BÁO CÁO NGHIÊN CỨU

HỒI QUY LOGISTIC

(logistic regression)

Học sinh 1: Trần Nguyễn Bảo Khang **Lớp:** 12CTin

Học sinh 2: Hoàng Việt **Lớp:** 12CTin

Học sinh 3: Trang Gia Khang **Lớp:** 12CTin

Học sinh 4: Nguyễn Quốc Anh **Lớp:** 12CTin

Thành phố Hồ Chí Minh, tháng 10 năm 2023

MỤC LỤC

1. Tại sao hồi quy logistic lại quan trọng?

Hồi quy logistic là một kỹ thuật quan trọng trong lĩnh vực trí tuệ nhân tạo và máy học (AI/ML). Mô hình ML là các chương trình phần mềm có thể được đào tạo để thực hiện các tác vụ xử lý dữ liệu phức tạp mà không cần sự can thiệp của con người. Mô hình ML được xây dựng bằng hồi quy logistic có thể giúp các tổ chức thu được thông tin chuyên sâu hữu ích từ dữ liệu kinh doanh của mình. Họ có thể sử dụng những thông tin chuyên sâu này để phân tích dự đoán nhằm giảm chi phí hoạt động, tăng độ hiệu quả và đổi chỉnh quy mô nhanh hơn. Ví dụ: doanh nghiệp có thể khám phá các mẫu hình cải thiện khả năng giữ chân nhân viên hoặc tạo ra thiết kế sản phẩm mang về nhiều lợi nhuận hơn.

2. Phân tích hồi quy hoạt động như thế nào?

Hồi quy logistic là một trong số ít các kỹ thuật phân tích hồi quy khác nhau thường được các nhà khoa học dữ liệu sử dụng trong máy học (ML). Để hiểu rõ về hồi quy logistic, trước tiên chúng ta phải hiểu kỹ thuật phân tích hồi quy cơ bản. Dưới đây là một ví dụ về phân tích hồi quy tuyến tính để cho thấy cách thức hoạt động của phân tích hồi quy.

2.1. Xác định câu hỏi

Bất kỳ quá trình phân tích dữ liệu nào cũng bắt đầu bằng một câu hỏi kinh doanh. Đối với hồi quy logistic, bạn nên giới hạn phạm vi câu hỏi để có được kết quả cụ thể:

- Những ngày mưa có ảnh hưởng đến doanh số hàng tháng của chúng ta không? (có hoặc không)
- Khách hàng đang thực hiện loại hoạt động thẻ tín dụng nào? (ủy quyền, gian lận hoặc có khả năng gian lận)

2.2. Thu thập dữ liệu lịch sử

Sau khi xác định câu hỏi, bạn cần xác định các yếu tố dữ liệu có liên quan. Sau đó, bạn sẽ thu thập dữ liệu trước đây cho tất cả các yếu tố. Ví dụ: để trả lời câu hỏi đầu tiên ở trên, bạn có thể thu thập dữ liệu doanh số hàng tháng và số ngày mưa mỗi tháng trong ba năm qua.

2.3. Đào tạo mô hình phân tích hồi quy

Bạn sẽ xử lý dữ liệu lịch sử bằng phần mềm hồi quy. Phần mềm sẽ xử lý các điểm dữ liệu khác nhau và kết nối chúng theo phương thức toán học bằng cách sử dụng phương trình. Ví dụ: nếu số ngày mưa trong ba tháng là 3, 5 và 8 còn doanh số trong những tháng đó là 8, 12 và 18, thuật toán hồi quy sẽ kết nối các yếu tố này với phương trình:

Doanh số = 2*(Số ngày mưa) + 2

2.4. Dự đoán các giá trị không xác định

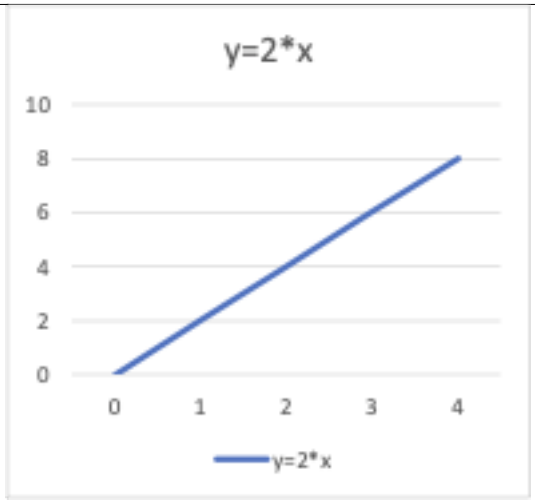
Đối với các giá trị không xác định, phần mềm sẽ áp dụng phương trình để dự đoán. Nếu bạn biết rằng tháng 7 sẽ có sáu ngày mưa, phần mềm sẽ ước tính giá trị doanh số trong tháng 7 là 14.

3. Mô hình hồi quy logistic hoạt động như thế nào?

Để hiểu rõ về mô hình hồi quy logistic, trước tiên chúng ta phải hiểu các phương trình và biến.

3.1. Phương trình

Trong toán học, phương trình cho ta mối quan hệ giữa hai biến: *x* và *y*. Bạn có thể sử dụng các phương trình hoặc hàm này để vẽ đồ thị theo trục x và trục y bằng cách nhập các giá trị khác nhau của *x* và *y*. Ví dụ: nếu bạn vẽ đồ thị cho hàm $y = 2x$, bạn sẽ có một đường thẳng như hình dưới đây. Do đó hàm này còn được gọi là hàm tuyến tính.



3.2. **Biến**

Trong thống kê, biến là các yếu tố dữ liệu hoặc thuộc tính có giá trị khác nhau. Bất kỳ phân tích nào cũng có một số biến nhất định là biến độc lập hoặc biến giải thích. Những thuộc tính này là nguyên nhân của một kết quả. Các biến khác là biến phụ thuộc hoặc biến đáp ứng; giá trị của chúng phụ thuộc vào các biến độc lập. Nhìn chung, hồi quy logistic khám phá cách các biến độc lập ảnh hưởng đến một biến phụ thuộc bằng cách xem xét các giá trị dữ liệu lịch sử của cả hai biến.

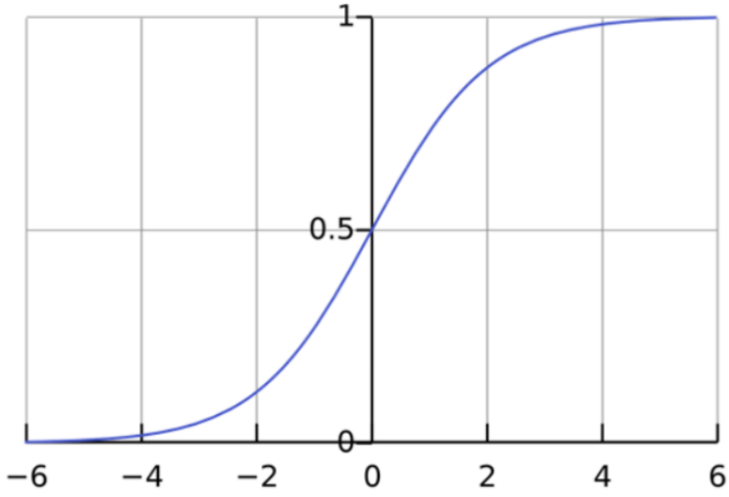
Trong ví dụ ở trên của chúng tôi, *x* được gọi là biến độc lập, biến dự đoán hoặc biến giải thích vì nó có một giá trị đã xác định. *Y* được gọi là biến phụ thuộc, biến kết quả hoặc biến đáp ứng vì giá trị của nó không xác định.

3.3. **Hàm hồi quy logistic**

Hồi quy logistic là một mô hình thống kê sử dụng hàm logistic, hay hàm logit trong toán học làm phương trình giữa *x* và *y*. Hàm logit ánh xạ *y* làm hàm sigmoid của *x*.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Nếu vẽ phương trình hồi quy logistic này, bạn sẽ có một đường cong hình chữ S như hình dưới đây.



Như bạn có thể thấy, hàm logit chỉ trả về các giá trị giữa 0 và 1 cho biến phụ thuộc, dù giá trị của biến độc lập là gì. Đây là cách hồi quy logistic ước tính giá trị của biến phụ thuộc. Phương pháp hồi quy logistic cũng lập mô hình phương trình giữa nhiều biến độc lập và một biến phụ thuộc.

Phân tích hồi quy logistic với nhiều biến độc lập

Trong nhiều trường hợp, nhiều biến giải thích ảnh hưởng đến giá trị của biến phụ thuộc. Để lập mô hình các tập dữ liệu đầu vào như vậy, công thức hồi quy logistic phải giả định mối quan hệ tuyến tính giữa các biến độc lập khác nhau. Bạn có thể sửa đổi hàm sigmoid và tính toán biến đầu ra cuối cùng như sau

$$y = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)$$

Ký hiệu β đại diện cho hệ số hồi quy. Mô hình logit có thể đảo ngược tính toán các giá trị hệ số này khi bạn cho nó một tập dữ liệu thực nghiệm đủ lớn có các giá trị đã xác định của cả hai biến phụ thuộc và biến độc lập.

3.4. Log của tỷ số odds

Mô hình logit cũng có thể xác định tỷ số thành công trên thất bại hay log của tỷ số odds. Ví dụ: nếu bạn đang chơi poker với bạn bè và thắng bốn ván trên mười ván, tỷ số chiến thắng của bạn là bốn phần sáu, hoặc 4/6, và đó là tỷ số thành công trên thất bại của bạn. Mặt khác, xác suất thắng là 4/10.

Về mặt toán học, tỷ số odds về mặt xác suất là $\frac{p}{1-p}$ và log của tỷ số odds là $\log\left(\frac{p}{1-p}\right)$. Bạn có thể biểu diễn hàm logistic bằng log của tỷ số odds như hình dưới đây:

$$Logit\ function = \log\left(\frac{p}{1-p}\right)$$

4. Phân tích hồi quy logistic có những loại nào?

Có ba cách tiếp cận phân tích hồi quy logistic dựa trên kết quả của biến phụ thuộc.

4.1. Hồi quy logistic nhị phân

Hồi quy logistic nhị phân phù hợp với các vấn đề phân lớp nhị phân chỉ có hai kết quả có thể xảy ra. Biến phụ thuộc chỉ có thể có hai giá trị, chẳng hạn như có và không hoặc 0 và 1.

Dù hàm logistic tính toán một phạm vi giá trị giữa 0 và 1, mô hình hồi quy nhị phân vẫn sẽ làm tròn kết quả đến các giá trị gần nhất. Nói chung, kết quả dưới 0,5 sẽ được làm tròn thành 0 và kết quả trên 0,5 sẽ được làm tròn thành 1, do đó hàm logistic trả về một kết quả nhị phân.

4.2. Hồi quy logistic đa thức

Hồi quy đa thức có thể phân tích các vấn đề có một số kết quả có thể xảy ra, miễn là số kết quả hữu hạn. Ví dụ: kỹ thuật này có thể dự đoán xem giá nhà sẽ tăng 25%, 50%, 75% hay 100% dựa trên dữ liệu dân số, nhưng sẽ không thể dự đoán được giá trị chính xác của một ngôi nhà.

Hồi quy logistic đa thức hoạt động bằng cách ánh xạ các giá trị kết quả cho các giá trị khác nhau giữa 0 và 1. Hàm logistic có thể trả về một khoảng dữ liệu liên tục như 0,1, 0,11, 0,12, v.v., do đó hồi quy đa thức cũng nhóm đầu ra đến các giá trị gần nhất có thể có.

4.3. Hồi quy logistic thứ tự

Hồi quy logistic thứ tự, hay mô hình logit có thứ tự, là một loại hồi quy đa thức đặc biệt cho các vấn đề trong đó các số đại diện cho các bậc chứ không phải là giá trị thực tế. Ví dụ: bạn sẽ sử dụng hồi quy thứ tự để dự đoán đáp án cho câu hỏi khảo sát yêu cầu khách hàng đánh giá dịch vụ của bạn ở mức kém, ổn, tốt hoặc xuất sắc dựa trên một giá trị số, chẳng hạn như số lượng mặt hàng họ mua từ bạn trong năm.

5. Làm thế nào để so sánh hồi quy logistic với các kỹ thuật ML khác?

Hai kỹ thuật phân tích dữ liệu phổ biến là phân tích hồi quy tuyến tính và học sâu.

5.1. Phân tích hồi quy tuyến tính

Như đã giải thích ở trên, hồi quy tuyến tính lập mô hình mối quan hệ giữa các biến phụ thuộc và độc lập bằng tổ hợp tuyến tính. Phương trình hồi quy tuyến tính là

$y = \beta_0 X_0 + \beta_1 X_1 + \beta_2 X_2 + \dots \beta_n X_n + \varepsilon$, trong đó β_1 đến β_n và ε là các hệ số hồi quy.

5.2. **Hồi quy logistic so với hồi quy tuyến tính**

Hồi quy tuyến tính dự đoán một biến phụ thuộc liên tục bằng một tập hợp các biến độc lập cho trước. Một biến liên tục có thể có một phạm vi giá trị, chẳng hạn như giá cả hoặc độ tuổi. Do đó, hồi quy tuyến tính có thể dự đoán giá trị thực của biến phụ thuộc. Kỹ thuật này có thể trả lời các câu hỏi như "Giá gạo sau 10 năm nữa sẽ là bao nhiêu?"

Không giống như hồi quy tuyến tính, hồi quy logistic là một thuật toán phân loại. Kỹ thuật này không thể dự đoán giá trị thực sự cho dữ liệu liên tục. Kỹ thuật này có thể trả lời các câu hỏi như "Liệu giá gạo trong 10 năm nữa có tăng 50% hay không?"

5.3. **Deep learning**

Học sâu sử dụng mạng nơ-ron hoặc các thành phần phần mềm mô phỏng bộ não con người để phân tích thông tin. Các phép toán học sâu dựa trên khái niệm toán học của vectơ.

5.4. **Hồi quy logistic so với học sâu**

Hồi quy logistic ít phức tạp và có cường độ điện toán ít hơn so với học sâu. Quan trọng hơn là nhà phát triển không thể điều tra hoặc sửa đổi các phép toán học sâu do tính chất phức tạp và dựa trên máy móc của chúng. Mặt khác, các phép toán hồi quy logistic lại minh bạch và dễ khắc phục sự cố hơn, Và làm sao bạn biết rằng những cây thu hải đường đẹp đẽ này không có tầm quan trọng ngang nhau?

Hercule Poirot, trong Vụ án bí ẩn ở Styles của Agatha Christie

Những câu chuyện trinh thám có rất nhiều manh mối giống như văn bản có từ ngữ. Tuy nhiên đối với người đọc kém, có thể khó biết cách cân nhắc các manh mối của tác giả theo thứ tự để thực hiện nhiệm vụ phân loại quan trọng: quyết định whodunnit. Trong chương này chúng tôi giới thiệu một thuật toán rất phù hợp để khám phá mối liên hệ giữa các tính năng hoặc tín hiệu và một số kết quả cụ thể: hồi quy logistic.

5.5. Hậu cần hồi quy

Thật vậy, hồi quy logistic là một trong những công cụ phân tích quan trọng nhất trong xã hội và khoa học tự nhiên. Trong xử lý ngôn ngữ tự nhiên, hồi quy logistic là thuật toán học máy được giám sát cơ bản để phân loại và cũng có vai trò rất quan trọng. mối quan hệ chặt chẽ với mạng lưới thần kinh. Như chúng ta sẽ thấy sau, một mạng lưới thần kinh có thể được xem như một chuỗi các bộ phân loại hồi quy logistic được xếp chồng lên nhau. Do đó, các kỹ thuật phân loại và học máy được giới thiệu ở đây sẽ đóng một vai trò quan trọng xuyên suốt cuốn sách.

Hồi quy logistic 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 thuộc một trong nhiều lớp.

Bởi vì toán học cho trường hợp hai lớp đơn giản hơn nên chúng ta sẽ mô tả đặc biệt này trường hợp hồi quy logistic trước tiên trong một số phần tiếp theo, sau đó tóm tắt ngắn gọn việc sử dụng hồi quy logistic đa thức cho nhiều hơn hai lớp.

Chúng ta sẽ giới thiệu toán học về hồi quy logistic trong một số phần tiếp theo. Nhưng hãy bắt đầu với một số vấn đề cấp cao.

Phân loại sáng tạo và phân biệt đối xử: Sự khác biệt quan trọng nhất giữa Bayes ngây thơ và hồi quy logistic là hồi quy logistic là một phân loại phân biệt đối xử trong khi Bayes ngây thơ là một phân loại tổng quát.

Đây là hai khuôn khổ rất khác nhau về cách thức để xây dựng mô hình học máy. Hãy xem xét một hình ảnh trực quan phép ẩn dụ: hãy tưởng tượng chúng ta đang cố gắng phân biệt hình ảnh con chó từ hình ảnh con mèo. Một mô hình sáng tạo sẽ có mục tiêu là hiểu con chó trông như thế nào và mèo trông như thế nào. Bạn có thể hỏi theo nghĩa đen như vậy một mô hình để 'tạo ra', tức là vẽ một con chó. Đưa ra một bài kiểm tra. Sau đó, hệ thống sẽ hỏi xem mô

hình con mèo hay mô hình con chó tốt hơn phù hợp (ít ngạc nhiên hơn) với hình ảnh và chọn nó làm nhãn cho nó.

Ngược lại, một mô hình phân biệt đối xử chỉ cố gắng học cách phân biệt các lớp (có thể không cần tìm hiểu nhiều về chúng). Vì vậy có lẽ tất cả những con chó trong dữ liệu huấn luyện đang đeo vòng cổ và mèo thì không. Nếu một đặc điểm đó ngăn cách rõ rang lớp học, mô hình được thỏa mãn. Nếu bạn hỏi như vậy làm mẫu những gì nó biết về mèo tất cả những gì nó có thể nói là họ không đeo vòng cổ.

Chính thức hơn, hãy nhớ lại rằng Bayes ngây thơ gán một lớp c cho một tài liệu d không bằng cách tính toán trực tiếp P(c|d) nhưng bằng cách tính toán khả năng và giá trị trước đó

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}} \tag{5.1}$$

Một mô hình sinh sản như Bayes ngây thơ sử dụng thuật ngữ khả năng này. người mẫu diễn tả cách tạo ra các tính năng của một tài liệu nếu chúng ta biết nó thuộc lớp c. Ngược lại, một mô hình phân biệt trong kịch bản phân loại văn bản này cố gắng phân biệt người mẫu để tính trực tiếp P(c|d). Có lẽ nó sẽ học cách gán trọng số cao cho tài liệu các tính năng trực tiếp cải thiện khả năng phân biệt giữa các lớp có thể, ngay cả khi nó không thể tạo ra một ví dụ về một trong các lớp. Các thành phần của bộ phân loại học máy theo xác suất: Giống như Bayes ngây thơ, hồi quy logistic là một bộ phân loại xác suất sử dụng máy được giám sát học hỏi. Trình phân loại học máy yêu cầu tập dữ liệu huấn luyện gồm m đầu vào/đầu ra cặp (x (i), y(i)). (Chúng tôi sẽ sử dụng chỉ số trên trong ngoặc đơn để chỉ các trường hợp riêng lẻ trong tập huấn luyện—để phân loại tình cảm, mỗi trường hợp có thể là một cá thể tài liệu cần được phân loại.) Sau đó, một hệ thống máy học để phân loại sẽ có bốn thành phần:

- Một đại diện tính năng của đầu vào. Đối với mỗi quan sát đầu vào x(i), cái này sẽ là một vectơ đặc trưng $[x_1, x_2, \ldots, x_n]$. Chúng ta thường đề cập đến tính năng tôi cho đầu vào x(j) như x(j)i, đôi khi được đơn giản hóa là xi, nhưng chúng ta cũng sẽ thấy ký hiệu fi , fi(x), hoặc, để phân loại nhiều lớp, fi(c, x).
- Hàm phân loại tính toán ˆy, lớp ước tính, thông qua p(y|x). TRONG Phần tiếp theo chúng tôi sẽ giới thiệu các công cụ sigmoid và softmax để phân loại.
- Hàm mục tiêu cho việc học, thường liên quan đến việc giảm thiểu lỗi trên ví dụ đào tạo. Chúng tôi sẽ giới thiệu hàm mất entropy chéo.
- Thuật toán tối ưu hóa hàm mục tiêu. Chúng tôi giới thiệu thuật toán giảm độ dốc ngẫu nhiên.

Hồi quy logistic có hai giai đoạn:

- Huấn luyện: Chúng tôi huấn luyện hệ thống (cụ thể là các trọng số w và b) bằng cách sử dụng ngẫu nhiên giảm độ dốc và mất entropy chéo.
- Kiểm tra: Cho ví dụ kiểm tra x, chúng tôi tính p(y|x) và trả về xác suất cao hơn nhãn y = 1 hoặc y = 0.

5.6. **The sigmoid function**

Mục tiêu của hồi quy logistic nhị phân là huấn luyện một bộ phân loại có thể tạo ra một kết quả nhị phân quyết định về lớp của một quan sát đầu vào mới. Ở đây chúng tôi giới thiệu sigmoid phân loại sẽ giúp chúng tôi đưa ra quyết định này. Hãy xem xét một quan sát đầu vào duy nhất x, mà chúng tôi sẽ biểu thị bằng một vectơ đặc trưng $[x_1, x_2, \ldots, x_n]$ (chúng tôi sẽ hiển thị các đặc trưng mẫu trong tiểu mục tiếp theo). Bộ phân loại đầu ra y có thể là 1 (có nghĩa là quan sát là thành viên của lớp) hoặc 0 (quan sát không phải là thành viên của lớp). Chúng tôi

muốn biết xác suất $P(y = 1|x)$ rằng quan sát này là một thành viên của lớp. Vì vậy có lẽ quyết định này là "tích cực tình cảm" so với "tình cảm tiêu cực", các đặc điểm thể hiện số lượng từ trong một tài liệu, $P(y = 1|x)$ là xác suất tài liệu có cảm xúc tích cực, và $P(y = 0|x)$ là xác suất tài liệu có cảm xúc tiêu cực.

Hồi quy logistic giải quyết nhiệm vụ này bằng cách học từ tập huấn luyện một vectơ của **trọng số (weights)** và **bias (bias term)**. Mỗi trọng lượng w_i là số thực và liên kết với một của các tính năng đầu vào x_i. Trọng số w_i thể hiện tầm quan trọng của tính năng đầu vào đó liên quan đến quyết định phân loại và có thể là dương (cung cấp bằng chứng cho thấy cá thể được phân loại thuộc về loại dương) hoặc âm (cung cấp bằng chứng rằng cá thể đang được phân loại thuộc về lớp âm). Như vậy chúng ta có thể mong đợi trong một sentiment task, từ "awsome" sẽ có trọng số dương cao và bias (bias term) **"**abysmal" có trọng số rất âm. Số hạng sai lệch (bias term), còn được gọi là số chặn, là chặn một số thực khác được thêm vào đầu vào có trọng số.

Để đưa ra quyết định về một phiên bản thử nghiệm—sau khi chúng ta đã tìm hiểu các trọng số trong quá trình huấn luyện—đầu tiên bộ phân loại sẽ nhân mỗi xi với trọng số của nó w_i, tổng hợp các tính năng có trọng số, và thêm số hạng sai lệch b. Kết quả z là một số biểu thị tổng có trọng số bằng chứng cho lớp.

$$z = \left(\sum_{i=1}^{n} w_i x_i\right) + b$$

Trong phần còn lại, chúng ta sẽ biểu diễn những tổng như vậy bằng cách sử dụng tích vô hướng từ đại số tuyến tính. Tích vô hướng của hai vectơ a và b, viết là a · b, là tổng tích các phần tử tương ứng của mỗi vectơ. (Chú ý rằng chúng ta biểu diễn vectơ bằng ký hiệu in đậm b). Do đó, sau đây là tương đương hình thành phương trình:

$$z = w.x + b$$

Nhưng lưu ý rằng không có gì trong phương trình trên buộc z phải là một xác suất hợp pháp, nghĩa là nói dối trong khoảng từ 0 đến 1. Trên thực tế, vì các trọng số có giá trị thực nên đầu ra thậm chí có thể là vô cực; z dao động từ -∞ đến ∞.

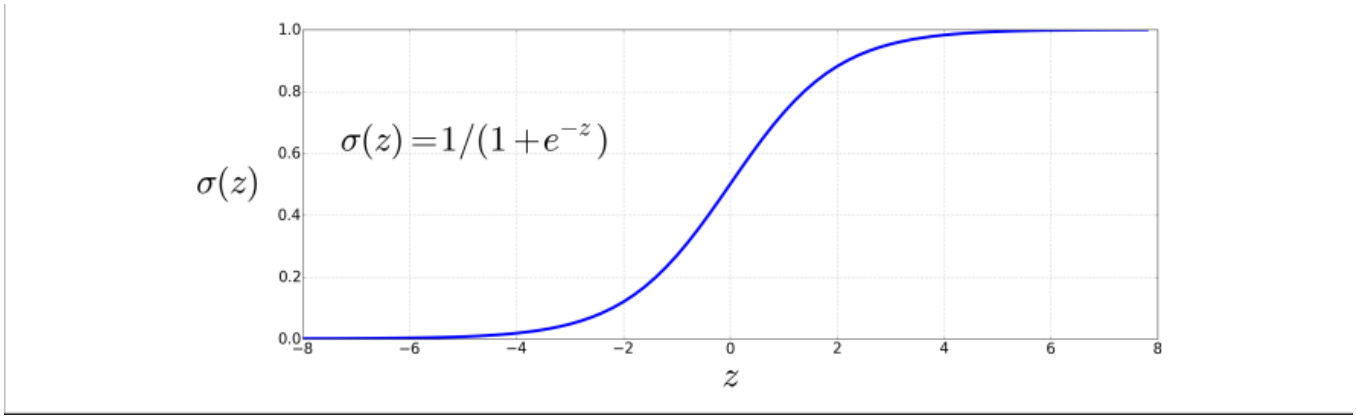


Figure 6.6.1: Hàm sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ nhận một giá trị thực và ánh xạ nó tới phạm vi (0,1). Nó gần như tuyến tính quanh 0 nhưng các giá trị ngoại lệ bị nén về 0 hoặc 1

Để tính xác suất, chúng ta sẽ chuyển z thông qua hàm sigmoid, σ(z). Các Hàm sigmoid (được đặt tên vì nó trông giống chữ s) còn được gọi là hàm logistic và đặt tên cho hồi quy logistic.

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)}$$

(Trong phần còn lại của báo cáo, chúng ta sẽ sử dụng ký hiệu exp(x) để chỉ e^x.) Đường sigmoid có một số lợi thế; nó lấy một số có giá trị thực và ánh xạ nó vào phạm vi (0,1), đây chính là điều chúng ta mong muốn đối với một xác suất. Bởi vì nó gần như tuyến tính xung quanh 0 nhưng phẳng dần về phía cuối, nó có xu hướng ép các giá trị ngoại lệ về 0 hoặc 1. Và nó khả vi, như chúng ta sẽ thấy trong phần 6.10 sẽ rất hữu ích cho việc học.

Nếu chúng ta áp dụng sigmoid cho tổng các đặc trưng có trọng số, chúng ta nhận được một số từ 0 đến 1. Để biến nó thành xác suất, chúng ta chỉ cần thực hiện chắc chắn rằng hai trường hợp $p(y = 1)$ và $p(y = 0)$, có tổng bằng 1. Chúng ta có thể làm điều này như sau:

$$\begin{aligned} P(y = 1) &= \sigma(w.x + b) \\ &= \frac{1}{1 + \exp(-(w.x + b))} \\ P(y = 0) &= 1 - \sigma(w.x + b) \\ &= 1 - \frac{1}{1 + \exp(-(w.x + b))} \\ &= \frac{\exp(-(w.x + b))}{1 + \exp(-(w.x + b))} \end{aligned}$$

Hàm sigmoid có tính chất:

$$1 - \sigma(x) = \sigma(-x)$$

vì vậy chúng ta cũng có thể biểu thị $P(y = 0)$ dưới dạng $\sigma(-(w \cdot x + b))$.

5.7. **Classification with Logistic Regression**

Do đó, hàm sigmoid ở phần trước cho chúng ta một cách để lấy một thể hiện x và tính xác suất $\mathrm{P}(\mathrm{y} = 1|\mathrm{x})$.

Làm cách nào để chúng tôi đưa ra quyết định về việc áp dụng lớp nào cho phiên bản thử nghiệm x? Vì với x cho trước, chúng ta nói có nếu xác suất $P(y = 1|x)$ lớn hơn 0,5 và không có trường hợp nào khác. Chúng ta gọi .5 là ranh giới quyết định:

$$decision(x) = \begin{cases} 1 \; if \; P(y = 1|x) > 0.5 \\ 0 \end{cases}$$

Hãy có một số ví dụ về việc áp dụng hồi quy logistic làm công cụ phân loại cho nhiệm vụ xử lý ngôn ngữ.

5.7.1 **Sentiment Classification**

Giả sử chúng ta đang thực hiện phân loại sentiment nhị phân trên văn bản đánh giá phim và chúng ta muốn biết liệu đây có phải là một văn bản tích cực + hay − cho một bài đánh giá. Chúng ta sẽ biểu thị mỗi quan sát đầu vào bằng 6 giá trị x_1 ... x_6 của đầu vào được hiển thị trong bảng sau; Hình 6.7.1.1 thể hiện các tính năng trong một mẫu mini.

Bây giờ hãy giả sử rằng chúng ta đã học được trọng số có giá trị thực cho

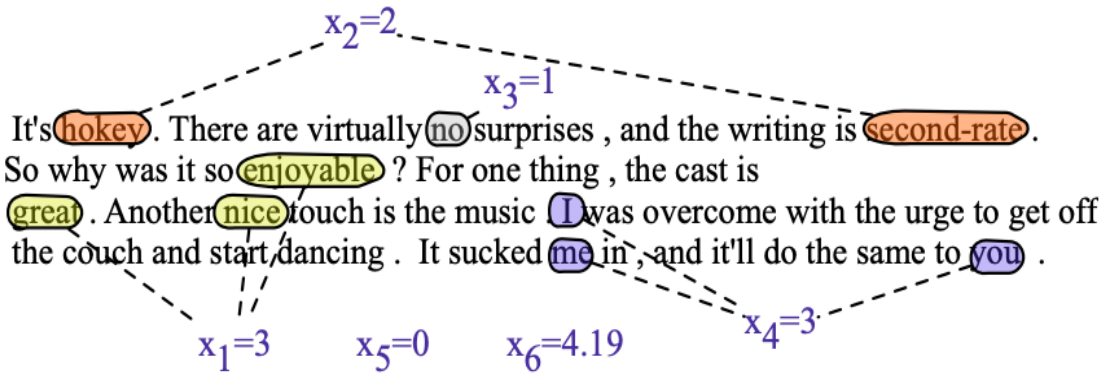


Figure 6.7.1 Một tài liệu thử nghiệm nhỏ mẫu hiển thị các đặc điểm được trích xuất trong vectơ x

Mỗi tính năng này và 6 trọng số tương ứng với 6 tính năng đó là $[2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$ trong khi b = 0.1. (Chúng ta sẽ thảo luận trong phần tiếp theo về cách các trọng số đã được học.) Ví dụ, trọng số w_1 cho biết tầm quan trọng của một làm nổi bật số lượng từ vựng tích cực (tuyệt vời, tốt đẹp, thú vị, v.v.) là một đoạn văn mang tính tích cực, trong khi w_2 cho chúng ta biết tầm quan trọng của từ vựng tiêu cực từ. Lưu ý rằng w_1 = 2,5 là dương, trong khi w_2 = −5,0, nghĩa là các từ phủ định có liên quan tiêu cực đến một quyết định có cảm xúc tích cực và cao gấp đôi quan trọng như những lời nói tích cực. Với 6 đặc điểm này và kết quả xem xét đầu vào x, P(+|x) và P(-|x) có thể được tính bằng phương trình:

$$p(+|x) = P(y = 1|x) = \sigma(w.x + b) = \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7].[3, 2, 1, 3, 0, 4.19] + 0.1) = 0.7$$

$$p(-|x) = P(y = 0|x) = 1 - \sigma(w.x + b) = 0.3$$

5.7.2 Other classification tasks and features

Hồi quy logistic thường được áp dụng cho tất cả các loại nhiệm vụ NLP và bất kỳ thuộc tính nào của đầu vào có thể là một tính năng. Hãy xem xét nhiệm vụ định hướng thời kỳ: thời kỳ quyết định sự định hướng nếu dấu chấm là phần cuối của câu hoặc một phần của từ, bằng cách phân loại từng dấu chấm thành một trong hai lớp EOS (cuối câu) và không phải EOS. Chúng ta có thể sử dụng các tính năng như x_1 bên dưới biểu thị rằng từ hiện tại là chữ thường (có thể có dấu dương trọng lượng), hoặc từ hiện tại có trong từ điển viết tắt ("Prof.") (có lẽ có trọng số âm).

Một đặc điểm cũng có thể thể hiện sự kết hợp khá phức tạp của cải. Ví dụ: một dấu chấm theo sau một từ viết hoa có thể là một dấu chấm EOS, nhưng nếu bản thân từ đó là St. và từ trước đó được viết hoa thì dấu chấm có thể là một phần của việc rút ngắn từ đường phố.

$$x_1 = \begin{cases} 1 \; if \; \text{Case}(w_i) = "Lower" \\ 0 \; otherwise \end{cases}$$

$$x_2 = \begin{cases} 1 \; if \; "w_i \in AcronymDict" \\ 0 \; otherwise \end{cases}$$

$$x_3 = \begin{cases} 1 \; if \; "w_i \in \text{St. and Case}(w_{i-1}) = Cap" \\ 0 \; otherwise \end{cases}$$

Thiết kế các tính năng: Các tính năng thường được thiết kế bằng cách kiểm tra quá trình đào tạo tập trung vào trực giác ngôn ngữ và tài liệu ngôn ngữ trong lĩnh vực này. MỘT phân tích lỗi cẩn thận trên tập huấn luyện hoặc tập phát triển của phiên bản đầu tiên của hệ thống thường cung cấp cái nhìn sâu sắc về các tính năng. Đối với một số nhiệm vụ, việc xây dựng các tính năng phức tạp là sự kết hợp của các tính năng nguyên thủy hơn sẽ đặc biệt hữu ích. Chúng tôi đã thấy một tính năng như vậy để định hướng thời kỳ ở trên, trong đó dấu chấm trên từ St. ít có khả năng là dấu chấm cuối câu nếu từ trước đó được viết hoa. Đối với hồi quy logistic và Bayes naive, những điều này các tính năng kết hợp hoặc tương tác tính năng phải được thiết kế bằng tay. Tính năng tương tác đối với nhiều tác vụ (đặc biệt khi giá trị tính năng có thể tham chiếu các từ cụ thể) chúng ta sẽ cần số lượng lớn các tính năng. Thông thường chúng được tạo tự động thông qua các mẫu tính năng, thông số kỹ thuật trừu tượng của các tính năng. Ví dụ: tính

năng mẫu bigram mẫu để định hướng dấu chấm có thể tạo ra một đặc điểm cho mỗi cặp từ xuất hiện trước một khoảng thời gian trong tập huấn luyện. Do đó, không gian đặc trưng rất thưa thớt, vì chúng ta chỉ phải tạo một tính năng nếu n-gram đó tồn tại ở vị trí đó trong tập huấn luyện. Các tính năng thường được tạo dưới dạng hàm băm từ các mô tả chuỗi. Mô tả người dùng của một tính năng như "bigram(bữa sáng kiểu Mỹ)" được băm thành một số nguyên duy nhất i trở thành số đặc trưng fi.Để tránh nỗ lực to lớn của con người trong việc thiết kế tính năng, nghiên cứu gần đây về NLP đã tập trung vào việc học biểu diễn: các cách học các tính năng một cách tự động theo cách không giám sát từ đầu vào. Chúng tôi sẽ giới thiệu các phương pháp biểu diễn học ở Chương 6 và Chương 7. Chia tỷ lệ các tính năng đầu vào: Khi các tính năng đầu vào khác nhau có cực kỳ khác nhau phạm vi giá trị, thông thường chúng ta phải điều chỉnh lại tỷ lệ để chúng có phạm vi tương đương. Chúng tôi chuẩn hóa các giá trị đầu vào bằng cách căn giữa chúng để tạo ra giá trị trung bình bằng 0 và giá trị tiêu chuẩn độ lệch của một (sự chuyển đổi này đôi khi được gọi là điểm z). Nghĩa là, nếu μ_i điểm z là giá trị trung bình của các giá trị của đặc trưng x_i trên m quan sát trong tập dữ liệu đầu vào, và σ_i là độ lệch chuẩn của các giá trị của đặc trưng x_i trên tập dữ liệu đầu vào, chúng ta có thể thay thế mỗi tính năng x_i bằng một tính năng mới x_0 ,ta tính được như sau:

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_i^{(j)}$$

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(x_i^{(j)} - \mu_i)^2}$$

$$\grave{x_i} = \frac{x_i - \mu_i}{\sigma_i}$$

Ngoài ra, chúng ta có thể chuẩn hóa các giá trị tính năng đầu vào nằm trong khoảng từ 0 đến 1:

$$\grave{x_i} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Việc có dữ liệu đầu vào với phạm vi có thể so sánh được sẽ hữu ích khi so sánh các giá trị trên đặc trưng. Chia tỷ lệ dữ liệu đặc biệt quan trọng trong các mạng nơ-ron lớn, vì nó giúp tăng tốc độ giảm độ dốc.

5.7.3 Processing many examples at once

Chúng ta đã trình bày các phương trình hồi quy logistic. Nhưng trong thực tế, tất nhiên chúng ta sẽ muốn xử lý toàn bộ tập kiểm tra với nhiều tập dữ liệu. Hãy Giả sử chúng ta có một bộ thử nghiệm bao gồm m ví dụ thử nghiệm mà mỗi mẫu chúng ta muốn phân loại. Chúng ta sẽ tiếp tục sử dụng ký hiệu từ trang 2, trong đó giá trị chỉ số trên trong ngoặc đơn đề cập đến chỉ mục mẫu trong một số bộ dữ liệu (dành cho đào tạo hoặc cho thử nghiệm). Vì vậy, trong trường hợp này mỗi ví dụ kiểm tra $x^{(i)}$ có vectơ đặc trưng [$x^{(1)}$,$x^{(2)}$,..,$x^{(m)}$]. (Chúng ta sẽ biểu diễn vectơ và ma trận bằng chữ in đậm.) Một cách để tính từng giá trị đầu ra $\widehat{y(i)}$ chỉ là để có một vòng lặp for và tính toán mỗi ví dụ kiểm tra một lần:

$$\boldsymbol{foreach}\ x^{(i)}\ in\ input\ \left[x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right]$$

$$y^{(i)} = \boldsymbol{\sigma}(\boldsymbol{w}.\boldsymbol{x^{(i)}} + b)$$

Khi đó, đối với ví dụ thử nghiệm đầu tiên, chúng ta sẽ tính toán riêng biệt $\widehat{y^{(i)}}$ như sau:

$$P(y^{(1)} = 1|x^{(1)}) = \sigma(\boldsymbol{w}.\boldsymbol{x^{(1)}} + b)$$

$$P(y^{(2)} = 1|x^{(2)}) = \sigma(\boldsymbol{w}.\boldsymbol{x^{(2)}} + b)$$

$$P(y^{(3)} = 1|x^{(3)}) = \sigma(\boldsymbol{w}.\boldsymbol{x^{(3)}} + b)$$

Nhưng hóa ra là chúng ta có thể sửa đổi một chút phương trình ban đầu của mình. Để làm điều này hiệu quả hơn nhiều. Chúng ta sẽ sử dụng ma trận số học để gán một lớp cho tất cả các ví dụ với một phép toán ma trận!

Đầu tiên, chúng ta sẽ gói tất cả các vectơ đặc trưng đầu vào cho mỗi đầu vào x vào một đầu vào duy nhất ma trận X, trong đó mỗi hàng i là một vectơ hàng bao gồm vectơ đặc trưng cho ví dụ đầu vào x(i) (tức là vectơ x (i)). Giả sử mỗi ví dụ có f đặc điểm và trọng số, do đó X sẽ là ma trận có hình dạng [m× f], như sau:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_1^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_f^{(2)} \\ x_1^{(3)} & x_2^{(3)} & \cdots & x_f^{(3)} \\ \cdots & & & \end{bmatrix}$$

Bây giờ nếu chúng ta giới thiệu b là một vectơ có độ dài m bao gồm độ lệch vô hướng số hạng b lặp lại m lần, b = [b_1,b_2,...,b_m], và $\hat{y} = [\widehat{y^{(1)}}, \widehat{y^{(2)}}, .., \widehat{y^{(m)}}]$ là vectơ của đầu ra (một vô hướng $\widehat{y^{(i)}}$ cho mỗi đầu vào $x_{(i)}$ và vectơ đặc trưng x của nó (i)) và đại diện vectơ trọng số w là vectơ cột, chúng ta có thể tính toán tất cả các kết quả đầu ra bằng một vectơ duy nhất phép nhân và phép cộng của ma trận:

$$\boldsymbol{y = Xw + b}$$

Phương trình trên tính toán điều tương tự như vòng lặp for của chúng tatrong phương trình trước. Ví dụ, mục đầu tiên của vectơ đầu ra y, sẽ chính xác là:

$$\hat{y}^{(1)} = [\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, ..., \mathbf{x}_f^{(1)}] \cdot [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_f] + b \qquad (5.13)$$

Lưu ý rằng chúng tôi phải sắp xếp lại X và w theo thứ tự chúng xuất hiện trong biểu thức. 5,5 đến làm cho các phép nhân xuất hiện đúng cách. Đây là phương trình. 5.12 một lần nữa với các hình dạng cho xem:

$$\begin{matrix} \mathbf{y} & = & \mathbf{X} & \mathbf{w} & + & \mathbf{b} \\ (m \times 1) & & (m \times f) & (f \times 1) & & (m \times 1) \end{matrix} \qquad (5.14)$$

Các trình biên dịch và phần cứng máy tính hiện đại có thể tính toán phép toán ma trận này rất hiệu quả, làm cho việc tính toán nhanh hơn nhiều, điều này trở nên quan trọng khi đào tạo hoặc thử nghiệm trên các tập dữ liệu rất lớn.

5.7.4 Choosing a classifier

Hồi quy logistic có một số lợi thế so với Bayes ngây thơ. Bayes ngây thơ có giả định độc lập có điều kiện quá mạnh. Hãy xem xét hai tính năng mà có mối tương quan chặt chẽ; trên thực tế, hãy tưởng tượng rằng chúng ta chỉ cần thêm cùng một tính năng f1 hai lần. Naive Bayes sẽ coi cả hai bản sao của f1 như thể chúng tách biệt, nhân chúng lên cả trong việc đánh giá quá cao bằng chứng. Ngược lại, hồi quy logistic còn nhiều hơn thế. mạnh mẽ với các tính năng tương quan; nếu hai đặc điểm f1 và f2 tương quan hoàn hảo, hồi quy sẽ chỉ gán một phần trọng số cho w1 và một phần cho w2. Như vậy khi có nhiều đặc điểm tương quan, hồi quy logistic sẽ

cho kết quả chính xác hơn xác suất hơn Bayes ngây thơ. Vì vậy, hồi quy logistic thường hoạt động tốt hơn trên tài liệu hoặc tập dữ liệu và là một mặc định phổ biến.

Mặc dù xác suất kém chính xác hơn nhưng Bayes ngây thơ vẫn thường đưa ra kết quả đúng quyết định phân loại. Hơn nữa, Bayes ngây thơ có thể hoạt động cực kỳ tốt (đôi khi còn tốt hơn cả hồi quy logistic) trên các tập dữ liệu rất nhỏ (Ng và Jordan, 2002) hoặc các tài liệu ngắn (Wang và Manning, 2012). Hơn nữa, Bayes ngây thơ là dễ triển khai và đào tạo rất nhanh (không có bước tối ưu hóa). Vì vậy nó vẫn là một cách tiếp cận hợp lý để sử dụng trong một số tình huống.

5.7.5 Multinomial logistic regression

Đôi khi chúng ta cần nhiều hơn hai lớp. Có lẽ chúng ta có thể muốn làm 3 chiều phân loại tình cảm (tích cực, tiêu cực hoặc trung tính). Hoặc chúng ta có thể chỉ định một số nhãn chúng tôi sẽ giới thiệu trong Chương 8, như phần lời nói của một từ (chọn từ 10, 30 hoặc thậm chí 50 phần khác nhau của lời nói) hoặc thực thể được đặt tên loại cụm từ (chọn từ các thẻ như người, địa điểm, tổ chức).

Trong những trường hợp như vậy, chúng tôi sử dụng hồi quy logistic đa thức, còn được gọi là hồi quy softmax (trong tài liệu NLP cũ hơn, đôi khi bạn sẽ thấy tên phân loại tối đa). Trong hồi quy logistic đa thức, chúng ta muốn gắn nhãn cho mỗi quan sát bằng một lớp k từ một tập hợp K lớp, với quy định chỉ có một trong các lớp này là cái đúng (đôi khi được gọi là phân loại cứng; một quan sát không thể có trong nhiều lớp). 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 đúng, 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 6= c. Một vectơ như y này, với một giá trị = 1 và phần còn lại là 0, được gọi là vectơ một nóng. Công việc của phân loại là tạo ra một vectơ ước lượng yˆ. Với mỗi lớp k, giá trị yˆ k sẽ là ước tính của bộ phân loại về xác suất p(yk = 1|x).

5.7.6 Softmax

Bộ phân loại logistic đa thức sử dụng sự khái quát hóa của sigmoid, được gọi là hàm softmax, để tính p(yk = 1|x). Hàm softmax lấy một vectơ z = [z1, z2,..., zK] của K giá trị tùy ý và ánh xạ chúng tới phân bố xác suất, với mỗi giá trị trong phạm vi [0,1] và tất cả các giá trị có tổng bằng 1. Giống như sigmoid, nó là một hàm số mũ.

Đối với vectơ z có chiều K, softmax được định nghĩa là:

$$\text{softmax}(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{K}\exp(\mathbf{z}_j)} \quad 1 \le i \le K \tag{5.15}$$

Do đó, softmax của vectơ đầu vào z = [z1, z2,..., zK] chính là một vectơ:

$$\text{softmax}(\mathbf{z}) = \left[\frac{\exp(\mathbf{z}_1)}{\sum_{i=1}^{K}\exp(\mathbf{z}_i)}, \frac{\exp(\mathbf{z}_2)}{\sum_{i=1}^{K}\exp(\mathbf{z}_i)}, \ldots, \frac{\exp(\mathbf{z}_K)}{\sum_{i=1}^{K}\exp(\mathbf{z}_i)}\right] \tag{5.16}$$

mẫu số PK tôi=1 exp(zi) được sử dụng để chuẩn hóa tất cả các giá trị thành xác suất. Vì vậy, ví dụ cho một vectơ:

$$\mathbf{z} = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$$

kết quả (làm tròn) softmax(z) là

$$[0.055, 0.090, 0.006, 0.099, 0.74, 0.010]$$

Giống như sigmoid, softmax có đặc tính nén các giá trị về 0 hoặc 1. Do đó, nếu một trong các đầu vào lớn hơn các đầu vào khác, nó sẽ có xu hướng đẩy xác suất của nó lên cao hơn. về 1 và loại bỏ xác suất của các đầu vào nhỏ hơn.

5.7.7 **Applying softmax in logistic regression**

Khi chúng ta áp dụng softmax cho hồi quy logistic, đầu vào sẽ (giống như đối với sigmoid) là tích vô hướng giữa vectơ trọng số w và vectơ đầu vào x (cộng với a Thiên kiến). Nhưng bây giờ chúng ta sẽ cần các vectơ trọng số wk và độ lệch bk riêng biệt cho mỗi K các lớp học. Do đó, xác suất của mỗi lớp đầu ra ˆyk của chúng ta có thể được tính như sau:

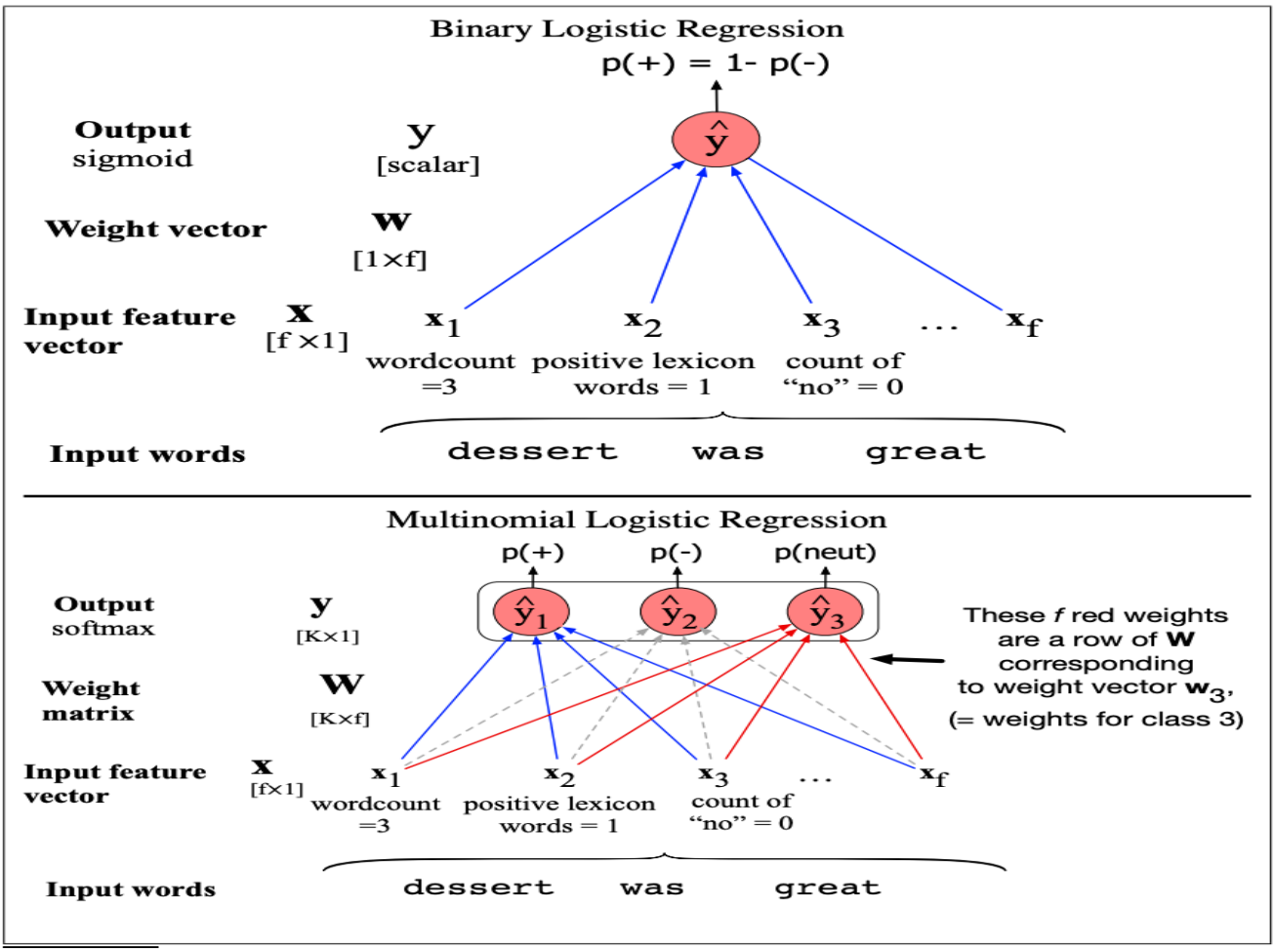
$$p(\mathbf{y}_k = 1|\mathbf{x}) = \frac{\exp(\mathbf{w}_k \cdot \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j \cdot \mathbf{x} + b_j)} \tag{5.17}$$

Dạng của phương trình. 5.17 có vẻ như chúng tôi sẽ tính toán từng đầu ra riêng biệt. Thay vào đó, việc thiết lập phương trình để tính toán hiệu quả hơn bằng phần cứng xử lý vectơ hiện đại là phổ biến hơn. Chúng tôi sẽ thực hiện điều này bằng cách đại diện cho tập hợp K vectơ trọng số làm ma trận trọng số W và vectơ thiên vị b. Mỗi hàng k của W tương ứng với vectơ trọng số wk. Do đó W có hình dạng [K × f], với K thì số lượng lớp đầu ra và f số lượng tính năng đầu vào. Vectơ thiên vị b có một giá trị cho mỗi lớp đầu ra K. Nếu chúng ta biểu diễn các trọng số theo cách này, chúng ta có thể tính toán yˆ, vectơ xác suất đầu ra cho mỗi lớp K, bằng một phương trình tao nhã duy nhất:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{5.18}$$

Nếu bạn tính toán ma trận số học, bạn có thể thấy rằng điểm ước tính của lớp đầu ra đầu tiên ˆy1 (trước khi chúng ta lấy softmax) sẽ trở thành w1 · x+b1.

Hình 5.3 cho thấy trực quan về vai trò của vectơ trọng số so với ma trận trọng số trong việc tính toán xác suất của lớp đầu ra cho nhị phân và đa thức hồi quy logistic.



Hồi quy logistic nhị phân và đa thức. Hồi quy logistic nhị phân sử dụng một vectơ trọng số đơn w và có đầu ra vô hướng ˆy. Trong hồi quy logistic đa thức chúng ta có K vectơ trọng số riêng biệt tương ứng với K lớp, tất cả được đóng gói thành một trọng số duy nhất ma trận W và đầu ra vectơ yˆ.

5.7.8 **Features in Multinomial Logistic Regression**

Các đặc điểm trong hồi quy logistic đa thức hoạt động giống như các đặc điểm trong hồi quy logistic nhị phân, với điểm khác biệt đã đề cập ở trên là chúng ta sẽ cần các vectơ trọng số riêng biệt và các thành kiến cho mỗi lớp K. Nhớ lại tính năng dấu chấm than nhị phân x5 của chúng tôi từ trang 4:

$$x_5 = \begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$$

Trong phân loại nhị phân, trọng số dương w5 trên một đối tượng sẽ ảnh hưởng đến bộ phân loại về phía y = 1 (tình cảm tích cực) và trọng số âm ảnh hưởng đến y = 0 (tình cảm tiêu cực) với giá trị tuyệt đối cho biết tầm quan trọng của tính năng này là. Ngược lại, đối với hồi quy logistic đa thức, với các trọng số riêng biệt cho từng lớp, một đặc điểm có thể là bằng chứng ủng hộ hoặc chống lại từng lớp riêng lẻ.

Ví dụ: trong phân loại tình cảm đa lớp 3 chiều, chúng ta phải chỉ định từng ghi lại một trong 3 lớp +, − hoặc 0 (trung tính). Bây giờ, một tính năng liên quan đến dấu chấm than có thể có trọng số âm đối với 0 tài liệu và trọng số dương đối với tài liệu + hoặc -:

Feature	Definition	$w_{5,+}$	$w_{5,-}$	$w_{5,0}$
$f_5(x)$	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	3.5	3.1	−5.3

Bởi vì các trọng số đặc trưng này phụ thuộc cả vào văn bản đầu vào và đầu ra lớp, đôi khi chúng tôi làm cho sự phụ thuộc này trở nên rõ ràng và biểu diễn chính các tính năng dưới dạng f(x, y): một hàm của cả đầu vào và lớp. Sử dụng ký hiệu như vậy f5(x) ở trên có thể được biểu diễn dưới dạng ba đặc trưng f5(x,+), f5(x,−) và f5(x,0), mỗi cái có một trọng lượng duy nhất. Chúng tôi sẽ sử dụng loại ký hiệu này trong mô tả của chúng tôi của CRF.

5.7.9 Learning in Logistic Regression

Các tham số của mô hình, trọng số w và độ lệch b, được học như thế nào? hậu cần

hồi quy là một ví dụ của phân loại có giám sát trong đó chúng ta biết chính xác gắn nhãn y (0 hoặc 1) cho mỗi quan sát x. Những gì hệ thống tạo ra thông qua phương trình. 5,5 là $\hat{y}$, ước tính của hệ thống về y thực. Chúng tôi muốn tìm hiểu các tham số (có nghĩa là w và b) làm cho y cho mỗi quan sát huấn luyện càng gần với y thực càng tốt. Điều này đòi hỏi hai thành phần mà chúng tôi đã báo trước trong phần giới thiệu về chương. Đầu tiên là thước đo mức độ gần gũi của nhãn hiện tại ($\hat{y}$) với vàng thật nhãn y. Thay vì đo lường sự giống nhau, chúng ta thường nói về điều ngược lại: khoảng cách giữa sản lượng hệ thống và sản lượng vàng, và chúng tôi gọi khoảng cách này hàm mất mát hoặc hàm chi phí. Trong phần tiếp theo chúng tôi sẽ giới thiệu sự mất mát hàm thường được sử dụng cho hồi quy logistic và cả mạng lưới thần kinh, sự mất mát entropy chéo. Điều thứ hai chúng ta cần là một thuật toán tối ưu hóa để cập nhật lặp đi lặp lại các trọng số sao cho cực tiểu hóa hàm mất mát này. Thuật toán tiêu chuẩn cho việc này là xuống dốc; chúng tôi sẽ giới thiệu thuật toán giảm độ dốc ngẫu nhiên trong Phần sau. Chúng tôi sẽ mô tả các thuật toán này cho trường hợp hồi quy logistic nhị phân đơn giản hơn trong hai phần tiếp theo và sau đó chuyển sang hồi quy logistic đa thức trong Mục 5.8.

5.7.10 The cross-entropy loss function

Chúng ta cần một hàm mất mát biểu thị, đối với một quan sát x, mức độ gần của bộ phân loại đầu ra ($\hat{y} = \sigma(w \cdot x+b)$) là đầu ra chính xác (y, là 0 hoặc 1). Chúng tôi sẽ gọi đây là:

$$L(\hat{y},y) = \text{How much } \hat{y} \text{ differs from the true } y \qquad (5.19)$$

Chúng tôi thực hiện điều này thông qua một hàm mất mát nhằm ưu tiên nhãn lớp chính xác của các ví dụ huấn luyện để có nhiều khả năng xảy ra hơn. Đây được gọi là khả năng tối đa có điều kiện ước tính: chúng tôi chọn các tham số w,b sao cho tối đa hóa xác suất log của

nhãn y thực trong dữ liệu huấn luyện dựa trên các quan sát x. Kết quả tổn thất hàm này là sự mất khả năng ghi nhật ký âm, thường được gọi là mất entropy chéo. Hãy suy ra hàm mất mát này, áp dụng cho một quan sát x. Chúng tôi muốn tìm hiểu các trọng số tối đa hóa xác suất của nhãn đúng p(y|x). Từ đó chỉ có hai kết quả riêng biệt (1 hoặc 0), đây là phân phối Bernoulli và chúng ta có thể biểu thị xác suất p(y|x) mà bộ phân loại của chúng tôi tạo ra cho một quan sát là sau (hãy nhớ rằng nếu y=1, phương trình 5.20 đơn giản hóa thành $\hat{y}$; nếu y=0, phương trình 5.20 đơn giản hóa thành $1-\hat{y}$):

$$p(y|x) = \hat{y}^y (1-\hat{y})^{1-y} \tag{5.20}$$

Bây giờ chúng ta lấy nhật ký của cả hai bên. Điều này sẽ trở nên hữu ích về mặt toán học, và không làm tổn thương chúng tôi; bất kỳ giá trị nào tối đa hóa xác suất cũng sẽ tối đa hóa xác suất log của xác suất:

$$\begin{aligned}\log p(y|x) &= \log\left[\hat{y}^y (1-\hat{y})^{1-y}\right] \\ &= y\log\hat{y} + (1-y)\log(1-\hat{y}) \end{aligned} \tag{5.21}$$

phương trình. 5.21 mô tả khả năng nhật ký cần được tối đa hóa. Để biến điều này thành một hàm mất mát (thứ mà chúng ta cần giảm thiểu), chúng ta sẽ chỉ cần bật dấu hiệu lên phương trình. 5,21. Kết quả là tổn thất entropy chéo LCE:

$$L_{\text{CE}}(\hat{y},y) = -\log p(y|x) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})] \tag{5.22}$$

Cuối cùng, chúng ta có thể áp dụng định nghĩa $\hat{y} = \sigma(w.x + b)$

$$L_{\text{CE}}(\hat{y},y) = -[y\log\sigma(\mathbf{w}\cdot\mathbf{x}+b) + (1-y)\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \tag{5.23}$$

Hãy xem liệu hàm mất mát này có thực hiện đúng với ví dụ của chúng ta trong Hình 5.2 hay không. Chúng tôi muốn tổn thất nhỏ hơn nếu ước tính của mô hình gần đúng và lớn hơn nếu mô hình bị nhầm lẫn. Vì vậy, trước tiên hãy giả sử nhãn vàng chính xác cho tình cảm
ví dụ trong Hình 5.2 là dương, tức là y = 1. Trong trường hợp này mô hình của chúng ta đang hoạt động tốt, vì từ phương trình. 5.7, nó thực sự đã mang lại cho ví dụ này xác suất dương cao hơn (0,70) hơn âm (0,30). Nếu chúng ta thế σ(w · x+b) = 0,70 và y = 1 vào phương trình. 5.23, bên phải vế của phương trình bị loại bỏ, dẫn đến sự mất mát tiếp theo (chúng ta sẽ sử dụng log để chỉ nhập ký tự nhiên khi cơ sở không được chỉ định):

$$\begin{aligned} L_{\text{CE}}(\hat{y},y) &= -[y\log\sigma(\mathbf{w}\cdot\mathbf{x}+b) + (1-y)\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \\ &= -[\log\sigma(\mathbf{w}\cdot\mathbf{x}+b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

Ngược lại, thay vào đó hãy giả sử rằng ví dụ trong Hình 5.2 thực sự là âm, tức là y = 0 (có lẽ người đánh giá tiếp tục nói "Nhưng mấu chốt là bộ phim này kinh khủng! Tôi xin bạn đừng nhìn thấy nó!"). Trong trường hợp này mô hình của chúng tôi bị nhầm lẫn và chúng tôi muốn
tổn thất sẽ cao hơn. Bây giờ nếu chúng ta thế y = 0 và 1−σ(w· x+b) = 0,31 từ phương trình. 5,7 vào phương trình. 5.23, vế trái của phương trình bị loại bỏ:

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1-y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -\log (.30) \\ &= 1.2 \end{aligned}$$

Chắc chắn rằng, tổn thất đối với phân loại thứ nhất (0,36) nhỏ hơn tổn thất đối với phân loại thứ hai phân loại (1.2).

Tại sao việc giảm thiểu xác suất nhật ký âm này lại làm được điều chúng ta muốn? Một bộ phân loại hoàn hảo sẽ gán xác suất 1 cho kết quả đúng (y=1 hoặc y=0) và xác suất 0 dẫn tới kết quả sai. Điều đó có nghĩa là nếu y bằng 1 thì ˆy càng cao (càng gần 1) thì bộ phân loại càng tốt; ˆy càng thấp (càng gần 0), bộ phân loại càng tệ. Thay vào đó, nếu y bằng 0 thì 1 − yˆ càng cao (gần 1), thì phân loại tốt hơn. Log âm của ˆy (nếu y đúng bằng 1) hoặc 1−yˆ (nếu y đúng y bằng 0) là thước đo tổn hao thuận tiện vì nó bắt đầu từ 0 (log âm của 1, không loss) đến vô cùng (log âm 0, mất vô hạn). Hàm mất mát này cũng đảm bảo rằng khi xác suất trả lời đúng là tối đa thì xác suất trả lời sai sẽ giảm thiểu; vì cả hai có tổng bằng một nên bất kỳ sự gia tăng nào về xác suất câu trả lời đúng sẽ phải trả giá bằng câu trả lời sai. Nó được gọi là sự mất mát entropy chéo, bởi vì phương trình. 5.21 cũng là công thức của entropy chéo giữa phân phối xác suất thực y và phân phối ước tính ˆy của chúng tôi.

Bây giờ chúng tôi biết những gì chúng tôi muốn giảm thiểu; trong phần tiếp theo, chúng ta sẽ xem cách tìm mức tối thiểu.

5.7.11 Gradient Descent

Mục tiêu của chúng tôi khi giảm độ dốc là tìm ra trọng số tối ưu: giảm thiểu tổn thất

chức năng chúng tôi đã xác định cho mô hình. Trong phương trình. 5.24 bên dưới, chúng tôi sẽ trình bày rõ ràng thực tế là hàm mất mát L được tham số hóa bởi các trọng số mà chúng ta sẽ đề cập đến trong học máy nói chung là θ (trong trường hợp hồi quy logistic θ = w,b).

Vì vậy, mục tiêu là tìm tập trọng số tối thiểu hóa hàm mất mát, tính trung bình trên tất cả các ví dụ:

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \frac{1}{m} \sum_{i=1}^{m} L_{\text{CE}}(f(x^{(i)}; \theta), y^{(i)}) \tag{5.24}$$

Làm thế nào chúng ta có thể tìm được giá trị nhỏ nhất của hàm mất mát này (hoặc bất kỳ) nào? Giảm dần độ dốc là một phương pháp tìm cực tiểu của hàm bằng cách tìm ra hướng nào (trong không gian của các tham số θ) độ dốc của hàm tăng mạnh nhất và di chuyển

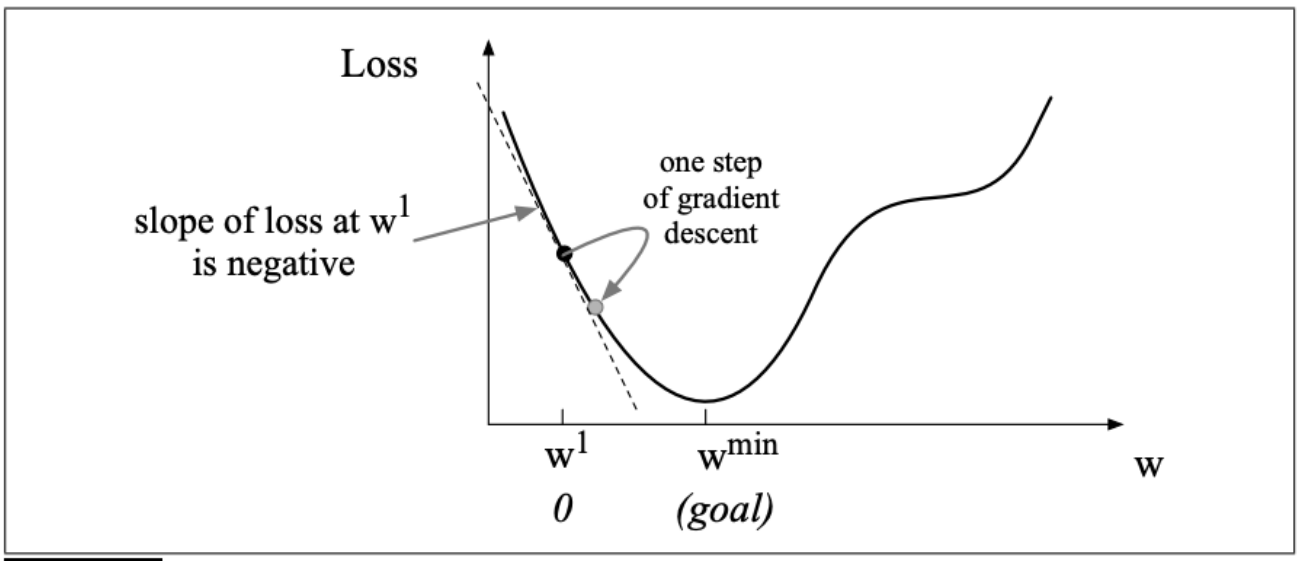
Ở hướng ngược lại. Trực giác cho thấy rằng nếu bạn đang đi bộ trong hẻm núi và cố gắng

để đi xuống con sông ở phía dưới nhanh nhất, bạn có thể nhìn xung quanh mình 360 độ, tìm hướng nào mặt đất dốc nhất, và đi bộ xuống dốc theo hướng đó.

Đối với hồi quy logistic, hàm mất mát này là hàm lồi thuận tiện. Hàm lồi có nhiều nhất một giá trị nhỏ nhất; không có cực tiểu cục bộ nào bị mắc kẹt, do đó, việc giảm độ dốc bắt đầu từ bất kỳ điểm nào được đảm bảo tìm thấy mức tối thiểu. (Ngược lại, tổn thất đối với mạng nơ-ron nhiều lớp là không lồi và việc giảm độ dốc có thể bị kẹt ở mức cực tiểu cục bộ để huấn luyện mạng nơ-ron và không bao giờ tìm thấy mức tối ưu toàn cục.)

Mặc dù thuật toán (và khái niệm độ dốc) được thiết kế để định hướng vectơ, trước tiên chúng ta hãy xem xét một cách trực quan hóa trường hợp tham số của hệ thống chỉ là một đại lượng vô hướng w, được minh họa trong Hình 5.4.

Cho một khởi tạo ngẫu nhiên của w ở một giá trị nào đó w1, và giả sử tổn thất hàm L tình cờ có dạng như Hình 5.4, chúng ta cần thuật toán cho biết liệu ở lần lặp tiếp theo chúng ta có nên di chuyển sang trái hay không (làm cho w2 nhỏ hơn w1) hoặc đúng (làm w2 lớn hơn w1) đạt cực tiểu.



Bước đầu tiên trong việc lặp lại việc tìm giá trị nhỏ nhất của hàm mất mát này bằng cách di chuyển w theo hướng ngược lại với độ dốc của hàm. Vì độ dốc âm nên chúng ta cần để di chuyển w theo hướng tích cực, sang phải. Ở đây các chỉ số trên được sử dụng cho các bước học tập, gieo 1 nghĩa là giá trị ban đầu của w (là 0), w2 giá trị ở bước thứ hai, v.v.

Thuật toán giảm độ dốc trả lời câu hỏi này bằng cách tìm độ dốc của hàm mất mát tại thời điểm hiện tại và di chuyển theo hướng ngược lại. Các gradient của hàm nhiều biến là một vectơ hướng theo hướng của mức tăng lớn nhất của hàm số. Độ dốc là sự khái quát hóa nhiều biến của độ dốc, do đó, đối với hàm một biến như trong Hình 5.4, chúng ta có thể một cách không chính thức hãy coi gradient là độ dốc. Đường chấm trong Hình 5.4 cho thấy độ dốc của hàm mất mát giả định tại điểm w = w1. Bạn có thể thấy rằng độ dốc của dấu chấm này dòng là âm. Do đó, để tìm mức tối thiểu, độ dốc giảm dần yêu cầu chúng ta đi theo hướng ngược lại: di chuyển w theo hướng tích cực.

Độ lớn của lượng di chuyển theo độ dốc giảm dần là giá trị của độ dốc d/dw tốc độ học L(f(x;w), y) được tính theo trọng số của tốc độ học η. Học tập cao hơn (nhanh hơn) tỷ lệ có nghĩa là chúng ta nên di chuyển w nhiều hơn trên mỗi bước. Sự thay đổi chúng tôi thực hiện trong tham số là tốc độ học nhân với độ dốc (hoặc độ dốc, trong biến đơn của chúng tôi ví dụ):

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x;w), y) \qquad (5.25)$$

Bây giờ, hãy mở rộng trực giác từ hàm một biến vô hướng w sang nhiều biến, bởi vì chúng ta không chỉ muốn di chuyển sang trái hoặc phải, chúng ta muốn biết vị trí trong không gian N chiều (của N tham số tạo nên θ) chúng ta nên di chuyển. Độ dốc chỉ là một vectơ như vậy; nó biểu thị các thành phần định hướng của độ dốc sắc nét nhất dọc theo mỗi N chiều đó. Nếu chúng ta chỉ tưởng tượng hai chiều trọng lượng (ví dụ với một trọng số w và một độ lệch b), thì gradient có thể là một vectơ có hai thành phần trực giao, mỗi thành phần cho chúng ta biết độ dốc của mặt đất theo chiều w và trong b kích thước. Hình 5.5 hiển thị trực quan giá trị của vectơ gradient 2 chiều được lấy tại điểm màu đỏ.

Trong hồi quy logistic thực tế, vectơ tham số w dài hơn 1 hoặc 2, vì vectơ đặc tính đầu vào x có thể khá dài và chúng ta cần trọng số wi cho mỗi xi. Đối với mỗi thứ nguyên/biến wi trong w (cộng với độ lệch b), độ dốc sẽ có một thành phần cho chúng ta biết độ dốc đối với biến

đó. Ở mỗi chiều Wi, chúng ta biểu thị độ dốc dưới dạng đạo hàm riêng ∂ /∂wi của hàm mất mát. Thiết yếu chúng ta đang hỏi: “Một sự thay đổi nhỏ trong biến số đó sẽ là bao nhiêu?”

ảnh hưởng đến hàm tổn thất tổng L?”

Khi đó, về mặt hình thức, gradient của hàm nhiều biến f là một vectơ trong đó mỗi thành phần biểu thị đạo hàm riêng của f đối với một trong các biến. Chúng ta sẽ sử dụng ký hiệu delta Hy Lạp ngược ∇ để chỉ độ dốc và biểu diễn ˆy dưới dạng f(x;θ) để làm cho sự phụ thuộc vào θ rõ ràng hơn:

$$\nabla L(f(x;\theta),y) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \\ \frac{\partial}{\partial b} L(f(x;\theta),y) \end{bmatrix} \tag{5.26}$$

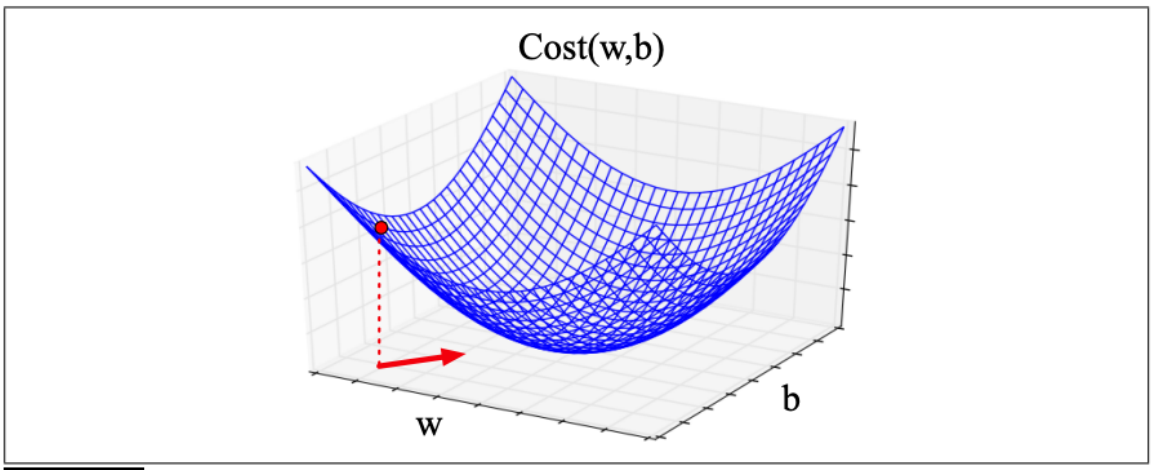
Do đó, phương trình cuối cùng để cập nhật θ dựa trên gradient là

$$\theta^{t+1} = \theta^t - \eta \nabla L(f(x;\theta),y) \tag{5.27}$$

5.7.12 The Gradient for Logistic Regression

Để cập nhật θ, chúng ta cần định nghĩa cho gradient ∇L(f(x;θ), y). Nhớ lại điều đó đối với hồi quy logistic, hàm mất entropy chéo là:

$$L_{\text{CE}}(\hat{y},y) = -[y \log \sigma(\mathbf{w}\cdot\mathbf{x}+b) + (1-y)\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \tag{5.28}$$



Trực quan hóa vectơ gradient tại điểm màu đỏ trong hai chiều w và b, hiển thị mũi tên màu đỏ trong mặt phẳng x-y chỉ theo hướng chúng ta sẽ đi tìm tối thiểu: hướng ngược lại của gradient (hãy nhớ rằng các điểm gradient theo hướng tăng chứ không giảm). Hóa ra đạo hàm của hàm này đối với một vectơ quan sát x là Eq. 5,29 (người đọc quan tâm có thể xem Phần 5.10 để biết cách suy ra phương trình này):

$$\frac{\partial L_{\text{CE}}(\hat{y},y)}{\partial \mathbf{w}_j} = [\sigma(\mathbf{w}\cdot\mathbf{x}+b)-y]\mathbf{x}_j$$
$$= (\hat{y}-y)\mathbf{x}_j \quad (5.29)$$

Đôi khi bạn cũng sẽ thấy phương trình này ở dạng tương đương:

$$\frac{\partial L_{\text{CE}}(\hat{y},y)}{\partial \mathbf{w}_j} = -(y-\hat{y})\mathbf{x}_j \quad (5.30)$$

Lưu ý trong các phương trình này rằng độ dốc đối với một trọng số wj biểu thị một giá trị rất trực quan: sự khác biệt giữa y thực và ước tính của chúng tôi yˆ = σ(w·x+b) cho quan sát đó, nhân với giá trị đầu vào tương ứng xj.

5.7.13 The Stochastic Gradient Descent Algorithm

Giảm độ dốc ngẫu nhiên là một thuật toán trực tuyến giúp giảm thiểu hàm mất mát bằng cách tính toán độ dốc của nó sau mỗi ví dụ huấn luyện và di chuyển θ ở bên phải hướng (ngược lại với hướng của gradient). (“Thuật toán trực tuyến” là thuật toán xử lý ví dụ đầu vào của nó bằng ví dụ, thay vì đợi cho đến khi nó nhìn thấy toàn bộ đầu vào.) Hình 5.6 thể hiện thuật toán.

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
  # where: L is the loss function
  # f is a function parameterized by θ
  # x is the set of training inputs $x^{(1)}$, $x^{(2)}$,..., $x^{(m)}$
  # y is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$,..., $y^{(m)}$

$\theta \leftarrow 0$
repeat til done # see caption
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
      Compute $\hat{y}^{(i)} = f(x^{(i)};\theta)$ # What is our estimated output $\hat{y}$?
      Compute the loss $L(\hat{y}^{(i)},y^{(i)})$ # How far off is $\hat{y}^{(i)}$ from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)};\theta),y^{(i)})$ # How should we move θ to maximize loss?
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ

Thuật toán giảm gradient ngẫu nhiên. Bước 1 (tính toán tổn thất) được sử dụng chủ yếu là để báo cáo mức độ chúng tôi đang làm trên bộ dữ liệu hiện tại; chúng ta không cần phải tính toán mất mát để tính toán độ dốc. Thuật toán có thể kết thúc khi nó hội tụ (hoặc khi định mức độ dốc <) hoặc khi tiến trình dừng lại (ví dụ khi tổn thất bắt đầu xảy ra lên trên một bộ đã được tổ chức).

Tốc độ học η là một siêu tham số phải được điều chỉnh. Nếu nó quá cao, người học sẽ thực hiện các bước quá lớn, vượt quá mức mất mát tối thiểu chức năng. Nếu nó quá thấp, người học sẽ thực hiện những bước quá nhỏ và quá lâu để đạt đến mức tối thiểu. Người ta thường bắt đầu với tốc độ học tập cao hơn và sau đó giảm từ từ để nó là một hàm của phép lặp k của quá trình huấn luyện; ký hiệu ηk có thể được dùng để chỉ giá trị của tốc độ học ở lần lặp k.

Chúng ta sẽ thảo luận về siêu tham số chi tiết hơn trong Chương 7, nhưng tóm lại, chúng là

một loại tham số đặc biệt cho bất kỳ mô hình học máy nào. Không giống như các tham số thông thường của một mô hình (các trọng số như w và b), được thuật toán học từ tập huấn luyện, siêu tham số là các tham số đặc biệt được thuật toán chọn nhà thiết kế ảnh hưởng đến cách thức hoạt động của thuật toán.

5.7.14 Mini-batch training

Giảm dần độ dốc ngẫu nhiên được gọi là ngẫu nhiên vì nó chọn một ngẫu nhiên ví dụ tại một thời điểm, di chuyển trọng lượng để cải thiện hiệu suất trên đơn lẻ đó ví dụ. Điều đó có thể dẫn đến những chuyển động rất giật cục, do đó người ta thường tính toán gradient trên các lô phiên bản huấn luyện thay vì một phiên bản đơn lẻ.

Ví dụ: trong đào tạo hàng loạt, chúng tôi tính toán độ dốc trên toàn bộ tập dữ liệu. Bằng cách xem rất nhiều ví dụ, việc đào tạo hàng loạt đưa ra ước tính tuyệt vời về hướng di chuyển trọng lượng, với chi phí phải mất nhiều thời gian xử lý mỗi lần ví dụ duy nhất trong tập huấn luyện để tính toán hướng hoàn hảo này.

Một sự thỏa hiệp là đào tạo theo đợt nhỏ: chúng tôi đào tạo trên một nhóm m ví dụ (có thể là 512 hoặc 1024) nhỏ hơn toàn bộ tập dữ liệu. (Nếu m là kích thước của tập dữ liệu, sau đó chúng tôi đang thực hiện giảm độ dốc hàng loạt; nếu m = 1, chúng ta quay lại thực hiện giảm độ dốc ngẫu nhiên.) Huấn luyện theo đợt nhỏ cũng có ưu điểm về tính toán hiệu quả. Các lô nhỏ có thể dễ dàng được vector hóa, chọn kích thước của lô nhỏ dựa trên tài nguyên tính toán. Điều này cho phép chúng tôi xử lý song song tất cả các mẫu trong một lô nhỏ và sau đó tích lũy phần lỗ, một điều không phải vậy. có thể với đào tạo cá nhân hoặc theo đợt.

Chúng ta chỉ cần xác định các phiên bản lô nhỏ của hàm mất entropy chéo chúng tôi đã xác định trong Phần 5.5 và độ dốc trong Phần 5.6.1. Hãy mở rộng tổn thất chéo entropy cho một ví dụ từ biểu thức. 5.22 đến lô nhỏ cỡ m. Chúng ta sẽ tiếp tục sử dụng ký hiệu x (Tôi) và y(i) nghĩa là tính năng đào tạo và nhãn đào tạo thứ i, tương ứng. Chúng ta giả định rằng các ví dụ huấn luyện là độc lập:

$$\begin{aligned}\log p(\text{training labels}) &= \log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}) \\ &= \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}) \\ &= -\sum_{i=1}^{m} L_{\text{CE}}(\hat{y}^{(i)}, y^{(i)}) \end{aligned} \tag{5.31}$$

Bây giờ hàm chi phí cho lô nhỏ gồm m mẫu là tổn thất trung bình cho mỗi ví dụ:

$$\begin{aligned}Cost(\hat{y}, y) &= \frac{1}{m}\sum_{i=1}^{m} L_{\text{CE}}(\hat{y}^{(i)}, y^{(i)}) \\ &= -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \sigma(\mathbf{w}\cdot\mathbf{x}^{(i)} + b) + (1 - y^{(i)}) \log\left(1 - \sigma(\mathbf{w}\cdot\mathbf{x}^{(i)} + b)\right) \end{aligned} \tag{5.32}$$

Độ dốc của lô nhỏ là mức trung bình của các độ dốc riêng lẻ từ biểu thức. 5.29:

$$\frac{\partial Cost(\hat{y},y)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left[\sigma(\mathbf{w}\cdot\mathbf{x}^{(i)}+b)-y^{(i)}\right]\mathbf{x}_j^{(i)} \quad (5.33)$$

Thay vì sử dụng ký hiệu tổng, chúng ta có thể tính toán độ dốc hiệu quả hơn ở dạng ma trận của nó, theo cách vector hóa mà chúng ta đã thấy ở trang 7, nơi chúng ta có một ma trận X có kích thước [m× f] đại diện cho m đầu vào trong lô và vectơ y có kích thước [m×1] thể hiện kết quả đầu ra chính xác:

$$\begin{aligned}\frac{\partial Cost(\hat{y},y)}{\partial \mathbf{w}} &= \frac{1}{m}(\hat{\mathbf{y}}-\mathbf{y})^{\intercal}\mathbf{X} \\ &= \frac{1}{m}(\sigma(\mathbf{X}\mathbf{w}+\mathbf{b})-\mathbf{y})^{\intercal}\mathbf{X}\end{aligned} \quad (5.34)$$

5.7.15 Regularization

Có một vấn đề với trọng số học khiến cho mô hình hoàn toàn phù hợp với dữ liệu huấn luyện. Nếu một đặc điểm có thể dự đoán hoàn hảo về kết quả vì nó xảy ra chỉ xảy ra trong một lớp, nó sẽ được gán một trọng số rất cao. Các trọng số cho các tính năng sẽ cố gắng khớp hoàn toàn các chi tiết của tập huấn luyện, trên thực tế là quá hoàn hảo, mô hình hóa các yếu tố ồn ào vô tình tương quan với lớp. Vấn đề này là được gọi là trang bị quá mức. Một mô hình tốt sẽ có khả năng khái quát hóa tốt từ dữ liệu huấn luyện đến tập kiểm tra không nhìn thấy, nhưng một mô hình phù hợp quá mức sẽ có khả năng khái quát hóa kém.

Để tránh trang bị quá mức, một thuật ngữ chính quy mới R(θ) được thêm vào mục tiêu chức năng trong phương trình. 5.24, dẫn đến mục tiêu sau đây cho một loạt m ví dụ (được viết lại một chút từ phương trình 5.24 để tối đa hóa xác suất loga thay vì giảm thiểu tổn thất và loại bỏ 1/m thuật ngữ không ảnh hưởng đến argmax):

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\sum_{i=1}^{m}\log P(y^{(i)}|x^{(i)}) - \alpha R(\theta) \quad (5.35)$$

Thuật ngữ chính quy hóa mới R(θ) được sử dụng để xử phạt các trọng số lớn. Vì vậy một thiết lập các trọng số khớp hoàn hảo với dữ liệu huấn luyện— nhưng sử dụng nhiều trọng số với giá trị cao để làm như vậy—sẽ bị phạt nhiều hơn cài đặt phù hợp với dữ liệu kém hơn một chút nhưng làm được điều đó bằng cách sử dụng trọng lượng nhỏ hơn. Có hai cách phổ biến để tính toán thuật ngữ chính quy này R(θ). Chính quy hóa L2 là một hàm bậc hai của các giá trị trọng số, được đặt tên vì nó sử dụng định mức (bình phương của) L2 của trọng số các giá trị. Chuẩn L2, ||θ||2, giống như khoảng cách Euclide của vectơ θ từ nguồn gốc. Nếu θ bao gồm n trọng số thì:

$$R(\theta) = ||\theta||_2^2 = \sum_{j=1}^{n}\theta_j^2 \quad (5.36)$$

Hàm mục tiêu chính quy L2 trở thành:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left[\sum_{i=1}^{m} \log P(y^{(i)} | x^{(i)}) \right] - \alpha \sum_{j=1}^{n} \theta_j^2 \qquad (5.37)$$

Chính quy hóa L1 là một hàm tuyến tính của các giá trị trọng số, được đặt tên theo chuẩn L1 ||W||1, tổng các giá trị tuyệt đối của các trọng số hoặc khoảng cách Manhattan (Khoảng cách Manhattan là khoảng cách bạn phải đi bộ giữa hai điểm trong thành phố với mạng lưới đường phố như New York):

$$R(\theta) = ||\theta||_1 = \sum_{i=1}^{n} |\theta_i| \qquad (5.38)$$

Hàm mục tiêu chính quy L1 trở thành:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left[\sum_{1=i}^{m} \log P(y^{(i)} | x^{(i)}) \right] - \alpha \sum_{j=1}^{n} |\theta_j| \qquad (5.39)$$

Những loại chính quy hóa này đến từ thống kê, trong đó chính quy hóa L1 được gọi là hồi quy Lasso (Tibshirani, 1996) và chính quy hóa L2 được gọi là hồi quy sườn và cả hai đều được sử dụng phổ biến trong xử lý ngôn ngữ. Việc chính quy hóa L2 dễ dàng hơn tối ưu hóa vì đạo hàm đơn giản của nó (đạo hàm của θ2 chỉ là 2θ), trong khi Chính quy hóa L1 phức tạp hơn (đạo hàm của |θ| không liên tục tại 0). Nhưng trong khi L2 thích các vectơ trọng số có nhiều trọng số nhỏ thì L1 lại thích các vectơ trọng số thưa thớt hơn. các giải pháp có một số trọng số lớn hơn nhưng nhiều trọng số hơn được đặt thành 0. Như vậy L1 việc chính quy hóa dẫn đến các vectơ trọng số thưa thớt hơn nhiều, tức là có ít tính năng hơn. Cả chính quy hóa L1 và L2 đều có cách diễn giải Bayesian như là các ràng buộc đối với trước về trọng lượng sẽ trông như thế nào. Chính quy hóa L1 có thể được xem như một Laplace trước về trọng lượng. Chính quy hóa L2 tương ứng với việc giả sử rằng các trọng số là được phân phối theo phân phối Gaussian với giá trị trung bình μ = 0. Trong phân phối Gaussian hoặc phân phối chuẩn, giá trị càng xa giá trị trung bình thì giá trị đó càng thấp xác suất (được chia tỷ lệ theo phương sai σ). Bằng cách sử dụng Gaussian trước các trọng số, chúng ta đang nói rằng các trọng số thích có giá trị 0. Một Gaussian cho trọng số θj là

$$\frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\theta_j - \mu_j)^2}{2\sigma_j^2} \right) \qquad (5.40)$$

Nếu chúng ta nhân mỗi trọng số với một Gaussian trước trọng số đó thì chúng ta sẽ tối đa hóa ràng buộc sau:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{m} P(y^{(i)} | x^{(i)}) \times \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\theta_j - \mu_j)^2}{2\sigma_j^2} \right) \qquad (5.41)$$

trong không gian log, với μ = 0, và giả sử 2σ 2 = 1, tương ứng với

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)}) - \alpha \sum_{j=1}^{n} \theta_j^2 \tag{5.42}$$

có dạng tương tự như phương trình. 5.37.

5.7.16 Learning in Multinomial Logistic Regression

Hàm mất mát cho hồi quy logistic đa thức khái quát hóa hàm mất mát cho hồi quy logistic nhị phân từ lớp 2 đến lớp K. Hãy nhớ lại rằng entropy chéo tổn thất đối với hồi quy logistic nhị phân (lặp lại từ phương trình 5.22) là:

$$L_{\text{CE}}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})] \tag{5.43}$$

Hàm mất mát cho hồi quy logistic đa thức khái quát hóa hai số hạng trong phương trình. 5.43 (một giá trị khác 0 khi y = 1 và một giá trị khác 0 khi y = 0) thành điều khoản K. Như chúng tôi đã đề cập ở trên, đối với hồi quy đa thức, chúng tôi sẽ biểu thị cả y và yˆ dưới dạng vectơ. Nhãn đúng y là một vectơ có K phần tử, mỗi phần tử tương ứng đến một lớp, với yc = 1 nếu lớp đúng là c, với tất cả các phần tử khác của y là 0. Và trình phân loại của chúng tôi sẽ tạo ra một vectơ ước tính với K phần tử yˆ, mỗi phần tử yˆ k trong đó biểu thị xác suất ước tính p(yk = 1|x). Hàm mất mát cho một ví dụ x đơn lẻ, tổng quát hóa từ hồi quy logistic nhị phân, là tổng của nhật ký của K lớp đầu ra, mỗi lớp được tính trọng số bằng xác suất yk của chúng (Phương trình 5.44). Điều này hóa ra chỉ là xác suất log âm của đúng lớp c (Phương trình 5.45):

$$L_{\text{CE}}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{k=1}^{K} \mathbf{y}_k \log \hat{\mathbf{y}}_k \tag{5.44}$$

$$= -\log \hat{\mathbf{y}}_c, \quad \text{(where } c \text{ is the correct class)} \tag{5.45}$$

$$= -\log \hat{p}(\mathbf{y}_c = 1|\mathbf{x}) \quad \text{(where } c \text{ is the correct class)}$$

$$= -\log \frac{\exp(\mathbf{w_c} \cdot \mathbf{x} + b_c)}{\sum_{j=1}^{K} \exp(\mathbf{w_j} \cdot \mathbf{x} + b_j)} \quad (c \text{ is the correct class}) \tag{5.46}$$

Làm thế nào chúng ta có được từ phương trình. 5,44 đến phương trình. 5 giờ 45? Bởi vì chỉ có một lớp (hãy gọi nó là c) là đúng, vectơ y chỉ nhận giá trị 1 cho giá trị k này, tức là có yc = 1 và yj = 0 ∀ j 6= c. Điều đó có nghĩa là các số hạng trong tổng trong biểu thức. 5,44 tất cả sẽ là 0 ngoại trừ cho thuật ngữ tương ứng với lớp thực sự c. Do đó sự mất mát entropy chéo chỉ đơn giản là log của xác suất đầu ra tương ứng với lớp chính xác và do đó chúng tôi cũng gọi phương trình. 5,45 khả năng mất nhật ký âm.

Tất nhiên, để giảm độ dốc, chúng ta không cần mất mát, chúng ta cần độ dốc của nó. Các gradient cho một ví dụ đơn lẻ hóa ra rất giống với gradient cho nhị phân hồi quy logistic, (yˆ−y)x, mà chúng ta đã thấy trong biểu thức. 5,29. Chúng ta hãy xem xét một phần của gradient, đạo hàm của một trọng số. Với mỗi lớp k, trọng số của lớp thứ i phần tử của đầu vào x là wk,i. Đạo hàm riêng của tổn thất đối với wk,ii? Đạo hàm này hóa ra chỉ là sự khác biệt giữa giá trị thực của lớp k (là 1 hoặc 0) và xác suất mà bộ phân loại đưa ra cho lớp k,

được tính trọng số bằng giá trị của xi đầu vào tương ứng với phần tử thứ i của trọng số vectơ cho lớp k:

$$\begin{aligned}\frac{\partial L_{\text{CE}}}{\partial \mathbf{w}_{k,i}} &= -(\mathbf{y}_k - \hat{\mathbf{y}}_k)\mathbf{x}_i \\ &= -(\mathbf{y}_k - p(\mathbf{y}_k = 1|\mathbf{x}))\mathbf{x}_i \\ &= -\left(\mathbf{y}_k - \frac{\exp\left(\mathbf{w_k}\cdot\mathbf{x} + b_k\right)}{\sum_{j=1}^{K}\exp\left(\mathbf{w_j}\cdot\mathbf{x} + b_j\right)}\right)\mathbf{x}_i \end{aligned} \tag{5.47}$$

Chúng ta sẽ quay lại trường hợp gradient cho hồi quy softmax này khi chúng tôi giới thiệu mạng lưới thần kinh trong Chương 7, và lúc đó chúng ta cũng sẽ thảo luận về nguồn gốc của gradient này trong phương trình Eq. ??–phương trình. ??.

5.7.17 interpreting models

Thông thường chúng ta muốn biết nhiều hơn là chỉ phân loại chính xác một quan sát.

Chúng tôi muốn biết tại sao bộ phân loại lại đưa ra quyết định như vậy. Đó là, chúng tôi muốn quyết định của mình có thể hiểu được. Khả năng diễn giải có thể khó xác định một cách chặt chẽ, nhưng Ý tưởng cốt lõi là với tư cách là con người, chúng ta nên biết tại sao thuật toán của chúng ta lại đưa ra kết luận như vậy. Bởi vì các đặc điểm của hồi quy logistic thường do con người thiết kế, Một cách để hiểu quyết định của bộ phân loại là hiểu vai trò của từng đặc điểm chơi trong quyết định. Hồi quy logistic có thể được kết hợp với các kiểm tra thống kê (kiểm tra tỷ lệ khả năng hoặc kiểm tra Wald); điều tra xem liệu một tính năng cụ thể có có ý nghĩa bởi một trong những thử nghiệm này hoặc việc kiểm tra độ lớn của nó (trọng số w liên quan đến đối tượng địa lý lớn đến mức nào?) có thể giúp chúng tôi giải thích lý do tại sao bộ phân loại thực hiện quyết định mà nó đưa ra. Điều này cực kỳ quan trọng để xây dựng các mô hình minh bạch. Hơn nữa, ngoài việc sử dụng nó như một công cụ phân loại, hồi quy logistic trong NLP và nhiều lĩnh vực khác được sử dụng rộng rãi như một công cụ phân tích để kiểm tra các giả thuyết về ảnh hưởng của các biến giải thích khác nhau (tính năng). Trong phân loại văn bản, có lẽ chúng ta muốn biết liệu những từ tiêu cực về mặt logic (không, không, không bao giờ) có nhiều khả năng liên quan đến cảm xúc tiêu cực hay không, hoặc liệu những đánh giá tiêu cực về phim có nhiều khả năng dẫn đến thảo luận về điện ảnh. Tuy nhiên, khi làm như vậy cần phải kiểm soát những yếu tố có thể xảy ra: các yếu tố khác có thể ảnh hưởng đến cảm xúc (thể loại phim, năm nó được thực hiện, có lẽ là độ dài của bài đánh giá bằng lời). Hoặc chúng ta có thể đang nghiên cứu mối quan hệ giữa các đặc điểm ngôn ngữ được trích xuất từ NLP và các đặc điểm phi ngôn ngữ. kết quả (tái nhập viện, kết quả chính trị hoặc doanh số bán sản phẩm), nhưng cần phải kiểm soát các yếu tố gây nhiễu (tuổi của bệnh nhân, quận bầu cử, nhãn hiệu của sản phẩm). Trong những trường hợp như vậy, hồi quy logistic cho phép chúng ta kiểm tra xem một số tính năng có liên quan đến một số kết quả ở trên và ngoài tác dụng của các tính năng khác.

5.7.18 Advanced: Deriving the Gradient Equation

Trong phần này chúng tôi đưa ra đạo hàm gradient của hàm entropy chéo LCE cho hồi quy logistic. Hãy bắt đầu với một số nội dung ôn tập nhanh về phép tính. Đầu tiên, đạo hàm của ln(x):

$$\frac{d}{dx}\ln(x) = \frac{1}{x} \tag{5.48}$$

Thứ hai, đạo hàm của sigmoid:

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z)) \tag{5.49}$$

Cuối cùng là quy tắc đạo hàm. Giả sử chúng ta đang tính đạo hàm của hàm tổng hợp f(x) = u(v(x)). Đạo hàm của f(x) là đạo hàm của u(x) đối với v(x) nhân đạo hàm của v(x) đối với x:

$$\frac{df}{dx} = \frac{du}{dv} \cdot \frac{dv}{dx} \tag{5.50}$$

Đầu tiên, chúng ta muốn biết đạo hàm của hàm mất đối với một đơn vị trọng số wj (chúng ta sẽ cần tính nó cho từng trọng số và độ lệch):

$$\begin{aligned}\frac{\partial L_{\text{CE}}}{\partial \mathbf{w}_j} &= \frac{\partial}{\partial \mathbf{w}_j} - [y \log \sigma(\mathbf{w}\cdot\mathbf{x}+b) + (1-y)\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \\ &= -\left[\frac{\partial}{\partial \mathbf{w}_j} y \log \sigma(\mathbf{w}\cdot\mathbf{x}+b) + \frac{\partial}{\partial \mathbf{w}_j}(1-y)\log[1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]\right]\end{aligned} \tag{5.51}$$

Tiếp theo, sử dụng quy tắc chuỗi và dựa vào đạo hàm của log:

$$\frac{\partial L_{\text{CE}}}{\partial \mathbf{w}_j} = -\frac{y}{\sigma(\mathbf{w}\cdot\mathbf{x}+b)}\frac{\partial}{\partial \mathbf{w}_j}\sigma(\mathbf{w}\cdot\mathbf{x}+b) - \frac{1-y}{1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)}\frac{\partial}{\partial \mathbf{w}_j}1-\sigma(\mathbf{w}\cdot\mathbf{x}+b) \tag{5.52}$$

Sắp xếp lại các điều khoản:

$$\frac{\partial L_{\text{CE}}}{\partial \mathbf{w}_j} = -\left[\frac{y}{\sigma(\mathbf{w}\cdot\mathbf{x}+b)} - \frac{1-y}{1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)}\right]\frac{\partial}{\partial \mathbf{w}_j}\sigma(\mathbf{w}\cdot\mathbf{x}+b) \tag{5.53}$$

Và bây giờ thay đạo hàm của sigmoid vào và sử dụng quy tắc chuỗi nhiều thời gian hơn, chúng tôi kết thúc với phương trình. 5,54:

$$\begin{aligned}\frac{\partial L_{\text{CE}}}{\partial \mathbf{w}_j} &= -\left[\frac{y-\sigma(\mathbf{w}\cdot\mathbf{x}+b)}{\sigma(\mathbf{w}\cdot\mathbf{x}+b)[1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]}\right]\sigma(\mathbf{w}\cdot\mathbf{x}+b)[1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]\frac{\partial(\mathbf{w}\cdot\mathbf{x}+b)}{\partial \mathbf{w}_j} \\ &= -\left[\frac{y-\sigma(\mathbf{w}\cdot\mathbf{x}+b)}{\sigma(\mathbf{w}\cdot\mathbf{x}+b)[1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]}\right]\sigma(\mathbf{w}\cdot\mathbf{x}+b)[1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]\mathbf{x}_j \\ &= -[y-\sigma(\mathbf{w}\cdot\mathbf{x}+b)]\mathbf{x}_j \\ &= [\sigma(\mathbf{w}\cdot\mathbf{x}+b)-y]\mathbf{x}_j\end{aligned} \tag{5.54}$$

5.7.19 Summary

Chương này giới thiệu mô hình hồi quy logistic của phân loại. • Hồi quy logistic là một trình phân loại máy học có giám sát, trích xuất các đặc điểm có giá trị thực từ đầu vào, nhân mỗi đặc điểm với một trọng số, tính tổng chúng và chuyển tổng đó qua hàm sigmoid để tạo ra xác suất. Một ngưỡng được sử dụng để đưa ra quyết định. • Hồi quy logistic có thể được sử dụng

với hai lớp (ví dụ: cảm xúc tích cực và tiêu cực) hoặc với nhiều lớp (hồi quy logistic đa thức, ví dụ để phân loại văn bản n-ary, ghi nhãn từng phần của giọng nói, v.v.). • Hồi quy logistic đa thức sử dụng hàm softmax để tính xác suất. • Các trọng số (vectơ w và độ lệch b) được học từ tập huấn luyện được dán nhãn thông qua hàm mất mát, chẳng hạn như mất mát entropy chéo, phải được giảm thiểu. • Giảm thiểu hàm mất mát này là một bài toán tối ưu lồi và các thuật toán lặp như giảm độ dốc được sử dụng để tìm trọng số tối ưu. • Việc điều chỉnh chính quy được sử dụng để tránh việc trang bị quá mức. • Hồi quy logistic cũng là một trong những công cụ phân tích hữu ích nhất vì khả năng nghiên cứu một cách minh bạch tầm quan trọng của từng đặc điểm riêng lẻ.