Análisis Matemático para Inteligencia Artificial

Verónica Pastor (vpastor@fi.uba.ar),
Martín Errázquin (merrazquin@fi.uba.ar)

Especialización en Inteligencia Artificial

12/05/2023

Colab

El mundo del Data Science y las competencias de Machine Learning cobraron mucha relevancia con la competencia de Netflix, que prometía **1M USD** a quien pudiera obtener una performance de un 10% por sobre su algoritmo de base para el problema de recomendación de películas.

En este contexto surgió una nueva variante de modelo en base a factorización de matrices.

¿Lo interesante? Con las dos clases de AM ya nos alcanza para entenderlo!

¿Por qué funciona la inversa? (1/2)

Demostremos que: $Ax = b \Rightarrow$ si $\exists A^{-1} \Rightarrow x^* = A^{-1} \cdot b$

Si $A \in \mathbb{R}^{n \times n}$ es una matriz definida positiva, entonces es invertible:
Dem.: (por método del absurdo)

Supongamos que el determinante de A es cero, y llegaremos a una solución absurda.

Si $det(A) = 0$ y queremos resolver un sistema $Ax = 0$ significa que además de la solución trivial hay infinitas soluciones, es decir $rango(A) < n$, entonces

$$\exists \tilde{x} \neq 0 : A\tilde{x} = 0 \Rightarrow \tilde{x}^T A \tilde{x} = \tilde{x}^T 0 = 0$$

y entonces no se cumple que $\tilde{x}^T A \tilde{x} > 0$.

↳ pero entonces A no es def. +

↳ ABSURDO

↳ q.e.d.

¿Por qué funciona la inversa? (2/2)

$<x,y> = x^T y$

Demostremos que:

$A = x \cdot x^T + \lambda I_k$ **es definida positiva**, es decir, $y^T A y > 0, \ \forall y \in \mathbb{R}^k - \{0\}$

$<y,x><x,y> \quad \lambda <y,y>$

$(y^T x)(x^T y) \quad \lambda (y^T y)$

Dem.: Sea $y \in \mathbb{R}^k - \{0\}$

$$y^T(x \cdot x^T + \lambda I_k)y = y^T x x^T y + y^T \lambda I_k y$$

(recordemos que el p.i. es $< u, v >= u^T v$), prop. asociativa,

$$< y, x >< x, y > + \lambda < y, y > = \underbrace{< x, y >^2}_{\geq 0} + \lambda ||y||^2 > 0.$$

$\therefore x \cdot x^T + \lambda I_k$ es definida positiva.

por p.i. $y \neq \vec{0} \Rightarrow ||y||^2 > 0$

Proyección

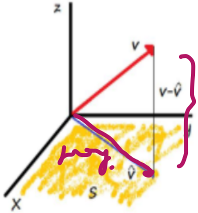
Sea $\mathbb{V}$ un EV y $S \subset \mathbb{V}$ un SEV. Una transformación lineal $\Pi : \mathbb{V} \to S$ es una proyección si $\Pi^2 = \Pi \circ \Pi = \Pi$
Esto significa que la matriz de transformación asociada a la proyección cumple con la propiedad de idempotencia: $[\Pi]^2 = [\Pi]$.

Proyección Ortogonal

Dado $\mathbb{V}$ un EV con p.i. y $S \subset \mathbb{V}$ un SEV, el objetivo es dado $v \in \mathbb{V}$ hallar $\tilde{v} \in S$ que sea "lo más parecido posible" a v.

$$\tilde{v} \in S : \tilde{v} = arg\ min_{s \in S} ||v - s||$$

Además vale que $\langle v - \tilde{v}, s \rangle = 0, \ \forall s \in S$



lo que "sobra"

Obs: La ortogonalidad de la proyección tiene que ver con el p.i. que se use.

Teorema de proyección

Sea $\mathbb{V}$ un EV de dimensión finita con p.i. $\langle .,. \rangle$, S un SEV. Dado $v \in \mathbb{V}$ existe un único $\tilde{v} \in S$ tal que

$$||v - \tilde{v}|| \leq ||v - u||, \quad \forall u \in \cancel{\mathbb{V}} \ \ S$$

¿Cómo hallar la proyección?

Sea $\mathbb{V}$ un EV de dimensión n con p.i. $\langle .,. \rangle$, y $S \subset \mathbb{V}$ un SEV, $dim(S) = m \geq 1$, y sea $B = \{s_1, ..., s_m\}$ una BON de S. Buscamos encontrar la proyección de $\tilde{v} \in S$ de $v \in \mathbb{V}$ ($\tilde{v} = \Pi_S(v)$).

Como $\tilde{v} \in S$, $\tilde{v} = \sum_{i=1}^{m} \alpha_i s_i \ \Rightarrow$ busco los coeficientes que minimizan $||v - \sum_{i=1}^{m} \alpha_i s_i||$. El problema puede escribirse como:

$$\Pi_S(v) = \sum_{i=1}^{m} \alpha_i s_i = B\alpha, \quad B = [s_1, ..., s_m], \ \alpha = [\alpha_1, ..., \alpha_m]^T$$

¿Cómo hallar la proyección?

Como por definición $\langle v - \Pi_S(v), s\rangle = 0$, $\forall s \in S$, debo resolver el siguiente sistema de ecuaciones:

(B)

"el resto" $\perp$ cada vector de la BON de S

$$\left(\begin{array}{ccc} | & & | \\ s_1 & \cdots & s_m \\ | & & | \end{array}\right)^T = B^T$$

$$\begin{cases} \langle \underbrace{v - \Pi_S(v)}_{\text{"el resto"}}, s_1\rangle = 0 \\ \vdots \\ \langle v - \Pi_S(v), s_m\rangle = 0 \end{cases} \Rightarrow \begin{cases} s_1^T(v - B\alpha) = 0 \\ \vdots \\ s_m^T(v - B\alpha) = 0 \end{cases} \Rightarrow \begin{pmatrix} -s_1^T- \\ \vdots \\ -s_m^T- \end{pmatrix}(v - B\alpha) = 0$$

$$\underbrace{B^T(v - B\alpha) = 0} \Leftrightarrow B^T v = B^T B\alpha \Leftrightarrow \alpha = (B^T B)^{-1} B^T v$$

$\Pi_S(v) = B \cdot \Pi_S \cdot v$

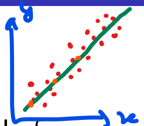
$$\Pi_S = B(B^T B)^{-1} B^T$$

Observación: Si B es una BON entonces $P_{\Pi_S} = BB^T$.

Aplicación: Cuadrados Mínimos

$\begin{pmatrix} - x_1 - \\ \vdots \\ - x_m - \end{pmatrix}$ # fila > # cols.

obs > # mediciones



Supongamos que tenemos un sistema sobredeterminado de la forma:

$$Ab = y,\ A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^n,\ y \in \mathbb{R}^m,\ m > n.$$

b es la solución de cuadrados mínimos

Como $m > n$, puede que no exista b que satisfaga todas las m ecuaciones, entonces busco la solución que más se acerque (busco $Proy_{Col(A)}y$)

$$b = (A^T A)^{-1} A^T y \ \Rightarrow \ P = A(A^T A)^{-1} A^T$$

Colab

Ya vimos que el ajuste por cuadrados mínimos funciona en la teoría, ahora veámoslo funcionar en la práctica!