

13 | 10/25

Término de aprendizaje de matrizes.

Punto 1

2025-2

1. Punto al modelo de regresión, buscamos resolver

diferentes modelos basados en el problema.

→ Definiciones como:

$f \in \mathbb{R}^N$ ; vector de salidas

$X \in \mathbb{R}^{N \times P}$ ; matriz de entradas

$\phi(X) \in \mathbb{R}^{N \times Q}$ ; matriz de características  
con  $\phi(x_n) \in \mathbb{R}^Q$

Finalmente  $w \in \mathbb{R}^Q$  como vector

de parámetros.

y presentamos el problema y su

de optimización o inferencia mediante  
asumencias dadas i.i.d.  
i) Minimizar cuadrados (O2S)

Buscamos minimizar el error cuadrático:

$$\text{f.g.: } J(w) = \|\Phi w - f\|^2$$

$$= (\Phi w - f)^T (\Phi w - f)$$

usando máxima verosimilitud log

el error gaussiano con media cero

$$\} \text{ variante de } f_n \sim N(\mu_n, 10^{-2})$$

(vi) Minimizar los residuos regularizados

(Ridge)

Se logra mediante regularización

$$f_{\text{fit}}: J(w) = \| \Phi w - f \|^2 + \lambda \| w \|^2$$

si  $\lambda > 0$  el parámetro lambda

de regularización debe ser menor a cero.

### (ii) Máxima verosimilitud (ML)

En este caso, asumimos

$$p(f|x, w) = \mathcal{N}(f| \bar{f}_w, \sigma^2 I)$$

que cada señal  $f_n$  es una muestra de una

distribución normal con media depende

de  $w$  y tiene varianza fija

y covarianza.

Una vez desembocada la de verosimilitud,

se maximiza con respecto a  $w$ , lo

que equivale a minimizar el error

cuadrático en OLS.

## IV) Máxima a posteriori (MAP)

En este caso, además del modelo de verosimilitud gaussiana, se introduce un conocimiento previo sobre los parámetros  $w$ , en forma de distribución  $f(w)$ .

$$f(w) = \mathcal{N}(w | 0, \alpha^{-1} I)$$

• Esto indica que, antes de observar los datos, se cree que  $w$  es una variable aleatoria con distribución gaussiana y centrada en cero.

De esta forma, en la distribución posterior

se obtiene el optimo bayesiano  $\hat{w}$ .

$$p(w) \propto p(\theta|w) p(w)$$

de forma que, y es igual a

Bayesian en el problema de optimización

bayesiano al de Ridge con  $\lambda = \sigma^2 \alpha$ .

# r) Regresión lineal bayesiana (Gaussian model)

Aquí, consideramos  $w$  como una variable

aleatoria con prior gaussiano y muestreo de verosimilitud gaussiana.

De la forma que:

$$\text{Prior : } w \sim N(0, \alpha^{-1} I)$$

$$\text{Likelihood : } p(t|w) = N(t | \Phi w, \beta^{-1} I)$$

De igual manera, la posterior sobre  $w$

es también gaussiana:

$$p(w|t) = N(w | m_w, s_w)$$

si  $m_w$  es el medio posterior

$s_w$  es la variancia posterior.

De esta forma, hemos visto, para una nueva entrada

$x^*$ , en predicción también que

una distribución normal y aleatoria.

Foto proporciona, además de una predicción puntual,

también una medida de incertidumbre.

## vi) Regresión Ridge Kernel (kernel ridge regression)

Este modelo permite modelar relaciones no lineales mediante el uso de funciones

Kernel. En lugar de usar directamente

los pesos  $w$ , expresamos el modelo

como una combinación de kernels:

$$f(x) = \sum_{n=1}^N \alpha_n k(x, x_n),$$

$$(w_n \text{ son los } \alpha = (K + \lambda I)^{-1} f)$$

$K$  es la matriz de Gram

construida sobre el Kernel  $k(x_i, x_j)$

## vii) Proceso gaussiano para regresión (GPR)

Este modelo bayesiano NO paramétrico

asume que, todo función  $f(x)$  es

distribuida como un proceso gaussiano

$$\text{f.g.: } f(x) \sim GP(0, k(x, x'))$$

Ahora, dado un conjunto de datos

se puede calcular la distribución condicional (predicción) para un nuevo punto.

$$\text{f.g.: } p(f_* | X_1, \dots, X_n) = N(f_*, \text{var}(f_*))$$

cuya predicción presenta

también una incertidumbre.



L. Con relación a los esquemas de  
 Regresión visto, discutimos algunos  
 características de rendimiento y formulación  
 matemática, de los algoritmos (Linear Reg.,  
 Lasso, elastic net, Kernel Ridge, etc...)

Regresión	Modelo matemático	Función de costo	Optimización	Ecuaciones de optimización	Finalización
-----------	-------------------	------------------	--------------	----------------------------	--------------

Lasso Reg.	$f = \Phi w + \gamma$	MSE (mean square error)	Análitica (closed-form)	O.L.S. ML (maximum likelihood)	Alta.
------------	-----------------------	----------------------------	----------------------------	--------------------------------------	-------

Lasso	Lineal + penalización de la norma 1. $f = \Phi w + \lambda \ w\ _1$	MSE + $\lambda \ w\ _1$	Gradiente descendente	Regularización L1	Media
-------	---	-------------------------	--------------------------	----------------------	-------

Elastic net	Lineal + penalización. $f = \Phi w + \lambda_1 \ w\ _1 + \lambda_2 \ w\ ^2$	MSE + $\lambda_1 \ w\ _1 + \lambda_2 \ w\ ^2$	Grad. descendente.	Interpolación Lasso - Ridge	Media
-------------	--	---	-----------------------	--------------------------------	-------

Kernel Ridge	$f(x) = \sum x_i k(x, x_i) + \lambda \ f - K\ ^2 + \lambda \ b\ ^2$	Dual (Kernel) /MAP.	Ridge kernelizado /MAP.	Ridge
--------------	---	---------------------------	----------------------------	-------

SGD Regressor	$f = \Phi w + \gamma$	MSE	Gradiente descendente iterativo O.S.	Alta
---------------	-----------------------	-----	---	------

Gaussian Ridge Ridge	$y = Xw + b$	Posterior sobre los pesos.	Regularización probabilística	Bayesiano Ridge	Media
-------------------------	--------------	-------------------------------	----------------------------------	--------------------	-------

· Gaussian Process  $f(x) \sim GP(0, k)$    
 Regresor   
 J. y razonamiento  
 negativo   
 Máximo  
 verosimilitud   
 - Proceso  
 Gausiano   
 - Limitada  
 por  $O(n^3)$

· Support vector regressor (SVR)  $f(x) = \sum_i (d_i + \hat{d}_i) k(x_i, x) + b$    
 $\begin{pmatrix} \min_{w, b, \epsilon, \hat{\epsilon}} \frac{1}{2} \|w\|^2 + \\ C \sum (\epsilon_i + \hat{\epsilon}_i) \end{pmatrix}$    
 · Programación  
 cuadrática (QP)   
 · Regulación  
 de margen; Media

· Random Forest  $\hat{f}(x) = \frac{1}{M} \sum_m h_m(x)$    
 Regresor   
 F. de voto por  
 cada nodo   
 $MSE = \frac{1}{N_{\text{muestra}}} \sum_i (y_i - \hat{y}_{im})^2$    
 · Bagging +  
 partición   
 · Ensemble no  
 paramétrico.   
 · Alta

Gradient Boosting  $\hat{f}(x) = \sum_{k=1}^K f_k(x)$    
 / XGBoost   
 $\sum_i L(y_i, \hat{y}_i) + \sum_k Q(f_k)$    
 $j: L(y_i, \hat{y}_i) = \frac{1}{2} (y_i - \hat{y}_i)^2$    
 $Q(f_k) = \sqrt{T + \frac{1}{2}} \lambda \sum_j w_j^2$    
 · Boosting por  
 gradientes   
 · Ensemble  
 aditivo   
 (Gradiente en el  
 espacio dimensional de  
 predictores)   
 · Muy  
 eficiente

### 3. RAPIDS (GPU accelerated data science)

Comenzamos definiendo. "RAPIDS . ai" y lo que nos ofrece ..

RAPIDS AI es un sistema de librerías de código abierto, esto gracias a que está enfocada en tener el desarrollo libre de software eficiente; esto gracias a permitir su aceleración por GPU, desarrollando para NVIDIA principalmente.

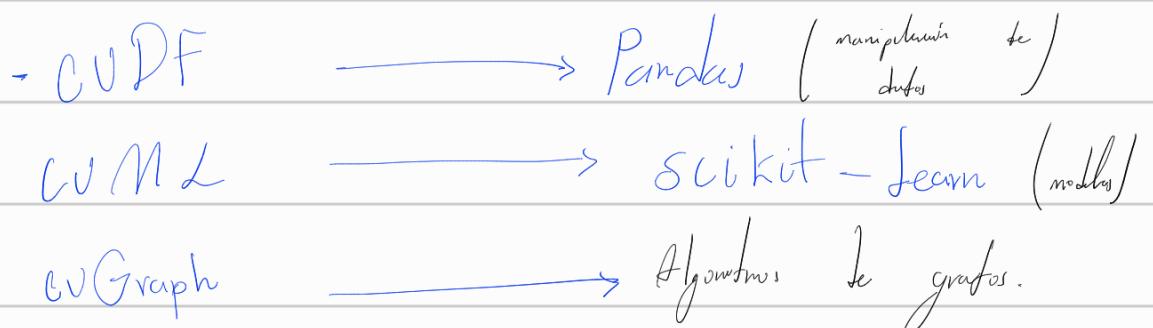
Alynnus cyntes:

- Está basado en CUDA que es una forma de para el desarrollo de aplicaciones preferidas por GPU.
- Reduce el tiempo de cálculo usando GPU.
- Debido a la alta demanda se implementaron los "métodos" ó modelos de aprendizaje de las librerías Scikit-learn y numpy/pandas para el manejo de datos y tensors.



Una vez se revisó la documentación sobre el RAPIDS, se vio que los signos de equivalencias con otros modelos definidos anteriormente como el linear, logístico, bayesiano, etc..

→ Vemos que, para acceder a los modelos definidos en la librería "Cuml", simplemente se accede al método de muestra clase.



La ventaja de CML reside en su paralelismo para datasets grandes.

- Finalmente, vemos una tabla comparativa entre diferentes modelos, su implementación en RAPIDS, y algunas de las hipótesis más relevantes.  
e.g., para los métodos linear regression, su implementación es mediante `Cuml - Linear Regression...`

# $\mathcal{L}_{\text{generalization}}$ CPV $\leftrightarrow$ GPV

Method	subset-learn	RAPIDS	Close	Hyperparameters
Linear Reg	Cvm L. Linear Regression	Ols	fit_intercept	
Ridge	Lemppänen por Lasso / elastic net	—	CD	alpha, max_iter, tol.
Lasso	Cvm L. Lasso	CD	alpha, l1_ratio	
Elasticnet	Cvm L. Elastic Net	CD	alpha, l1_ratio	
Kernel Ridge	X	—	—	—
SGD Regressor	Cvml. SGD	SGD CPV - equivalent	lr, epochs, penalty	
Bayesian Ridge	X	—	—	—
Gaussian Process Reg	X	—	—	—
SVR	Cvml. SVR	SMO	C_epsilon, kernel_gamma	
Random Forest Reg	Cvml. Random Forest Regressor	Bayesian GPV	n_estimators, max_depth	
Gradient Boosting Reg	Cvml. XGB Regressor	Boosting	learning_rate, max_depth, subsample	



A. Finalmente, trabajaremos en el nfl - log - data - bowl 2026 prediction, intentando implementar RAPIDS, una vez comprendido el dataset.

a)

- Entender que el objetivo es predecir el movimiento de jugadores durante partidos en jugadas de la NFL usando datos de seguimiento a 10 Hz.

u.1 Prever, Imponer las bases de datos para el EDA.

