

# Trabajo Práctico de Optimización no Lineal

## Introducción a Investigación Operativa y Optimización - Segundo Cuatrimestre 2025

La fecha límite de entrega es el 05/12 a las 23:59. En el caso de haber errores graves, podrán contar con una fecha de reentrega. Se recomienda utilizar como base el documento de Colab que se encuentra en este [link](#).

### Regresión Lasso

En esta sección utilizaremos el dataset `transfermarkt_fbref_201920.csv` con datos sobre gran variedad de atributos de jugadores de fútbol, para la temporada 2019-2020. El objetivo será implementar un modelo de regresión lineal para predecir el valor de un jugador a partir de sus atributos.

**Ejercicio 1.** Regresión Lineal Ordinaria.

1. Elegir dos métodos entre: método de gradiente con mitad de paso óptimo, gradiente conjugado o ADAM. Utilizarlos para hallar  $\beta$  y comparar su desempeño en relación a cantidad de iteraciones y tiempo de ejecución. Utilizar `k_max=5000` o inferior.
2. Para  $\beta$  hallado en el punto anterior, calcular el cociente de determinación ( $R^2$ ) para los datos de entrenamiento y de testeo. ¿Qué se observa? ¿Cómo lo interpretarían?
3. Interpretar  $\beta$ : ¿qué atributos benefician el valor del jugador? ¿Qué atributos lo perjudican? ¿Cuánta variabilidad hay entre las coordenadas de  $\beta$ ? Se pueden utilizar las funciones `etiquetas_beta` y `graficar_betas` de la plantilla del Colab.

**Ejercicio 2.** Regresión Lasso.

1. Primera adaptación de ADAM a Lasso: implementar una función `adam_penalizado` que aplique el método ADAM general, siguiendo el pseudocódigo visto en clase, añadiendo un argumento obligatorio `lambda` para el valor de regularización  $\lambda$  y que, en cada iteración, sume a `d` el subgradiente de  $\lambda\|\beta\|_1$ .
2. Aplicar `adam_penalizado` a la función:

$$f(\beta) = \sum_{i=1}^N \left( y^{(i)} - \sum_{j=1}^M \beta_j x_j^{(i)} \right)^2$$

con  $\lambda = 10$ ,  $\alpha = 0,01$  y 2000 iteraciones. ¿Es satisfactorio el tiempo de cómputo?

3. Segunda adaptación de ADAM a Lasso: identificar un cuello de botella en el desempeño de `adam_penalizado`. Implementar `lasso_adam`, que mejore el desempeño. Probar `lasso_adam` con  $\lambda = 10$ ,  $\alpha = 0,01$  y 2000 iteraciones. Comparar el tiempo de cómputo con la versión del ítem anterior.  
**Pista:** ¿podríamos aprovechar que nos interesa minimizar una función en particular?

4. Para el  $\beta$  obtenido usando `lasso_adam` con  $\lambda = 10$ , calcular el cociente de determinación ( $R^2$ ) para los datos de entrenamiento y de testeo. ¿Qué diferencia se observa respecto a Regresión Lineal?
5. Interpretar el  $\beta$  obtenido usando `lasso_adam` con  $\lambda = 10$ . ¿Qué atributos benefician y perjudican el valor de un defensor? ¿Cuánta variabilidad hay entre las coordenadas de  $\beta$ ? Comparar con la interpretación del  $\beta$  obtenido con Regresión Lineal. En cada método, ¿cuántos atributos tienen un peso con valor absoluto mayor que 0,05? (para esto se puede usar la función `etiquetas_beta_umbral`).
6. Implementar la función `lasso_coordenadas` que aplique minimización por coordenadas para hallar  $\beta$  de la Regresión Lasso, con  $\lambda = 10$ . Comparar la interpretación de  $\beta$  con la obtenida en el ejercicio anterior y el coeficiente de determinación para el conjunto de testeo.

**Ejercicio 3.** Realizar una Regresión Lasso para predecir el valor de los jugadores de otra(s) posición(es). Para esto, utilizar `lasso_adam` o `lasso_coordenadas`. Observar qué sucede para distintos valores de  $\lambda$  y, a partir de los  $\beta$  hallados, interpretar cuáles son las características que más influyen en el valor de los jugadores de esa posición.

## Máquina de soporte vectorial (*Support Vector Machine*)

En esta sección, el objetivo es implementar SVM para clasificación, adaptando algunos de los métodos de optimización que hemos visto en la cursada. Utilizaremos el dataset `fifa_players.csv` que contiene los atributos de jugadores de fútbol, extraídos del FIFA 18.

**Ejercicio 4.** Clasificando la posición de los jugadores.

1. Implementar `subgradiente_svm`, que calcule el subgradiente de:

$$f(w, b) = \lambda \frac{1}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(wx_i + b))$$

tomando como argumentos la matriz de atributos  $X$ , el vector de target  $y$ ,  $w$ ,  $b$  y el parámetro de regularización  $\lambda$ .

2. Implementar `metodo_gradiente_svm` y `adam_svm` que apliquen el Método del Gradiente con paso fijo y ADAM, respectivamente, para hallar los valores de  $w$  y  $b$ . El Método del Gradiente con paso fijo debe tomar como argumento la longitud del paso  $\alpha$ , con valor por defecto 0,01.
3. Aplicar ambos métodos para clasificar arqueros y delanteros en base de los atributos *dribbling* y *penalties*, utilizando `k_max` por lo menos 500. Para cada uno, visualizar el resultado del método y calcular la precisión de la clasificación para los conjuntos de entrenamiento y testeо. Probar con distintos valores para los parámetros. ¿ADAM converge siempre a la solución esperada? ¿Por qué creen que sucede esto?
4. Elegir otras dos posiciones<sup>1</sup> y otro par de atributos y repetir el ítem anterior. Para filtrar las posiciones y los atributos elegidos, pueden utilizar la función `filtrar_posicion_atributos`. Para visualizar los atributos elegidos, pueden usar `graficar`. No es necesario que los datos sean estrictamente linealmente separables. Pueden probar con distintas elecciones de posiciones y atributos.

---

<sup>1</sup>En este [link](#) pueden encontrar las definiciones de las posiciones