

Técnicas de Aprendizaje No Supervisado

A. M. Alvarez-Meza, Ph.D.

D. F. Collazos-Huertas, Ph.D(c)

amalvarezme@unal.edu.co, dfcollazos@unal.edu.co

Minería de datos

Departamento de Matemáticas y Estadística

Universidad Nacional de Colombia-sede Manizales



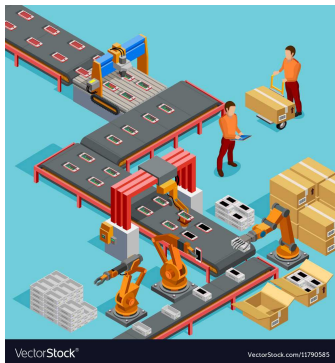
Contenido

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número optimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número óptimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

Aprendizaje no supervisado

La mayoría de las aplicaciones de **Machine Learning** hoy en día se basan en el aprendizaje supervisado, **la gran mayoría de los datos disponibles no están etiquetados**.



Tenemos las características de entrada **X** , pero no tenemos las etiquetas **y**

Aprendizaje no supervisado: Ejemplo

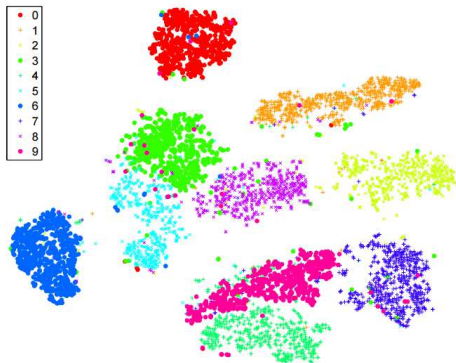
- **Objetivo:** Deseamos crear un sistema que tome algunas fotos de cada artículo en una línea de producción y detecte qué artículos están defectuosos.
- **Solución:** Podemos crear **un sistema que tomará fotos automáticamente, y esto proporcionaría miles de fotos todos los días.** Así construimos un conjunto de datos grande en pocas semanas.
- **Drawback:** **¡No hay etiquetas!** Si entrenamos un clasificador binario que prediga si un artículo es defectuoso o no, debemos etiquetar cada imagen como "*defectuosa*" o "*normal*". Esta es una tarea **larga, costosa y tediosa.**
- **Posible resultado:** El conjunto de datos etiquetado será bastante pequeño y el **rendimiento del clasificador será decepcionante.**

Contenido

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número óptimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

Clustering

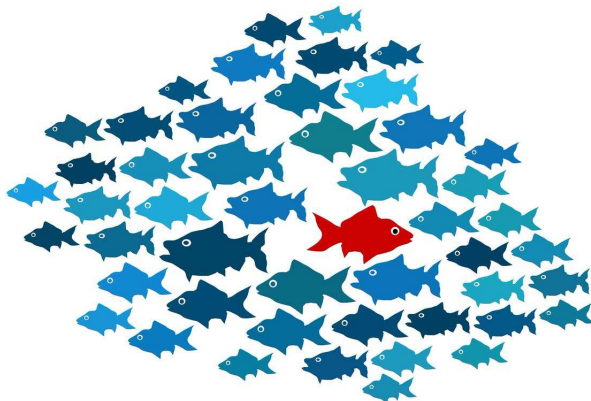
El objetivo es **agrupar instancias/muestras similares en grupos**.



Esta es una gran herramienta para el **análisis de datos, segmentación de imágenes, aprendizaje semi-supervisado, reducción de dimensionalidad** y más.

Anomaly detection

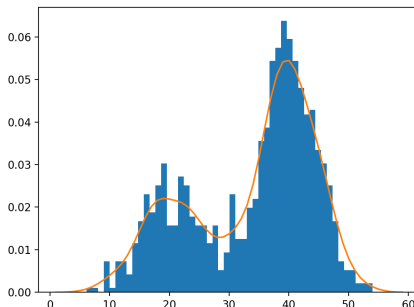
El objetivo es **aprender cómo se ven los datos “normales”** y usar esto **para detectar instancias anormales**.



Como por ejemplo, elementos defectuosos en una línea de producción o
una nueva tendencia en una serie de tiempo.

Density estimation

Consiste en **estimar la función de densidad de probabilidad (PDF)** del **proceso aleatorio que generó el conjunto de datos**.

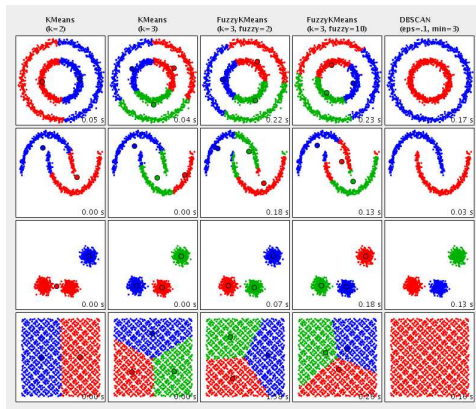


Se usa para la detección de anomalías: **es probable que las instancias ubicadas en regiones de muy baja densidad sean anomalías**. También es útil para el análisis y visualización de datos

Contenido

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering**
- 4 K-Means
- 5 Encontrar el número optimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

Clustering

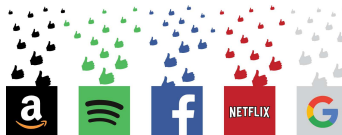


Clustering es la tarea de identificar instancias similares y asignarlas a agrupaciones, es decir, grupos de instancias similares. Al igual que en la clasificación, cada instancia se asigna a un grupo. Sin embargo, esta es una tarea sin supervisión.

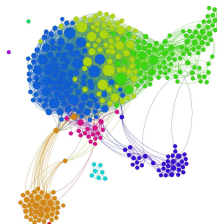


Clustering: Aplicaciones I

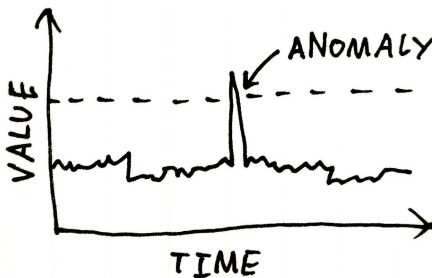
- 1 **Segmentación de clientes:** puede agrupar a sus clientes en función de sus compras, su actividad en su sitio web, etc.



- 2 **Análisis de datos:** al analizar un nuevo conjunto de datos, a menudo es útil descubrir primero grupos de instancias similares.

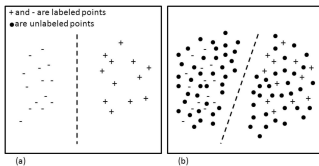


- ③ **Reducción de dimensión:** El vector de características x de cada instancia se puede reemplazar por el **vector de sus afinidades de cluster**.
- ④ **Detección de anomalías:** cualquier instancia que tenga una **baja afinidad con todos los grupos** probablemente sea una anomalía.



Clustering: Aplicaciones III

- 5 **Aprendizaje semi-supervisado:** si solo tiene unas pocas etiquetas, puede realizar la agrupación y propagar las etiquetas a todas las instancias en el mismo cluster.



- 6 **Segmentación de imagen:** al agrupar los píxeles de acuerdo con su color, y luego reemplazar el color de cada píxel con el color medio de su grupo.



Contenido

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means**
- 5 Encontrar el número optimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

K-Means: Definición

- El algoritmo **K-Means** es un algoritmo simple **capaz de agrupar este tipo de conjunto de datos de manera muy rápida y eficiente**, a menudo en solo unas pocas iteraciones.
- Fue propuesto por *Stuart Lloyd* en los Laboratorios Bell en 1957 como una **técnica para la modulación de código de pulso**, pero solo se publicó fuera de la compañía en 1982, en un artículo titulado "*Cuantización de mínimos cuadrados en PCM*".

En 1965, Edward W. Forgy había publicado prácticamente el mismo algoritmo, por lo que K-Means a veces se conoce como **Lloyd-Forgy**.



Algoritmo K-Means

El algoritmo K-Means es uno de los algoritmos de agrupación más **rápidos**, pero también uno de los más **simples**:

- 1 Primero inicialice k centroides al azar: se seleccionan al azar k instancias distintas del conjunto de datos y los centroides se colocan en sus ubicaciones.
- 2 **Repita hasta la convergencia** (es decir, hasta que los centroides dejen de moverse): Asigne cada instancia al centroide más cercano.
- 3 Actualice los centroides para que sean la media de las instancias que se les asignan.

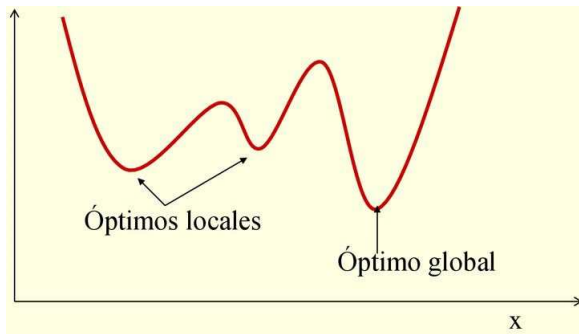
Algoritmo K-Means en otras palabras

- Suponga que **le dieron los centroides**: podría etiquetar fácilmente todas las instancias en el conjunto de datos **asignando cada una de ellas al grupo cuyo centroide es el más cercano**.
- Por el contrario, si se **le dieran todas las etiquetas de instancia**, podría localizar fácilmente todos los centroides **calculando la media de las instancias para cada grupo**.
- Pero **no le dan las etiquetas ni los centroides**.
 - Comience colocando los centroides al azar (por ejemplo, eligiendo k instancias al azar y usando sus ubicaciones como centroides).
 - Luego etiquete las instancias, actualice los centroides, etiquete las instancias, actualice los centroides, etc., hasta que los centroides dejen de moverse.

Se garantiza que el algoritmo converge en un número finito de pasos
(generalmente bastante pequeño).

Variabilidad del K-Means

Desafortunadamente, aunque se garantiza que el algoritmo converge, puede **no converger** a la solución correcta (es decir, puede converger a un **óptimo local**).



Esto depende de la inicialización del centroide.



K-Means ++

En lugar de inicializar los **centroides al azar**, es preferible inicializarlos usando el algoritmo llamado **K-Means ++**:

- 1 Tome un centroide c_1 , elegido uniformemente al azar del conjunto de datos.
- 2 Tome un nuevo centroide c_i , eligiendo una instancia x_i con probabilidad: $D(x_i)^2 / \sum_{j=1}^m D(x_j)^2$ donde $D(x_i)$ es la distancia entre la instancia x_i y el centroide más cercano que ya se eligió. **Esto asegura que las instancias que están más lejos de los centroides ya elegidos sean mucho más probables como centroides.**
- 3 Repita el paso anterior **hasta que se hayan elegido todos los centroides k .**

El resto del algoritmo K-Means++ es solo K-Means normal. Con esta inicialización, es mucho menos probable que el algoritmo K-Means converja a una solución subóptima.



- **Accelerated K-Means:** Esto se logra [explotando la desigualdad del triángulo](#) (dados los tres puntos A, B, C , la distancia AC es siempre tal que $AC \leq AB + BC$) y manteniendo seguimiento de los límites inferior y superior para las distancias entre instancias y centroides.
- **Mini-Batch K-Means:** En lugar de usar el conjunto de datos completo en cada iteración, el algoritmo es capaz de usar **mini-lotes**, moviendo los centroides ligeramente en cada iteración. [Esto acelera el algoritmo típicamente en un factor de 3 o 4 y hace posible agrupar grandes conjuntos de datos que no caben en la memoria.](#)



Contenido

- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número optimo de clusters**
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering

Encontrar el número óptimo de clusters

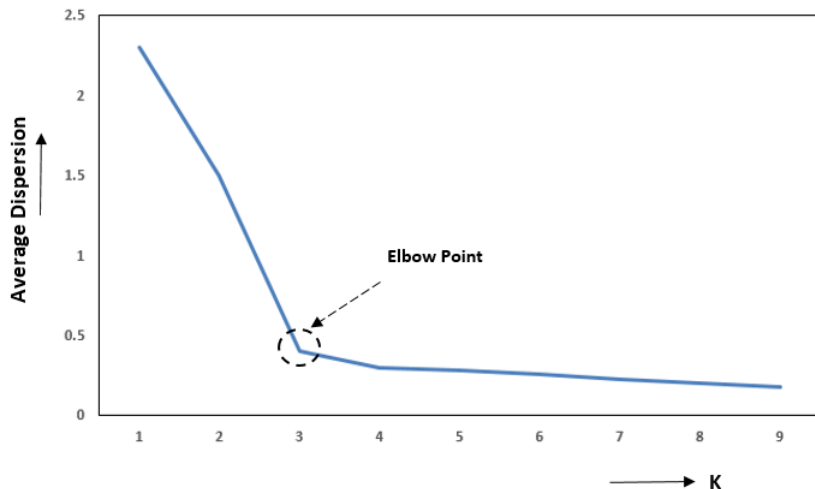
Hasta ahora, hemos establecido el número de grupos k a 5 **porque era obvio al observar los datos que este es el número correcto de grupos**. Pero, en general, no será tan fácil saber cómo configurar k , y **el resultado podría ser bastante malo si lo configura en el valor incorrecto**.



Criterio del codo

También llamado **criterio “L-curve”**.

Elbow Method for selection of optimal “K” clusters



El **Silhouette score** de una instancia es igual a $(b - a) / \max(a, b)$ donde a es la distancia media a las otras instancias en el mismo grupo (es la distancia media dentro del grupo), y b es la distancia media del cluster más cercano, que es la distancia media a las instancias del siguiente clúster más cercano.

El Silhouette coefficient puede variar entre -1 y $+1$:

- Un coeficiente cercano a $+1$ significa que la instancia está bien dentro de su propio grupo y lejos de otros grupos.
- Un coeficiente cercano a 0 significa que está cerca de un límite de grupo.
- Un coeficiente cercano a -1 significa que la instancia puede haber sido asignada al grupo incorrecto.

Encontrar el número óptimo de clusters



- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número óptimo de clusters
- 6 Limitaciones del K-Means**
- 7 Aplicaciones de clustering

Limitaciones del K-Means

A pesar de sus muchos méritos, más notablemente rápido y escalable, **K-Means no es perfecto.**

- Como vimos, **es necesario ejecutar el algoritmo varias veces para evitar soluciones subóptimas**, además de que **necesita especificar el número de clústeres**, lo que puede ser bastante complicado.
- Además, K-Means **no se comporta muy bien cuando los grupos tienen diferentes tamaños**, diferentes densidades o formas no esféricas.

Dependiendo de los datos, los diferentes algoritmos de agrupación pueden funcionar mejor. **Por ejemplo, en estos tipos de grupos elípticos, los modelos de mezcla gaussiana funcionan muy bien.**

Limitaciones del K-Means



- 1 Definición de aprendizaje no supervisado
- 2 Algunas tareas y algoritmos de aprendizaje no supervisado
- 3 Clustering
- 4 K-Means
- 5 Encontrar el número óptimo de clusters
- 6 Limitaciones del K-Means
- 7 Aplicaciones de clustering**

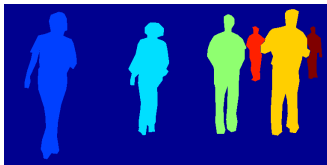
Clustering para segmentación de imagen

La segmentación de imágenes es la tarea de **dividir una imagen en múltiples segmentos**.

- **Segmentación semántica:** todos los píxeles que forman parte del mismo tipo de objeto se asignan al mismo segmento.



- **Segmentación de instancia:**, todos los píxeles que forman parte del mismo objeto individual se asignan al mismo segmento.



Clustering para segmentación de imagen



- **Clustering para pre-procesamiento:** Clustering puede ser un enfoque eficiente para la reducción de la dimensionalidad, en particular como un paso de pre-procesamiento antes de un algoritmo de aprendizaje supervisado.
- **Clustering para aprendizaje semi-supervisado:** Otro caso de uso para clustering es en el aprendizaje semi-supervisado cuando tenemos muchas instancias sin etiquetar y muy pocas instancias etiquetadas.



Este algoritmo define grupos como regiones continuas de alta densidad:

- 1 Para cada instancia, el algoritmo cuenta cuántas instancias se encuentran dentro de una pequeña distancia ϵ de él (ϵ -neighborhood).
- 2 Si una instancia tiene al menos un `min_samples` en su vecindario ϵ (incluida ella misma), entonces se considera una **instancia central**.
- 3 Todas las instancias cercanas a una instancia central pertenecen al mismo clúster. Esto puede incluir otras instancias centrales.
- 4 Cualquier instancia que no sea una instancia central y no tenga una en su vecindario **se considera una anomalía**.

Este algoritmo funciona bien si todos los grupos son lo suficientemente densos y están bien separados por regiones de baja densidad.



En resumen,

- DBSCAN es un algoritmo muy simple pero potente, capaz de identificar **cualquier número de clústeres**, de **cualquier forma**, es **robusto para los valores atípicos** y tiene solo dos hiperparámetros (ϵ y `min_samples`).
- Sin embargo, si la densidad varía significativamente entre los grupos, puede ser imposible capturar todos los grupos correctamente.

Otros algoritmos de clustering I

- ① **Agglomerative clustering:** *construye una jerarquía de agrupaciones de abajo hacia arriba.* Piense en muchas burbujas diminutas que flotan en el agua y se unen gradualmente entre sí hasta que solo haya un gran grupo de burbujas.
- ② **Birch:** *construye una estructura de árbol durante el entrenamiento que contiene información suficiente para asignar rápidamente cada nueva instancia a un clúster, sin tener que almacenar todas las instancias en el árbol:* esto le permite usar memoria limitada, mientras maneja grandes conjuntos de datos.
- ③ **Mean-shift:** este algoritmo comienza colocando un círculo centrado en cada instancia, luego, para cada círculo, *calcula la media de todas las instancias ubicadas dentro de él y desplaza el círculo de modo que esté centrado en la media.* A continuación, itera este paso de desplazamiento medio hasta que todos los círculos dejan de moverse.

- 4 **Affinity propagation**: este algoritmo utiliza un sistema de votación, donde las instancias votan por instancias similares para ser sus representantes, y una vez que el algoritmo converge, cada representante y sus votantes forman un grupo. Este algoritmo puede detectar cualquier cantidad de grupos de diferentes tamaños.
- 5 **Spectral clustering**: este algoritmo toma una matriz de similitud entre las instancias y crea un embedding de baja dimensión a partir de ella, luego usa otro algoritmo de agrupación en este espacio de baja dimensión (i.e., K-Means).

Spectral clustering

