Análisis Matemático para Inteligencia Artificial

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

Especialización en Inteligencia Artificial

13/5/2022

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)

Recordar A simétrica y $x^T A x > 0$

Demostremos que:

Si $A \in \mathbb{R}^{n \times n}$ es una matriz definida positiva, entonces es invertible:

Dem.: (por método del absurdo) A tiene inversa $\Leftrightarrow \det(A) \neq 0$

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

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

$$\begin{bmatrix} *x & *y & *z \\ 0 & *y & *z \\ 0 & 0 & 0 \end{bmatrix} \Big\} \; r(A) < n$$

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

¿Por qué funciona la inversa? (2/2)

Demostremos que:

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

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

A

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

$\lambda y^T y$

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

$$< y, x >< x, y > +\lambda < y, y >=< x, y >^2 + \lambda ||y||^2 > 0.$$

→prop $<y,x> = <x,y>$

>0 >0

>0 por el modelo

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

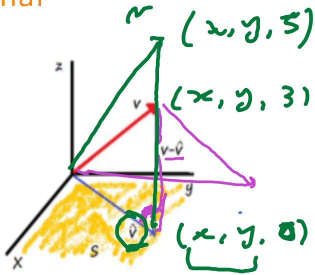
Proyección Ortogonal

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 $\Pi^2 = \Pi \circ \Pi = \Pi$ $\pi(\pi(v)) = v$ $v \to \pi(v)$
Esto significa que la matriz de transformación asociada a la proyección cumple con la propiedad de idempotencia: $[\Pi]^2 = [\Pi]$. $A . A = A$

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

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

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

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

¿Cómo hallar la proyección?

Sea $\mathbb{V}$ un EV de dimensión $\overset{4}{n}$ con p.i. $\langle .,. \rangle$, y $S \subset \mathbb{V}$ un SEV, $m<n$
$dim(S) = \overset{3}{m} \geq 1$, 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:

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

matriz

$s_i \in \mathbb{V}$

$$\begin{array}{cc} & m \\ n & \begin{bmatrix} \downarrow & \downarrow & \downarrow \\ & & \\ \end{bmatrix} \end{array}$$

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

$B \in K^{n \times m}$

$B^T \in K^{m \times n}$

$$\left\{\begin{array}{c} \langle v - \Pi_S(v), s_1\rangle = 0 \\ \vdots \\ \langle v - \Pi_S(v), s_m\rangle = 0 \end{array}\right. \Rightarrow \left\{\begin{array}{c} s_1^T(v - B\alpha) = 0 \\ \vdots \\ s_m^T(v - B\alpha) = 0 \end{array}\right. \Rightarrow \overbrace{\begin{pmatrix} -s_1^T- \\ \vdots \\ -s_m^T- \end{pmatrix}}^{B^T} (v - B\alpha) = 0$$

$s_i \cdot s_j = \begin{cases} 1 \\ 0 \end{cases}$

$$B^T(v - B\alpha) = 0 \Leftrightarrow B^T v = B^T B\alpha \Leftrightarrow \alpha = (\underbrace{B^T B}_{\in K^{m \times m}})^{-1} B^T v$$

La matriz de proy.: $[P_{\Pi_S}] = B\underbrace{(B^T B)^{-1} B^T}_{\alpha}$ $\qquad [\Pi_S] \cdot v =$

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

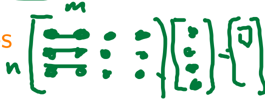
Aplicación: Cuadrados Mínimos

→ más ec. que incógnitas

Supongamos que tenemos un sistema sobredeterminado de la forma:

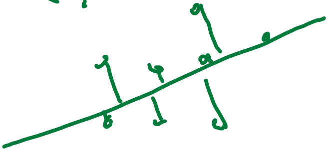
$$Ab = y, \; A \in \mathbb{R}^{n \times m}, \; b \in \mathbb{R}^m, \; y \in \mathbb{R}^m, \; m > n.$$

y 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 $Proy_{Col(A)} y$)

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



Colab

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