



BOURBAKI

COLEGIO DE MATEMÁTICAS

Índice

01. Introducción _____ pág. 03

01. ¿Qué es el aprendizaje no-supervisado?

pág. 03

02. Notación matemática _____ pág. 04

02. Instalación de Python y un entorno de trabajo para Machine Learning _____ pág. 06

01. Instalación de Miniconda [3] _____ pág. 06

02. Creación de un entorno virtual para análisis de datos _____

pág. 08

03. Jupyter Notebooks para ML en la Nu-

be _____ pág. 11

03. Algoritmos de cercanía _____ pág. 12

01. Preludio supervisado: K-nearest neighbours

pág. 12

02. Motivación: aglomeración jerárquica

pág. 13

03.	2-means	_____	pág. 15
04.	K-means	_____	pág. 16
05.	Estandarización de los datos	—	pág. 18
04.	Analizar el bienestar en el entorno laboral	_____	pág. 19
05.	Complementos del aprendizaje no-supervisado	_____	
	pág. 20		
01.	K-medians, K-medoids	_____	pág. 20
02.	Variables categóricas y el algoritmo KA-MILA	_____	pág. 21
03.	Sobre-ajuste y los métodos de la silueta y del codo	_____	pág. 21
04.	Densidades mixtas	_____	pág. 22
06.	Referencias	_____	pág. 23

Ø1 Introducción

Las siguientes notas son la bitácora de un curso de 8 horas que impartimos junto a Ana Isabel Ascencio. Además de este documento los invitamos a consultar el Github del curso [en este link](#).

El curso es una invitación al aprendizaje no-supervisado y sus aplicaciones, nuestro ejemplo principal es el algoritmo de K-means sin embargo hablaremos de algunos otros, el curso está dividido en clases de la siguiente manera:

1. ¿Qué es el aprendizaje no-supervisado y dónde aplicarlo? (una hora)
2. Un vistazo a Python y la visualización de datos mediante bibliotecas (una hora)
3. Descripción formal del algoritmo de K-means (dos horas)
4. Implementación del algoritmo de K-means para una base de datos de condiciones laborales (dos horas)
5. Dudas y complementos (dos horas)

¿Qué es el aprendizaje no-supervisado?

El aprendizaje no-supervisado consta de una amplia familia de algoritmos que no requieren entrenarse con bases de datos etiquetadas a diferencia de otros algoritmos como por ejemplo la regresión lineal. Las tareas que realizan son bastante diversas desde la aglomeración (clustering en inglés), la reducción de la dimensión y algunas otras. En este curso nos concentraremos en el problema de la aglomeración.

Notación matemática

- En estas notas denotaremos por \mathbb{R} al conjunto de los números reales es decir todos los números negativos o positivos que conocemos (incluyendo algunos como π etc.).
- Un subconjunto importante de los números reales es \mathbb{N} llamada el conjunto de los números naturales y consiste en los siguientes números $\mathbb{N} = \{1, 2, 3, \dots\}$.
- Si $d \in \mathbb{N}$ es un número natural entonces denotaremos por \mathbb{R}^d al conjunto de vectores (x_1, x_2, \dots, x_d) de tamaño d con entradas en \mathbb{R} . Por ejemplo \mathbb{R}^2 es el plano cartesiano.
- Si $x = (x_1, \dots, x_d), y = (y_1, \dots, y_d) \in \mathbb{R}^d$ entonces el producto punto de x, y se denota por $\langle x, y \rangle$ y es el número real

$$\langle x, y \rangle = x_1 y_1 + \dots + x_d y_d$$

- Si $x \in \mathbb{R}^d$ es un punto en el espacio euclíadiano denotaremos por $\|x\|_2 = \sqrt{\langle x, x \rangle}$ y lo llamaremos la norma de x .
- Si $x, y \in \mathbb{R}^d$ son dos puntos en el espacio euclíadiano denotaremos por $d(x, y) = \|x - y\|_2$ a la distancia entre x e y . Si por ejemplo $y = (0, 0, \dots, 0)$ entonces $d(x, 0) = \|x\|_2$.

02 Instalación de Python y un entorno de trabajo para Machine Learning

La forma más rápida y eficiente de instalar Python, manejar sus librerías y evitar problemas de dependencia(por actualizaciones y versiones) es mediante el uso de Miniconda que además de la instalación de Python, instala conda^[1] un sistema gestor de paquetes y de entornos virtuales ¹ [2].

Una forma alternativa es utilizar Anaconda, que al igual que Miniconda, es soportada por la misma compañía Anaconda. Ambas alternativas instalan la misma versión de Python y de conda. La razón por la que se recomienda Miniconda es porque permite instalar solamente lo que se requiere para este curso, con la opción de descargar otros paquetes cuando se requieran. Anaconda por el contrario instala por defecto, una cantidad excesiva de paquetes que raramente son utilizados, lo cual requiere demasiado espacio y tiempo.

Instalación de Miniconda [3]

¹Los entornos virtuales permiten controlar las versiones de software (en nuestro caso python y sus librerías) usado para análisis o aplicaciones

Descarga aquí el archivo correspondiente a tu sistema operativo, ya sea que sea que MacOS, Windows y Linux, eligiendo la última versión de Python.

Instalación para Windows

1. Da doble clic en el archivo descargado.
2. Sigue las instrucciones que se muestren aceptando las opciones que se proponen por defecto (si es necesario se pueden cambiar después).
3. Desde el menú de Inicio de Windows abre el programa Anaconda Prompt.

Instalación para MacOS

1. En la Terminal de tu Mac, navega hasta el directorio en donde descargaste el instalador y corre la siguiente línea: bash Miniconda3-latest-MacOSX-x86_64.sh
2. Sigue las instrucciones, aceptando las opciones por defecto (si es necesario se pueden cambiar después)
3. Cierra la Terminal y vuélvela a abrir, para que los cambios sean actualizados.

Instalación para Linux

1. En la Terminal de Linux, navega hasta el directorio en donde descargaste el instalador y corre la siguiente línea:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

2. Sigue las instrucciones, aceptando las opciones por defecto (si es necesario se pueden cambiar después)
3. Cierra la Terminal y vuélvela a abrir, para que los cambios sean actualizados.

Para todos los sistemas operativos:

4. Para ver todos los paquetes que instalados por defecto en la distribución de Miniconda, corre la siguiente línea:

```
conda list
```

Creación de un entorno virtual para análisis de datos

Un entorno virtual es un ambiente de trabajo aislado, lo que permite instalar determinadas librerías o versiones de librerías sin que afecte al resto del sistema principal o de otros ambientes.

Por defecto, Miniconda crea un entorno llamado base que estará activo cuando abramos la terminal. Este entorno contiene todos los programas enlistados cuando se utilizó el comando conda list. Aunque podríamos usar este entorno, una mejor práctica es crear un entorno nuevo. Para ello es conveniente cambiar la configuración de conda para que no abra automáticamente el entorno base.

Desactivar el entorno base

- En Mac y Linux, se debe correr la linea que se muestra a continuación y enseguida cerrar y volver a abrir la terminal:

```
conda config --set auto_activate_base false
```

- Los usuarios de Windows deberán desactivar manualmente el entorno base utilizando:

```
conda deactivate
```

Creación de un entorno específico para análisis de datos mediante conda-forge

El entorno que vamos a crear lo titularemos MachineLearning, en éste instalaremos las librerías que necesitaremos para nuestros análisis.

1. Crea el ambiente de trabajo mediante la siguiente línea:

```
conda create -n MachineLearning
```

2. Actívalo con la siguiente instrucción:

```
conda activate MachineLearning
```

3. Agrega el canal ²

```
conda config --env --add channels conda-forge
```

Se recomienda utilizar el canal conda-forge, ya que es resultado de un esfuerzo colectivo con una gran variedad de librerías y paquetes que son cuidadosamente actualizados y donde se asegura tener versiones compatibles para macOS, Linux y Windows[4].

Puedes ver los canales instalados mediante la siguiente línea:

```
conda config --show channels
```

Cada canal tiene al menos una dirección URL asociada donde se localizan los repositorios de paquetes y librerías. Puedes ver estas direcciones utilizando:

```
conda info
```

4. Instala las librerías Pandas³, Scikit-learn⁴, Seaborn, Matplotlib⁵,

²Los repositorios desde donde podemos descargar las librerías de Python se llaman canales, Anconda, Inc provee por defecto el canal llamado default, sin embargo permite a cualquier usuario crear su propio canal si necesita otros paquetes que no están incluidos en el canal default, por esta razón existe una gran variedad de canales disponibles en la web desde donde se pueden descargar librerías de Python.

³Pandas es la principal librería de python utilizada Analisis de datos. Al instalar Pandas, por defecto se instala Numpy, sobre la cual funciona Pandas. Numpy es ampliamente usada en Ciencia de Datos porque permite el fácil manejo de matrices y sus operaciones.

⁴Scikit-learn es una importante librería para machine learning

⁵Matplotlib es las principales librerías de Python usadas para graficar y visualizar información

Plotly, Yellowbrick y Jupyter Notebooks⁶ copiando el siguiente comando (es importante copiarlo en una sola línea, separando mediante un espacio, cada librería a ser instalada):

```
conda install pandas matplotlib notebook jupyter_contrib_nbextensions  
scikit-learn yellowbrick plotly plotly=4.12.0
```

Jupyter Notebooks para ML en la Nube

Una alternativa altamente recomendable es utilizar el entorno que ofrece Google Colab ya que permite ejecutar código de Python utilizando notebooks de Jupyter a cualquier persona con una cuenta de Google, sin la necesidad de tener que instalar nada en la computadora del estudiante, y ejecutar el código directamente en los servidores alojados en la nube de Google. Google Colab es altamente compatible con el repositorio del curso que se encuentra en Github

⁶Jupyter notebook es una aplicación que nos ayudará a hacer nuestros análisis paso a paso, crear visualizaciones e incluir comentarios, como si se tratara de nuestro cuaderno de apuntes.

03 Algoritmos de cercanía

Este capítulo incluye los detalles técnicos del algoritmo *K-means*. Para comenzar el curso recordaremos brevemente cómo funciona el algoritmo *K-nearest neighbours* el cuál es la versión supervisada de *K-means*.

Preludio supervisado: K-nearest neighbours

K-nearest neighbours es un algoritmo simple que aprovecha cuando la dimensión del problema es suficientemente baja, por ejemplo el algoritmo de recomendación de Netflix se basa en esta idea.

Sea $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ nuestra base de datos, recordemos que $x_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$.

Definition 01.1. (1-NN) Sea $x \in \mathbb{R}^d$,

1. Calculemos las siguientes distancias: $d(x, x_1) =: d_1, \dots, d(x, x_N) =: d_N$

2. Calculamos y llamaremos d_x al menor de los números d_1, \dots, d_N , sea

$$d_x = d_i.$$

3. Suponiendo que $(x_i, y_i) \in S$, predecimos (x, y) .

Exercise 01.1. *Dibujar las fronteras de clasificación para*

$$S = \{(1, 0, -1), (-1, 0, -1), (0, 1, 1), (0, -1, 1)\}$$

Exercise 01.2. *Si por un momento definimos $d(x, x') = |x_1 - x'_1| + \dots + |x_d - x'_d|$,*

dibujar las fronteras de clasificación de 1-NN en el plano cartesiano.

Definition 01.2. (K-NN) Sea $x \in \mathbb{R}^d$ y $K \leq N$

1. Calculemos las siguientes distancias: $d(x, x_1) =: d_1, \dots, d(x, x_N) =: d_N$
2. Ordenamos las distancias de menor a mayor: $d_{i_1} \leq d_{i_2} \leq \dots \leq d_{i_K} \leq \dots \leq d_{i_N}$.
3. Definimos a y como aquella última coordenada que más se repita en los siguientes vectores en S : $(x_{i_1}, y_{i_1}), \dots, (x_{i_K}, y_{i_K})$.
4. Suponiendo que $(x_i, y_i) \in S$, predecimos (x, y) .

Exercise 01.3. *Si $K = N$ a qué corresponde nuestro algoritmo K-NN?*

Exercise 01.4. *Cuáles son las desventajas de un K cercano a 1? Cuáles son las desventajas de un K cercano a N ?*

Exercise 01.5. *Qué dificultad computacional podría encontrar K-NN?*

Motivación: aglomeración jerárquica

Existen dos paradigmas distintos en los algoritmos de clusterización, cada uno de ellos sigue alguna de las siguientes ideas:

- Si dos elementos están cerca entonces deben pertenecer al mismo cluster.
- Si dos elementos están en el mismo cluster entonces están cerca.

La aglomeración jerárquica es un algoritmo que sigue el primero de estos paradigmas mientras que K -means es un algoritmo que sigue el segundo.

De acuerdo al problema que queremos de resolver será mejor seguir el primero o el segundo de los paradigmas. En el curso explicaremos por qué en nuestro ejemplo es necesario utilizar el segundo.

Definition 02.1. Sean $A, B \subseteq \mathbb{R}^d$ dos conjuntos finitos de puntos que debemos pensar como posibles clusters, definimos dos tipos distintos de distancias entre ellos:

$$1. D_1(A, B) = \min_{a \in A, b \in B} d(a, b)$$

$$2. D_2(A, B) = \max_{a \in A, b \in B} d(a, b)$$

Remark 02.1. Es importante notar que cuando $A = \{a\}, B = \{b\}$, entonces $D_1(A, B) = D_2(A, B) = d(a, b)$.

Definition 02.2. (Aglomeración jerárquica) Sea $S = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^d$, $K \in \mathbb{N}$ y $\lambda > 0$, buscamos separar a S en K conjuntos disjuntos C_1, \dots, C_K :

1. En el primer paso, definimos N -clusters de la siguiente manera $C_i^0 =$

$$\{x_i\}$$

2. Calculamos todas las distancias $D_j(C_i^0, C_k^0)$ (para $j = 1, 2$ dependiendo qué distancia deseamos utilizar) para todo $i \neq k$.

3. En el tiempo t iteramos el algoritmo que define clusters de la siguiente manera: $C_i^t = C_{i_1} \cup \dots \cup C_{i_l}$ siempre y cuando $D_j(C_{i_s}^{t-1}, C_{i_r}^{t-1}) < \lambda$

4. Detenerse cuando hayamos encontrado K -clusters.

Exercise 02.2. Supongamos que $S = \{1, 2, 3, 4\}$, aplicar el algoritmo de agrupación jerárquica con $\lambda = 2, K = 2$.

2-means

El primer algoritmo no-supervisado que estudiaremos es el llamado 2-means, este algoritmo lo utilizaremos también para resolver un problema de clasificación binaria.

Es fundamental notar que en los algoritmos supervisados se hacía intenso uso de la información que las coordenadas y_i de los puntos $(x_i, y_i) \in S$ nos daban. Esta vez debemos prescindir de esa información, es decir: sea $S = \{x_1, \dots, x_N\}$ con $x_i \in \mathbb{R}^d$, buscamos partir a nuestra base de datos en dos subconjuntos disjuntos: $S_{+1}, S_{-1} \subseteq S$, $S_{+1} \cap S_{-1} = \emptyset$.

Definition 03.1. Sea $S = \{x_1, \dots, x_N\}$ con $x_i \in \mathbb{R}^d$ y $S' \subseteq S$ un subconjunto arbitrario de S ,

- Definimos el centroide de S' de la siguiente forma: $\mu_{S'} := \underset{\mu \in \mathbb{R}^d, x_i \in S'}{\operatorname{argmin}} (\mu, x_i)^2$

Definition 03.2. (2-Means) Sea $S = \{x_1, \dots, x_N\}$ con $x_i \in \mathbb{R}^d$,

1. Comenzemos con cualquier par de puntos $\mu_1^0, \mu_2^0 \in \mathbb{R}^d$

2. Definimos

$$S_1^0 = \{x \in S : 1 = \underset{i \leq 2}{\operatorname{argmin}} (d(\mu_i, x))\}$$

$$S_2^0 = \{x \in S : 2 = \underset{i \leq 2}{\operatorname{argmin}} (d(\mu_i, x))\}$$

3. Una vez encontrados S_1^0 y S_2^0 debemos actualizar los centroides:

$$\mu_1^1 := \frac{1}{|S_1^0|} \sum_{x \in S_1^0} x$$

$$\mu_2^1 := \frac{1}{|S_2^0|} \sum_{x \in S_2^0} x$$

4. Repetir inductivamente para encontrar $S_1^i, S_2^i, \mu_1^i, \mu_2^i$.

Exercise 03.1. Sea $S = \{1, 2, 3, 4\} \subseteq \mathbb{R}$, aplicar el algoritmo de 2-means.

Definition 04.1. (K-Means) Sea $S = \{x_1, \dots, x_N\}$ con $x_i \in \mathbb{R}^d$ y $K \in \mathbb{N}$.

1. Comenzemos con cualquier familia de K centroides $\mu_1^0, \mu_2^0, \dots, \mu_K^0 \in \mathbb{R}^d$,

para cada $i \leq K$, definimos

$$S_i^0 = \{x \in S : i = \underset{j \leq 2}{\operatorname{argmin}}(d(\mu_j, x))\}$$

2. Una vez encontrados $S_1^0, S_2^0, \dots, S_K^0$ debemos actualizar los centroides:

$$\mu_i^1 := \frac{1}{|S_i^0|} \sum_{x \in S_i^0} x$$

3. Repetir inductivamente para encontrar $S_1^t, S_2^t, \dots, S_K^t, \mu_1^t, \mu_2^t, \dots, \mu_K^t$.

Remark 04.1. *En general no hay ningún resultado matemático para garantizar la convergencia de este algoritmo, lo más que es posible garantizar es el lema siguiente. Por tanto lo que se debe de hacer en la práctica es inicializar con centroides aleatorios algunas veces para obtener buenos resultados.*

Definition 04.2. Sea C_1, \dots, C_K una clusterización de algún conjunto de puntos S , definimos la función objetivo de la siguiente manera:

$$G_{means}(S : C_1, C_2, \dots, C_K) = \sum_{i \leq K} \sum_{x \in C_i} d(x, \mu(C_i))$$

Remark 04.2. *Es facil convencerse de que es posible re-escribir la función G_{means} de la siguiente manera:*

$$G_{means}(S: C_1, C_2, \dots, C_K) = \min_{\mu_1, \dots, \mu_K \in \mathbb{R}^d} \sum_{i \leq K} \sum_{x \in C_i} d(x, \mu_i)^2$$

Lemma 04.3. *Cada iteración del algoritmo de K-means no aumenta la función G_{means} .*

Estandarización de los datos

Para que el algoritmo de K -means sea más eficiente algunas veces es recomendable estandarizar los datos, existen distintos métodos para hacerlo, en esta sección definimos dos comúnmente utilizados con los que experimentaremos en nuestro data-set.

Definition 05.1. (Estandarización vía Z-score) Sea $S = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^d$, definimos la estandarización de los datos de acuerdo al Z -score de la siguiente manera: $S' = \{x'_1, \dots, x'_N\} \subseteq \mathbb{R}^d$ donde para cada $j \leq d$, si \bar{x}_j es la media aritmética del conjunto $\{x_{1,j}, \dots, x_{N,j}\}$ y σ_j su desviación estándar, entonces

$$x'_{i,j} = \frac{x_{i,j} - \bar{x}_j}{\sigma_j}$$

Definition 05.2. (Estandarización vía min-max) Sea $S = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^d$, definimos la estandarización de los datos de acuerdo al min-max de la siguiente manera: $S' = \{x'_1, \dots, x'_N\} \subseteq \mathbb{R}^d$ donde para cada $j \leq d$, si $x_{max}, x_{min} \in \mathbb{R}$ son los valores máximos y mínimos de las entradas en S , entonces $x'_{i,j} = \frac{x_{i,j} - x_{min}}{x_{max} - x_{min}}$

Ø4 Analizar el bienestar en el entorno laboral

A continuación incluimos una breve descripción del data set que utilizaremos para ejemplificar la implementación de *K-means*. Los datos fueron obtenidos de un Challenge planteado por Oze Energies para intentar predecir el bienestar en un entorno laboral utilizando la siguiente información: el entorno térmico, la calidad del aire, la humedad, la luminosidad y la concentración de dióxido de carbono en los entornos de trabajo. Pueden consultar los detalles en el [siguiente link](#). Es importante mencionar que inicialmente el challenge se planteo como un problema supervisado sin embargo inspirados en la aplicación en México de la denominada Nom 035 la cuál es una encuesta a nivel nacional que pretende conocer las condiciones laborales al rededor del país pensamos en plantear un problema similar al que muchas compañías se enfrenta a raíz de este cambio.

El conjunto de datos utilizados está disponible [siguiente link](#)

05 Complementos del aprendizaje no-supervisado

Esta sección incluye un compendio de temas que buscan complementar el conocimiento de los estudiantes sobre el aprendizaje no-supervisado.

K-medians, K-medoids

Existen otros dos algoritmos idénticos en estructura a K -means que se actualizan utilizando alteraciones de los centroides tales como los medoids o las medianas, esta vez estos elementos se solicita que pertenezcan al conjunto S , las funciones de pérdida asociadas se definen de la siguiente manera:

Definition 01.1. Sea C_1, \dots, C_K una clusterización de algún conjunto de puntos S , definimos la función objetivo del algoritmo de K -medoids de la siguiente manera:

$$G_{medoid}(S: C_1, C_2, \dots, C_K) = \min_{\mu_1, \dots, \mu_K \in S} \sum_{i=1}^K \sum_{x \in C_i} d(x, \mu_i)$$

Definition 01.2. Sea C_1, \dots, C_K una clusterización de algún conjunto de puntos S , definimos la función objetivo del algoritmo de K -medias de la siguiente manera:

$$G_{medias}(S : C_1, C_2, \dots, C_K) = \min_{\mu_1, \dots, \mu_K \in S} \sum_{i=1}^K \sum_{x \in C_i} d(x, \mu_i)^2$$

Variables categóricas y el algoritmo KAMILA

La propiedad fundamental que se utiliza en los algoritmos de cercanía es el concepto de distancia euclíadiana, la cual se comporta extremadamente bien y por ello es posible sugerir un algoritmo tan simple. Desafortunadamente algunas veces no es posible suponer que nuestros ejemplos $x \in \mathbb{R}^d$, pensemos por ejemplo en el caso de variables categóricas. Existe un algoritmo que sigue la misma estructura que K -means sin la necesidad de hacer hipótesis de orden en nuestras características, su nombre es KAMILA y recomendamos a los interesados investigar más sobre sus detalles e implementación, desafortunadamente este algoritmo excede los objetivos de este curso.

Sobre-ajuste y los métodos de la silueta y del codo

Densidades mixtas

Así como las distribuciones de probabilidad son muy importantes en aprendizaje supervisado, existe una familia de distribuciones llamadas densidades mixtas que son muy importantes para el aprendizaje no-supervisado.

Definition 04.1. Sean $\Omega = \{\omega_1, \dots, \omega_n\}$, \mathbb{P}_i K espacios de probabilidad sobre el mismo conjunto Ω y $\alpha_1, \dots, \alpha_K \geq 0$ tales que $\sum_{i \leq K} \alpha_i = 1$. Definimos una nueva función

$$\mathbb{P}(\omega) = \sum_{i \leq K} \alpha_i \cdot \mathbb{P}_i(\omega)$$

Es posible pensar en un problema de aprendizaje no-supervisado como un problema de aproximación de una distribución mixta donde cada uno de los \mathbb{P}_i es la distribución de el cluster i . Existen diversas técnicas para utilizar este enfoque por ejemplo el de la máxima verosimilitud.

06 Referencias

- [1] «Conda — conda 4.8.3.post5+125413ca documentation». <https://docs.conda.io/projects/conda/en/latest/> (accedido mar. 26, 2020).
- [2] Anaconda is Bloated - Set up a Lean, Robust Data Science Environment with Miniconda and Conda-Forge.
- [3] «Miniconda — Conda documentation». <https://docs.conda.io/en/latest/miniconda.html> (accedido mar. 26, 2020).
- [4] «A brief introduction — conda-forge 2019.01 documentation». <https://conda-forge.org/docs/user/introduction.html> (accedido mar. 26, 2020).



escuela-bourbaki.com