Regresia liniară

(AI notes)

Problemele de regresie identifică relația de dependență dintre datele de ieșire și datele de intrare ale unei probleme (pe baza unor caracteristici a datelor, se dorește prezicerea unor valori asociate acestor date).

Valorile prezise sunt de tip continuu.

Există relații de dependență: liniare/neliniare.

Exemple:

- predicția acțiunilor la bursă în funcție de anumiți indicatori economici
- predicția consumului de înghețată în funcție de temperatura și de numărul de copii dintr-o tabără

Regresia liniară - model liniar care presupune că variabila de ieșire poate fi calculată ca o combinație a variabilelor de intrare.

Datele se caracterizează prin **atribute** (x = (x1, x2, ..., xn)) și **output** (atât atributele cât și outputul sunt valori numerice continue).

Regresorul (modelul liniar de predicție): y = f(x, w) = w0 + w1 * x1 + w2 * x2 + ...

forma din liceu: y = f(a, b) = b0 + b1 * a1 + b2 * a2 + ...

Metodologia rezolvării unei probleme de regresie (liniară):

Antrenare

> Input: un set de exemple etichetate (x^i, y^i), cu $i \in \{1, 2, \ldots, trainDataSize\}$, x^i - vectorul de atribute asociate unui exemplu, y^i etichetata asociata exemplului x^i (valoare numerica reala/float)

> Output: un model de regresie = regresor (adica valorile optime ale coeficientilor w din ecuatia de regresie $f(x, w) = w_0 + w_1 * x_1 + w_2 * x_2 + \ldots$)

Algoritm:

- **LeastSquare** (metoda celor mai mici pătrate)
- **GradientDescent**

Testare

> Input: un exemplu ne-etichetat (x_{new}), cu x_{new} - vectorul de atribute asociate acelui exemplu

> Output: valoarea prezisa pentru exemplul x_{new}

> Algoritm: Folosirea regresorului invatat (a coeficientilor) pentru a calcula valoarea outputului
>
> $$y_{new} = f(x_{new}, w)$$

Metoda celor mai mici pătrate

Presupunem cazul unei regresii univariate (un exemplu are un singur atribut), deci $f(x, w) = w_0 + w_1 * x_1$. Se dorește identificarea valorilor optime pentru coeficienții **w = [w0, w1]**, știindu-se un set de **n** exemple de antrenament de forma (x^i, y^i), cu $i = 1, 2, \ldots, n$, $x^i = (x_1^i)$.

Se definește o funcție de cost:

$$cost(x) = \sum_{i=1}^{n} (y^i_{computed} - y^i)^2$$

Se identifică punctul de minim al acestei funcții, care duce la valorile optime pentru **w**:

$$w_1 = \frac{n \sum_{i=1}^{n} (x^i * y^i) - \sum_{i=1}^{n} x^i * \sum_{i=1}^{n} y^i}{n \sum_{i=1}^{n} (x^i)^2 - (\sum_{i=1}^{n} x^i)^2}$$

$$w_0 = \frac{\sum_{i=1}^{n} y^i - w_1 \sum_{i=1}^{n} x^i}{n}$$

Pentru cazul unei regresii multivariate (m atribute):

$f(x, w) = w_0 + w_1 * x_1 + w_2 * x_2 + \ldots w_m * x_m)$

și un set de date cu **n** exemple:

$\boldsymbol{w} = (X^T X)^{-1} X^T Y$, unde $X = (x_j^i)$, $Y = (y^i)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

Evaluarea performanței regresorului:

- the absolute difference (L1 distance):

$$Error = \frac{1}{n} \times \sum_{i=1}^{n} |y^i - y^i_{computed}| = MeanAbsoluteError(MAE)$$

- the square difference (L2 distance):

$$Error = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (y^i - y^i_{computed})^2} = RootMeanSquareError(RMSE)$$

Pași în rezolvarea unei probleme de regresie:

- plot pentru distribuția datelor & plot pentru "verificarea" liniarității (dacă legătura dintre y și x este liniară)
- împărțire date pe train și test
- învățare model (cu tool generic și cu tool de least square)
- plot rezultate
- calcul metrici de performanță