**Học thống kê – Statistical Learning (SL)**

**1 Học máy – Machine Learning (ML)**

Một trong những công cụ hữu ích trong phân tích dữ liệu là *học máy* – Machine Learning (ML): tìm ra quy luật đơn giản để dự báo chính xác một số đặc trưng cơ bản của của dữ liệu mới là dữ liệu không có trong tập mẫu hay dữ không được học.

ML muốn học tìm ra một số *quy tắc* nhằm giải thích dữ liệu. Hơn thế nữa, các quy tắc có tính tổng quát hóa, nghĩa là, các quy tắc này không chỉ mô tả đúng những dữ liệu được học (đã có trong CSDL), mà còn có khả năng mô tả đúng cả những dữ liệu mới (không được học) có cùng phân phối.

Ta định nghĩa bài toán học máy theo mô hình PAC (Probability Approximately Correct) của Valiant như sau:

***Định nghĩa***: Đặt là miền xác định các mẫu chưa được gán nhãn. Không mất tính tổng quát, ta xem nhãn là một giá trị trong tập {0, 1}. Một mẫu được gán nhãn là cặp.

Học (giám sát) được định nghĩa như một phân phối trên tập mẫu được gán nhãn. Mục đích là tìm ra hàm đánh nhãn đúng hầu hết các mẫu được lấy theo phân phối .

***Định nghĩa***: cho hàm và một phân phối trên tập mẫu đã gán nhãn, sai số của trên được xác định theo công thức:

.

Trong thực hành, ta có thể định nghĩa sai số của trên tập mẫu hữu hạn D theo:

.

(Trong trường hợp là số thực, ta thay trong hàm sai số trên bằng , với cho trước.)

Thuật toán học được quan sát một số mẫu được gán nhãn rút ra từ và mục tiêu là tìm ra hàm *f* với sai số (đo trên ) nhỏ nhất có thể.

Chất lượng một thuật toán học được đo theo thời gian chạy thuật toán và số mẫu cần có để tìm được giả thiết *f* tốt.

***Định nghĩa***: Thuật toán *A* được gọi là PAC-learn học lớp hàm d-chiều C nếu với mọi α, β > 0, tồn tại m là hàm đa thức theo d, 1/α, log(1/β) sao cho với mọi phân phối trên tập mẫu gán nhãn, A nhận vào m mẫu gán nhãn lấy theo phân phối và kết xuất một giả thiết *f* ∈ C sao cho với xác suất 1 - β:

.

Nếu thuật toán A được gọi là *có thể hiện thực* (nghĩa là có một số hàm *f* trong lớp hàm C gán nhãn dữ liệu hoàn hảo).

Nếu A có thời gian chạy là hàm đa thức theo d, 1/α, log(1/β), thuật toán *A là thuật toán hiệu quả*.

Nếu có một thuật toán PAC-learn C, C được gọi là PAC-learnable.

**2 Hồi quy tuyến tính**

Hồi quy là học giám sát, tuy đơn giản nhưng rất phổ biến trong phân tích dữ liệu. Trong trường hợp này, C là lớp hàm tuyến tính. Nếu dữ liệu có quan hệ tuyến tính, C sẽ là PAC-learnable.

***2.1 Hồi quy một biến***

Hàm hồi quy tuyến tính trong trường hợp này có dạng:

,

với a, b là các hằng thực và ε*i* là biến ngẫu nhiên biểu diễn hiệu quả biến động ngẫu nhiên. Giả sử

ε1, ε2, …

độc lập và có phân phối đều với σ > 0.

Do thừa số lỗi ε*i* có trung bình 0,

.

Vì thế bài toán ước lượng E*yt* dẫn về bài toán ước lượng tham số a, b. Và đường

,

gọi là *đường hồi quy*.

Cho tập dữ liệu

Ω = {(x1, y1), …, (xn, yn)},

độ “khớp” của đường hồi quy a + bx là:

,

tính theo điểm (xi, yi) và đường hồi quy tại điểm x = xi.

Để đơn giản cho bài toán tối ưu, thay vì dùng sai số truyệt đối, ta sử dụng sai số bình phương, và cực tiểu hóa hàm mục tiêu:

.

Đường khớp nhất là đường cực tiểu hóa hàm mục tiêu trên:

.

Ta cần tìm các tham số của đường hồi quy sao cho đường khớp nhất với tập mẫu được học nên ta gọi là bài toan tối ưu tham số:

.

Điểm cực trị là điểm tại đó đạo hàm bằng 0. Vì thế, ta sẽ thiết lập và giải các phương trình đạo hàm

.

Trong bài toán đang xét, ta có hệ phương trình đạo hàm sau:

.

Giải hệ trên ta được:

,

với .

Đặt

,

là giá trị dự báo. Hệ số tương quan

,

dùng để đo phần biến thiên tổng thể trong dữ liệu theo đường hồi quy. Phần còn lại là sai số. Nếu R2 gần 1, dữ liệu có quan hệ rất gần tuyến tính. Nếu R2 gần 0, dữ liệu hầu như ngẫu nhiên.

***2.2 Lớp hàm đa thức***

Trong trường hợp dữ liệu không có tương quan tuyến tính rõ rệt, R2 nửa phía 0, có thể học (hồi quy) thử với lớp các hàm đa thức:

.

Các giá trị của a0, a1, a2, …, am là giá trị để hàm Q định nghĩa bởi:

,

đạt cực tiểu.

Ta cũng sẽ dùng phương pháp bình phương tối tiểu giải bài toán tối ưu sau:

.

Tương tự phương pháp tìm đường hồi quy, ta cần thiết lập và giải hệ phương trình vi phân sau:

.

Sau một số bước biến đổi, ta có hệ phương trình tuyến tính chuẩn tắc theo ai, i = 0, 1, 2, …, m như sau:

.

Hệ có nghiệm duy nhất nếu định thức của ma-trận A khác 0, với A là:

.

***2.3 Hồi quy tuyến tính nhiều biến***

Bây giờ, ta mở rộng phương pháp hồi quy tuyến tính một biến cho trường hợp vec-tơ mẫu d-chiều. Một cách tổng quát, ta muốn xây dựng hàm có dạng:

.

Ta cũng sử dụng phương pháp bình phương tối tiểu để học mô hình từ tập mẫu

.

Đặt

.

Cũng như trong trường hợp một biến, ta giải bài toán tối ưu

.

Bằng cách thiết lập hệ phương trình đạo hàm riêng , ta được hệ chuẩn tắc:

,

,

…

.

Hệ trên có nghiệm duy nhất nếu định thức ma-trận (d+1) x (d+1) tương ứng khác 0.

**2.3 Chuỗi thời gian (TS – Time Series)**

Một ứng dụng phổ biến của phân tích dữ liệu là bài toán dự đoán theo thời gian, bài toán chuỗi thời gian – Time Series (TS).

TS là một tiến trình ngẫu nhiên thay đổi theo thời gian, thường được quan sát trong một khoảng thời gian cố định. Ví dụ nhiệt độ từng ngày, thu nhập theo năm, …là những ví dụ về TS. Công cụ cơ bản để mô hình hóa TS là hồi quy. Ở đây ta tiếp cận một số phương pháp tựa hồi quy khác. Ta bắt đầu bằng ví dụ sau.

Một TS là dãy biến ngẫu nhiên {Xt} theo thời gian t = 0, 1, 2, … theo một phân phối (mô hình) ngẫu nhiên nào đó. Vì thế, mô hình hóa TS là nhận dạng mẫu phân phối này. Giả sử mô hình liên quan đến một xu hướng trong một chuỗi thời gian cố định. Xu hướng giả sử là hàm không-ngẫu nhiên theo thời gian và diễn tả giá trị trung bình của chuỗi. Nếu loại bỏ được xu hướng, ta chỉ còn lại một số không phải trung bình của chuỗi nữa, và ta muốn mô hình cấu trúc phụ thuộc của nó. Trường hợp đơn giản nhất là chuỗi thời gian còn lại sau khi loại xu hướng, chỉ gồm các biến ngẫu nhiên độc lập. Tuy nhiên, thường thì không độc lập và ta cần tìm sự phụ thuộc giữa các biến này. Sự phụ thuộc được đo theo hiệp phương sai, Cov(X1, X2), của 2 biến ngẫu nhiên:

Cov(X1, X2) = E[(X1 - μ1)(X2 - μ2)],

với

μi = E(Xi)

là kỳ vọng hay trung bình xác suất.

Nếu X1 và X2 độc lập, Cov(X1, X2) = 0. Nếu hiệp phương sai dương, trung bình X1 cao thì trung bình của X2 cũng cao. Chẳng hạn, nếu X1 là thu nhập và X2 là thuế thu nhập của người đó, thì Cov(X1, X2) sẽ dương. Không thể suy ra thuế thu nhập nếu chỉ biết thu nhập, nhưng dễ lý giải tại sao người có thu nhập cao phải đóng thuế nhiều hơn. Về mặt toán học, (X1 - μ­1) là độ lệch của thu nhập theo trung bình, tương tự cho (X2 - μ2). Hiệp phương sai là kỳ vọng của tích 2 đại lượng này. Nếu cả 2 cùng dấu thì tích sẽ dương cho thấy quan hệ dương. Lưu ý là hiệp phương sai là trường hợp tổng quát của phương sai:

Var(X) = Cov(X, X).

Một khái niệm rất gần với hiệp phương sai là tương quan:

,

là phiên bản không thứ nguyên của hiệp phương sai.

Nếu X1, X2 độc lập, corr(X1, X2) = 0, và ta nói X1 và X2 không tương quan.

Ta có . Nếu tương ứng với X2 phụ thuộc tuyến tính vào X1. ρ > 0, X1 và X2 tương quan dương; ρ < 0, tương quan âm.

Tương quan dùng đo sự phụ thuộc trong TS. Theo đó, gọi

ρ(t, h) = corr(Xt, Xt+h)

là hàm tự tương quan của TS. Đo sự phụ thuộc giữa các TS tại những thời điểm khác nhau. TS là tĩnh (hay tĩnh yếu) nếu E(Xt) và ρ(t, h) là hằng số. Khi phân tích TS, cần loại x hướng để có chuỗi tĩnh. Chẳng hạn, ta khử xu hướng chuỗi Xi bằng hồi quy tuyến tính để xác định xu hướng tuyến tính a + bt, để lại thừa số sai số εt mà ta mô hình như biến độc lập và phân phối đồng nhất. Trong phân tích TS, εt được gọi là nhiễu ngẫu nhiên. Nó là TS trung tâm đơn gian nhất. Tổng quát hơn, ta hy vọng chuỗi trung tâm ít nhất ổn định, với một cấu trúc tương quan vẫn giữ nguyên theo thời gian. Có nhiều cách kiểm định cho tính ổn định, nhưng đơn giản nhất là vẽ sai số εt theo t xem chúng có xuất hiện theo một mô hình (ngẫu nhiên) nhất quán hay không. Nếu nó mở rộng hay thu hẹp phân phối thì là không ổn định. Hiện tượng này gọi là không đồng nhất, ngược lại, được cho là ổn định. Nếu ST được cho là ổn định, nên thử mô hình cấu trúc hiệp phương sai. Mô hình đơn giản nhất là tiến trình tự hồi quy:

.

Tham số p được gọi là bậc tiến trình tự hồi quy, viết tắt AR(p). Ta có thể tính các tham số của tiến trình tự hồi quy bằng giá trị quan sát được Xt theo nhiều yếu tố dự báo. Trước hết là yếu tố thời gian t; sau đó là các yếu tố Xt-1, …, Xt-p. Điểm cốt lõi của phương pháp AR(p) là chọn được đúng p. Có 2 cách:

(**c1**) Cách thứ nhất là xem xét giá trị R2. Vì theo lý, cứ thêm thông tin thì dự báo sẽ tốt hơn, nên thêm yếu tố dự báo thì sẽ tăng R2. Tuy nhiên, nếu R2 chỉ tăng một lượng nhỏ thì lượng thông tin thêm vào không đáng quan tâm. Vì thế, cách làm là thêm Xt-1, rồi Xt-2, … cho đến khi R2 không tăng nữa. Nhắc lại ở đây, công thức tính R2 là

,

với lần lượt là giá trị quan sát, dự đoán và trung bình. Nên theo nguyên lý “*tạo mô hình đơn giản nhất có thể, không hy sinh khả năng dự đoán*”.

(**c2**) Cách thứ hai là xem xét phần thặng dư trong mô hình TS, nghĩa là ước lượng sai số εt. Khi đã xác định các tham số hồi quy a, b, ci, ta sử dụng công thức

,

để ước lượng sai số. Mục tiêu của ta là gom đủ yếu tố dự báo vào mô hình để nắm được cấu trúc độc lập, nên ta muốn εt là chuỗi nhiễu không tương quan. Như vậy ta cần kiểm tra hàm tự tương quan của thặng dư. Giả sử hàm tự tương quan là tiến trình ổn định, ta có thể kiểm tra bằng đồ thị hàm thặng dư.

**3 Mạng nơ-ron**

Ở trên, ta đã nghiên cứu các phương pháp học đơn giản nhất, hồi quy tuyến tính để xây dựng các mô hình dự đoán giá trị đầu ra khi biết vec-tơ giá trị đầu vào . Phương pháp hồi quy tuyến tính sẽ rất tốt nếu dữ liệu có quan hệ tuyến tính. Tuy nhiên trong thực tế, nhiều bài toán dữ liệu thường không có quan hệ tuyến tính cách tường mình, hoặc do bản chất dữ liệu đã không tuyến tính; hoặc do ảnh hưởng nhiểu; hoặc cả hai. Các bài toán phân lớp là điển hình, ở đó, cần xây dựng mô hình “gán nhãn” cho một vec-tơ đầu vào: . Mặt khác, như đã lưu ý, ma-trận thiết lập từ các phương trình đạo hàm có thể suy biến, mặc dù có thể sử dụng một số kỹ thuật đại số tuyến tính để khắc phục, nhưng đó cũng là vấn đề khi cài đặt hồi quy tuyến tính tổng quát.

***3.1 Hồi quy logistic***

Các phương pháp hồi quy (tuyến tính, tựa tuyến tính) như trình bày trước, học từ tập mẫu để tìm mô hình có đầu ra là giá trị thực: . Hồi quy tuyến tính, như ta đã thấy, cho kết quả tốt khi giữa biến kết quả (biến phụ thuộc), , và các biến tiên lượng (biến độc lập), , có quan hệ tuyến tính.

Trong thực tế, có nhiều bài toán, kiểu như bài toán phân loại, ở đó, đầu ra là các “nhãn” rời rạc. Không mất tính tổng quát, ta xét bài toán học để gán nhãn cho một đối tượng, được mô tả bằng một vec-tơ các giá trị thực , bằng một nhẵn 0 hoặc 1:

.

Để dễ hình dung, giả sử ta được cho tập dữ liệu đã gán nhãn

.

Xét hàm tuyến tính

.

Để gán nhãn cho vec-tơ đầu vào , ta sử dụng một giá trị ngưỡng θ = 0:

,

thì hàm sign gán nhãn hoàn hảo cho tập mẫu Ω được cho.

Điểm cốt lõi ở đây là ta cần tìm ra được hàm có khả năng tách tập mẫu Ω thành 2 tập tách biệt và một ngưỡng θ để gán nhãn cho vec-tơ đầu vào . Có một số cách để tìm hàm phân tách , chẳng hạn phương pháp SVM – Support Vector Machine, sẽ giới thiệu sau, dùng các kỹ thuật thống kê để tìm những siêu phẳng cắt tập mẫu thành những tập tách biệt nhất có thể. Ở đây, ta sẽ sử dụng kỹ thuật học máy rất phổ biến trong những năm gần đây: *mạng nơ-ron* – Neural Network (NN).

Mạng nơ-ron là mô hình có nguồn gốc rất gần từ thống kê, đặc biệt với hồi quy tuyến tính. Mô hình mạng nơ-ron đơn giản nhất là mạng *perception* hay hồi quy logistic. Trước hết, ta có các khái niệm sau.

Gọi là xác suất của một biến cố, chẳng hạn vec-tơ được gán nhãn trong ví dụ bài toán gán nhãn ta đang xét,

.

Đặt là tỉ số giữa xác xuất và :

.

như trên mô tả khả năng của biến cố xảy ra.

Cho tập mẫu được gán nhãn Ω, mạng perception sẽ học từ tập mẫu Ω để tìm ra một quan hệ tuyến tính giữa biến tiên lượng với log-odds, ký hiệu là :

,

vì thế mạng perception được gọi là hồi quy logistic.

Giả sử, quan hệ là:

.

Ta có

,

với .

Và xác suất ta cần tính là đại lượng:

=.

Trở lại bài toán học hồi quy logistic. Xét lớp hàm logistic

,

với .

Bài toán học của ta là tìm các hệ số từ tập mẫu gán nhãn quan sát được Ω.

Đặt

,

.

Sử dụng phương pháp MLE (Maximun Likelihood Estimator), cực đại hóa hàm sau để tìm các hệ số :

.

***Thuật toán học – hồi quy logistic***

Để đơn giản và sử dụng kí hiệu vec-tơ, ma-trận sau này, ta kí hiệu lại , và vec-tơ trọng số:

.

Ta cũng biểu diễn lại vec-tơ các biến tiên lượng

,

với biến đầu tiên là hằng số và luôn bằng 1. Khi ấy,

.

Ta sẽ dùng phương pháp giảm đạo hàm – Gradient Descent (GD) để tìm là các hệ số hàm hồi quy logistic F xấp xỉ tập mẫu:

.

Phương pháp GD gồm các bước chính sau:

1. Chọn ngẫu nhiên một điểm trong không gian trọng số;
2. Tính độ dốc mặt lỗi tại **;**
3. Cập nhật lạitheo hướng dốc nhất của mặt lỗi;
4. Xem điểm mới tính này như điểm mới

(lặp lại quá trình trên cho điến khi điều kiện dừng thỏa.)

Giả sử

,

với là hàm logistic như định nghĩa ở trên. Hàm này có đạo hàm là

.

Để tìm được sao cho mô hình khớp nhất với tập Ω, ta sẽ cực tiểu hàm lỗi

với .

bằng cách giải phương trình vi phân

.

(Ta sẽ bỏ qua các chỉ số mẫu cho đơn giản các công thức.)

Trường hợp 1:

.

• Thừa số thứ nhất,

.

• Thừa số thứ 2:

.

• Thừa số thứ 3:

Từ , ta có

.

Cuối cùng, ta có công thức cập nhật trọng số (bước (3) của tiến trình GD):

,

,

được gọi là vec-tơ biến thiên trọng;

là vec-tơ đạo hàm được tính như trên,

gọi là hệ số học.

Hay ta viết công thức cập nhật từng trọng số như sau:

,

.

Để giải bài toán phân lớp tổng quát, trong đó, thay vì như ví dụ trên, chỉ phân loại 2 nhóm, , bây giờ ta phải gán cho vec-tơ đầu vào một nhãn thuộc tập .

Cách ngây thơ nhất là dựa trên phân loại 2 nhóm như trên, theo đó tính xác suất từng nhóm tương ứng rồi chọn nhóm có xác suất cao nhất làm nhãn cho :

.

Việc quyết định nhóm vẫn dựa trên ngưỡng θ: giả sử có xác suất cao nhất, nếu thì gán nhãn k cho .

Rõ ràng phương pháp là đơn giản, dễ hiểu, nhưng nó chỉ có thể chấp nhận được khi số lớp ít. Trong trường hợp lớn, như 26 lớp chữ cái hay nhận dạng cả ngàn người chẳng hạn, cần một cách tiếp cận hiệu quả hơn.

Trước hết, ta sẽ biểu diễn nhóm thứ bằng một vec-tơ mà chỉ có phần tử thứ bằng 1, tất cả các phần tử khác bằng 0: . Kỹ thuật biểu diễn này được gọi là biểu diễn bằng *vec-tơ 1-hot*. Khi ấy, hồi quy logistic là mô hình

.

Và vec-tơ được gán nhãn

.

Trong trường hợp này, tương tự như trường hợp phân loại 2 lớp, để tìm các hệ số của mô hình, ta cực tiểu hóa hàm lỗi:

,

với là ma-trận trọng số được tạo từ d vec-tơ trọng số được cập nhật đồng thời:

,

.

***3.2 Mạng nơ-ron***

Hồi quy logistic như khảo sát trên có thể xem như một mở rộng rất tốt của hồi quy tuyến tính khi giải cả bài toán phân lớp; và nếu đầu ra không phải các nhãn lớp mà là một số thực, , thì bằng một vài kỹ thuật đơn giản, ta có thể “co” đầu ra về đoạn [0, 1], rồi ta có thể sử dụng hồi quy logistic. Hơn thế nữa, đầu ra có thể là một vec-tơ như trường hợp ta dùng biểu diễn vec-tơ 1-hot cho bài toán phân loại nhiều lớp, . Tương tự, ta cũng hoàn có thể dùng hồi quy logistic cho bài toán đầu ra là một vec-tơ thực, .

Tuy nhiên, hồi quy logistic chỉ thực sự tốt khi dữ liệu *khả tách*. Trong trường hợp dữ liệu không có quan hệ khả tách, việc áp dụng hồi quy logistic như trình bày trên sẽ không còn hiệu quả, thậm chí không thể tìm ra bộ trọng , nghĩa là thuật toán lặp sẽ không dừng tự nhiên được. Chẳng hạn, sửa lại tập

.

trong ví dụ của hồi quy logistic thành:

,

thì không thể tìm được tập trọng phân loại hoàn hảo các mẫu trong Ω.

Bằng cách thêm một hàm trung gian, ta có thể giải bài toán này theo kiểu 2 bước:

* Bước 1: dùng hồi quy logistic để tách kết quả sơ lược;
* Bước 2: dùng thêm hồi quy logistic để phân loại.

Đây chính là kỹ thuật *mạng nơ-ron truyền thẳng*, mà ta sẽ gọi tắt là *mạng truyền thẳng* hay đơn giản là *mạng nơ-ron*, sẽ được trình bày trong phần kế.

***3.2.1 Mạng truyền thẳng – Neural Network (NN)***

***3.2.1.a Ánh xạ NN***

Trước hết, ta định nghĩa mạng nơ-ron là một ánh xạ từ vào :

,

trong đó, được tính như sau:

**NN**( //thuật toán FORWARD – lan truyền tiến

(B1) //Tính

• tính

• tính

(B2) //Tính

• tính

• tính

Trong đó, , , với cho trước, là các ma-trận thực được cho; và là hàm logistic định nghĩa trong phần hồi quy logistic.

A, B là các ma-trận trọng số mà ta phải tìm.

***3.2.1.b Công thức học***

Như trong hồi quy logistic, trọng số, ta sẽ ký hiệu hay đơn giản chỉ , thay cho các trọng số , hay công thức cập nhật trọng số:

,

,

với ký hiệu đạo hàm hàm lỗi; và các ký hiệu mang ý nghĩa như cũ: biến thiên trọng và hệ số học.

Bây giờ, cho tập n mẫu:

.

Đặt

.

Ta sẽ thiết lập công thức cập nhật các trong số A và B mà thực chất chỉ cần xác định công thức tính đạo hàm theo các trọng số.

*Đạo hàm theo*

Từ công thức chuỗi đạo hàm:

,

ta tính từng thành phần bên vế phải.

•

Một thành phần trong vec-tơ kết quả không ảnh hưởng gì đến sai số của các thành phần kết quả khác. Để đơn giản, ta bỏ qua các chỉ số cả các thành phần của và chỉ viết . Nếu kết quả thành phần đang xét là và giá trị đúng phải là , thì sai số bình phương là:

.

Bất kỳ thay đổi nào của cũng làm thay đổi giá trị của , do không đổi, và tỷ lệ thay đổi này bằng , nghĩa là

.

•

Số hạng thứ hai trong chuỗi đạo hàm, , chính là đạo hàm hàm theo . Do với là hàm logistic, bất kỳ giá trị nào của , ảnh hưởng của đối với chính là độ dốc của hàm logistic tại điểm đó, và được tính theo công thức:

.

•

Số hạng thức ba, , trong công thức diễn tả sự thay đổi của phụ thuộc vào thay đổi của trọng số. Do

,

ta tính được:

. Với , ta có,

,

Nghĩa là thay đổi cùng tỷ lệ với trọng số . Điều này có nghĩa độ dốc của hàm theo , là 1, bất chấp giá trị của .

. Với các trong số khác, ta có

.

Tức là khi trọng số thay đổi thì thay đổi một lượng bằng với giá trị tương ứng. Độ dốc là

Tóm lại, nếu đặt

,

thì đạo hàm hàm lỗi theo trọng số được tính theo công thức:

.

*Đạo hàm theo A*

Từ

,

ta cũng sẽ lần lượt tính các số hạng vế phải.

•

Số hạng thứ nhất này chính là đạo hàm hàm lỗi đối với kết quả của hàm . Các kết quả này không tạo ra lỗi nhưng nó góp phần tác động vào lỗi kết quả chung cuộc, .

Ta cũng sử dụng công thức chuỗi đạo hàm

(ta lược bỏ chỉ số j cho đơn giản).

Công thức cho thấy ảnh hưởng của kết quả trung gian vào hàm lỗi là một tổng, theo tất cả kết quả cuối cùng , của tích 3 số hạng.

Hai số hạng đầu có ý nghĩa tương tự như trong phần trước. Chúng diễn tả sự thay đổi của lỗi theo kết quả của cùng với thay đổi của .

Bằng cách ký hiệu cho công thức ta đặt trong phần trước,, để chỉ rõ tương ứng với nào, ta có thể viết lại phương trình trên cho hai số hạng đầu:

.

(Công thức này có vẻ dễ hiểu hơn.)

Bây giờ, ta xét số hạng thứ 3 trong tổng .

Số hạng này, , biểu diễn tác động của kết quả trung gian trên tổng trọng hóa các dữ liệu đầu vào . Khi kết quả trung gian tăng, tổng trọng hóa cũng tăng một lượng bằng

.

Tóm lại, ta có

.

•

Số hạng thứ 2 trong chuỗi đạo hàm, diễn tả sự thay đổi trong kết quả trung gian đối với tổng trọng hóa, các dữ liệu đầu vào . Do , ta có

.

Đặt

,

Ta có

.

•

Số hạng cuối cùng, , chính là thay đổi của đối với trọng số . Nó phụ thuộc vào hay các hệ số .

Đối với , ta có

.

Với các khác,

.

Tóm lại, đạo hàm hàm lỗi theo trọng số được tính theo công thức:

.

***3.3 Một số lưu ý***

***3.3.1 Về hệ số học η***

Ta thấy, tại mỗi bước học , trọng số được cập nhật theo biến thiên trọng tỉ lệ với đạo hàm hàm lỗi :

,

trong đó là hệ số học được chọn theo kinh nghiệm người thiết kế mạng. Nếu quá lớn, thuật toán có thể không hội tụ do điểm trọng số dao động liên tục quanh điểm cực trị; để đảm bảo hội tụ về cực trị, nên nhỏ, nhưng nếu quá nhỏ, tốc độ hội tụ rất chậm.

Ban đầu, khi mạng nơ-ron mới được khám phá, các tác giả gợi ý thiết lập , nghĩa là chỉ cập nhập chừng 10% theo độ dốc. Sau đó, một loạt những cải tiễn đã được đề xuất. Một số sử dụng thêm thông tin “độ cong” của hàm lỗi bằng cách tính thêm đạo hàm bậc hai như phương pháp SECANT, nhưng không thành công do chi phí tính thêm đạo hàm bậc hai là cao, nhất là khi hàm không phải hàm logistic. Các cải tiến khác theo hướng “học thích nghi”, nghĩa là hệ số học cũng là hàm theo bước lặp, , và theo heuristic “nếu hàm lỗi các bước trước đang đang thì bước kế sẽ giảm mạnh”. Heuristic này được cài đặt và trở thành công thức “chuẩn” cho mạng nơ-ron, gọi là quy tắc học delta-bar-delta.

Scott Fahlman tiếp cận theo heuristic trực quan hơn khi ông giả sử nếu hàm lỗi có dạng parabol thì cực tiểu hàm lỗi sẽ nằm ngay vị trị có đạo hàm bằng 0. Qua 3 điểm xác định một parabol, hay qua thông tin độ dốc cũng như biến thiên trọng 2 bước trước ta cùng xác định được một parabol. Như vậy, hàm lỗi sẽ được xấp xỉ bằng chuỗi các parabol như sau:

Gọi

. là giá trị đạo hàm ở bước lặp trước,

. là giá trị đạo hàm ở bước lặp hiện hành,

thì giá trị đạo hàm ở bước lặp kế sẽ bằng 0:

.

Gọi

. là giá trị biến thiên trọng ở bước trước, và

. biến thiên trọng sẽ được cập nhật theo công thức:

, hay

.

Như vậy, chỉ cần thêm 3 biến tạm để lưu các giá trị đạo hàm và biến thiên trọng của các bước trước, ta không còn băn khoăn về hệ số học nữa. Tuy nhiên, có một lưu ý nhỏ ở đây, khi hàm lỗi dốc gần như thẳng đứng, . Trong trường hợp này, ta sử dụng lại công thức gốc, với :

.

***3.3.2 Về hàm g(.)***

Trong công thức ánh xạ, lan truyền tiến, giá trị trung gian (cũng như giá trị kết quả) được ép theo hàm truyền là hàm logistic:

.

Trong thực tế, có thể chọn bất cứ hàm nào làm hàm truyền miễn thỏa:

1. Là hàm bị chặn;
2. Đơn điệu tăng;
3. Liên tục và trơn.

Các hàm này có dạng hình chữ S nên gọi chung là hàm *Sigmoid*.

Bên cạnh hàm logisctic, 2 hàm sigmoid sau thường được sử dụng khi cần đầu ra thuộc [-1, 1]:

• Hàm hyperpol, là hàm có liên quan mật thiết với hàm logistic :

,

với đạo hàm

.

• Hàm tang-hyperpol có tốc độ tiến đến giới hạn nhanh:

,

với đạo hàm

,

.

**4 Máy học vec-tơ hỗ trợ – Support Vector Machine (SVM)**

Mạng nơ-ron là công cụ mạnh không chỉ giải quyết tốt bài toán xấp xỉ một hàm thực mà còn giải rất tốt bài toán phân lớp. Tuy nhiên, khi bài toán phân lớp có số lớp nhỏ và tập dữ liệu học lớn, và dữ liệu có quan hệ phân tách tuyến tính, hoặc có thể tuyến tính hóa, mô hình máy học vec-tơ hỗ trợ - Support Vector Machine (SVM), thường hiệu quả hơn. Để giới thiệu phương pháp SVM, ta bắt đầu bằng bài toán phân 2 lớp.

Cho tập dữ liệu huấn luyện (training set) gồm dữ liệu , với là vector đặc trưng (feature vector), và là nhãn (label) hay lớp (class) tương ứng. Ta tìm cách gán nhãn cho mỗi vec-tơ đầu vào .

Mục đích của ta là tìm ra một “ranh giới phân lớp” – classification boundary phân chia các điểm dữ liệu thành 2 lớp riêng biệt. Giả sử ranh giới này là một siêu phẳng – hyperplane P có phương trình:

,

và hàm dấu , là hàm xác định bởi:

,

thì điểm sẽ được gán nhãn vào lớp:

.

Ta định nghĩa khoảng cách từ một điểm trong không gian chiều, , tới siêu phẳng là:

***4.1 Bộ phân lớp lề cực đại***

Khoảng cách nhỏ nhất từ các điểm ảnh trong tập mẫu đến siêu phẳng , ký hiệu là margin – lề:

.

Nhiệm vụ tìm siêu phẳng P được đưa về giải bài toán tối ưu tham số có ràng buộc sau:

thỏa

.

Khi ấy, điểm sẽ được gán nhãn theo

.

***4.2 Bộ phân lớp lề mềm***

Trong thưc tế, do ảnh hưởng từ nhiễu, tập dữ liệu thường không được tách bạch hoàn toàn: hầu hết các điểm dữ liệu nằm 2 bên của lề cực đại xác định như trên, nhưng thi thoảng, có một số điểm nằm lẫn lộn trong khoảng lề cực đại phân cách 2 lớp, và chúng cũng có thể đan xen nhau, nghĩa là điểm của lớp này lại gần lề phía lớp kia hơn. Để hạn chế gán nhãn sai cho những trường hợp này, ta làm cho lề “mềm” đi bằng cách giới thiệu thêm khái niệm mềm – slack, :

.

Khi ấy, bài toán tối ưu lề cực đại ở trên được đưa về bài toán tối ưu:

thỏa các điều kiện:

,

và

.

***4.3 SVM với nhân tử***

Trong trường hợp dữ liệu không có quan hệ tuyến tính, người phân tích có thể tuyến tính hóa bộ dữ liệu mẫu trước khi cho học bằng SVM. Việc tuyến tính hóa phụ thuộc nhiều vào kinh nghiệm người phân tích. Tuy nhiên, thay vì giải bài toán tối ưu lề như trên, ta giải bài toán đối ngẫu của nó. Từ bài toán đối ngẫu, một số cải tiến SVM – gọi chung là *SVM với nhân tử* được giới thiệu như những cải tiến của SVM. Trước tiên ta bắt đầu với bài toán đối ngẫu.

***4.3.1 Bài toán đối ngẫu***

Bài toán đối ngẫu cho bộ phân lớp lề mềm được thiết lập:

thỏa

.

Khi ấy, điểm sẽ được gán nhãn theo

.

***4.3.2 Nhân tử***

Nếu trong không gian gốc dữ liệu khó phân biệt tuyến tính thì dùng phép biến đổi phi tuyến (non-linear transformation) để chuyển sang không gian mới (dễ phân biệt tuyến tính hơn). Trong không gian mới:

thỏa

và

.

Nếu lựa chọn khéo, ta có thể tính được:

,

mà không cần phải tính tường minh .

Khi đó ta chỉ cần làm việc trên không gian gốc (cùng với hàm kernel )

Ví dụ, xét phép biến đổi:

Ta có:

.

***4.3.3 Các nhân tử phổ biến***

Sau đây là một số nhân tử phổ biến:

• Nhân tử tuyến tính:

**.**

• Nhân tử đa thức:

.

• Nhân tử Gauss hay RF – Radial basic function:

.

• Nhân tử Sigmoid:

.

***4.4 Liên hệ với hồi quy logistic***

Bài toán phân lớp lề mềm có thể được phát biểu lại thành:

,

trong đó:

,

được gọi là mất mát lề - hinge loss.

Và ta có thể giải bằng kỹ thuật hồi quy logistic hay tổng quát hơn, bằng mạng nơ-ron như trình bày trước. Tuy nhiên, khi dữ liệu có quan hệ (gần) tuyến tính và dữ liệu lớn, SVM thích hợp hơn cho ài toán phân lớp.

**5 Hồi quy di truyền**

Mạng nơ-ron là một công cụ mạnh, đặc biệt, với sự phát triển của phần cứng cùng với công nghệ hợp tác tính toán trên cloud, mạng nơ-ron ngày càng trở nên phổ biến, điển hình với mạng học sâu – Deep Learning, một mạng nơ-ron nhiều lớp.

Bên cạnh đó, một công cụ cũng rất tổng quát là *hồi quy di truyền* – Genetic Regression (GR), mà sẽ học từ lớp hàm C không tuyến tính, sẽ được giới thiệu trong phần cuối này.

GR là thuật toán máy tính mô phỏng tiến hóa tự nhiên của Darwin, đặc biệt phép lai theo kiểu Menden. Hồi quy di truyền là một tổng quát hóa của *thuật giải di truyền* – Genetic Programming (GA) với cấu trúc di truyền không cố định.

***5.1 Các thành tố di truyền***

Cho

. là tập các phép toán và các hàm cơ sở. Ví dụ

;

. là tập các ký hiệu kết thúc.

Một cá thể là một biểu thức số học được tạo thành từ các phép toán trong tập theo quy tắc văn phạm của biểu thức. Ví dụ,

là một biểu thức.

Trong cài đặt, để dễ xử lý, thường người ta biểu diễn biểu thức theo kiểu Balan ngược, nghĩa là toán tử để sau toán hạng, nên còn có thể khác là *biểu thức postfix*. Chẳng hạn, F có biểu diễn Balan ngược là

.

***5.2 Các phép di truyền***

Có 2 phép di truyền chính là *phép lai* – Crossover, và *phép đột biến* – Mutation.

• Phép lai nhận vào 2 biểu thức , gọi là các *biểu thức cha-mẹ* – Parent, và tạo ra 2 biểu thức mới , gọi là *biểu thức con-cái* – Children, như sau:

(ta sẽ sử dụng 2 biểu thức cha-mẹ sau:

. và

.

minh họa cho thuật toán).

***5.2.1 Crossover***

1. Phát sinh 2 điểm lai ngẫu nhiên , với là số ký hiệu (phép toán hay ký hiệu kết thúc, không tính dấu phẩy (“,”) và dấu ngoặc (“(“ hay “)”)). Giả sử .
2. Cắt các biều thức cha-mẹ thành các biểu thức thành phần tại các điểm lai (2 biểu thức con hoàn chỉnh và 2 biểu thức con khuyết cần “ghép” cho hoàn chỉnh). Ví dụ từ và , ta được:

.

.

.

. .

trong đó ký hiệu điểm ghép.

1. Ghép biểu thức con hoàn chỉnh của biểu thức cha-mẹ này vào điểm ghép của biểu thức cha-mẹ kia, và ngược lại. Chẳng hạn, ví dụ trên ta có:

. ;

. .

• Phép đột biến nhận vào một biểu thức và tạo ra biểu thức mới . Có 2 dạng đột biến: đột biến thay thế – Replace Mutation (RM) và đột biến rút gọn – Compact Mutation (CM).

***5.2.2 RM – Replace Mutation***

(ta minh họa bằng này)

1. Phát sinh ngẫu nhiên điểm đột biến .
2. . Nếu điểm đột biến là một ký hiệu kết thúc , thuật toán chọn ngẫu nhiên một ký hiệu kết thúc và thay trong bằng để được . Ví dụ đang xét, giả sử , và chọn được , ta được .

. Nếu điểm đột biến là phép toán , thuật toán chọn ngẫu nhiên một phép toán cùng ngôi , và thay bằng để được . Ví dụ đang xét, giả sử , thì là phép toán 1-ngôi, và chọn được phép toán 1-ngôi , thì ta được

.

***5.2.3 CM – Compact Mutation***

(ta minh họa với trong phép toán này.)

* Quét biểu thức và thực hiện các phép tính có thể thực hiện cho đến khi không thể rút gọn được nữa. Khi có một hằng trị mới thì thểm vào tập ký hiệu kết thúc: . Ví dụ:

.

, và

, và

, và .

***5.3 Lược đồ hồi quy di truyền***

**Bước 1**: Khởi tạo quần thể gồm P biểu thức ban đầu với trị “thích nghi” tương ứng. Trị thích nghi là tổng sai số của biểu thức đo trên toàn bộ tập mẫu , và bằng .

**Bước 2**: Lặp các bước sau cho tới khi điều kiện dừng thỏa

1. Chọn ngẫu nhiên cặp cha-mẹ , và thực hiện phép lai trên từng cặp , được gồm 2p biểu thức con-cái. Tính trị thích nghi cho các biểu thức con-cái mới sinh ra.
2. Chọn ngẫu nhiên cá thể trong quần thể , và thực hiện các phép đột biến được . Tính trị thích nghi cho các biểu thức mới đột biến ra.
3. .
4. “Chọn lọc tự nhiên” chỉ giữ lại P cá thể có trị thích nghi tốt nhất.

**Bước 3**: các biểu thức tốt nhất có thể được chọn làm mô hình cần tìm.

**Tham khảo**

[1] Nguyễn Đình Thúc. Mạng Nơ-ron: Phương pháp & Ứng dụng. NXB GD, 2000.

[2] Mark M. Meerschaert. Mathemathical Modeling Fourth Edition. AP 2013.