matemáticas:
  - Los vectores propios solo existen para matrices cuadradas y no para todas. En el caso de que una matriz n x n, el número de vectores propios es n.
  - Si se escala un vector propio antes de multiplicarlo por la matriz, se obtiene un múltiplo del mismo eigenvector.
  - Esto se debe a que si se escala un vector multiplicándolo por cierta cantidad, lo único que se consigue es cambiar su longitud pero la dirección es la misma.
  - Todos los vectores propios de una matriz son perpendiculares (ortogonales) entre ellos, independientemente de las dimensiones que tengan.

Valores y vectores propios (3)

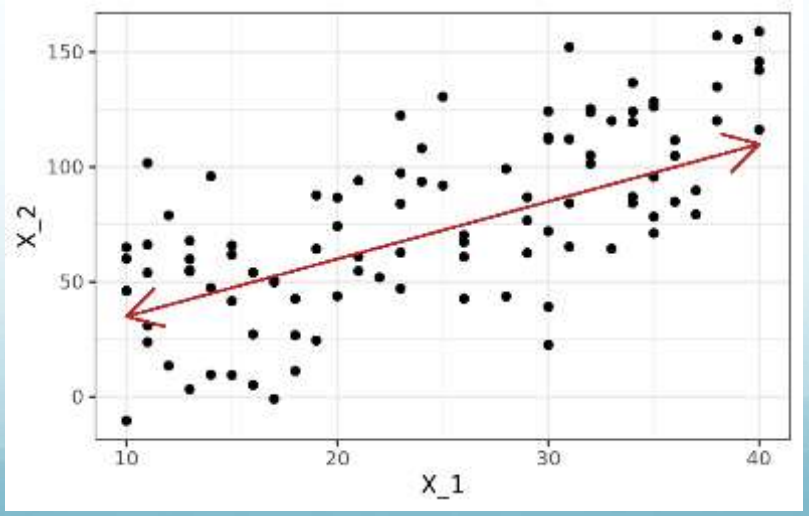
- Debido a esta última propiedad un vector propio puede cambiar su longitud, pero sigue siendo vector propio.
- Por eso, se suelen mantener normalizados (norma 1).
- En el ejemplo anterior, el vector $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ tiene norma
- $\sqrt{3^2 + 2^2}$=$\sqrt{13}$, por lo tanto, dividiendo el vector por su norma se obtiene un vector de norma 1:
- $\begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$

Valores y vectores propios (4)

- Cuando se multiplica una matriz por alguno de sus vectores propios se obtiene un múltiplo del vector original, es decir, el resultado es ese mismo vector multiplicado por un número.
- Al valor por el que se multiplica el vector propio se le conoce como valor propio.
- A todo vector propio le corresponde un valor propio y viceversa.
- Los vectores propios suelen ordenarse de acuerdo al orden ascendente de sus valores propios asociados.
- Es decir, el primer vector propio es el que corresponde al valor propio de mayor valor.

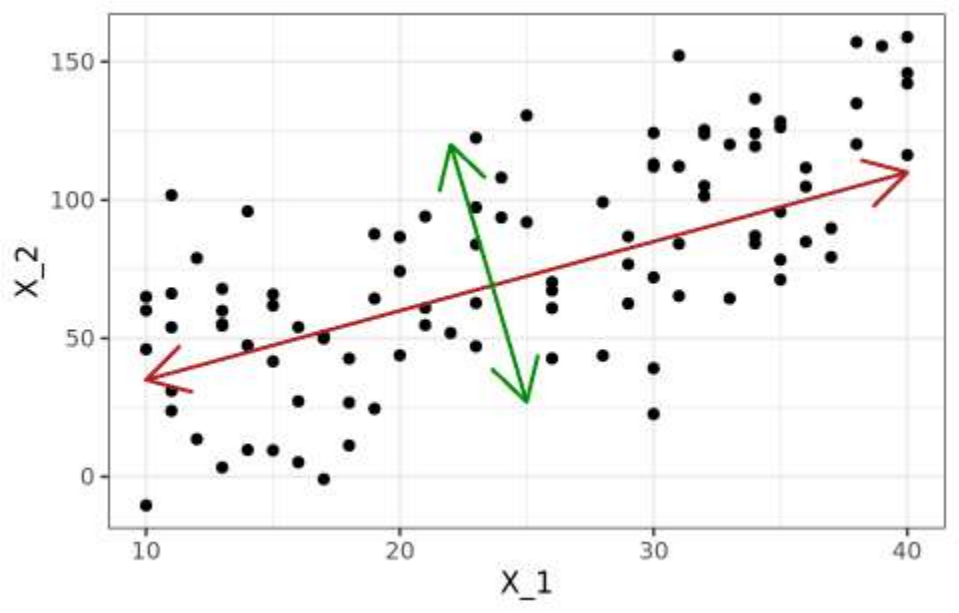
Interpretación geométrica de las componentes principales

- PCA es una técnica que escoge una nueva base ortogonormal para describir los vectores del espacio.
- El primer vector de esa base (llamado componente principal), se escoge como la dirección de mayor varianza de los datos.



Interpretación geométrica de las componentes principales

- Iterativamente se escogen las siguientes componentes principales de manera que minimicen la varianza restante en los datos (las que no han modelado las componentes principales previamente escogidas).



Formulación matemática para obtener las PCA

- Considerando la matriz de datos X (de M filas e I columnas)
- Primero se centra (es decir, a cada casilla se le resta el promedio de la columna a la que pertenece)
- Las proyecciones más relevantes (Z_i) se obtienen como una combinación lineal de las variables originales, por ejemplo, la primera proyección:

$$Z_1 = w_{11}x^{(1)} + w_{12}x^{(2)} + w_{1I}x^{(I)}.$$

- Donde el vector $w_1 = (w_{11}, w_{12}, \dots, w_{1I})$ es el primer vector de la base, y es llamado primera componente principal.

Formulación matemática para obtener las PCA (2)

- Como dijimos anteriormente, queremos que cada componente principal tenga norma uno.
- Por lo tanto, necesitamos que $\|w_p\| = 1, p = 1, \ldots, I$.
- Formalmente, para encontrar la primera componente principal w_1 necesitamos que:

$$\begin{aligned} \text{Maximizar} \quad & \|Xw\|_2^2 \\ \text{Sujeto a} \quad & \|w\|_2^2 = 1 \end{aligned}$$

- $\|Xw\|_2^2$ se expresa matricialmente como $(Xw)^T Xw$.
- Esto es la varianza de la proyección de los datos sobre el vector w.

Formulación matemática para obtener las PCA (3)

- En otras palabras estamos escogiendo la dirección del vector unitario que maximiza la varianza sobre la matriz que contiene los datos (previamente centrados).
- Para encontrar la segunda componente principal debemos

$$\begin{aligned} \text{Maximizar} \quad & \left\| \hat{X} w \right\|_2^2 \\ \text{Sujeto a} \quad & \|w\|_2^2 = 1 \end{aligned}$$

- Donde $\hat{X} = X - X w_1 w_1^T$, es decir le sacamos a los datos la influencia de la primera componente principal.

Formulación matemática para obtener las PCA (4)

- El resultado es que las I componentes principales se obtienen de los vectores propios de la matriz de covarianzas $X^T X$.
- Siendo la i-ésima componente principal el vector propio asociado al i-ésimo valor propio (como ya dijimos se ordenan los valores propios de mayor a menor).

T-SNE

- Es un algoritmo para reducir dimensionalidad que se ejecuta en dos pasos:
  1. Se construye una distribución de probabilidad para los paresde muestras (x_i, x_j) en el espacio original:

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

donde σ_i^2 es la varianza de la gaussiana centrada en x_i.

De esta forma las muestras semejantes reciben alta probabilidad de ser escogidas, mientras que las muestras muy diferentes reciben baja probabilidad de ser escogidas.

T-SNE (2)

2. Se lleva los puntos del espacio de alta dimensionalidad al espacio de baja dimensionalidad, para esto, se crean dos observaciones y_i, y_j que representan a x_i, x_j pero en una dimensionalidad menor. Y se construye en modelo de probabilidad para estos pares y_i, y_j:

$$q_{j|i} = \frac{\exp\left(\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp(\|y_i - y_k\|^2)},$$

- Luego para que y_i, y_j representen fielmente a x_i, x_j se debe minimizar la divergencia Kullback-Leibler entre $p_{j|i}$ y $q_{j|i}$.

T-SNE (3)

- Esto es, minimizar:
- $C = \sum_i KL(p_i||q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$.
- Esta función de costo se puede minimizar con gradiente descendente:
- $\frac{\delta C}{\delta y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{j|i}\right)\left(y_i - y_j\right)$

Perplexity

- Para estimar el valor de la varianza σ_i^2, se usa una búsqueda binaria con una perplexity fijada por el usuario.
- La perplexity puede entenderse como una medida del número de observaciones vecinas que tienen que emplearse en cada estimación local.
- Suele ser recomendable seleccionar valores entre 5 y 50, por defecto su valor es 30 en sklearn.

Manifold

- La reducción de dimensionalidad que consigue el *t-SNE* está basada en el concepto matemático de *manifold*.
- Un *manifold* se define como una superficie *d*-dimensional que reside dentro de un espacio *D*-dimensional, siendo $d < D$.
- Un ejemplo en 3 dimensiones sería el siguiente: Un hilo enredado formando de ovillo, sería un *manifold* 1D en un espacio 3D.
- T-SNE asume que existe un manifold de menor dimensión donde se alojan los datos. En aquellos casos en los que se cumple esta hipótesis, entonces, el *t-SNE* es capaz de proyectar correctamente los datos en esa dimensionalidad inferior.

Manifold (2)

- En la realidad, la asunción de *manifold* no puede cumplirse siempre, de hecho, si así fuese, cualquier set de datos con *N*-dimensiones, podría considerarse como un *manifold* de dimensión *M*, siendo $M < N$.
- Una vez proyectados los datos en ese espacio *M*-dimensional, se podría repetir el proceso sucesivamente hasta concluir que los datos originales de dimensión *N* pertenecen a un *manifold* de dimensión 1.
- Dado que esto no es cierto para todos los sets de datos, la asunción de *manifold no siempre se cumple*. Es en estos casos, en los que *t-SNE* no resulta útil reduciendo la dimensionalidad.

Limitaciones y desventajas

- Está diseñado para reducir los datos a 2 o 3 dimensiones. No debe aplicarse si la reducción de dimensionalidad se hace a espacios d > 3.
- La baja interpretación de su algoritmo (black box), es decir, permite visualizar muy bien datos, pero no genera, por ejemplo, una serie de ecuaciones fácilmente interpretables como hace el PCA.

Limitaciones y desventajas (2)

- No es un proceso totalmente determinista, por lo que, a pesar de emplear los mismos datos, los resultados pueden variar.
- Su algoritmo no es incremental, en otras palabras, no puede aplicarse sobre un set de datos y después actualizarlo con unas pocas observaciones nuevas. Se tiene que ejecutar de nuevo todo el algoritmo incluyendo todas las observaciones (las viejas y las nuevas).