

Análisis Matemático para Inteligencia Artificial

Martín Errázquin (merrazquin@fi.uba.ar)

Especialización en Inteligencia Artificial

Clase 8

1 Optimización con restricciones

- Esquema general
- Optimización Convexa

2 *Optimización matemática*

- *Fused Logistic Regression*

3 Wrap-up

Optimización con restricciones

Optimización con restricciones de igualdad

Dado el problema de optimización con restricciones:

$$\begin{aligned} & \text{opt } f(x_1, \dots, x_n) \\ & \text{s.a. } g_1(x_1, \dots, x_n) = 0 \\ & \quad \vdots \\ & \quad g_m(x_1, \dots, x_n) = 0 \end{aligned} \tag{1}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ con $i = 1, \dots, m$.

Se denomina **función Langrangiana** a la función de $n + m$ variables \mathcal{L} definida por:

$$\mathcal{L}(\bar{\lambda}, \bar{x}) = f(\bar{x}) + \lambda_1 g_1(\bar{x}) + \dots + \lambda_m g_m(\bar{x}), \quad \text{con } \bar{\lambda} = (\lambda_1, \dots, \lambda_m)$$

Teorema de condición necesaria de Lagrange

Sea $D \subseteq \mathbb{R}^n$ abierto. Sea (1) con $m < n$, donde f, g_i $i = 1, m$ son funciones definidas de D en \mathbb{R} , con derivadas parciales primeras continuas en D y

$$B = \{\bar{x} \in D : g_i(\bar{x}) = 0, i = 1, m\}$$

el conjunto de soluciones factibles.

Entonces, si \bar{x}^* es un óptimo local de (1) tal que la matriz jacobiana de $\bar{g}(\bar{x}^*)$ verifica $|J\bar{g}(\bar{x}^*)_m| \neq 0$, existen m números reales $\lambda_1^*, \dots, \lambda_m^*$ tales que son solución del sistema:

$$\nabla f(\bar{x}^*) + \sum_{i=1, m} \lambda_i^* \nabla g_i(\bar{x}^*) = \bar{0}$$

Las soluciones factibles que verifican esta ecuación se denominan *puntos estacionarios*.

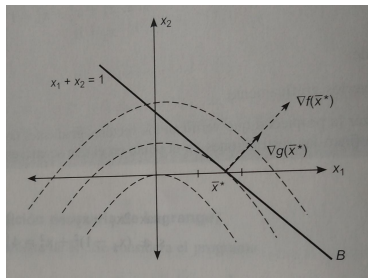
Los números $\lambda_1, \dots, \lambda_m$ se denominan **multiplicadores de Langrange** asociados a las m restricciones en el punto \bar{x}^* .

Ejemplo

$$\begin{aligned} \min f(x_1, x_2) &= x_1^2 + 2x_2 \\ \text{s.a. } x_1 + x_2 &= 1 \end{aligned}$$

Así definimos:

$$g(x_1, x_2) = x_1 + x_2 - 1.$$



- El conjunto de soluciones factibles es la recta $x_2 = -x_1 + 1$.
- Las curvas de nivel de la función $f(x_1, x_2) = x_1^2 + 2x_2$ son parábolas de la forma $x_1^2 + 2x_2 = k, k \in \mathbb{R}$.

$$\therefore \bar{x}^* = (1, 0) \text{ y } f(\bar{x}^*) = 1.$$

Calculamos los vectores gradientes: $\nabla f(x_1, x_2) = (2x_1, 2)$ y $\nabla g(x_1, x_2) = (1, 1)$, ambos vectores son l.d. en \bar{x}^* .

Optimización con restricciones de desigualdad

Condiciones necesarias de primer orden de Fritz-John

Dado el problema de optimización con restricciones:

$$\begin{aligned} \text{opt } & f(x_1, \dots, x_n) \\ \text{s.a. } & g_1(x_1, \dots, x_n) \leq 0 \\ & \vdots \\ & g_m(x_1, \dots, x_n) \leq 0 \end{aligned} \tag{2}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ con $i = 1, \dots, m$. Sea \bar{x}^* un punto tal que $I = \{i : g_i(\bar{x}^*) = 0\}$, f, g_i diferenciables en \bar{x}^* . Entonces, si $g_i, i \notin I$ son continuas en \bar{x}^* se verifica que es solución y existen escalares $\lambda_0, \lambda_i, i \in I$ no todos nulos tales que:

$$\lambda_0 \nabla f(\bar{x}^*) + \sum_{i=1, m} \lambda_i \nabla g_i(\bar{x}^*) = \bar{0}$$

para $0 \leq \lambda_0, \lambda_i, i \in I$, y $g_i(\bar{x}^*) \leq 0, i = 1, m$.

Dualidad en Optimización: Problema dual

Dado el problema primal:

$$\begin{aligned} \min f(x_1, \dots, x_n) \\ \text{s.a. } g(x_1, \dots, x_n) \leq b \\ x_i \geq 0 \end{aligned}$$

Tenemos el problema dual:

$$\begin{aligned} \max D(\lambda_1, \dots, \lambda_m) \\ \text{s.a. } h(\lambda_1, \dots, \lambda_m) \geq b \\ \lambda_j \leq 0 \end{aligned}$$

- Se define $D(\lambda_1, \dots, \lambda_m) \doteq \inf_{\bar{x}} \mathcal{L}(\bar{\lambda}, \bar{x})$.
- Si \bar{x}^* es una solución factible del problema primal e $\bar{\lambda}^*$ es una solución del dual entonces $f(\bar{x}^*) \geq D(\bar{\lambda}^*)$.
- Si un problema no tiene un óptimo finito entonces el otro no es factible.

Ejemplo (1/2)

Sea

$$\begin{aligned} \min \quad & 3x_1 + 4x_2 + 5x_3 \\ \text{s.a.} \quad & -2x_1 - 1x_2 - x_3 + 7 \leq 0 \\ & -x_1 - 3x_2 - 2x_3 + 8 \leq 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

El lagrangiano resulta:

$$\mathcal{L}(\bar{\lambda}, \bar{x}) = 3x_1 + 4x_2 + 5x_3 + \lambda_1(-2x_1 - 1x_2 - x_3 - 7) + \lambda_2(-x_1 - 3x_2 - 2x_3 - 8)$$

de donde para que $D(\lambda_1, \lambda_2)$ no pueda valer $-\infty$ los términos de x_1, \dots, x_3 deben ser no negativos, resultando en el sistema de inecuaciones:

$$\begin{cases} 3 - 2\lambda_1 - \lambda_2 \geq 0 \\ 4 - \lambda_1 - 3\lambda_2 \geq 0 \\ 5 - \lambda_1 - 2\lambda_2 \geq 0 \end{cases}$$

Ejemplo (2/2)

Luego, como el mínimo de \mathcal{L} se da para $x_1 = x_2 = x_3 = 0$, resulta $D(\lambda_1, \lambda_2) = \mathcal{L}(\bar{\lambda}, \bar{0}) = 7\lambda_1 + 8\lambda_2$.

Teniendo entonces las formulaciones de D y las restricciones, reformulamos el problema:

$$\begin{aligned} \max \quad & 7\lambda_1 + 8\lambda_2 \\ \text{s.a.} \quad & 3 - 2\lambda_1 - \lambda_2 \geq 0 \\ & 4 - \lambda_1 - 3\lambda_2 \geq 0 \\ & 5 - \lambda_1 - 2\lambda_2 \geq 0 \\ & \lambda_1, \lambda_2 \geq 0 \end{aligned}$$

Nota: en estos casos de problemas lineales se da la condición de *dualidad fuerte*, por la cual el óptimo del dual coincide con el óptimo del primal.

Función Convexa

Def: Un subconjunto $S \subset \mathbb{R}^n$ es **convexo** si para cada par de puntos $x, y \in S, \alpha \in [0, 1]$ se verifica que $z = \alpha x + (1 - \alpha)y \in S$.

Obs: X_1, X_2 son dos conjuntos convexos, entonces,

- $X_1 \cap X_2$ es convexo.
- $X_1 + X_2 = \{x_1 + x_2 \in \mathbb{R}^n : x_1 \in X_1, x_2 \in X_2\}$.
- si L es una transformación lineal, $L(X_1)$ es convexa.

Función Convexa

Def: Sea $M \subset \mathbb{R}^n, M \neq \emptyset$ convexo, $f : M \rightarrow \mathbb{R}$. Entonces se dice que:

- f es convexa en M sii $\forall x, y \in M, \forall \alpha \in [0, 1]$ se verifica que:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

Desigualdad de Jensen

- f es estrictamente convexa en M sii $\forall x, y \in M, \forall \alpha \in (0, 1)$ se verifica que:

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y)$$

Condiciones para convexidad de funciones diferenciables

Sea $M \subset \mathbb{R}^n$, $M \neq \emptyset$ abierto convexo, $f : M \rightarrow \mathbb{R}$ diferenciable, se dice que:
 f es convexa en $M \Leftrightarrow \forall x, y \in M, f(y) \geq f(x) + \nabla f(x)(y - x)$
O bien, $(\nabla f(y) - \nabla f(x))(y - x) \geq 0$

Prop: Sea $M \subset \mathbb{R}^n$, $M \neq \emptyset$ abierto convexo, $f : M \rightarrow \mathbb{R}$ $f \in \mathcal{C}^2$. Entonces,
 f es convexa en M si $\forall x \in M, y^T Hf(x)y \geq 0$, para cualquier $y \in \mathbb{R}^n$.

Condiciones Suficientes de Optimalidad Global

$$\begin{aligned} & \text{opt } f(x_1, \dots, x_n) \\ & \text{s.a. } g_1(x_1, \dots, x_n) = 0 \\ & \quad \vdots \\ & \quad g_m(x_1, \dots, x_n) = 0 \end{aligned}$$

con $m < n$ donde f y $g_i, i = 1, \dots, m$ son funciones \mathcal{C}^1 en un conjunto abierto $D \in \mathbb{R}^n$. Entonces se verifica que si f es convexa en el conjunto de soluciones factibles B y, las funciones g_i son lineales, **todos los puntos estacionarios son mínimos globales.**

Programación Lineal

Un problema de programación lineal es donde tanto la función objetivo como las funciones que definen las restricciones son lineales. La forma general es:

$$\begin{aligned} & \text{opt } c_1x_1 + \dots + c_nx_n \\ & \text{s.a. } a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ & \quad \vdots \\ & a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ & d_{11}x_1 + \dots + d_{1n}x_n \geq e_1 \\ & \quad \vdots \\ & d_{r1}x_1 + \dots + d_{rn}x_n \geq e_r \\ & g_{11}x_1 + \dots + g_{1n}x_n = b_1 \\ & \quad \vdots \\ & g_{s1}x_1 + \dots + g_{sn}x_n = b_s \end{aligned}$$

Propiedades de la Programación Lineal

Definición: Sea $S \subset \mathbb{R}^n$ convexo $S \neq \emptyset$, $\bar{x}^* \in S$ es un punto extremo de S si \bar{x}^* no puede expresarse como combinación lineal convexa de puntos de S distintos de él.

- 1 Es un problema convexo ya sea de minimización o maximización.
- 2 La solución óptima, si existe, es global.
- 3 Nunca existen óptimos locales que no sean globales.
- 4 Puede tener o no solución, en caso de existir se encuentra en único punto o en infinitos.

Objetivo: Se busca hallar $\min \bar{c}\bar{x}$, s.a. $A\bar{x} = \bar{b}$; $\bar{x} \geq 0$,
donde $A \in \mathbb{R}^{m \times n}$, $m < n$, $rg(A) = m$.

Métodos estándar de solución: Algoritmo Simplex y revised Simplex, Interior Point (IPM).

Programación cuadrática

En el caso de que la función objetivo sea una cuadrática convexa:

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.a.} \quad & A x \leq b \end{aligned}$$

donde $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^d$.

La matriz $Q \in \mathbb{R}^{d \times d}$ es simétrica y definida positiva, y por lo tanto la función objetivo es convexa. El langrangiano está dado por:

$$L(x, \lambda) = \frac{1}{2} x^T Q x + c^T x + \lambda^T (A x - b)$$

Para obtener $D(\lambda) = \min_x L(x, \lambda)$ buscamos x derivando L y despejando:

$$\nabla L(x) = Qx + (c + A^T \lambda) = 0 \Rightarrow x = -Q^{-1}(c + A^T \lambda)$$

Y substituyendo obtenemos la lagrangiana dual:

$$D(\lambda) = -\frac{1}{2}(c + A^T \lambda)^T Q^{-1}(c + A^T \lambda) - \lambda^T b.$$

Optimización matemática

El problema

Se tiene un vector de múltiples features, todas categóricas, al cual se le aplica un [CatBoost Encoding](#) para obtener "embeddings" de cada valor. Luego, se aplica una [regresión logística](#).

En producción, el modelo se despliega en un mini servidor donde debe realizar predicciones de verdadero-falso para *una observación a la vez*.

Debido a limitaciones de la librería Category Encoders, que asume un DataFrame como entrada, el código se ve aproximadamente así:

```
def predict(self, input_dict):  
    x = to_dataframe(input_dict)  
    x = self.encoder.transform(x)  
    prediction = self.regressor.predict(x)  
    return prediction > self.threshold
```

Y esto tiene (en benchmark) una latencia promedio de 39 ms.

Para cada feature $i = 1, \dots, p$, el encoder construye un dataframe con dos columnas: COUNT y SUM. La regla de transformación es:

$$\tilde{X}(X) = \begin{cases} \frac{SUM(X) + \mu \cdot a}{COUNT(X) + a} & \text{si } COUNT(X) > 1 \\ \mu & \text{si } COUNT(X) = 1 \end{cases}$$

donde $\mu = \bar{y}$ es la media global y $a = 1$ es un hp. de regularización.

Luego, pasando a la regresión logística:

- ❶ cada feature i es multiplicada por su peso β_i : $s = \sum_{i=1}^p \beta_i \cdot \tilde{X}_i$
- ❷ se suma el intercept: $z = s + z_0$
- ❸ se aplica la función logística $\sigma(t) = \frac{1}{1+e^{-t}}$: $\hat{y} = \sigma(z)$
- ❹ se devuelve la comparación con un threshold aprendido $pred = \hat{y} > \theta$

- 1 Se pueden usar diccionarios comunes \rightarrow se puede evitar instanciar un DataFrame.
- 2 Si $\tilde{X} = \mu$ cuando $COUNT = 1$, no tiene sentido diferenciarlo de $COUNT = 0$ (cuando no está presente) $\rightarrow \tilde{X}$ deja de ser partida.
- 3 Si no se pregunta por el valor de $COUNT$ no tiene sentido guardar SUM y $COUNT$ separados \rightarrow se puede guardar el embedding \tilde{X}_i precomputado.
- 4 Cada feature tiene un mapeo propio \rightarrow en vez de \tilde{X}_i se puede guardar precomputado $\beta_i \cdot \tilde{X}_i$ (aunque se debe guardar un $\beta_i \cdot \mu_i \forall i$).
- 5 Para un $\theta \in (0, 1)$ fijo $\exists \sigma^{-1}(\theta) = \text{logit}(\theta)$, luego $\sigma(z) > \theta \Leftrightarrow z > z_\theta \rightarrow$ se puede precomputar $\text{logit}(\theta)$ y evitar computar $\sigma(z)$.
- 6 Separando $z = s + z_0$ tenemos $z > \text{logit}(\theta) \Leftrightarrow s > \text{logit}(\theta) - z_0 \rightarrow$ podemos precomputar $z_\theta = \text{logit}(\theta) - z_0$ y evitar computar esa suma.

El resultado

Tras la drástica reducción en operaciones, tenemos que en total el modelo está realizando solamente un dict lookup por cada feature (inevitable con categóricas), una suma y una comparación.

El código se ve aproximadamente así:

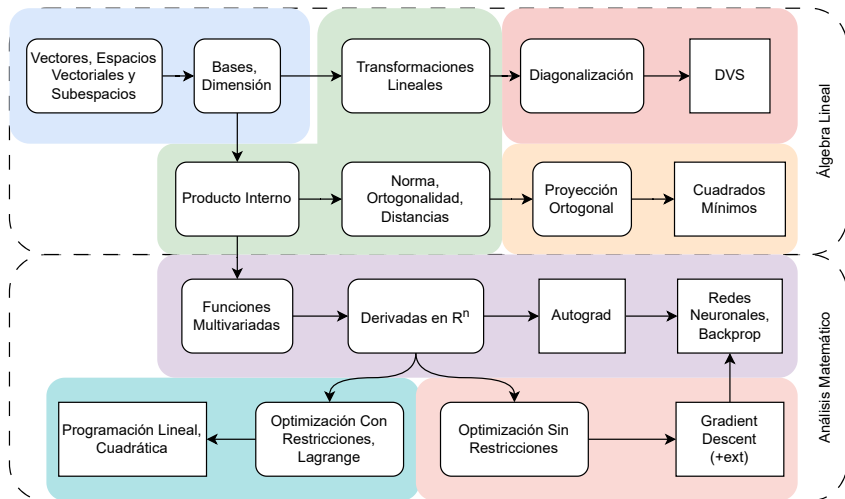
```
def predict(self, input_dict):  
    s = sum(  
        self.embeddings[feature].get(value, self.mus[feature])  
        for feature, value in input_dict.items()  
    )  
    return s > self.z_threshold
```

Y esto tiene (en benchmark) una latencia promedio de 0.01 ms, es decir **un speedup de 3800x**. En el caso de tiempos máximos, la versión original fue de 192 ms vs 0.487 ms del fused.

Como extra, el modelo fused ocupa la mitad de espacio en disco.

Wrap-up

Lo recorrido



¿Qué buscamos?

- Objetivo primario 1: proveer el sustento teórico matemático requerido para poder comprender la formulación de los modelos clásicos.
- Objetivo primario 2: entrenar lectura e interpretación de lenguaje matemático formal y avanzado.
- Objetivo secundario: cuando sea posible, mostrar aplicaciones directas de la teoría vista.
- Bonus: concientizar sobre importancia y ventajas de optimización en tiempo y memoria, y cómo la matemática es la vía para la misma.

POR FAVOR revisen que tengan **todas** las autoevaluaciones **aprobadas**. Recuerden que ustedes son los responsables de su aprobación.

Una vez que llega el deadline se descargan las notas al momento y por más que sigan abiertas las autoevaluaciones los resultados de las mismas **no son tenidos en cuenta**.

Recordar que está disponible la encuesta de clase! Completarla es cortito y sirve para ir monitoreando el estado del curso.

¿Dónde encontrarla? En la hoja de notas (e.g. "Notas CEIA 10Co2024"), abajo de todo, junto al link de las grabaciones de las clases.