3.3. Thuật toán logistic regression

3.3.1. linear regression.

3.3.1.1 giới thiệu bài toán.

Một trong những model (mô hình) đơn giản nhất của bài toán hồi quy (regression) là hồi quy tuyến tính (linear regression). Mô hình hồi quy tuyến tính là một mô hình liên quan đến sự kết hợp tuyến các đầu vào:

$y \approx f(x,w) = w^T x = w_0 + w_1 x_1 + ... + w_D x_D$

trong đó:

- $w = (w_0, w1, ..., w_D)^T$ là vector weight (hay thường được gọi là parameters)
- $x = (1, x1, ..., x_D)^T$ là vector đầu vào.

trong một số sách thì: $y \approx f(x,w) = w^T \phi(x)$ trong đó $\phi(x)$ là 1 hàm transform đầu vào của x. ví dụ như: $\phi(x)_j = e^{-\frac{(x-\mu_j)^2}{2s^2}}$. điều đáng chú ý ở đây nếu $\phi(x)$ là 1 hàm phi tuyến thì model trở thành model hồi quy phi tuyến [3].

3.3.1.2 Xây dựng bài toán.

Cũng giống như các bài toán supervide linear regression cũng đi tìm những tham số cho model chi tiết là vector weight sao cho

$y \approx \hat{y} = f(x,w) = w^T x$

để tìm ra parameters ta cần một phép đánh giá phù hợp với bài toán. Với bài toán regression nói chung, ta mong muốn rằng sự sai khác e giữa giá trị thực y và giá trị dự đoán ŷ là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt. ta gọi hàm đánh giá đó là Loss function:

$$L(w) = \frac{1}{2N} \sum_{i=0}^{N} (y_i - w^T x_i)^2 = \frac{1}{2N} \|y - (w^T X)^T\|_2^2 = \frac{1}{2N} \|y - X^T w\|_2^2$$

với $y = [y_0, y_0, ..., y_{N-1}]^T$

để hàm loss function nhỏ nhất theo w thì nghiệm ta cần tìm là $w = \underset{w}{argmin} L(w)$. Chúng ta có thể sử dụng phương pháp đạo và giải nghiệm bằng cách cho đạo hàm bằng = 0. Tiểu luận xin được để xuất phương pháp Stochastic Gradient Descent (SGD) .

thuật toán SGD với linear regression:

```
Khởi Tạo:
    X - ma trận đầu vào
    y - mục vector mục tiêu
    Datas = (X,y)
    lr  - learning rate
Lặp:
    For epoch from 0 to epochs:
        Datas = shuffle(Datas)
        for x,y in Datas:
            w ← w − lr ∗ ∇_w L(w, x, y)
        endfor
    endfor
        return:
            return w
```

với : $\nabla_w L(w, x, y) = x(x^T w - y)$ và shuffle(Datas) là hàm xáo trộn lại cặp (input, output) để đảm bảo tính ngẫu nhiên. Việc này cũng ảnh

hưởng tới hiệu năng của SGD. Đây cũng chính là lý do thuật toán này có chứa từ stochastic (ngẫu nhiên) [4]

3.3.1.2 logistic regression.

thuật toán loggistic regression là thuật toán thuộc nhóm binary classification. trong bài toán phân loại nhị phân mô hình xác xuất là được đánh giá cao nhất. với output sẽ được biểu diễn là y = 1 với nhãn thuộc positive và y = 0 đối với nhãn negative. Do đó mô hình sẽ đi tính $p(y=1/x,\theta)$ vậy nhiệm vụ của logistic regression là đi tìm bộ tham số θ dựa trên tập training sao cho :

$y \approx p(y/x,\theta)$

a) tìm $p(y/x,\theta)$

Theo bayes ta có :

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)\,p(y=1)}{p(x,\theta)}$$

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)*p(y=1)}{p(x,\theta/y=1)\,p(y=1)+p(x,\theta/y=0)\,p(y=0)}$$

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)}{p(x,\theta/y=1)+p(x,\theta/y=0)\frac{N_0}{N_1}}$$

$$p(y=1/x,\theta)=\frac{1}{1+\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}}$$

Nhận xét : $\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\in(0,+\infty)$ vậy nên ta biến đổi căn bản:

$$\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}=\exp(\ln(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}))$$

ta đặt : $\ln(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1})=z$ vậy ta có: $p(y=1/x,\theta)=\frac{1}{1+e^z}$

nhận xét : $z=\ln(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1})\in(-\infty,+\infty)$ vậy nhiệm vụ mới chúng ta là từ đầu vào x làm sao dự đoán được $z=\ln(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1})$ bằng bao nhiêu ? vậy kết luận lại là bài toán quay lại là bài toán regression vậy nên thuật toán có tên logistic regression. Để đơn giản hóa bài thuật toán ta sử dụng linear regression để dự đóan z hay

$$z \approx w^T x+b \Rightarrow p(y=1/x,\theta)\approx\frac{1}{1+e^{w^T x+b}}$$

b) xây dựng hàm Lossfunction:

- **độ bất định:** trong lí thuyết thông tin thì độ bất định là độ đo tính mập mờ của 1 sự kiện. độ bất định càng cao thì độ mập mờ của thông tin càng cao hay độ bất định càng thấp thì thông tin càng có ý nghĩa. độ bất định có quan hệ mật thiết với xác suất.
- các tính chất của độ bất định:
  - Nếu sự kiện X chỉ có 1 trường hợp sảy ra là x $\Rightarrow p(x)=1$ kéo theo độ bất định $d(x)=0$.
  - xác suất sảy ra của 1 trường hợp của 1 sự kiện càng lớn thì độ bất định càng nhỏ.
  - và số trường hợp sảy ra càng lớn thì trung bình độ bất định cả sự kiện càng lớn.

ta gọi độ bất định của trường hợp x trong sự kiện X là d(x). hàm phụ thuộc của độ bất định vào xác suất là d(x) = g(p(x)) giả sử g(p(x)) khả vi và liên tục. dựa vào tính chất của độ bất định ta có:

- $d(x)\geq 0\Rightarrow g(p(x))\geq 0$
- $p(x)=1\Rightarrow g(p(x))=0\Rightarrow d(x)=0$
- có tính cộng

đặt biến phụ $u=p(x_k), v=p(x_k/x_h)$

$\Rightarrow uv=p(x_k,x_h)\Rightarrow g(uv)=g(u)-g(v)$

ta vi phân 2 vế theo biết u ta được :

$$vg'(uv)=g'(u)\Rightarrow vug'(uv)=ug'(u)\Rightarrow ug'(u)=const\Rightarrow g'(u)=mln(u)+C$$

mặt khác ta có: $g'(1)=mln(1)+C=0\Rightarrow C=0$

ta chọn m = 1.

vậy d(x) = ln(p(x)).

- như phần a đã trình bày thì chúng đang cố gắng tạo ra 1 model sao cho $y\approx p(y/x,\theta)$ ta sử hàm lossfunction là tổng độ bất định của tất cả các phép thử :

$$L(x,w)=\sum_{x\in C_1}\ln(p(y=1/x,\theta))+\sum_{x\in C_2}\ln(1-p(y=1/x,\theta))$$

hay lossfunction thường được viết :

$$L(x,w,y)=\sum_{i=0}^{N} y_i\ln(p(x_i)+(1-y_i)\ln(1-p(x_i)))$$