

Unidad VII - Técnicas de Agrupamiento Partitivas

Germán Braun

Facultado de Informática - Universidad Nacional del Comahue

german.braun@fi.uncoma.edu.ar

4 de octubre de 2025

Agenda

- 1 Aprendizaje No Supervisado
- 2 Clustering

Aprendizaje No Supervisado

AA Landscape

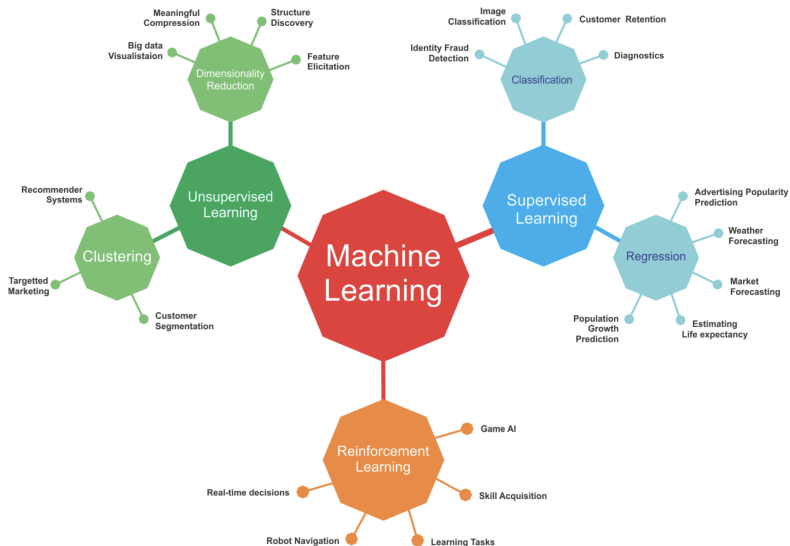


Figura 1.1: Imagen de [?]

- El dataset no está etiquetado: $\{x^{(1)} \dots x^{(n)}\}$

Aprendizaje No Supervisado

- El dataset no está etiquetado: $\{x^{(1)} \dots x^{(n)}\}$
- El objetivo es encontrar estructuras “**interesantes**” en los datos → *clusters*

Aprendizaje No Supervisado

- El dataset no está etiquetado: $\{x^{(1)} \dots x^{(n)}\}$
- El objetivo es encontrar estructuras “**interesantes**” en los datos → *clusters*
- Los algoritmos descubren patrones ocultos or agrupan datos sin necesidad de intervención humana.

Aprendizaje No Supervisado

- El dataset no está etiquetado: $\{x^{(1)} \dots x^{(n)}\}$
- El objetivo es encontrar estructuras “**interesantes**” en los datos → *clusters*
- Los algoritmos descubren patrones ocultos or agrupan datos sin necesidad de intervención humana.
- A diferencia del supervisado, el aprendizaje no supervisado puede ser útil para la detección de anomalías, segmentación de clientes, entre otros ...

Aprendizaje No Supervisado

- El dataset no está etiquetado: $\{x^{(1)} \dots x^{(n)}\}$
- El objetivo es encontrar estructuras “**interesantes**” en los datos → *clusters*
- Los algoritmos descubren patrones ocultos or agrupan datos sin necesidad de intervención humana.
- A diferencia del supervisado, el aprendizaje no supervisado puede ser útil para la detección de anomalías, segmentación de clientes, entre otros ...

Son computacionalmente más complejos debido a que requieren de un dataset más grande para producir una salida satisfactoria

Clustering

- Organizar ejemplos en grupos (*clusters*)

Intuición y Formalización

- Organizar ejemplos en grupos (***clusters***)
- Luego del entrenamiento, nuevos ejemplos son asignados a los grupos correctos

Intuición y Formalización

- Organizar ejemplos en grupos (**clusters**)
- Luego del entrenamiento, nuevos ejemplos son asignados a los grupos correctos

Dado un dataset de entrada, S el conjunto de posibles clusters (finito), $c^{(i)}$ el cluster de $x^{(i)}$ y $\mu_{c^{(i)}}$, la media del cluster $c^{(i)}$:

$$\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}, \quad x^{(i)} \in \mathbb{R}^d$$

$$S = \{c \mid c \in \{1, 2, \dots, k\}^n\}$$

$$\mu_k = \frac{\sum_{i=1}^n x^{(i)} \cdot 1(c^{(i)} = k)}{\sum_{i=1}^n 1(c^{(i)} = k)}$$

Intuición y Formalización

- Organizar ejemplos en grupos (**clusters**)
- Luego del entrenamiento, nuevos ejemplos son asignados a los grupos correctos

Dado un dataset de entrada, S el conjunto de posibles clusters (finito), $c^{(i)}$ el cluster de $x^{(i)}$ y $\mu_{c^{(i)}}$, la media del cluster $c^{(i)}$:

$$\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}, \quad x^{(i)} \in \mathbb{R}^d$$

$$S = \{c \mid c \in \{1, 2, \dots, k\}^n\}$$

$$\mu_k = \frac{\sum_{i=1}^n x^{(i)} \cdot 1(c^{(i)} = k)}{\sum_{i=1}^n 1(c^{(i)} = k)}$$

El objetivo es

$$\min_{c \in S} \sum_{i=1}^n \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

Idea

- 1 Setear k como el número de clusters inicial

Idea

- 1 Setear k como el número de clusters inicial
- 2 Tomar k puntos como “**centroides**” (aleatoriamente) del conjunto de entrenamiento

Idea

- 1 Setear k como el número de clusters inicial
- 2 Tomar k puntos como “**centroides**” (aleatoriamente) del conjunto de entrenamiento
- 3 Agrupar cada observación a su centroide más cercano

Idea

- 1 Setear k como el número de clusters inicial
- 2 Tomar k puntos como “**centroides**” (aleatoriamente) del conjunto de entrenamiento
- 3 Agrupar cada observación a su centroide más cercado
- 4 Actualizar el centroide a el centro de todos las observaciones agrupada hasta el momento

Idea

- 1 Setear k como el número de clusters inicial
- 2 Tomar k puntos como “**centroides**” (aleatoriamente) del conjunto de entrenamiento
- 3 Agrupar cada observación a su centroide más cercado
- 4 Actualizar el centroide a el centro de todos las observaciones agrupada hasta el momento
- 5 Repetir los pasos 3 y 4 un número fijo de veces hasta que todos los centroides se estabilicen

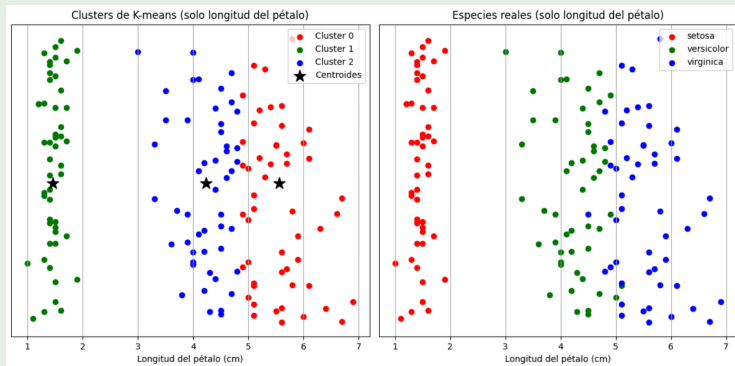
Idea

- 1 Setear k como el número de clusters inicial
- 2 Tomar k puntos como “**centroides**” (aleatoriamente) del conjunto de entrenamiento
- 3 Agrupar cada observación a su centroide más cercado
- 4 Actualizar el centroide a el centro de todos las observaciones agrupada hasta el momento
- 5 Repetir los pasos 3 y 4 un número fijo de veces hasta que todos los centroides se estabilicen
- 6 Evaluación:
 - **accuracy** con etiquetas reales
 - **método del codo (Elbow)**: el codo es el punto donde dejar de agregar clusters mejora la performance
 - **coeficiente de Silhouette**: separación de clusters (1/0/-1)

K-means (cont'd)

Agrupar flores de acuerdo a sus características

Supongamos que queremos agrupar las flores de iris analizando sus características: *longitud del sépalo*, *ancho del sépalo*, *longitud del pétalo*, *ancho del pétalo*



El gráfico muestra los 3 clusters generados, uno para cada flor del dataset: *iris setosa*, *versicolor* y *virginica*, usando el algoritmo **k-means**.

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:
 - Dada una entrada **x**, se distribuyen pesos **w** en cada una de las neuronas de la red

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:
 - Dada una entrada **x**, se distribuyen pesos **w** en cada una de las neuronas de la red
 - Se calcula la **similitud** para indicar cual de las neuronas es más “**parecida**” a la entrada (**distancia euclideana/producto punto**)

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:
 - Dada una entrada **x**, se distribuyen pesos **w** en cada una de las neuronas de la red
 - Se calcula la **similitud** para indicar cual de las neuronas es más “**parecida**” a la entrada (**distancia euclideana/producto punto**)
 - En base a esta similitud se procede a competir

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:
 - Dada una entrada **x**, se distribuyen pesos **w** en cada una de las neuronas de la red
 - Se calcula la **similitud** para indicar cual de las neuronas es más “**parecida**” a la entrada (**distancia euclideana/producto punto**)
 - En base a esta similitud se procede a competir
 - **Inhibición Lateral**: la neurona con mayor similitud, apaga las demás

Idea

- Las neuronas de la red **compiten** basadas en la proximidad al patrón de entrada
- La neurona “**ganadora**” (*winner-takes-all*) ajusta su peso para hacerse más próxima
- Funcionamiento:
 - Dada una entrada **x**, se distribuyen pesos **w** en cada una de las neuronas de la red
 - Se calcula la **similitud** para indicar cual de las neuronas es más “**parecida**” a la entrada (**distancia euclideana/producto punto**)
 - En base a esta similitud se procede a competir
 - **Inhibición Lateral**: la neurona con mayor similitud, apaga las demás
 - **Competencia de Recursos**: las neuronas de salida se activan según el grado de similaridad con la entrada

Bibliografía y material de referencia



Harrington, Peter. Machine learning in action. *Simon and Schuster*, 2012.



Alpaydin, Ethem. Introduction to machine learning. 3era Edición *MIT Press*, 2020.



Brett Lantz. Machine Learning with R. *Packt Publishing*, 1997.



Tom M. Mitchell. Machine Learning. *WCB McGraw-Hill*, 1997.



Witten I., Frank E., Hall, M., Pal C.. Data Mining: Practical Machine Learning Tools and Techniques. 4th Edition *WMorgan Kaufmann. Elsevier*, 2017.



Michael A. Nielsen. Neural Networks and Deep Learning. 4th Edition *Determination Press*, 2015.

<http://neuralnetworksanddeeplearning.com>

Bibliografía y material de referencia



Afshine Amidi, Shervine Amidi. CS 229 — Machine Learning.
<https://stanford.edu/~shervine/teaching/cs-229/>



Andrew Ng. Stanford CS229 - Machine Learning Course.
<https://www.youtube.com/playlist?list=PLoROMvovdv4rMiGQp3WXShtMGgzqpFVfbU>



Andrew Ng. Deep Learning AI.
<https://www.deeplearning.ai/resources/>



Kilian Weinberger. Machine Learning for Intelligent Systems. <https://www.cs.cornell.edu/courses/cs4780/2018fa/syllabus/>



Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine Learning. Springer. 1995



Paul J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University, 1974.



David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, *Learning representations by back-propagating errors*, Nature, vol. 323, pp. 533–536, 1986.
DOI: 10.1038/323533a0

FIN!

Gracias!

- Agustina Buccella
- Laura Cecchi
- Sandra Roger