**Chương 5: Logistic Regression**

Logistic Regression có thể được sử dụng để phân loại một quan sát thành một trong hai lớp (như "tình cảm tích cực" và "tình cảm tiêu cực") hoặc vào một trong nhiều lớp. Một ví dụ đơn giản khác là phân biệt hình ảnh có phải con mèo hay không. Khác với Navie Bayer (a generative classifier) sẽ cố gắng hiểu con mèo như thế nào, con chó như thế nào từ đó phân biệt, Logistic Regression (a discriminative Classifiers) sẽ cố gắng học cách phân biệt qua các lớp.

Logistic Regression là một mô hình học máy có giám sát (supervised machine learning). Nó yêu cầu một kho tài liệu gồm m đầu vào/đầu ra (x(i), y(i)). Nó gồm 2 pha:

+ **training**: đào tạo hệ thống (cụ thể là trọng số w và b) bằng cách sử dụng ngẫu **gradient descent** and **cross-entropy loss**.

+ **test**: Với một ví dụ kiểm tra x, tính p (y|x) và trả về xác suất cao hơn nhãn y = 1 hoặc y = 0.

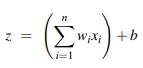
1. **The sigmoid function**

Mục tiêu của hồi binary Logistic Regression là đào tạo một bộ phân loại có thể tạo quyết định đầu vào có thuộc về lớp hay không ( ảnh có phải mèo hay không). Hàm sigmoid sẽ giúp chúng ta đưa ra quyết định này. Từ một đầu vào dưới dang vector [x1, x2,..., xn], đầu ra sẽ là y=1 nếu nó thuộc lớp và y=0 nếu nó không thuộc lớp.

Hồi quy logistic giải quyết công việc này bằng cách tìm hiểu, từ một tập huấn luyện, một vectơ của trọng số (**weights**) và một thuật ngữ thiên vị (**bias term**). Trong đó:

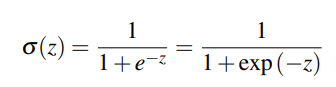
* Mỗi trọng lượng **wi** là một số thực và được liên kết với một của các tính năng đầu vào **xi**. Nó thể hiện mức độ quan trọng của tính năng đầu vào đó là đối với quyết định phân loại.
* **Bias term** hay còn được gọi là **intercept**, là một số thực khác được thêm vào các đầu vào có trọng số.

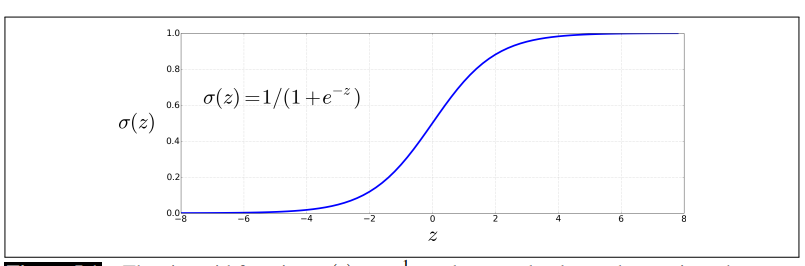
Để đưa ra quyết định về một phiên bản thử nghiệm, sau khi đã tìm hiểu các weights trong đào tạo, bộ phân loại đầu tiên nhân mỗi **xi** với trọng lượng của nó **wi**, tính tổng weights các tính năng, và cộng thêm bias term b. Z sẽ biểu thị tổng trọng số của các bằng chứng cho lớp.



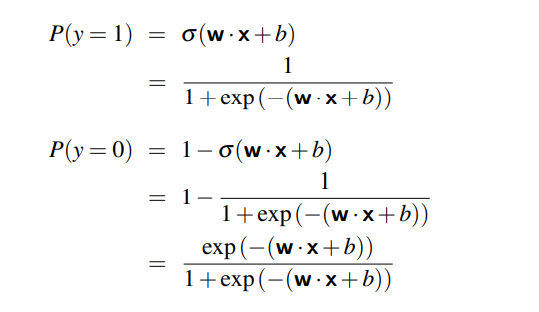


Như đã nói ở trên đầu ra sẽ là y= 1 hoặc y= 0, vì vậy ta cần tìm giá trị xác xuất giữa 0 và 1. Nhưng trên thực tế, vì weights có giá trị thực, đầu ra thậm chí có thể là số âm, vì vậy z có thể nằm trong khoảng từ −∞ đến ∞. Để có thể lấy được giá trí xác xuất giữa 0 và 1, ta cần chuyển z qua hàm sigmoid hay còn được gọi là hàm logistic. Hàm sigmoid được hiện thị như sau:



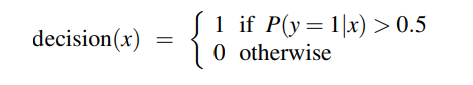


Sau khi áp dụng hàm sigmoid, ta nhận được một số từ 0 đến 1. Để biến nó thành xác suất, ta chỉ cần chắc chắn rằng hai trường hợp, p (y = 1) và p (y = 0), tổng bằng 1.

****

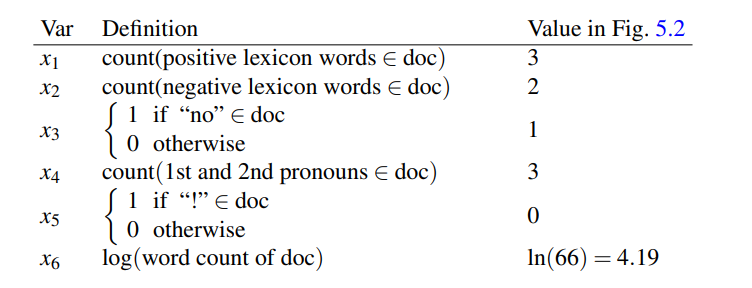
1. **Classification with Logistic Regression**

Sau khi có xác suât P, ta cần xác định đầu vào có thuộc lớp hay không theo công thức sau:

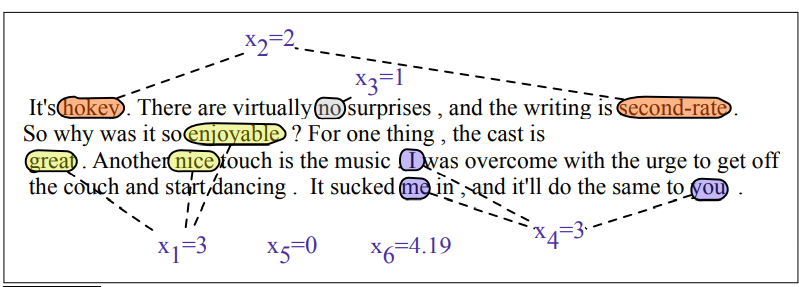
****

**Example 1:** Sentiment Classification

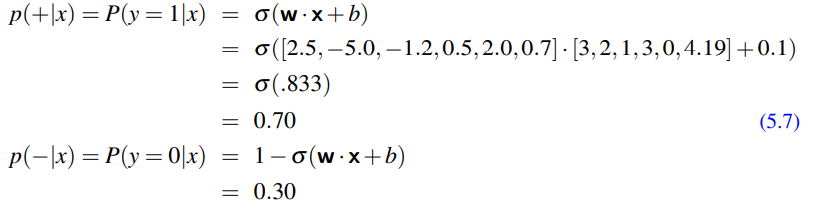
Giả sử chúng tôi đang phân loại tình cảm nhị phân trên văn bản đánh giá phim và muốn biết nên chỉ định lớp tình cảm + hoặc - cho một bài đánh giá. Chúng tôi sẽ đại diện cho mỗi quan sát đầu vào bằng 6 đặc điểm x1 ... x6 và được hiển thị trong bảng sau:



Từ bảng trên ta có thể phân tích đoạn đánh giá sau:



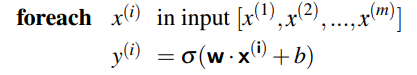
Giả sử rằng chúng ta đã biết được weignts có giá trị thực cho từng tính năng này và 6 trọng số tương ứng với 6 đặc điểm là [2.5, −5.0, −1.2, 0.5, 2.0, 0.7], và bias term b = 0,1. Ta có thể tính được xác xuất P cho câu trên như sau:



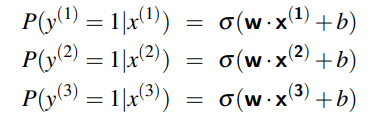
**Example 2:** Processing many examples at once

Hãy giả sử có một bộ thử nghiệm bao gồm m ví dụ thử nghiệm. Mỗi ví dụ x(i) có một vectơ feature **x**(i), 1 <i <m.

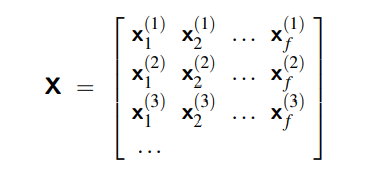
Một cách để tính toán từng giá trị đầu ra ˆy(i) , ta có thể sử dụng một vòng lặp for:



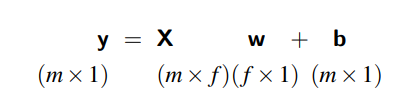
Với từng ví dụ riêng biệt ta có thể tính xác suất cua chúng như sau (3 ví dụ đầu tiên):

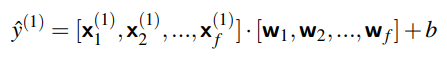


Một cách khác để tính toán giá trị đầu ra một cách hiệu quả hơn đó là sử dụng ma trận. Đầu tiên chúng ta cần đóng gói các đầu vào thành một ma trận có kích thước (m\* f) trong đó f là là số feature and weights của từng ví dụ.

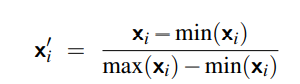


Tiếp đó chúng ta cần một vecto b có kích thước (m\*1) sau đó thực hiện theo công thức để được đầu ra ^y có kích thước (m\*1)



Ví dụ để tính ^y(1) : 

**Note:** Khi các giá trị đầu vào có phạm vi giá trị khác nhau, chúng ta cầnnormalize đầu vào về các giá trị giuwac 0 và 1. Việc normalize dữ liệu này rất quan trọng trong các mạng nơ-ron lớn, vì nó giúp tăng tốc độ của gradient descent.

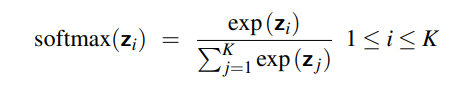
****

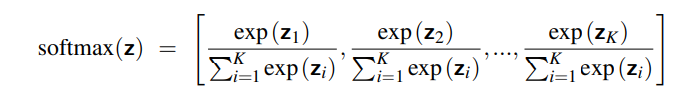
1. **Multinomial logistic regression**

Trong các trường hợp phân loại nhiều hơn hai lớp, chúng ta cần sử dụng multinomial logistic regression hay còn được gọi là softmax regression (maxent classifier). Trong multinomial logistic regression, nếu muốn gắn nhãn mỗi quan sát bằng một lớp k từ một tập hợp các lớp K, theo quy định rằng chỉ một trong các lớp này là đúng, hãy sử dụng biểu diễn sau: đầu ra y cho mỗi đầu vào x sẽ là một vectơ có độ dài K. Nếu lớp c là lớp chính xác, chúng ta sẽ đặt yc = 1, và đặt tất cả các phần tử khác của y là 0, tức là yc = 1 và yj = 0 ∀ j ≠ c. Một vectơ như y này, với một giá trị = 1 và phần còn lại là 0, được gọi là **one-hot vector**. Công việc của phân loại là tạo ra một vectơ ước lượng ˆy. Đối với mỗi lớp k, giá trị ^yk sẽ là ước tính của trình phân loại về xác suất p (yk = 1 | x)

**Softmax:** Multinomial logistic regression sử dụng tổng quát hóa sigmoid, được gọi là hàm softmax, để tính p (yk = 1 | x). Đầu vào sẽ là một vecto z = [z1, z2, ..., zK] với K giá trị tùy ý và ánh xạ chúng thành phân phối xác suất, với mỗi giá trị trong phạm vi (0,1) và tất cả các giá trị tổng bằng 1.

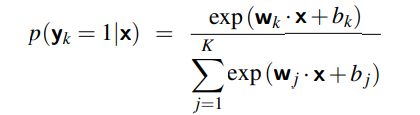
Với 1 vector z có K chiều ta có:



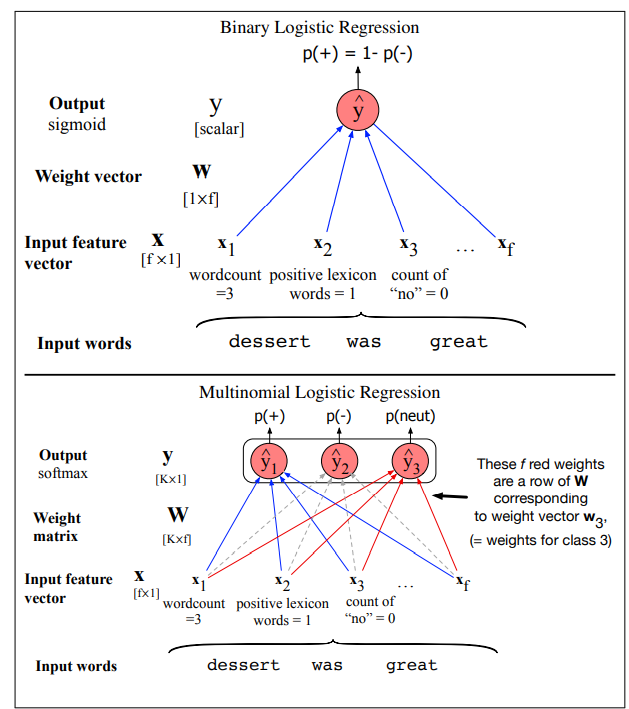


Cũng giống như hàm sigmoid, đầu vào của hàm softmax sẽ là tích giữa vectơ weights w và vectơ đầu vào x cộng với bias term b. Điểm khác biệt ở đây là cần các vectơ weights wk và bias bk riêng biệt cho mỗi K các lớp.





Sự khác biệt giữa binary logistic regression và multinomial logistic regression được thể hiện trong hình dưới đây



1. **The cross-entropy loss function**

Loss function thể hiện sự khác nhau giữ đầu ra của bộ phân loại với đầu ra đúng (^y, y). Kết quả là mất mát hàm là mất khả năng log âm, thường được gọi là cross-entropy loss function.





1. **Gradient Descent**

**Gradient descent là một phương pháp tìm giá trị tối thiểu của một hàm bằng cách tìm ra hướng nào cuar độ dốc của hàm cao nhất.**