PROGRAMA INSTITUCIONAL DE FORMACIÓN DOCENTE

Fundamentos de Machine Learning

Cómo aprenden las máquinas

Ildeberto de los Santos Ruiz

https://isantosruiz.github.io/home

Tecnológico Nacional de México
Instituto Tecnológico de Tuxtla Gutiérrez

Turix-Dynamics Diagnosis and Control Group

Los algoritmos

Una clase para aprendizaje supervisado

className.m

```
classdef className
    properties
        params
    end

    methods
        function obj = className(args)
            obj.params = [];
        end

        function obj = fit(obj, X, y)  % also named train()
            obj.params = [];
        end

        function y = predict(obj, X)
            y = [];
        end
    end
end
```

Regresión lineal

$$y = c_1 f_1(x) + c_2 f_2(x) + \cdots + c_n f_n(x)$$

Regresión lineal

$$y = c_1 f_1(x) + c_2 f_2(x) + \cdots + c_n f_n(x)$$

$\boldsymbol{x}$	$\boldsymbol{y}$
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

Regresión lineal

$$y = c_1 f_1(x) + c_2 f_2(x) + \cdots + c_n f_n(x)$$

x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

$\Rightarrow$

$$\begin{aligned} c_1 f_1(x_1) + c_2 f_2(x_1) + \cdots + c_n f_n(x_1) &= y_1 \\ c_1 f_1(x_2) + c_2 f_2(x_2) + \cdots + c_n f_n(x_2) &= y_2 \\ &\vdots \\ c_1 f_1(x_m) + c_2 f_2(x_m) + \cdots + c_n f_n(x_m) &= y_m \end{aligned}$$

Regresión lineal

$$y = c_1\, f_1(x) + c_2\, f_2(x) + \cdots + c_n\, f_n(x)$$

$\boldsymbol{x}$	$\boldsymbol{y}$
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

$$\begin{aligned}
c_1\, f_1(x_1) + c_2\, f_2(x_1) + \cdots + c_n\, f_n(x_1) &= y_1 \\
c_1\, f_1(x_2) + c_2\, f_2(x_2) + \cdots + c_n\, f_n(x_2) &= y_2 \\
&\vdots \\
c_1\, f_1(x_m) + c_2\, f_2(x_m) + \cdots + c_n\, f_n(x_m) &= y_m
\end{aligned}$$

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Regresión lineal

$$y = c_1\, f_1(x) + c_2\, f_2(x) + \cdots + c_n\, f_n(x)$$

$\boldsymbol{x}$	$\boldsymbol{y}$
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

$$\Longrightarrow \quad \begin{aligned} c_1\, f_1(x_1) + c_2\, f_2(x_1) + \cdots + c_n\, f_n(x_1) &= y_1 \\ c_1\, f_1(x_2) + c_2\, f_2(x_2) + \cdots + c_n\, f_n(x_2) &= y_2 \\ &\vdots \\ c_1\, f_1(x_m) + c_2\, f_2(x_m) + \cdots + c_n\, f_n(x_m) &= y_m \end{aligned}$$

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{y}$$

Regresión lineal

$$y = c_1 f_1(x) + c_2 f_2(x) + \cdots + c_n f_n(x)$$

$\boldsymbol{x}$	$\boldsymbol{y}$
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

➠

$$\begin{aligned} c_1 f_1(x_1) + c_2 f_2(x_1) + \cdots + c_n f_n(x_1) &= y_1 \\ c_1 f_1(x_2) + c_2 f_2(x_2) + \cdots + c_n f_n(x_2) &= y_2 \\ &\vdots \\ c_1 f_1(x_m) + c_2 f_2(x_m) + \cdots + c_n f_n(x_m) &= y_m \end{aligned}$$

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{y} \quad ➠ \quad \boldsymbol{c} = \left(\boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{y}$$

Regresión lineal

$$y = c_1 f_1(x) + c_2 f_2(x) + \cdots + c_n f_n(x)$$

x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_m	y_m

$\Rightarrow$

$$\begin{aligned} c_1 f_1(x_1) + c_2 f_2(x_1) + \cdots + c_n f_n(x_1) &= y_1 \\ c_1 f_1(x_2) + c_2 f_2(x_2) + \cdots + c_n f_n(x_2) &= y_2 \\ &\vdots \\ c_1 f_1(x_m) + c_2 f_2(x_m) + \cdots + c_n f_n(x_m) &= y_m \end{aligned}$$

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{y}$ $\Rightarrow$ $\boldsymbol{c} = \left(\boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{y}$ $\Rightarrow$ `c = linsolve(A,y)`

Codificando la regresión lineal

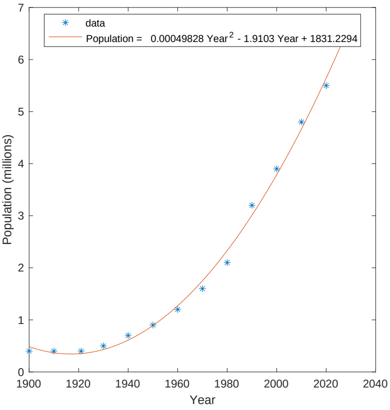
MATLAB

```
data = readtable('chiapas_population.csv');
x = data.Year;
y = data.Population;
A = [x.^2,x.^1,x.^0];
c = linsolve(A,y);
polinomio = @(x) c(1)*x.^2 + c(2)*x + c(3);

polinomio(2025)           % ans = 6.1701

x_test = (1900:2030)';
plot(x,y,'*',x_test,polinomio(x_test))
xlabel('Year')
ylabel('Population (millions)')
legend("data","Population =" + ...
    poly2str(c,'Year'))
yhat = polinomio(x);
r = y - yhat;             % residuos
RMSE = sqrt(mean(r.^2))   % error típico
```

Codificando la regresión lineal

MATLAB

```
data = readtable('chiapas_population.csv');
x = data.Year;
y = data.Population;
A = [x.^2,x.^1,x.^0];
c = linsolve(A,y);
polinomio = @(x) c(1)*x.^2 + c(2)*x + c(3);

polinomio(2025)          % ans = 6.1701

x_test = (1900:2030)';
plot(x,y,'*',x_test,polinomio(x_test))
xlabel('Year')
ylabel('Population (millions)')
legend("data","Population =" + ...
    poly2str(c,'Year'))
yhat = polinomio(x);
r = y - yhat;            % residuos
RMSE = sqrt(mean(r.^2))  % error típico
```



Una clase para regresión polinomial

polynom.m

```
classdef polynom < handle
    properties
        coeff
    end
    methods
        function obj = fit(obj,x,y,n)
            A = zeros(numel(y),n+1);
            for k = 0:n
                A(:,k+1) = x.^k;
            end
            obj.coeff = linsolve(A,y);
        end
        function y = predict(obj,x)
            y = zeros(size(x));
            n = numel(obj.coeff) - 1;
            for k = 0:n
                y = y + obj.coeff(k+1) * x.^k;
            end
        end
    end
end
```

MATLAB

```
x = data.Year;
y = data.Population;

model = polynom;
model.fit(x,y,2)

model.predict(2025)
xx = (1900:2030)';
yy = model.predict(xx);
plot(x,y,'*',xx,yy)
```

Regresión con las funciones del toolbox

MATLAB

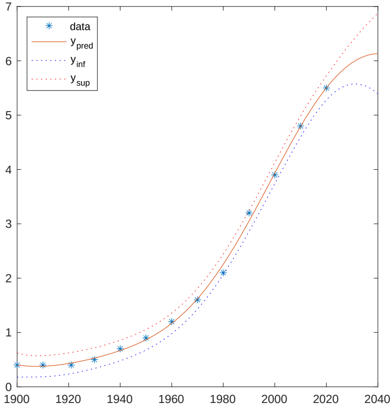
```
% Regresión lineal
model = fitlm(x,y,'quadratic');
model.predict(2025)                                 % ans = 6.1701

% Regresión lineal generalizada
model = fitglm(x,y,'quadratic');
model.predict(2025)                                 % ans = 6.1701

% Especificando la forma del modelo
modelspec = 'Population ~ 1 + Year + Year^2';
model = fitglm(data,modelspec)
model.predict(2025)                                 % ans = 6.1701
```

Regresión con procesos gaussianos

Es una regresión **no paramétrica** que permite hacer **predicciones con una estimación de la incertidumbre** asociada a esas predicciones.

MATLAB

```
data = readtable('chiapas_population.csv');
x = data.Year;
y = data.Population;
model = fitrgp(x,y);

disp("Población estimada para 2025:")
disp(model.predict(2025))
[m,s] = model.predict(2025);
disp("Intervalo de confianza:")
disp([m-1.96*s, m+1.96*s])

x_test = (1900:2040)';
[y_pred,s] = model.predict(x_test);
y_inf = y_pred - 1.96*s;
y_sup = y_pred + 1.96*s;

plot(x,y,'*',x_test,y_pred,...
    x_test,y_inf,'b:',x_test,y_sup,'r:')
legend('data','y_{pred}',...
    'y_{inf}','y_{sup}',Location='best')
```



Regresión multivariada con procesos gaussianos

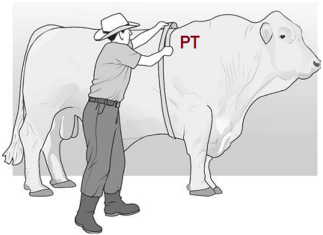
Considere el dataset **patients.xls**. Interpolando el peso a partir de la edad, la estatura y el género, ¿cuál sería el peso de un hombre de 51 años con una estatura de 180 cm?

MATLAB

```
cm2in = @(cm) cm/2.54;
lb2kg = @(lb) lb*0.454;

t = readtable('patients.xls');
Manhood = double(t.Gender=="Male");                      % masculinidad
model = fitrgp([t.Age,t.Height,Manhood],t.Weight);

lb2kg(model.predict([51,cm2in(180),1]))                  % ans = 81.7032

[m,s] = model.predict([51,cm2in(180),1])
disp("Peso esperado (kg): " + lb2kg(m))
disp("Intervalo de confianza al 95%:")
disp(lb2kg([m-1.96*s,m+1.96*s]))
```

Actividad 4: Regresión

Con el dataset **bovine.csv** generar un modelo de regresión para estimar el **peso vivo** (PV) en bovinos a partir de las mediciones de **perímetro torácico** (PT). Elaborar una tabla para PV desde 1.20 m hasta 1.80 m con incrementos de un centímetro en PT.




```
PT = model.predict(PT)
```

Clustering

Busca patrones ocultos en conjuntos de datos cuyas respuestas no están etiquetadas y permite explorar los datos cuando no se sabe qué información contienen.

Clustering

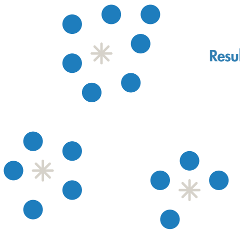
k-Means

How it Works

Partitions data into k number of mutually exclusive clusters. How well a point fits into a cluster is determined by the distance from that point to the cluster's center.

Best Used...

- When the number of clusters is known
- For fast clustering of large data sets



Result: Cluster centers

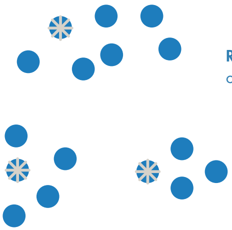
k-Medoids

How It Works

Similar to k-means, but with the requirement that the cluster centers coincide with points in the data.

Best Used...

- When the number of clusters is known
- For fast clustering of categorical data
- To scale to large data sets

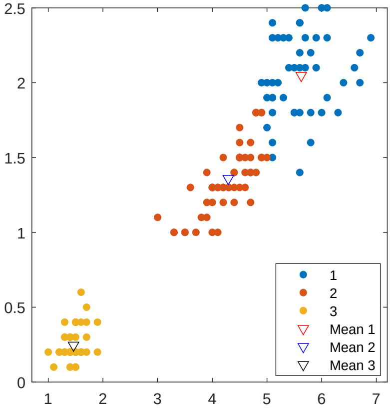


Result: Cluster centers that coincide with data points

Clustering: *k*-Means

MATLAB

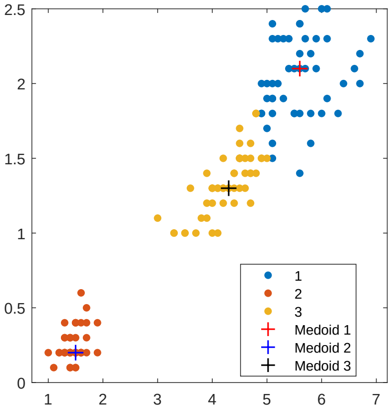
```
data = load('fisheriris.mat');
x = data.meas(:,3:4);
% PetalLength, PetalWidth

[idx,m] = kmeans(x, 3);
gscatter(x(:,1), x(:,2), idx)
hold on
plot(m(1,1), m(1,2), 'rv', ...
    DisplayName='Mean 1')
plot(m(2,1), m(2,2), 'bv', ...
    DisplayName='Mean 2')
plot(m(3,1), m(3,2), 'kv', ...
    DisplayName='Mean 3')
hold off
```



Clustering: *k*-Medoids

MATLAB

```matlab
data = load('fisheriris.mat');
x = data.meas(:,3:4);
% PetalLength, PetalWidth

[idx,m] = kmedoids(x, 3);
gscatter(x(:,1), x(:,2), idx)
hold on
plot(m(1,1), m(1,2), 'r+', ...
    DisplayName='Medoid 1', ...
    MarkerSize=10, LineWidth=1)
plot(m(2,1), m(2,2), 'b+', ...
    DisplayName='Medoid 2', ...
    MarkerSize=10, LineWidth=1)
plot(m(3,1), m(3,2), 'k+', ...
    DisplayName='Medoid 3', ...
    MarkerSize=10, LineWidth=1)
hold off
```



Modelo de mezcla gaussiana

MATLAB

```
data = load('fisheriris.mat');
X = data.meas(:,3:4);
gm = fitgmdist(X, 3);
i = gm.cluster(X);
clusterName = "Cluster " + i;
[X1,X2] = meshgrid(linspace(0,7,50),...
    linspace(0,3,50));
P = gm.pdf([X1(:),X2(:)]);
P = reshape(P, size(X1));
pcolor(X1, X2, P)
hold on
gscatter(X(:,1), X(:,2), clusterName)
shading interp
hold off
l = legend; l.Location = 'best';
axis([0,7,0,3])
```